

Aus dem Departement für Informatik
Universität Freiburg (Schweiz)

Optimierung automatisierter Kompaktlager in Entwurf und Steuerung

INAUGURAL-DISSERTATION

zur Erlangung der Würde eines Doctor scientiarum informaticarum
der Mathematisch-Naturwissenschaftlichen Fakultät
der Universität Freiburg in der Schweiz

vorgelegt von
Andreas Klinkert
aus
Lütisburg (SG)

Nr. 1353
Druckerei der Universität Freiburg
2001

Von der Mathematisch-Naturwissenschaftlichen Fakultät der Universität Freiburg in der Schweiz
angenommen, auf Antrag der Herren

Prof. Dr. Jacques Pasquier-Rocha, Universität Freiburg, Schweiz (Präsident der Jury)

Prof. Dr. Heinz Gröflin, Universität Freiburg, Schweiz (Referent)

Prof. Dr. Thomas M. Liebling, Ecole Polytechnique Fédérale de Lausanne, Schweiz (Korreferent)

Freiburg, den 31. August 2001

Der Leiter der Doktorarbeit

Der Dekan

Prof. Dr. Heinz Gröflin

Prof. Dr. Alexander von Zelewsky

ABSTRACT

This thesis investigates the optimization of automated high-density storage systems at design stage and in control logic development. In modern storage technology, automated high-density storage systems offer a space saving alternative to traditional storage rack systems. They comprise several floors subdivided into parallel storage aisles and linking cross-aisles. Pallet movements are computer-controlled and executed by elevators and transfer carriages integrated in each aisle. Layout and operation of a specific warehouse are customer-dependent.

Two main issues are investigated. A central problem in designing complex storage systems is to estimate their performance in a reliable way. Performance analysis is needed in order to evaluate various design alternatives differing in size, layout and other parameters. Common performance measures are based on maximum throughput (i.e. number of incoming and outgoing pallets in a given time period). Due to the complexity of automated high-density storage systems, no standard analytical method for performance analysis is available. Also, the use of simulation methods is questionable, not only because of the large number of processes to be simulated, but also because the control logic is often not fully specified at the design stage.

In this work, a linear optimization model for performance analysis is developed which is independent of the control system and needs a minimum of input data. The storage system is described by a network model where palette movements are represented by multi-commodity flows. An optimal allocation strategy and resulting processor loads can be calculated by flow optimization, leading to performance upper bounds. This multi-commodity flow model is then enhanced in order to take into account idle processor moves directly affecting processor loads, and pallet sojourn times in the system which affect storage strategy. It is shown that idle processor moves can be modeled via special transportation problems embedded in the flow model. Furthermore, constraints based on Little's formula are introduced in order to link stock levels, pallet sojourn times and throughput. Numerical results show the validity of the model. In addition to calculating performance upper bounds, the model offers support in identifying bottleneck resources and designing efficient storage and routing strategies.

The second issue concerns the development of an optimized control logic for automated high-density warehouses. Controlling such storage systems is a complex task involving a multitude of decisions at different levels. Three main logical components can be distinguished: determination of the transportation orders based on the storage and retrieval orders (storage strategy, routing), determination of the relocations (forced relocations, reorganization) and determination of the schedule (sequencing and scheduling of the transportation actions). This thesis focuses on opti-

mizing the control system at the schedule level. An optimization model is developed which generates an optimized schedule based on a given set of transportation orders. The transportation orders determine the actions to be performed by each processor. The model's objective is to sequence and schedule the actions in order to minimize total makespan. In the following, this problem is referred to as the storage problem.

As a first result, it is shown that the set of all admissible schedules (i.e. the solution space) can be fully described by means of conjunctive and disjunctive precedence constraints. A precedence constraint expresses the condition that there must be a given time lag between the starting times of two operations. A disjunctive precedence constraint corresponds to a pair of precedence constraints of which at least one must hold.

Models with disjunctive precedence constraints can be formulated as disjunctive graph problems. The nodes represent the operations and the weighted arcs the precedence constraints. Disjunctive precedence constraints correspond to a disjunctive arc set, i.e. a pair of disjunctive arcs of which one has to be selected. Disjunctive graphs are a well-known concept for modeling sequencing problems. Up to now, they have been used mainly in the context of the famous job-shop problem. This problem can be expressed in a disjunctive graph where a complete acyclic selection of disjunctive arcs has to be determined which minimizes the length of a maximal path in the resulting graph. A selection is called complete if it contains at least one arc of each disjunctive set.

A distinctive property of disjunctive job-shop graphs is that each disjunctive set contains two arcs with identical extremities but reverse orientation. Modeling the storage problem with disjunctive precedence constraints leads to disjunctive graphs for which this property does not hold. Motivated by this fact, this thesis introduces a generalization of classical disjunctive graphs, allowing disjunctive pairs of arcs with different extremities.

The structural properties of generalized disjunctive graphs are studied. A main result is that the feasibility problem for generalized disjunctive graphs - in contrast to the feasibility problem in the job-shop case - is NP-hard. The feasibility problem addresses the question whether a generalized disjunctive graph has an admissible, i.e. complete, positive acyclic selection of disjunctive arcs. Furthermore, it is shown that the feasibility problem is also NP-hard for the disjunctive graphs resulting from the storage problem. This has severe consequences for the design of optimization algorithms for the storage problem and solution methods developed for the job-shop problem cannot be applied to the storage problem.

Constructive heuristics like priority dispatch rule methods are hardly applicable since the question whether a partial solution can be extended to an admissible solution is NP-hard. Iterative improvement methods based on the exchange of critical disjunctive arcs are problematic as well since exchanging critical arcs in disjunctive graphs can lead to inadmissible solutions. Bottleneck approaches like the well-known shifting bottleneck algorithm are not applicable because the sequencing of single processors typically produces inadmissible solutions. The design of an

appropriate neighborhood for iterative improvement is difficult since, in general, several arcs on different processors have to be exchanged in order to obtain an admissible neighbor solution. A crucial question is which arcs need to be exchanged additionally when exchanging one critical arc. This problem leads to the concept of forced and forbidden arcs. A partial selection forces, respectively forbids a disjunctive arc if all admissible selections including this partial selection must contain, respectively cannot contain this arc.

For solving the storage problem, a neighborhood is defined which is based on forced and forbidden arcs and the concept of 1-job-storage-graphs. These graphs are a relaxation of storage graphs in the sense that only disjunctive arcs are considered which are incident to a given job. Of special interest are 1-job-storage-graphs that are conjunctively closed, i.e. in which no forced arcs exist. An important result is that a conjunctively closed, positive acyclic 1-job-storage-graph always has an admissible selection. Moreover, it is shown that the feasibility problem for 1-job-storage-graphs is polynomially solvable and the forced arcs can be calculated in polynomial time. These results allow the design of a relatively complex neighborhood involving the calculation of the forced arcs set in a 1-job-storage-graph and its completion to an admissible selection. Based on this neighborhood, a descent method is devised which improves an existing solution by iteratively exchanging critical arc sets. Numerical results are favorable and demonstrate considerable optimization potential. For realistic problem data and arbitrary starting sequences, schedule improvements in the order of thirty to fifty percent can be achieved in a few minutes of computation time.

ZUSAMMENFASSUNG

Die vorliegende Dissertation beschäftigt sich mit der Optimierung automatisierter Kompaktlager in der Phase des Entwurfs und bei der Entwicklung der Lagersteuerung. Automatisierte Kompaktlager bilden in der modernen Lagertechnik eine platzsparende Alternative zu herkömmlichen Zeilenregallagern. Ein automatisiertes Kompaktlager besteht aus mehreren Lagerebenen, welche gegliedert sind in parallel aneinanderliegende Lagerkanäle und rechtwinklig dazu angeordnete Verbindungsquergänge. Der Transport des palettisierten Lagerguts geschieht mit Hilfe von integrierten Kanalfahrzeugen, Quergangswagen und Aufzügen.

Im Zusammenhang mit automatisierten Kompaktlagern werden zwei Problemstellungen bearbeitet. In der Entwurfsphase ist die Problematik der Leistungsanalyse von zentraler Bedeutung. Die Evaluation verschiedener Entwurfsalternativen bedingt eine Methodik zur Abschätzung der zu erwartenden Lagerleistung. Als Leistungsmass wird die maximale Anzahl Ein- und Auslagerungen betrachtet, welche in einem bestimmten Zeitraum möglich ist. Auf Simulation basierende Ansätze sind aufgrund der Komplexität der Steuerungsprozesse nur schwer realisierbar. In dieser Arbeit wird ein Lineares Optimierungsmodell für die Leistungsanalyse entwickelt, welches steuerungsunabhängig ist und ein Minimum von Eingabedaten benötigt. Zur formalen Darstellung der Lagersysteme wird ein Netzwerkmodell entworfen, in welchem die Lagerbewegungen als Mehrgüterflüsse dargestellt werden. Durch Flussoptimierung lässt sich die Belastung der Förderprozessoren bei optimaler Allokationsstrategie abschätzen, woraus eine obere Schranke für die Lagerleistung resultiert. Eine Schwierigkeit bei der Abschätzung der Prozessorenbelastung besteht darin, den Aufwand für die reihenfolgeabhängigen Anfahrtswege zu bestimmen. Es wird gezeigt, wie mit Hilfe von speziellen Transportproblemen, welche in das Lineare Programm eingebettet sind, der Leerfahrtenaufwand der Förderprozessoren abgeschätzt werden kann. Als weitere Modellkomponente wird die Lagerstrategie abhängig gemacht von den Verweilzeiten der Paletten im Lager. Auf der Basis des Little'schen Gesetzes wird ein System von linearen Restriktionen entworfen, welches einen Zusammenhang zwischen Lagerbestand, Verweilzeit und Lagerleistung herstellt. Numerische Resultate zeigen die Validität des Modells. Neben der Bestimmung einer oberen Leistungsschranke bietet das Modell Unterstützung bei der Erkennung von Engpass-Ressourcen und beim Entwurf von optimierten Lager-, Routen- und Reorganisationsstrategien.

Als zweite Problemstellung beschäftigt sich diese Arbeit mit dem Entwurf einer optimierten Steuerung für automatisierte Kompaktlager. Die Steuerung eines Lagersystems bildet eine komplexe Aufgabe, welche eine Vielzahl von Entscheidungsdimensionen umfasst. Auf der Stufe der dispositiven Steuerung lassen sich drei Hauptfunktionen unterscheiden: Bestimmung der Transportaufträge aufgrund der Ein- und Auslagerungsaufträge (Lagerstrategie, Routenwahl),

Bestimmung der Umlagerungen (erzwungene Umlagerungen, Reorganisation) und Bestimmung des Ablaufplans (Reihenfolge und Terminierung der Transportaktionen). In dieser Arbeit wird die Optimierung der Steuerungsprozesse auf der Ebene der Ablaufpläne betrachtet. Es wird ein Optimierungsmodell entwickelt, welches ausgehend von einer Menge von Transportaufträgen einen optimierten Ablaufplan generiert. Durch die Transportaufträge wird für jeden Förderprozessor die Menge der auszuführenden Aktionen vorgegeben. Ziel des Optimierungsmodells ist die Bestimmung der Ausführungsreihenfolge und der Startzeitpunkte dieser Aktionen, so dass die Gesamtdauer des Lagerprozesses (makespan) minimiert wird. Dieses Optimierungsproblem wird im folgenden als Lager-Problem bezeichnet.

Als erstes Resultat wird gezeigt, dass sich die Menge aller zulässigen Ablaufpläne (d.h. der Lösungsraum) vollständig beschreiben lässt mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen. Eine Präzedenzbedingung besagt, dass zwei Operationen zeitlich nacheinander ausgeführt werden müssen, wobei zwischen den Startzeitpunkten der Operationen eine bestimmte minimale Zeitspanne verstreichen muss. Eine disjunktive Präzedenzbedingung entspricht einem Paar von Präzedenzbedingungen, von denen mindestens eine zutreffen muss.

Modelle mit disjunktiven Präzedenzbedingungen lassen sich in Form von disjunktiven Graphen darstellen. Die Knoten repräsentieren die Operationen und die gewichteten Bogen die Präzedenzbedingungen. Disjunktive Präzedenzbedingungen entsprechen einer disjunktiven Bogenmenge, d.h. einem Paar von disjunktiven Bogen, von denen einer ausgewählt wird. Disjunktive Graphen bilden ein bekanntes Konzept zur Modellierung von Reihenfolgeproblemen und wurden bis anhin vor allem im Zusammenhang mit dem Job-Shop-Problem verwendet. Das Job-Shop-Problem lässt sich als disjunktives Graphenproblem darstellen, bei welchem es darum geht, eine vollständige, azyklische Selektion von disjunktiven Bogen zu bestimmen, so dass die Länge eines längsten Weges im resultierenden Graphen minimal ist. Eine Selektion heisst vollständig, wenn sie aus jeder disjunktiven Menge mindestens einen Bogen enthält.

Disjunktive Job-Shop-Graphen haben die Eigenschaft, dass die disjunktiven Mengen aus zwei Bogen bestehen, welche dieselben Knoten verbinden, aber entgegengesetzte Richtungen aufweisen. Die Modellierung des Lager-Problems mit Hilfe von disjunktiven Präzedenzbedingungen führt zu disjunktiven Graphen, bei welchen diese Eigenschaft nicht zutrifft. Dies führt zu einer Verallgemeinerung des Konzepts der disjunktiven Graphen. Als neuer Modellierungsansatz werden in dieser Arbeit verallgemeinerte disjunktive Graphen betrachtet, welche sich von klassischen disjunktiven Graphen dadurch unterscheiden, dass die disjunktiven Mengen zwei Bogen mit unterschiedlichen Start- und Endknoten umfassen.

Die strukturellen Eigenschaften von verallgemeinerten disjunktiven Graphen werden ausführlich untersucht. Als wichtiges Resultat wird gezeigt, dass das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen NP-hart ist. Unter dem Zulässigkeitsproblem wird die Frage verstanden, ob ein verallgemeinerter disjunktiver Graph eine zulässige, d.h. eine vollständige, positiv azyklische Selektion von disjunktiven Bogen besitzt. Weiter wird gezeigt, dass diese kom-

plexitätstheoretische Eigenschaft auch für die disjunktiven Lager-Graphen zutrifft, welche bei der Modellierung des Lager-Problems resultieren. Dies hat weitreichende Konsequenzen für den Entwurf eines Optimierungsverfahrens für das Lager-Problem. Eine Folge davon ist, dass ein Grossteil der Lösungsansätze, welche im Zusammenhang mit dem Job-Shop-Problem entwickelt wurden, nicht auf das Lager-Problem übertragbar sind. Heuristische Eröffnungsmethoden wie beispielsweise Prioritätsregeln-Verfahren sind kaum anwendbar, da die Frage, ob eine bestehende Teillösung zu einer zulässigen Lösung erweitert werden kann, NP-hart ist. Verbesserungsverfahren, welche auf dem Austausch von kritischen disjunktiven Bogen beruhen, sind ebenfalls problematisch, da der Austausch kritischer Bogen in verallgemeinerten disjunktiven Graphen zu unzulässigen Lösungen führen kann. Bottleneck-Verfahren im Stil des bekannten Shifting-Bottleneck-Algorithmus sind nicht anwendbar, da die Sequenzierung einzelner Prozessoren typischerweise unzulässige Lösungen produziert. Der Entwurf eines geeigneten Nachbarschaftsbegriffs für ein Verbesserungsverfahren erweist sich als schwierig, da im allgemeinen mehrere disjunktive Bogen auf verschiedenen Prozessoren ausgetauscht werden müssen, damit eine zulässige Nachbarlösung resultiert.

Beim Entwurf eines Nachbarschaftsbegriffs stellt sich die Frage, welche Bogen beim Austausch eines kritischen Bogens zusätzlich ausgetauscht werden müssen, damit eine zulässige Nachbarsелеktion entsteht. Zur Bewältigung dieses Problems wird das Konzept der erzwungenen und verbotenen disjunktiven Bogen eingeführt. Ein disjunktiver Bogen heisst erzwungen bzw. verboten von einer Teilselektion, wenn alle zulässigen Selektionen, welche diese Teilselektion umfassen, diesen Bogen enthalten müssen bzw. nicht enthalten können.

Zur Lösung des Lager-Problems wird ein Nachbarschaftsbegriff entworfen, welcher einerseits auf dem Konzept der erzwungenen und verbotenen Bogen basiert, und andererseits auf der Idee der 1-Job-Lager-Graphen. Diese bilden eine Relaxation der Lager-Graphen, bei welchen nur disjunktive Bogen betrachtet werden, die mit einem ausgewählten Job inzidieren. Betrachtet werden 1-Job-Lager-Graphen, welche konjunktiv abgeschlossen sind, d.h. in welchen keine disjunktiven Bogen erzwungen werden. Als wichtiges Resultat wird gezeigt, dass ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph stets eine zulässige Selektion besitzt. Weiter wird gezeigt, dass das Zulässigkeitsproblem für 1-Job-Lager-Graphen polynomial lösbar ist und die Bestimmung der erzwungenen Bogen mit polynomialem Aufwand möglich ist. Diese Resultate ermöglichen den Entwurf eines relativ komplexen Nachbarschaftsbegriffs, bei welchem ausgehend von einem 1-Job-Lager-Graphen durch Bestimmung der erzwungenen Bogen und Ergänzung zu einer zulässigen Selektion eine Nachbarlösung konstruiert wird. Auf der Basis dieses Nachbarschaftsbegriffs wird ein iteratives Abstiegsverfahren entwickelt, welches eine bestehende Lösung sukzessive durch Austausch von kritischen Bogenmengen verbessert. Die numerischen Resultate sind vielversprechend und zeigen ein beträchtliches Optimierungspotential. Für realistische Problemdata und zufällige Startreihenfolgen wird bei einem Rechenaufwand von wenigen Minuten eine Verbesserung der Ablaufpläne im Bereich von dreissig bis fünfzig Prozent erzielt.

INHALTSVERZEICHNIS

Kapitel 1	Elemente der Lagertechnik	1
1.1.	Lager und Logistik	1
1.2.	Lagersysteme	4
1.2.1.	Lagergut	4
1.2.2.	Lagereinrichtungen	6
1.2.3.	Transporteinrichtungen	7
1.3.	Typen von Lagersystemen	12
1.3.1.	Ueberblick über Lagertypen	15
1.3.2.	Zeilen- und Blockregallager	18
1.4.	Betrieb und Steuerung von Lagersystemen	21
1.4.1.	Lagerhaltung und Lagersteuerung	22
1.4.2.	Dispositive Steuerung	27
Kapitel 2	Automatisierte Kompaktlager	35
2.1.	Systembeschreibung und Funktionsweise	35
2.1.1.	Systemübersicht	36
2.1.2.	Transporteinrichtungen	37
2.2.	Steuerung	39
2.2.1.	Transportrouten	39
2.2.2.	Transportaktionen	41
2.3.	Netzwerk-Modell zur Systembeschreibung	46
2.3.1.	Beschreibung	46
2.3.2.	Beispiel	49
2.3.3.	Formale Definition des System-Graphen	51
Kapitel 3	Modell zur Leistungsanalyse	55
3.1.	Problembeschreibung	56
3.1.1.	Entwurf automatisierter Kompaktlager	56
3.1.2.	Leistungsanalyse	58
3.2.	Ressourcenallokation mittels Mehrgüterflüssen	62
3.2.1.	Ein- und Auslagerungsflüsse	62
3.2.2.	Leistungsabschätzung	66
3.3.	Leerfahrten	69
3.4.	Rotation und Verweilzeiten	79
3.5.	Modellübersicht	84
3.6.	Numerische Resultate	87

3.6.1. Beispiel 1	90
3.6.2. Beispiel 2.	95
3.6.3. Zusammenfassung	99
Kapitel 4 Modell zur Steuerungsoptimierung	101
4.1. Problembeschreibung	102
4.2. Modellierungsansatz: Disjunktive Präzedenzbedingungen	103
4.2.1. Konjunktive und disjunktive Präzedenzbedingungen.	103
4.2.2. Job-Shop-Problem	106
4.3. Modell für das Lager-Problem	109
4.3.1. Synchronisation und kontextabhängige Operationsdauer	109
4.3.2. Modellübersicht	114
4.4. Räumliche Nebenbedingungen	116
Kapitel 5 Verallgemeinerte disjunktive Graphen	121
5.1. Disjunktive Präzedenzbedingungen und disjunktive Graphen	123
5.1.1. Disjunktive Präzedenzmodelle	124
5.1.2. Disjunktive Graphen	126
5.1.3. Literatur	128
5.2. Definitionen und Eigenschaften	130
5.2.1. Definitionen	130
5.2.2. Eigenschaften	132
5.3. Spezielle Klassen von disjunktiven Graphen	135
5.3.1. Klassische disjunktive Graphen.	137
5.3.2. Job-Shop-Graphen.	137
5.3.3. Lager-Graphen	140
5.3.4. 1-Job-Lager-Graphen	143
5.4. Konjunktive Erweiterung	144
5.4.1. Einführung.	144
5.4.2. Definitionen	149
5.4.3. Eigenschaften	150
5.4.4. Global verbotene und erzwungene Bogen	157
5.4.5. Allgemeines Prinzip der konjunktiven Erweiterung.	162
5.4.6. Relationen	167
5.4.7. Fixpunkttheorie	171
5.4.8. Komplexität	173
5.5. Zulässigkeitsproblem	174
5.5.1. Einführung.	174
5.5.2. Verallgemeinerte disjunktive Graphen	176
5.5.3. Klassische disjunktive Graphen.	185
5.5.4. Job-Shop-Graphen.	189
5.5.5. Lager-Graphen	189
5.5.6. 1-Job-Lager-Graphen	197

Kapitel 6	Heuristik zur Steuerungsoptimierung	203
6.1.	Optimierung in disjunktiven Graphen	204
6.1.1.	Eröffnungsverfahren	205
6.1.2.	Verbesserungsverfahren	205
6.2.	Vorbereitung	208
6.3.	Nachbarschaftsbegriff	212
6.4.	Beispiel zum Nachbarschaftsbegriff	218
6.4.1.	Illustration zu Definition 6.5.	220
6.4.2.	Illustration zu Definition 6.6.	222
6.4.3.	Konstruktion einer Nachbarselektion, Fall 1.	224
6.4.4.	Konstruktion einer Nachbarselektion, Fall 2.	228
6.4.5.	Illustration zu Satz 6.10.	232
6.5.	Heuristik	234
6.6.	Zulässige Startlösung	235
6.7.	Numerische Resultate	237
Kapitel 7	Konklusion	243
	Definitionen und Notationen	247
	Literaturverzeichnis	249
	Danksagung	255
	Curriculum Vitae	257
	Anhang	

KAPITEL 1

Elemente der Lagertechnik

Als Einführung werden in diesem Kapitel einige grundlegende Konzepte der Lagertechnik besprochen. Abschnitt 1.1. beschäftigt sich mit der Bedeutung des Lagerns in der Logistik und den verschiedenen Lagerfunktionen innerhalb eines Produktionsbetriebs. In Abschnitt 1.2. werden die funktionalen Komponenten eines Lagersystems diskutiert und verschiedene elementare Begriffe aus der Lagertechnik erläutert. Abschnitt 1.3. gibt eine Uebersicht über die gebräuchlichsten Lagertypen. Ein spezielles Gewicht wird den Kompaktlagern und insbesondere den dynamischen Blocklagern zugewiesen. In Abschnitt 1.4. wird der Betrieb und die Steuerung von Lagersystemen diskutiert. Der Begriff der Lagersteuerung wird abgegrenzt, und die verschiedenen Aufgaben und Komponenten der Lagersteuerung werden besprochen. Besondere Beachtung wird der Ebene der dispositiven Steuerung geschenkt, welche die wesentlichen Entscheidungsmechanismen der Lagersteuerung umfasst.

Die Ausführungen dieses Kapitels dienen als Grundlage für die anschliessenden Problemstellungen bei der Optimierung automatisierter Kompaktlager.

1.1. Lager und Logistik

Die Logistik befasst sich allgemein mit der Gestaltung, Steuerung und Kontrolle von Flüssen von Objekten zwischen verschiedenen Orten. Die Aufgabe der Logistik besteht in der Bereitstellung

der richtigen Objekte zum richtigen Zeitpunkt am richtigen Ort. Als Objekte der Logistik werden primär physische Güter betrachtet, es können aber auch Menschen, Dienstleistungen oder Informationen darunter verstanden werden.

Die Logistik umfasst zwei Hauptfunktionen: Transport und Lagerung. Transportvorgänge beziehen sich auf die räumliche Dimension der Logistik, die Bewegung der Objekte an den richtigen Ort, und Lagervorgänge auf die zeitliche Dimension, die Bereitstellung der Objekte zum richtigen Zeitpunkt. Transporte sind notwendig, wenn Objekte sich nicht am gewünschten Ort befinden, und Lagerung, wenn Objekte sich nicht zur gewünschten Zeit an einem Ort befinden. Bei der Betrachtung von Objektflüssen ist Lagerung generell an denjenigen Stellen notwendig, wo zu einem bestimmten Zeitpunkt die kumulierten Zuflüsse von Objekten die kumulierten Abflüsse übersteigen (vgl. [Corsten, H. (1996)]).

Die Unternehmenslogistik befasst sich speziell mit den Objektflüssen, welche im Rahmen eines Unternehmens auftreten. In einem Produktionsunternehmen beginnt der Materialfluss typischerweise bei der Beschaffung der Rohmaterialien, durchläuft die Produktion und endet bei den Fertigprodukten, welche zu den Abnehmern transportiert werden. Die Aufgaben der Unternehmenslogistik lassen sich entsprechend unterteilen in die Bereiche Beschaffung, Produktion und Distribution (und eventuell Entsorgung).

Die Lagerung stellt eine wesentliche Komponente der Logistik eines Produktionsunternehmens dar. Entsprechend der Position im Produktionsablauf werden die verschiedenen Lager in einem Produktionsunternehmen unterteilt in Beschaffungslager, Produktionslager und Distributionslager.

Beschaffungslager

Beschaffungslager (bzw. Eingangslager, Rohmateriallager, Werkstofflager) dienen der Lagerung von Ausgangsmaterialien, welche ausserhalb des Unternehmens beschafft werden, und versorgen die Produktion gemäss den Fertigungsaufträgen. Sie haben unter anderem zur Funktion, die Produktionsprozesse zu einem gewissen Grade von den Zulieferern abzukoppeln und somit unvorhergesehene Schwankungen oder Engpässe auf dem Beschaffungsmarkt auszugleichen. Sie ermöglichen weiter eine preisgünstige Beschaffung der Ausgangsmaterialien, beispielsweise durch Sammelbestellungen, Mengenrabatte für Grossbestellungen oder Einkauf bei günstigem Angebot.

Produktionslager

Produktionslager (bzw. Zwischenlager, Pufferlager, Halbfabrikateler) gleichen die technologisch oder organisatorisch bedingten Asynchronitäten zwischen den einzelnen Arbeitsgängen der Produktionsprozesse aus. Produktionslager sind aufgrund ihrer Pufferfunktion typischerweise Kurzeitlager mit relativ niedrigen Lagerbeständen. Die Notwendigkeit und der Umfang der

Zwischenlager werden weitgehend von der Flexibilität des Produktionsapparates bestimmt. Diese ist abhängig von der Einsetzbarkeit der Produktionsmittel, den Losgrößen, den Umrüstzeiten, der Intelligenz, Anpassungs- und Reaktionsfähigkeit der Produktionssteuerung sowie dem Automatisierungsgrad der Produktion. Durch Flexibilisierung und Automatisierung der Produktion können die Lagerbestände in den Zwischenlagern reduziert und gewisse Zwischenlager möglicherweise gänzlich eliminiert werden.

Distributionslager

Distributionslager (bzw. Handelslager) speichern die Erzeugnisse eines Unternehmens für die unternehmensübergreifende Warenverteilung. Ihre Funktion ist im wesentlichen, Differenzen zwischen Produktionsangebot und Verbrauchernachfrage auszugleichen. Spezielle Distributionslager mit weiteren Funktionen sind Saisonlager, Sortimentslager und Umschlagslager. Saisonlager werden benötigt, wenn Fertigstellungs- und Verbrauchszeitpunkt der Endprodukte erhebliche, unvermeidbare zeitliche Verschiebungen aufweisen. Sortimentslager sind dazu bestimmt, die Sortimente der Produktion in bedarfsgerechte Handelssortimente umzuwandeln. Umschlagslager sind erforderlich, wenn bei der Distribution ein Wechsel der Güter von einem Transportsystem zu einem andern notwendig ist.

Lagerung ist stets mit Kosten verbunden. Einerseits entstehen Kosten aus der Erstellung und dem Betrieb der Lagersysteme und andererseits aufgrund der Kapitalbindung durch die Lagerbestände. Typischerweise bildet in einem Produktionsunternehmen die Lagerung einen wesentlichen Kostenfaktor. Seit geraumer Zeit wird immer deutlicher erkannt, dass für ein Unternehmen im Bereich der Lagerung oftmals ein grosses Sparpotential vorhanden ist. Neue Produktionsstrategien im Sinne der Just-in-Time-Philosophie, welche darauf abzielen, die Lagerbestände möglichst klein zu halten oder sogar zu eliminieren, setzen sich immer mehr durch. Eine Verminderung der Lagerbestände bedeutet einerseits weniger Kosten aufgrund von Kapitalbindung und Lagerbetrieb, und erlaubt andererseits eine Verringerung der Durchlaufzeiten, wodurch schneller auf die Anforderungen des Absatzmarktes reagiert werden kann.

Aus diesen grundlegenden Veränderungen in der Produktionsphilosophie entstehen neue Anforderungen an die Lagersysteme. Die Lagerbestände der einzelnen Artikel werden kleiner, aber die Anzahl der verschiedenen Artikel nimmt häufig aufgrund der Produktdiversifizierung zu. Die Ein- und Auslagerungsaufträge werden typischerweise immer kleiner, aber dafür um so häufiger. Die Lagerzugriffe sollten möglichst schnell und störungsfrei abgearbeitet werden. Die Lager sollten möglichst platzsparend erstellt und kostengünstig betrieben werden. Technische Fortschritte ermöglichen einen stets wachsenden Automatisierungsgrad der Lager. Eine effiziente Lagerverwaltung setzt voraus, dass Lager nicht als isolierte Funktionseinheiten, sondern als integrierte Komponenten der gesamten Unternehmenslogistik betrachtet werden.

1.2. Lagersysteme

In diesem Abschnitt wird der Begriff des Lagers oder Lagersystems geklärt, und es werden die funktionalen Komponenten eines Lagersystems erläutert. Darüber hinaus werden einige grundlegende Begriffe aus der Lagertechnik eingeführt und erklärt. Weitere Begriffsklärungen finden sich beispielsweise in [Vogt, G. (1996)] und [Schulze, L. (1990)].

Unter Lagerung wird der geplante und kontrollierte zeitliche Aufenthalt von Objekten an einem bestimmten Ort verstanden. Die Gesamtheit der technischen und organisatorischen Einrichtungen, welche dem Zweck der Lagerung dienen, wird als Lager oder Lagersystem bezeichnet.

1.2.1. Lagergut

Die in einem Lager gelagerten Objekte bilden das Lagergut. Je nach Aggregatzustand unterscheidet man festes, flüssiges und gasförmiges Lagergut. Falls das Lagergut unterteilt ist in diskrete Objekte, welche technisch als Einheiten gehandhabt und/oder organisatorisch als Einheiten identifiziert und verwaltet werden, spricht man von Stückgut, andernfalls je nach Form des Lagerguts von Schüttgut oder Fliessgut. Bei Stückgutlagern wird bisweilen unterschieden zwischen Lagereinheit, Lade- bzw. Transporteinheit, Einlagerungs- und Auslagerungseinheit. Die Lagereinheit ist die Einheit, welche einen einzelnen Lagerplatz belegt und organisatorisch bei der Lagerplatzverwaltung als elementares Objekt verwaltet wird. Die Lade- bzw. Transporteinheit entspricht der Einheit, welche von den lagerinternen Transportmitteln bei den Lade- und Transportvorgängen als Einzellast behandelt wird. Die Ein- bzw. Auslagerungseinheit ist die Einheit, welche beim Eintritt in bzw. Austritt aus dem Lager als einzelnes Objekt registriert und verwaltet wird. Oft sind diese beiden Einheiten identisch und werden kurz als Lagereinheiten bezeichnet.

Zur besseren Handhabung des Lagerguts werden häufig Lagergeräte (bzw. Lagermittel, Lagerhilfsmittel) eingesetzt. Gängige Lagergeräte sind Paletten, Rollpaletten, Behälter, Gitterboxen, Kästen und Kassetten. Lagergeräte nehmen eine gewisse Menge Lagergut auf, um es handlicher zu machen für den Transport und die Lagerung. Typischerweise bildet ein Lagergerät zusammen mit dem aufgenommenen Lagergut eine Lagereinheit bzw. eine Transport-, Ein- oder Auslagerungseinheit.

Das Lagergut setzt sich üblicherweise aus verschiedenen Typen von Objekten zusammen. Diese Typen werden als Artikel bezeichnet. Das Sortiment (bzw. die Artikelmenge) entspricht der Menge aller im Lager vorhandenen Artikel.

Bei Betrachtung der Funktionsweise von Lagern muss zwischen zwei verschiedenen Systemen unterschieden werden: Einheitenlager und Kommissionierlager.

Einheitenlager

Bei einem Einheitenlager wird das Lagergut innerhalb des Lagers nicht verändert oder umstrukturiert. Die Einlagerungs-, Lager- und Auslagerungseinheiten sind identisch.

Kommissionierlager

Bei einem Kommissionierlager entsprechen die Auslagerungseinheiten nicht den Einlagerungseinheiten. Innerhalb des Lagers wird eine Umstrukturierung vorgenommen, indem den einzelnen Lagereinheiten Teilmengen entnommen werden und diese aufgrund der Kommissionieraufträge neu zusammengestellt werden zu auftragsbezogenen Auslagerungseinheiten.

Unter dem Begriff Kommissionieren wird gemäss VDI Richtlinie 3590 das Zusammenstellen von bestimmten Teilmengen aus einer bereitgestellten Gesamtmenge aufgrund von Bedarfsinformationen (Aufträgen) verstanden. In diesem Sinne erfüllt jedes Lager die Funktion des Kommissionierens. Der Unterschied zwischen Einheiten- und Kommissionierlagern ergibt sich aus der Art des Kommissionierens. Falls bei der Zusammenstellung der Aufträge nur komplette Lagereinheiten entnommen werden, spricht man von Einheitenlager, falls einzelne Lagereinheiten angebrochen und Teilmengen davon entnommen werden, von Kommissionierlager.

Häufig besteht ein Lagersystem aus zwei räumlich und funktional getrennten Teilbereichen, von denen einer ein Kommissionierlager und der andere ein Einheitenlager darstellt. Im Kommissionierlager befinden sich einige wenige Lagereinheiten der benötigten Artikel, welche bei der Zusammenstellung der Aufträge angebrochen werden. Das Einheitenlager dient als Reservelager, aus welchem bei Bedarf neue, komplette Lagereinheiten an das Kommissionierlager übergeben werden. In diesem Sinne ist die Unterscheidung zwischen Kommissionier- und Einheitenlager häufig auch eine Frage der Systemabgrenzung.

Beim Kommissionieren werden zwei Strategien unterschieden, welche mit den altertümlich anmutenden Begriffen "Mann zur Ware" und "Ware zum Mann" bezeichnet werden. Im ersten Fall bewegen sich die Kommissionierer durch das Lager zu den einzelnen Lagereinheiten und stellen so die Aufträge zusammen, im zweiten Fall werden die benötigten Lagereinheiten an Kommissionierplätze transportiert, wo die Auftragszusammenstellung stattfindet.

In den folgenden Ausführungen werden ausschliesslich Einheitenlager betrachtet, und es wird darauf verzichtet, auf die spezifischen Problemstellungen im Zusammenhang mit Kommissionierlagern einzugehen.

Für die systematische Beschreibung eines Lagers ist es hilfreich, das Lager in drei funktionale Hauptkomponenten zu unterteilen: Lagereinrichtungen, Transporteinrichtungen und Steuerung. Diese drei Teilsysteme werden in den folgenden Abschnitten näher betrachtet. Weiter wird auch auf die Schnittstelle des Lagers zum umgebenden System, den sogenannten Warenein- und -ausgang, eingegangen.

1.2.2. Lagereinrichtungen

Funktional gesehen stellen die Lagereinrichtungen die Menge aller Lagerplätze dar, welche zur Aufnahme der Lagereinheiten bestimmt sind. Technisch gesehen umfassen sie alle baulichen und konstruktiven Einrichtungen, durch welche diese Lagerplätze realisiert werden.

Grundsätzlich wird bei Stückgutlagern unterschieden zwischen Bodenlagerung (bzw. Stapellagerung) und Regallagerung. Bei der Bodenlagerung werden die Lagereinheiten ohne Lagereinrichtung auf dem Boden platziert, wobei oft mehrere Lagereinheiten - eventuell mit Hilfe eines Lagergeräts - aufeinander gestapelt werden. Bei der Regallagerung sind Lagereinrichtungen in Form von Regalen vorhanden, auf welchen die Lagereinheiten platziert werden können.

Unter dem Begriff Regal (bzw. Lagergestell) wird allgemein eine Konstruktion verstanden, welche eine Menge von aneinander angrenzenden Lagerplätzen bereitstellt (vgl. [Cormier, G., Gunn, E. A. (1992)] und [Vogt, G. (1996)]). In der Praxis existiert eine Vielzahl von Regalkonstruktionen, welche sich unter anderem unterscheiden bezüglich der konstruktiven Ausführung, der Art der Lagereinheiten, der verwendeten Lagergeräte, den Dimensionen und der Zugriffsart. Als einige der gebräuchlichsten Formen seien hier erwähnt: Palettenregale (Einplatz oder Mehrplatzsystem), Fachbodenregale, Behälterregale (Kassettenregale), Kragarmregale und Wabenregale. Eine ausführliche Beschreibung diverser Regalkonstruktionen ist zu finden in [Vogt, G. (1996)].

Im engeren Sinne wird der Begriff Regal meistens für Lagergestelle verwendet, bei denen Direktzugriff auf jede Lagereinheit besteht. In diesem Fall werden die Lagereinheiten nebeneinander auf dem Lagergestell angeordnet, wobei weder mehrere Lagereinheiten aufeinander gestapelt noch hintereinander aufgereiht werden. Die entsprechenden Lager werden als Zeilenregallager bezeichnet. Ein typisches Beispiel dieses Lagertyps sind die weit verbreiteten Hochregallager. Falls im Gegensatz dazu mehrere Lagereinheiten hintereinander angeordnet werden, spricht man von Blocklagern, Kompaktlagern oder Kanallagern. In diesem Falle werden die einzelnen Lagereinrichtungen, in welchen die Lagereinheiten hintereinander angeordnet werden, als Lagerkanäle (bzw. Lagerkorridore) bezeichnet. In einem Lagerkanal ist der Direktzugriff nur auf die vorderste Lagereinheit möglich, und der Zugriff auf dahinter liegende Lagereinheiten bedingt die Umlagerung der davor platzierten Einheiten. In Abschnitt 1.3. werden die verschiedenen Lagertypen im einzelnen diskutiert.

Damit auf die gelagerten Einheiten zugegriffen werden kann, muss in einem Lager Platz freigehalten werden, welcher den Zugang zu den verschiedenen Lagerplätzen gewährt. Diese Zugangswege werden als Lagergassen (bzw. Regalbediengassen) bezeichnet oder entsprechend der räumlichen Anordnung beispielsweise als Quergänge.

In einem weiteren Sinne kann auch die gesamte Gebäudekonstruktion eines Lagers zu den Lagereinrichtungen gezählt werden. Gebräuchliche Gebäudeformen sind Flachbau, Hochflachbau, Einzweckhochbau, Geschossbau, Traglufthalle sowie Lager im Freien ohne Gebäude (für weitere Details siehe beispielsweise [Jansen, R. (1990/91)]).

Eingangs- und Ausgangspunkte

Die Schnittstelle eines Lagers zum umgebenden System wird gebildet durch sogenannte Ein- und Ausgangspunkte (bzw. Annahme- und Abgabepunkte, Ein- und Auslagerungspunkte). An einem Eingangspunkt werden die Lagereinheiten vom umgebenden System in das Lager aufgenommen und an einem Ausgangspunkt an das umgebende System abgegeben. Ein Lager kann mehrere Ein- und Ausgangspunkte haben. Die Abgrenzung zwischen dem Lagersystem und dem Umsystem ist nicht immer klar vorgegeben. Sie hängt einerseits ab von den Lager- und Transporteinrichtungen, und andererseits davon, welche Transportprozesse in die Logik der Lagersteuerung miteinbezogen sind und welche ausgegrenzt werden.

1.2.3. Transporteinrichtungen

Neben der Funktion des Lagerns, realisiert durch die Lagereinrichtungen, hat ein Lagersystem auch Transportfunktionen. Die dafür vorgesehenen Transporteinrichtungen (bzw. Fördereinrichtungen) umfassen alle technischen Einrichtungen in einem Lager, welche dem Transport von Lagereinheiten innerhalb des Lagers dienen.

Es sei darauf hingewiesen, dass im Rahmen dieser Arbeit die Begriffe des Transportierens und Förderns gleichbedeutend verwendet werden. Bisweilen wird eine Unterscheidung gemacht, wobei Fördern eher im innerbetrieblichen und Transportieren im zwischenbetrieblichen Kontext verwendet wird (vgl. [Favarger, M., Bamert, A., Rupper, P. (1982)]).

Transportfunktionen

Prinzipiell können in einem Lager drei elementare Transportfunktionen (bzw. Lagerbewegungen, Lagerprozessfunktionen) unterschieden werden: Einlagerung, Auslagerung und Umlagerung. Bei einer Einlagerung übernimmt das Lagersystem eine Lagereinheit an einer bestimmten Stelle (Einlagerungspunkt, Eingangspunkt) vom umgebenden System, transportiert sie an einen freien Lagerplatz und stellt sie dort ab. Bei einer Auslagerung wird eine Lagereinheit an ihrem Lagerplatz aufgenommen, zu einer bestimmten Stelle (Auslagerungspunkt, Ausgangspunkt) transportiert und dort an das umgebende System abgegeben. Eine Umlagerung entspricht dem Transport einer Lagereinheit innerhalb des Lagers von einem Lagerplatz zu einem andern Lagerplatz.

Zusätzlich zu den erwähnten Transportfunktionen wird in der Lagertechnik häufig der Begriff des Lagerspiels verwendet. Ein Lagerspiel entspricht der Kombination einer Einlagerung und einer daran anschliessenden Auslagerung. In einem Lagerspiel übernimmt das Transportmittel eine Lagereinheit an einem Eingangspunkt, bewegt sich zu einem Lagerplatz, lädt die Last ab, bewegt sich anschliessend zu einem andern Lagerplatz, lädt die neue Lagereinheit auf, bewegt sich zu einem Ausgangspunkt und übergibt die Last. Das Konzept der Lagerspiele wird oft verwendet im Zusammenhang mit der Leistungsabschätzung von Lagersystemen (siehe Kapitel 3).

Uebersicht über Transporteinrichtungen

Für die lagerinterne Förderung existiert eine fast unüberschaubare Vielzahl von verschiedenen Transporteinrichtungen. Zusätzlich werden entsprechend der technischen Entwicklung laufend neue Konstruktionen entwickelt. Der folgende Abschnitt soll einen Ueberblick geben über die wichtigsten Transporteinrichtungen. Ziel dabei ist weniger der Anspruch auf Vollständigkeit als das Aufzeigen verschiedener Kriterien, nach welchen Transporteinrichtungen klassifiziert werden können. Weitere Informationen über Transporteinrichtungen finden sich beispielsweise in [Budde, R., Schwarz, W. (1988)], [Martin, H. (1990)] und [Vogt, G. (1996)].

Die Art der Transporteinrichtungen eines Lagers steht in engem Zusammenhang mit dem Typ des Lagerguts (z.B. Stückgut), den verwendeten Lagergeräten (z.B. Paletten) und der Art der Lagereinrichtungen (z.B. Hochregale). Im folgenden werden vor allem Stückgutlager betrachtet.

Eine erste Unterscheidung bei den Transporteinrichtungen wird gemacht zwischen Stetigförderern und Unstetigförderern. Beispiele für Stetigförderer sind Fließbänder und Rollenbahnen. Unstetigförderer lassen sich in Bezug auf ihre technischen und funktionalen Eigenschaften nach verschiedenen Kriterien einteilen. Nachfolgend werden einige Merkmale aufgelistet, welche in der Lagertechnik traditionellerweise zur Klassifizierung der Unstetigförderer angewendet werden:

- Förderer für Bodenlagerung und Förderer für Regallager
- flurfreie und flurgebundene Förderer
- gleislose und gleisgebundene Förderer
- regalunabhängige und regalgebundene Förderer

Als Förderer für Bodenlagerung kommen vorwiegend Kranen und Stapler zum Einsatz. Förderer für Regallager sind im wesentlichen Kranen, Stapler, Förderwagen, Hochregalstapler und Regalbediengeräte in diversen Ausführungen.

Flurfreie Förderer sind Einrichtungen, welche zur Fortbewegung nicht an den Boden gebunden sind. Es handelt sich im allgemeinen um Kranen verschiedener Bauart (Stapelkranen, Mobilkranen, etc.). Flurgebundene Förderer (auch Flurförderzeuge genannt) können unterteilt werden in gleislose und gleisgebundene Einrichtungen.

Gleislose Flurförderzeuge sind typischerweise Stapler, welche in diversen Bauarten mit unterschiedlichen Funktionalitäten existieren (Gabelstapler, Front- und Seitenstapler, Hochregalstapler, Mehrpaletten-Gabelstapler, etc.). Gleisgebundene Förderer sind einerseits Regalbediengeräte verschiedenster Art und andererseits Förderwagen, wie sie beispielsweise in automatisierten Kompaktlagern eingesetzt werden.

Regalgebundene Förderer können im Gegensatz zu regalunabhängigen Förderern nur einzelne Regale bedienen. Meist sind sie technisch mit der Regalkonstruktion verbunden, wie beispielsweise die weit verbreiteten Regalbediengeräte (RBG) bei Hochregallagern oder Hubbalkenkon-

strukturen. Gassengebundene Förderer sind technisch an eine bestimmte Lagergasse gebunden und bedienen die Regale auf beiden Seiten der Gasse.

Die aufgeführten Unterscheidungskriterien entstammen der traditionellen Lagertechnik und basieren im wesentlichen auf technischen Merkmalen der Fördereinrichtungen. Mit dem technologischen Fortschritt und den damit verbundenen Automatisierungsmöglichkeiten in der Lagertechnik gewann der Einsatz von automatisierten Lagern in neuerer Zeit immer mehr an Bedeutung. Es wurde eine Vielzahl von neuartigen, vollautomatisierten Lagersystemen entwickelt, deren Transportsysteme computergesteuert sind und technologische Merkmale aufweisen, welche in den traditionellen Kategorien nicht erfasst werden. Zeitgleich mit dieser Automatisierungstendenz wuchs das Interesse an platzsparenden Alternativen zu den herkömmlichen Zeilenregallagern (z.B. Hochregallager). Dies gilt speziell für die hochindustrialisierten Ländern Europas, wo der Boden eine zunehmend knappe und kostenintensive Ressource darstellt. Aus diesem Grund wuchs in der Lagertechnik der Trend zur Kompaktlagerung, und es entstanden neuartige Lagersysteme, welche die Automatisierungsmöglichkeiten mit den Vorteilen der kompakten Lagerung verbinden. Diese Lagertypen werden unter dem Begriff automatisierte Kompaktlager oder automatisierte Blocklager zusammengefasst. Beispiele dafür sind dynamische Blocklager, Kanal- oder Tunnellager und Satellitenlager. Automatisierte Kompaktlager werden in Abschnitt 1.3. näher besprochen.

Merkmale automatisierter Transportsysteme

Vorangehend wurden einige Merkmale aufgeführt, nach welchen sich Transporteinrichtungen in Lagern kategorisieren lassen. Weiter wurde erwähnt, dass in jüngerer Zeit diverse neue Lagerkonzepte entworfen wurden, insbesondere verschiedene Typen von automatisierten Kompaktlagern. Die Transporteinrichtungen dieser Lager unterscheiden sich teilweise wesentlich von den herkömmlichen Systemen, sowohl was die technischen Installationen betrifft, als auch in Bezug auf die automatisierte, computerbasierte Steuerung. Dieser Abschnitt versteht sich als Beitrag zur Charakterisierung automatisierter Transporteinrichtungen. Es werden gewisse Eigenschaften solcher Systeme herauskristallisiert, welche speziell beim Entwurf der Steuerungslogik eine wesentliche Rolle spielen.

Zur Charakterisierung der Transporteinrichtungen automatisierter Lager (und insbesondere automatisierter Kompaktlager) werden neben den traditionellen Kategorien folgende Merkmale vorgeschlagen:

- Anzahl Stufen des Transportsystems
- Transfer zwischen den Transportmitteln
- Wählbarkeit der Transportmittel
- Wählbarkeit der Transportroute
- Ladekapazität der Transportmittel

Diese Eigenschaften sowie ihre Implikationen auf die Steuerungslogik werden im folgenden erläutert.

Anzahl Stufen des Transportsystems

Traditionelle Lagersysteme haben typischerweise ein einstufiges Transportsystem. Das bedeutet, dass eine elementare Lagerbewegung (d.h. eine Ein-, Aus- oder Umlagerung) von einem einzigen Transportmittel ausgeführt wird. Dies ist beispielsweise der Fall bei nicht-automatisierten Lagern, welche von personengesteuerten Staplern bedient werden, oder bei automatisierten Hochregallagern, welche mit Regalbediengeräten ausgestattet sind.

Im Gegensatz dazu haben automatisierte Kompaktlager häufig ein mehrstufiges Transportsystem. Als Beispiel seien die dynamischen Blocklager angeführt, welche in dieser Arbeit untersucht werden (vgl. Kapitel 2). Das Transportsystem eines dynamischen Blocklagers umfasst Förderwagen in allen drei räumlichen Dimensionen: Aufzüge (Senkrechtförderer), Quergangwagen und Korridorwagen. Für die Ausführung einer Ein- oder Auslagerung werden alle drei Typen von Transportmitteln benötigt. Bei einer Auslagerung beispielsweise wird eine Lagereinheit zunächst von einem Korridorwagen, dann von einem Quergangwagen und schliesslich von einem Aufzug transportiert. In mehrstufigen Transportsystem erfolgt ein Lasttransfer zwischen verschiedenen Transportmitteln. Damit ein solcher stattfinden kann, müssen die Bewegungen der verschiedenen Transportmittel aufeinander abgestimmt werden, was zu entsprechenden Bedingungen in der Steuerungslogik führt.

Transfer zwischen den Transportmitteln

Der Transfer zwischen den verschiedenen Transportmitteln eines mehrstufigen Transportsystems kann auf verschiedene Weise erfolgen. Ein wichtiges Unterscheidungsmerkmal ist, ob der Transfer synchron oder asynchron geschieht. Bei synchronem Transfer müssen sich die beiden kommunizierenden Transportmittel gleichzeitig für eine bestimmte Zeitspanne an dem Ort befinden, wo die Lastübergabe stattfindet. Dies ist beispielsweise der Fall in dynamischen Blocklagern. Bei asynchronem Transfer kann eine Last übergeben werden, ohne dass die Bewegungen der beteiligten Transportmittel aufeinander abgestimmt werden müssen. Asynchroner Transfer kann realisiert werden, indem zwischen zwei kommunizierenden Transportmitteln ein Puffer geschaltet wird, in welchem die Lagereinheiten kurzzeitig gelagert werden können. Asynchroner Transfer liegt auch vor, wenn das übernehmende Transportmittel ein Stetigförderer ist, welcher dauernd in einem Zustand ist, der die Uebernahme einer Last ermöglicht. Dies ist beispielsweise der Fall bei Durchlaufslagern (bzw. Fließslagern), welche eine Form von dynamischen Blocklagern darstellen, bei welchen die Kanalförderer mittels Rollenbahnen realisiert sind.

Allgemein lässt sich sagen, dass speziell der synchrone Lasttransfer und die damit verbundene Abgleichung der Transportmittelbewegungen einen neuen Planungsaspekt in der Lagersteuerung darstellt, welcher die Komplexität der Steuerungslogik wesentlich erhöhen kann.

Wählbarkeit der Transportmittel

In vielen Lagern ist die Wahl der Transportmittel für eine elementare Lagerbewegung (d.h. eine Ein-, Aus- oder Umlagerung) fest vorgegeben. Betrachte beispielsweise ein Hochregallager, das mit regalgebundenen Bediengeräten ausgestattet ist. In diesem Fall ist jedem Lagerplatz (und jedem Ein- bzw. Ausgangspunkt) genau ein Regalbediengerät zugeordnet, welches diesen Platz bedient. Bei vorgegebenen Lagerbewegungen ist eindeutig bestimmt, welche Transportmittel zur Ausführung verwendet werden.

Im Gegensatz dazu besteht bei gewissen Lagersystemen eine Wahlmöglichkeit für die Transportmittel. Es gibt beispielsweise Satellitenlager, bei welchen von einer Lagergasse aus mehrere funkgesteuerte Satelliten eingesetzt werden für den Transport innerhalb der Lagerkanäle. In diesem Fall muss eine Wahl getroffen werden, welcher Satellit wann in welchem Kanal eingesetzt wird. Auch in dynamischen Blocklagern besteht eine gewisse Wahlmöglichkeit für die Transportmittel. Es ist beispielsweise möglich, dass für eine Umlagerung mehrere Transportrouten existieren, auf welchen die Lagereinheit von der Startposition an die Zielposition befördert werden kann. Jede mögliche Route führt zu einer unterschiedlichen Auswahl von beteiligten Förderwagen. Die Wählbarkeit der Transportmittel führt zu einer zusätzlichen Entscheidungsdimension, welche von der Lagersteuerung bewältigt werden muss.

Wählbarkeit der Transportroute

Im vorangehenden Abschnitt wurde darauf hingewiesen, dass in gewissen Lagersystemen eine Wahlmöglichkeit besteht für die Transportroute einer Lagerbewegung. Bei der Wählbarkeit der Transportroute können zwei Fälle unterschieden werden.

Im ersten Fall werden Lagerbewegungen (d.h. Ein-, Aus- und Umlagerungen) betrachtet, deren Start- und Zielort vorgegeben ist. Es gibt Lagersysteme, bei welchen aufgrund der Anordnung der Transporteinrichtungen verschiedene Transportwege zwischen einem vorgegebenen Start- und Zielort existieren. In diesem Fall besteht eine Wahlmöglichkeit für die Transportroute zwischen den vorgegeben Punkten.

Der zweite Fall bezieht sich auf Lagerbewegungen, für welche der Startpunkt und/oder der Zielpunkt nicht vorgegeben ist. Dies ist häufig der Fall bei der Steuerung von Lagersystemen. Typischerweise wird dem Steuersystem als Input eine Menge von Lagerbewegungen (ein sogenannter Lagerauftrag) vorgegeben, welche vom Transportsystem auszuführen sind. Bei Einlagerungen wird normalerweise der Eingangspunkt vorgegeben, an welchem die Lagereinheit abgeholt werden muss, der Zielort jedoch, d.h. der Lagerplatz, wo die Einheit eingelagert wird, ist häufig nicht spezifiziert und wird vom Steuersystem bestimmt. Bei Auslagerungen wird üblicherweise der Artikel und eventuell der Ausgangspunkt vorgegeben, nicht jedoch die spezifische Lagereinheit des Artikels, welche ausgelagert werden soll. Die Wahl einer konkreten Lagereinheit ist wiederum Aufgabe des Steuerungssystems. Somit besteht bei der Steuerung von Lagersystemen im allgemeinen eine Wahlmöglichkeit bezüglich der Start- und/oder Zielpositionen der Lagerbewe-

gungen. Diese Art der Routenwählbarkeit ist eine Konsequenz des Steuerungskonzepts und hängt nicht von der physischen Auslegung der Transporteinrichtungen ab. Sie existiert prinzipiell in jedem Lager und bildet eine Entscheidungsdimension auf der Ebene des Steuersystems. In Abschnitt 1.4. wird auf diese Thematik weiter eingegangen im Zusammenhang mit der Steuerung von Lagersystemen.

Ladekapazität der Transportmittel

Häufig sind die Transportmittel eines Lagersystems so ausgelegt, dass eine einzelne Lagereinheit pro Fahrt transportiert wird. Dies ist beispielsweise der Fall bei Hubstaplern, Hochregalstaplern und Regalbediengeräten. Es existieren jedoch Lagersysteme mit Transportmitteln, welche eine grössere Ladekapazität haben und mehrere Lagereinheiten gleichzeitig transportieren können. Als Beispiel seien Mehrpalettenstapler, Mehrpaletten-Aufzüge und Mehrpaletten-Regalbediengeräte erwähnt. Falls ein Transportmittel mehrere Lagereinheiten gleichzeitig transportieren kann, entsteht eine neue Entscheidungsdimension für das Steuersystem. Es muss festgelegt werden, welche Lagereinheiten in welcher Reihenfolge für eine Transportfahrt zusammengefasst werden. Damit verbunden ist ein Routenproblem, da ein Sammeltransport auf mehrere Arten zusammengestellt werden kann, wobei jede Möglichkeit zu einer unterschiedlichen Fahrtroute des Transportmittels führt.

Ziel dieses Abschnittes war, zu zeigen, dass in modernen Lagersystemen und speziell in automatisierten Kompaktlagern Transporteinrichtungen eingesetzt werden, deren Steuerung wesentlich komplexer ist als bei traditionellen Lagern. Es wurden verschiedene Merkmale hervorgehoben, durch welche sich diese Lagertypen von den traditionellen Systemen unterscheiden. Weiter wurde gezeigt, welchen Einfluss diese Eigenschaften auf die Entscheidungsdimensionen und die Komplexität des Steuersystems haben.

1.3. Typen von Lagersystemen

In diesem Abschnitt wird eine Übersicht über die gebräuchlichsten Lagertypen gegeben. Es werden verschiedene Unterscheidungskriterien aufgeführt, nach denen die Vielfalt der in Praxis verwendeten Lagertypen systematisch beschrieben und klassifiziert werden können. Die betrachteten Merkmale beziehen sich auf die technischen Systemkomponenten (Lagereinrichtung und Transportsystem) und nicht auf die logistische Funktion der Lager innerhalb der Materialflussskette eines Unternehmens.

Die vorgeschlagene Klassifikationssystematik ist speziell aus dem Blickpunkt der Lagersteuerung konzipiert. Hauptkriterien sind die Lageranordnung und die Lagereinrichtungen sowie die

eingesetzten Fördersysteme und Bedienprinzipien. Ein spezielles Gewicht wird den Kompaktlagern und insbesondere den dynamischen Blockregallagern zugewiesen.

Eine grossangelegte Übersicht über die häufigsten Lagertypen ist in [Vogt, G. (1996)] zu finden, wobei die Klassifikation der verschiedenen Lagersysteme von einem leicht anderen Gesichtspunkt aus erfolgt und von der hier vorgeschlagenen abweicht. Jeder Lagertyp wird ausführlich beschrieben, mit Abbildungen dokumentiert, und es werden viele technische Details erwähnt, welche für den Praktiker hilfreich sind. Weitere Lagerklassifikationen finden sich beispielsweise in [Martin, H. (1990)], [Schulze, L. (1990)], [Schulze, L. (1995)], [Jansen, R. (1990/91)], [Pörsch, M. (1988)] und [Beschaffung (1991a)].

Bei der Lageranordnung muss prinzipiell unterschieden werden zwischen Zeilenlagern und Blocklagern (bzw. Kompaktlagern). Dieser Unterschied wird im folgenden näher erläutert.

Zeilenlager und Blocklager

Die Unterscheidung zwischen Zeilen- und Blocklagern basiert auf der räumlichen Anordnung der Lagerplätze und der dazwischenliegenden Lagergassen.

In Zeilenlagern (bzw. Reihenlagern) sind die Lagerplätze in ein- oder doppelreihigen parallelen Lagerzeilen angeordnet, zwischen welchen jeweils eine Lagergasse liegt. Charakteristisch für Zeilenlagern ist, dass nicht mehrere Lagereinheiten hintereinander abgestellt werden. Falls keine Lagereinheiten aufeinandergestapelt werden, besteht die Möglichkeit des Direktzugriffs auf jede Lagereinheit. Die untenstehende Abbildung zeigt schematisch die Struktur eines Zeilenlagers.

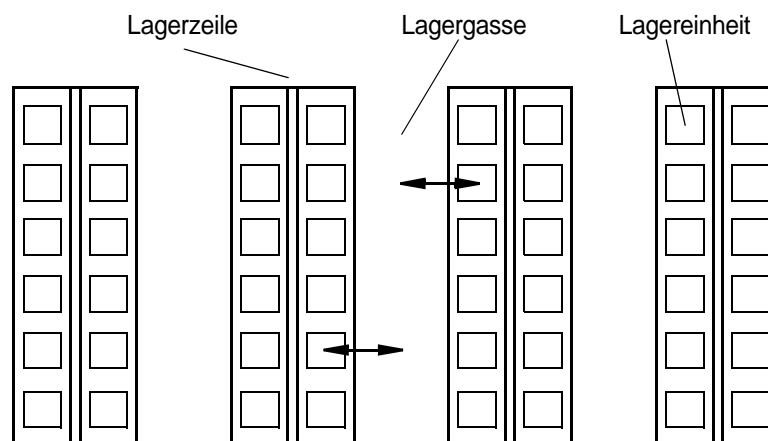


Abbildung 1.1. *Struktur eines Zeilenlagers*

In Blocklagern (bzw. Kompaktlagern) werden im Gegensatz dazu mehrere Lagereinheiten hintereinander angeordnet. Ein Blocklager setzt sich zusammen aus einzelnen Lagerblöcken, welche von einander abgetrennt sind durch dazwischenliegende Lagergassen. Eine Kolonne von hinter-

einanderliegenden Lagereinheiten (bzw. die Lagereinrichtung, in welcher diese Lagereinheiten platziert werden) wird als Lagerkanal bezeichnet. Charakteristisch für Blocklager ist, dass nur Direktzugriff auf die vordersten Lagereinheiten in einem Lagerkanal besteht. Falls mehrere Lagereinheiten aufeinandergestapelt werden, besteht nur Zugriff auf die vordersten, obersten Lagereinheiten. Wenn Lagereinheiten bewegt werden sollen, auf welche kein Direktzugriff besteht, müssen zuvor gewisse Umlagerungen ausgeführt werden.

Blocklager bilden eine platzsparende Alternative zu Zeilenregallagern, da die Zahl der Lagergasen wesentlich kleiner ist und somit weniger Platz für die Zugangswege zu den Lagerplätzen freigehalten werden muss. Als Nachteil resultiert daraus, dass nicht auf alle Lagereinheiten direkt zugegriffen werden kann und zusätzliche Umlagerungen notwendig sind. Durch diese zusätzlichen Transportbewegungen wird der Lagerprozess tendenziell verlangsamt und die Komplexität der Steuerung erhöht sich. Daraus wird ersichtlich, dass gerade in automatisierten Blocklagern ein Bedarf besteht, die tendenzielle Verlangsamung der Lagerzugriffe und den damit verbundenen Leistungsverlust durch eine Optimierung der Steuerungslogik auszugleichen.

Die folgende Abbildung zeigt schematisch die Struktur eines Blocklagers.

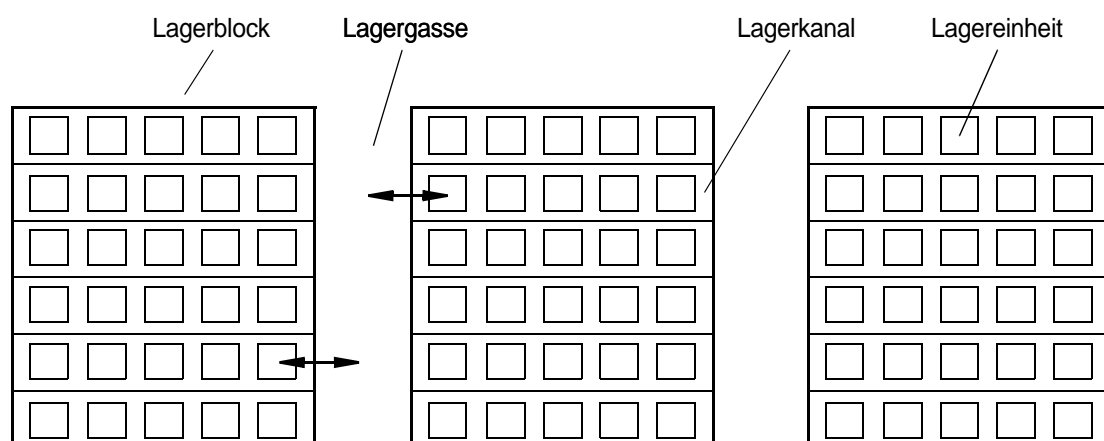


Abbildung 1.2. Struktur eines Blocklagers (bzw. Kompaktlagers)

Stapellager und Regallager

Als zweites Kriterium bei der Klassifizierung der Lagertypen wird unterschieden zwischen Stapel- und Regallagern. Diese Unterscheidung bezieht sich auf die Art der Lagereinrichtungen.

Stapellager (bzw. Bodenlager) verwenden keine oder nur sehr einfache Lagereinrichtungen. Die Lagereinheiten werden direkt auf dem Boden gelagert, wobei häufig mehrere Lagereinheiten aufeinandergestapelt werden. Die Stapelung kann mit oder ohne Lagergeräten erfolgen. Stapellager sind aufgrund der fehlenden Lagereinrichtungen kostengünstige Lagersysteme. Als Nachteil

besteht die Einschränkung, dass nur auf die vordersten, obersten Lagereinheiten eines Stapels zugegriffen werden kann.

In Regallagerlagern werden im Gegensatz dazu Regale eingesetzt, auf welchen die Lagereinheiten platziert werden. Bei entsprechender Anordnung kann mit Hilfe von Regalen ein Direktzugriff auf alle Lagereinheiten realisiert werden, wobei nicht auf das Uebereinanderlagern von Einheiten verzichtet werden muss. Gleichzeitig erlauben Regale den Einsatz von technisch hochstehenden, automatisierten Transporteinrichtungen wie beispielsweise Regalbediengeräten.

1.3.1. Ueberblick über Lagertypen

Die untenstehende Tabelle gibt eine Uebersicht über die gebräuchlichsten Lagertypen. Die Lagersysteme sind einerseits unterteilt in Zeilenlager, Blocklager und andere Lagertypen, und andererseits in Stapellager und Regallager. Bei Blockregallagern wird unterschieden zwischen statischen und dynamischen Lagern. Zu jeder Kategorie werden weitere gängige Bezeichnungen für die entsprechenden Lagertypen und einige Beispiele gegeben.

Anordnung:	Lagereinrichtung:	
	Stapellager	Regallager
Zeilenlager	Zeilenstapellager <i>Weitere Bezeichnungen:</i> Boden- / Flächenlager in Zeilen- / Linien- / Reihenanordnung <i>Beispiele:</i> <ul style="list-style-type: none"> • ohne Lagergerät: Containerstapel, Röhrenstapel, Rollenstapel • mit Lagergerät: Palettenstapel, Gitterboxenstapel 	Zeilenregallager <i>Weitere Bezeichnungen:</i> Hochregallager, Flachregallager, Reihenlager, Linienlager <i>Beispiele:</i> <ul style="list-style-type: none"> • Palettenregal, Einplatzsystem • Palettenregal, Mehrplatzsystem • Fachbodenregal • Behälterregal (Kassettenregal, Kleinteilelager, AKL-System) • Kragarmregal • Wabenregal (Langgutregal)
Blocklager	Blockstapellager <i>Weitere Bezeichnungen:</i> Boden- / Flächenlager in Blockanordnung, Blocklager <i>Beispiele:</i> <ul style="list-style-type: none"> • ohne Lagergerät: Containerstapel, Röhrenstapel, Rollenstapel • mit Lagergerät: Palettenstapel, Gitterboxenstapel 	Statische Blockregallager <i>Weitere Bezeichnungen:</i> Statische Block- / Kompaktlager <i>Beispiele:</i> <ul style="list-style-type: none"> • Einfahrregal (Drive-In-Regal) • Durchfahrregal
		Dynamische Blockregallager <i>Weitere Bezeichnungen:</i> Dynamische / automatische Block- / Kompakt- / Kanal- / Tunnellager <i>Beispiele:</i> <ul style="list-style-type: none"> • Einschubregal • Durchlaufregal (Fliesslager) • Satellitenlager • Kanalfahrzeuglager
Andere Lager		Verschiebe- und Umlaufregallager <i>Beispiele:</i> <ul style="list-style-type: none"> • Horizontal umlaufendes Fachbodenregal (Horizontal-Karussell, Rotary Rack) • Paternosterregal (Umlaufregal, Grossraumschrank) • Verschiebeumlaufregal Mehrgeschossige Regallager <i>Beispiele:</i> <ul style="list-style-type: none"> • Mehrgeschossiges Fachbodenregal (Podestanlage) • Bühnenlager

Abbildung 1.3. Klassifikation der häufigsten Lagertypen

Die in der Tabelle aufgeführten Lagertypen werden im folgenden kurz beschrieben.

Zeilenstapellager

In Zeilenstapellagern sind die Lagerplätze in Zeilen angeordnet und die Lagereinheiten werden ohne (oder mit minimalen) Lagereinrichtungen auf dem Boden gestapelt. Häufig werden mehrere

Lagereinheiten (ohne oder mit Lagergerät) aufeinander gestapelt, so dass jeweils nur die obersten Einheiten einer Lagerzeile direkt zugreifbar sind.

Zeilenregallager

In Zeilenregallagern werden die Lagereinheiten auf Regalen gelagert, welche in Zeilen angeordnet sind. Es besteht üblicherweise Direktzugriff auf jede Lagereinheit. Zeilenregallager gehören zu den am häufigsten verwendeten Lagertypen. Ein bekannter Vertreter sind die weit verbreiteten Hochregallager. Bei Palettenregalen wird unterschieden zwischen Einplatz- und Mehrplatzsystemen. Bei Einplatzsystemen sind die Regalflächen unterteilt in einzelne Regalabteile (bzw. Regalfächer), welche genau eine Palette aufnehmen können. Bei Mehrplatzsystemen werden mehrere Paletten nebeneinander in einem Regalabteil gelagert. Behälterregale sind im Gegensatz zu Palettenregalen für die Aufnahme von Lagergeräten in Form von Behältern verschiedener Art und Grösse konzipiert. In Kragarmregalen werden vor allem Langgüter (z.B. Röhren, Metallstäbe, Armierungseisen, etc.) gelagert, wobei die Lagereinheiten quer auf die hervorstehenden Kragarme gelegt werden. Wabenregale sind ebenfalls primär für Langgüter konzipiert, wobei die Lagereinheiten in diesem Fall der Länge nach in röhrenartige Wabenbehälter eingesteckt werden.

Blockstapellager

In Blockstapellagern sind mehrere Lagerplätze hintereinander in Blöcken angeordnet und die Lagereinheiten werden ohne (oder mit minimalen) Lagereinrichtungen auf dem Boden gestapelt. Häufig werden mehrere Lagereinheiten (ohne oder mit Lagergerät) aufeinander gestapelt, so dass jeweils nur die obersten, vordersten Einheiten eines Lagerblocks (bzw. eines Lagerkanals) direkt zugreifbar sind.

Statische und dynamische Blockregallager

Diese Lagersysteme werden im nächsten Abschnitt separat diskutiert.

Andere Lager

Neben den erwähnten Lagertypen existiert eine Vielzahl von weiteren Lagersystemen, welche hier nicht alle besprochen werden können. Es seien nur zwei relativ verbreitete Kategorien erwähnt: Verschiebe- und Umlauflager sowie mehrgeschossige Regallager. Verschiebe- und Umlauflager zeichnen sich dadurch aus, dass die Regale nicht ortfest installiert, sondern je nach Lagertyp auf verschiedene Weise verschiebbar sind. Es kann unterschieden werden zwischen horizontal verschiebbaren, horizontal umlaufenden und vertikal umlaufenden Regalen (sogenannten Paternosterregalen). In mehrgeschossigen Regallagern sind die Regale auf verschiedenen unabhängigen Lagerebenen (bzw. Stockwerken) angeordnet, welche miteinander verbunden sind (beispielsweise durch Aufzüge, Treppen, etc.).

1.3.2. Zeilen- und Blockregallager

In diesem Abschnitt wird eine Uebersicht über die wichtigsten Zeilen- und Blockregallager gegeben. Bei den Blockregallagern wird unterschieden zwischen statischen und dynamischen Lagern. Dynamische Blockregallager sind im Gegensatz zu statischen Blockregallagern mit speziellen Fördereinrichtungen ausgestattet, welche die Lagereinheiten innerhalb der Kanäle transportieren. Als weiteres Unterscheidungsmerkmal werden Struktur und Anzahl Stufen des Transportsystems untersucht.

Allgemein lassen sich die Transportbewegungen in einem Lager unterteilen in Vertikalförderung und Querfördererung. Vertikalförderung bezieht sich auf Bewegung der Lagereinheiten in der vertikalen Dimension und Querfördererung auf den Transport in den Lagergassen. Als zusätzliche Dimension kommt bei Blocklagern die Förderung innerhalb der Lagerkanäle hinzu. Bei der Vertikal- und Querfördererung wird unterschieden zwischen kombinierter und separater Förderung. Bei kombinierter Förderung werden die Bewegungen in den beiden Dimensionen von selben Transportmittel ausführt, während bei separater Förderung zwei verschiedene Transportmittel eingesetzt werden.

Lager mit kombinierter Vertikal- und Querfördererung haben ein zweistufiges bzw. einstufiges Transportsystem, je nachdem, ob Kanalförderzeuge eingesetzt werden oder nicht. Bei separater Vertikal- und Querfördererung liegt ein drei- bzw. zweistufiges System vor, wiederum in Abhängigkeit vom Einsatz von Kanalförderern. Allgemein lässt sich sagen, dass die Komplexität der Lagersteuerung wächst, je mehr Stufen ein Transportsystem umfasst.

Als Literatur zum Thema der automatisierten Blockregallager seien die folgenden Beiträge erwähnt: [Birkholz, F. (1996)], [Blumenschein, K. (1991)], [Favarger, M., Bamert, A., Rupper, P. (1982)], [FM (1990)], [Gödicke, D. (1991)], [Gremm, F. (1989)], [Hollenstein, E. (1996)], [Kahrman, V. (1989)], [Logistik im Unternehmen (1989)], [Neue Verpackung, (1990)], [Schulte-Zurhausen, M. (1982)] und [Weber, R. (1990)].

Die untenstehende Abbildung zeigt einen Ueberblick über die verschiedenen Zeilen- und Blocklager. Lager mit zweistufigem Transportsystem sind hellgrau und solche mit dreistufigem System dunkelgrau hervorgehoben. Die Lagertypen werden anschliessend im einzelnen diskutiert.

Lagertyp:	Kanalförderer:	Vertikal- und Querförderung:	
		kombiniert: Stapler / Regalbediengerät	separat: unabhängige Vertikal- und Querförderer
Zeilenregallager	-	Beispiele: • Hochregallager mit Staplerbedienung • Hochregallager mit Regalbediengeräten (RBG) • Hochregallager mit Hubbalkenbedienung	Beispiele: • SYSTORE-Lager
statische Blockregallager	keine	Beispiele: • Einfahrlager • Durchfahrlager	-
dynamische Blockregallager	Rollenbahnen / Rolluntersätze angetrieben durch Schwerkraft / Getriebemotoren	Beispiele: • Einschublager • Durchlauf Lager (Fliesslager)	Beispiele: • DYNASTORE-Lager
	Satelliten	Beispiele: • Satellitenlager mit Staplerbedienung • Satellitenlager mit Regalbediengeräten	Beispiele: • Satellitenlager mit Etagenfahrzeugen und Aufzügen
	Kanalfahrzeuge (Kulis)		Beispiele: • ACTIW-Lager

Abbildung 1.4. Uebersicht über Zeilen- und Blockregallager

Zeilenregallager

Einige Eigenschaften und Beispiele von Zeilenregallager wurden schon im vorangehenden Abschnitt besprochen. Die meisten Zeilenregallager haben eine kombinierte Vertikal- und Querförderung. Dies gilt beispielsweise für die weit verbreiteten Hochregallager, welche je nach Ausführung mit Hochregalstaplern, Regalbediengeräten oder Hubbalken bedient werden. Eine Ausnahme stellen die sogenannten SYSTORE-Lager dar, eine moderne Art von Hochregallagern, welche mit separaten Vertikal- und Querförderern bedient werden.

Statische Blockregallager

In statischen Blockregallagern sind die Lagerplätze blockweise auf Regalen angeordnet. Sie unterscheiden sich von dynamischen Blockregallagern dadurch, dass in den Lagerkanälen keine Fördereinrichtungen vorhanden sind. Statische Blockregallager sind üblicherweise nicht automatisiert und werden von personengesteuerten Staplern bedient. Als Lagergeräte werden meistens Paletten verwendet. Es wird unterschieden zwischen Einfahr- und Durchfahrregalen. Bei Einfahrregalen werden die Lagerkanäle nach dem LIFO-Prinzip bedient (last in first out), d.h. die Einlagerungen erfolgen auf der selben Kanalseite wie die Auslagerungen. Bei Durchfahrregalen

sind die Lagerkanäle durchgängig, und die Stapelfahrzeuge können die Lagerkanäle traversieren. Die Kanäle können sowohl nach dem LIFO- als auch nach dem FIFO-Prinzip (first in first out) bedient werden. Beim FIFO-Prinzip werden die Paletten auf der einen Kanalseite eingelagert und auf der anderen Seite ausgelagert.

Dynamische Blockregallager

Dynamische Blocklager zeichnen sich dadurch aus, dass sie über spezielle Fördereinrichtungen verfügen, welche den Transport innerhalb der Lagerkanäle übernehmen. Als Lagergeräte werden meistens Paletten verwendet. Es können drei Arten von Kanalförderern unterschieden werden: Rollenbahnen bzw. Rolluntersätze, Satelliten und Kanalfahrzeuge.

Kanalförderung mit Rollenbahnen oder Rolluntersätzen

Rollenbahnen sind fest installierte Transporteinrichtungen in den Lagerkanälen. Sie sind meistens von der Schwerkraft angetrieben, d.h. die Lagerkanäle besitzen eine leichte Neigung, so dass die Paletten zur tieferliegenden Kanalseite rollen. Es wird unterschieden zwischen Einschublager und Fliesslager.

Bei Einschublager sind die Lagerkanäle in zwei Hälften geteilt, wobei die Kanalmitte erhöht ist und die beiden Kanalenden tiefer liegen. Die Paletten werden beidseitig in die Kanäle eingeschoben. Durch die Schwerkraft sammeln sich die eingelagerten Paletten stets an den beiden Kanalenden. Die Bedienung der Lagerkanäle erfolgt nach dem LIFO-Prinzip (last in first out), da die Paletten auf derselben Kanalseite eingelagert und ausgelagert werden.

In Fliesslager sind die Lagerkanäle durchgängig und nur auf eine Seite geneigt. Die Paletten werden am höheren Kanalende eingelagert und am tieferen Ende entnommen. Die Kanäle werden dementsprechend nach dem FIFO-Prinzip bedient, d.h. die Paletten fließen von einer Kanalseite zur anderen.

An Stelle von Rollenbahnen können auch Rolluntersätze zum Einsatz kommen. In diesem Fall werden die Lagereinheiten auf fahrbare Untersätze gestellt, mit Hilfe welcher sie durch den Lagerkanal rollen. Rolluntersätze sind in der Regel ebenfalls mittels Schwerkraft angetrieben. Lagersysteme mit motorenbetriebenen Rollenbahnen oder Rolluntersätzen finden sich eher selten.

Dynamische Blocklager mit Rollenbahnen oder Rolluntersätzen können mit kombinierter oder separater Vertikal- und Querförderung ausgestattet sein. Bei kombinierter Förderung werden häufig Stapler zur Bedienung eingesetzt. Ein erfolgreiches, modernes Kanallagersystem ist das DYNASTORE-Lager. Es verfügt über separate automatisierte Gassen- und Vertikalförderer.

Kanalförderung mit Satelliten

Satelliten sind technologisch hochstehende, eigenständige Kanalförderzeuge, welche nicht an einzelne Lagerkanäle gebunden sind. Sie werden meistens mit Hilfe sogenannter Satellitenstapler bedient. Der Stapler transportiert den Satelliten zu einem Lagerkanal und setzt ihn dort ab. Der Satellit fährt eigenständig in den Kanal ein, positioniert sich unter einer Palette und transportiert diese an das Kanallende. Dort übernimmt der Stapler die Palette und den Satelliten. Es wird unterschieden zwischen staplergebundenen und funkgesteuerten Satelliten.

Staplergebundene Satelliten sind mit einem Kabel fest mit dem Stapler verbunden und werden von diesem gesteuert. Beim Einsatz des Satelliten muss der Stapler am Kanallende warten, bis der Satellit aus dem Kanal zurückkehrt. Funkgesteuerte Satelliten sind bei ihrem Einsatz nicht abhängig von der Präsenz des Staplers. Sie werden von diesem durch Funksignale gesteuert. Dadurch ist der Stapler nicht gezwungen, während des Satelliteneinsatzes am Kanallende zu warten. Häufig bedient ein Stapler mehrere funkgesteuerte Satelliten, die er in verschiedenen Korridoren aussetzt und anschliessend wieder abholt.

Eher selten anzutreffen sind Satellitenlager mit Regalbediengeräten oder separaten Etagenfahrzeugen und Aufzügen.

Kanalförderung mit Kanalfahrzeugen

Kanalfahrzeuge (auch Kulis genannt) sind eigenständig angetriebene Förderwagen, welche in die Lagerkanäle integriert sind und den Palettentransport in den Kanälen übernehmen. Der bekannteste und erfolgreichste Blocklagertyp mit Kanalfahrzeugen ist das sogenannte ACTIW-Lager. Ein ACTIW-Lager besitzt ein dreistufiges, automatisiertes Transportsystem, bestehend aus Aufzügen, Quergangswagen und Kanalfahrzeugen (Korridorwagen). Da jeder Lagerkanal und jeder Quergang mit einem eigenen Förderwagen ausgestattet ist, umfasst eine solches Lager eine enorme Anzahl (Hunderte bis Tausende) von autonom betriebenen und individuell gesteuerten Transportmitteln, deren Bewegungen zu koordinieren sind. Entsprechend komplex sind die Entscheidungsmechanismen bei der Steuerung dieser Lagersysteme. Diese Arbeit beschäftigt sich mit der Leistungsabschätzung und Steuerungsoptimierung von ACTIW-Lagern. Eine ausführliche Besprechung dieser Lagersysteme findet sich in Kapitel 2.

1.4. Betrieb und Steuerung von Lagersystemen

Der Betrieb und die Steuerung von Lagersystemen umfasst eine Vielzahl von Funktionen, welche je nach Gesichtspunkt unterschiedlich definiert und strukturiert werden können. Zur Beschreibung dieser Funktionen werden in der Literatur unter anderem die Begriffe Lagerverwaltung,

Lagerhaltung und Lagersteuerung verwendet. Die Auslegung dieser Begriffe sowie die Strukturierung der entsprechenden Funktionen ist nicht einheitlich und fällt je nach Autor und Betrachtungsstandpunkt unterschiedlich aus.

In diesem Abschnitt wird versucht, die verschiedenen Begriffe, welche im Zusammenhang mit dem Betrieb von Lagersystemen verwendet werden, vom Standpunkt dieser Arbeit aus zu klären. Ziel dabei ist, den Begriff der Lagersteuerung abzugrenzen und die funktionalen Komponenten einer Lagersteuerung zu beschreiben. Die Systemanalyse basiert einerseits auf den Arbeiten verschiedener Autoren und andererseits auf Ueberlegungen, welche im Rahmen dieser Arbeit gemacht wurden im Zusammenhang mit den Problemen der Leistungsabschätzung und Steuerungsoptimierung bei automatisierten Kompaktlagern. Literatur zum Thema findet sich in unter anderem in [Martin, H. (1990)], [Aggteleky, B. (1990)], [Schulze, L. (1995)], [Reichert, H. (1989)], [Schardt, W. (1991)] sowie in diversen weiteren Beiträgen aus [Kapoun, J. (1993 - 1996)] und [Kapoun, J. (1993 - 1997)].

Es sei darauf hingewiesen, dass die nachfolgenden Systembetrachtungen unter dem speziellen Gesichtspunkt dieser Arbeit erfolgen. Somit wird vor allem auf diejenigen Aspekte der Lagersteuerung eingegangen, welche beim Steuerungsentwurf für automatisierte (Kompakt-)Lager von Bedeutung sind, und andere Aspekte, welche im Rahmen dieser Arbeit weniger wichtig sind, werden bewusst weggelassen.

1.4.1. Lagerhaltung und Lagersteuerung

Als Komponente der Logistikkette eines Unternehmens stellt ein Lager eine Einrichtung dar, in welcher Material bis zum Zeitpunkt des Bedarfs zwischenzeitlich gelagert werden kann. Auf der Ebene der Materialflüsse bildet ein Lager eine Systemkomponente, welcher Material zugeführt wird in Form von Einlagerungen und aus welcher Material abfließt in Form von Auslagerungen.

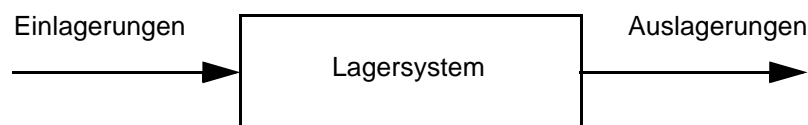


Abbildung 1.5. *Lagersystem als Komponente der Materialflussskette*

Bei der Diskussion von Betrieb und Steuerung von Lagersystemen muss unterschieden werden zwischen einer Betrachtungsebene, welche Lager auf der Ebene der Materialflüsse untersucht, und einer Betrachtungsebene, welche auf die internen Prozesse eines Lagersystems eingeht. Dementsprechend lassen sich die Aufgaben der Lagerverwaltung aufteilen in Lagerhaltung und Lagersteuerung.

Lagerhaltung

Der Begriff der Lagerhaltung bezieht sich auf diejenigen Funktionen der Lagerverwaltung, welche sich auf die Verwaltung und Kontrolle der Materialflüsse beziehen. Die Lagerhaltung ist unabhängig von der Struktur des Lagersystems und beschäftigt sich nicht mit den internen Prozessen des Lagersystems. Die Funktionen der Lagerverwaltung lassen sich in drei Hauptkomponenten aufteilen: Artikel- und Bestandesverwaltung, Einlagerungskomponente (bzw. Beschaffungskomponente) und Auslagerungskomponente (bzw. Absatzkomponente).

Artikel- und Bestandesverwaltung

Aufgabe der Artikelverwaltung ist es, die verschiedenen Lagerartikel mit den zugehörigen Produktdaten zu registrieren und die entsprechenden Informationen auf adäquate Weise verfügbar zu machen. Mögliche Produktdaten sind beispielsweise Artikelname, Identifikationscode, Artikelbeschreibung, Preise, Gewicht, Volumen, Lieferantendaten, statistische Daten, usw. In der Bestandesverwaltung werden die aktuellen Lagerbestände der Artikel festgehalten und entsprechend den Ein- und Ausgängen fortwährend nachgeführt. Die Artikel- und Bestandesverwaltung umfasst ausschliesslich Datenverwaltungsfunktionen und wird heutzutage typischerweise in Form einer elektronischen Datenbank realisiert.

Einlagerungskomponente

Aufgabe der Einlagerungskomponente ist die Verwaltung und Generierung der Einlagerungsaufträge. Je nach Funktion eines Lagers innerhalb der Unternehmenslogistik (d.h. Beschaffungslager, Produktions- oder Distributionslager) können Einlagerungsaufträge auf verschiedene Weise ausgelöst werden. Die Beschaffung bei einem Beschaffungslager kann beispielsweise auf der Basis einer bestimmten Beschaffungsstrategie erfolgen. Diese kontrolliert die aktuellen Lagerbestände und löst aufgrund gewisser Kriterien Materialbestellungen aus, welche nach Eintreffen der Ware zu Einlagerungsaufträgen führen. Bei Produktions- und Distributionslagern sind die Einlagerungsaufträge typischerweise abhängig von den umgebenden Produktions- und Distributionsprozessen und werden ausgelöst beim Eintreffen von neuem Material.

Die Einlagerungskomponente umfasst neben den Datenverwaltungsfunktionen (Verwaltung der Einlagerungsaufträge) auch Entscheidungsfunktionen bei der Generierung der Einlagerungsaufträge: Es muss entschieden werden, wann und in welcher Reihenfolge die Einlagerungsaufträge an das Steuersystem übergeben werden. Auf die Formulierung der Einlagerungsaufträge und deren Uebergabe an die Lagersteuerung wird weiter unten noch eingegangen.

Auslagerungskomponente

Aufgabe der Auslagerungskomponente ist die Verwaltung und Generierung der Auslagerungsaufträgen. Analog zur Einlagerungskomponente können Auslagerung je nach Funktion eines Lagers ebenfalls auf verschiedene Weise ausgelöst werden. Ebenso umfasst die Auslagerungs-

komponente sowohl Datenverwaltungs- als auch Entscheidungsfunktionen. Auslagerungsaufträge werden zur Ausführung an die Lagersteuerung weitergegeben.

Abschliessend sei bemerkt, dass die Lagerhaltung in modernen Unternehmen immer häufiger als eine Teilfunktion der gesamten Logistik- und Produktionsplanung gesehen wird und weniger als Verwaltungsfunktion eines einzelnen Lagersystems. Je nach Unternehmensorganisation ergibt sich daraus eine unterschiedliche Abgrenzung zwischen den Ein- und Auslagerungskomponenten eines Lagersystems und den entsprechenden Funktionen in der unternehmensweiten Logistikplanung. Die Thematik der Lagerbewirtschaftung und Lagerhaltung ist ein weites Feld, welches ausführlich in der Literatur diskutiert wird. Im Rahmen dieser Arbeit wird nicht weiter darauf eingegangen, da der Fokus dieser Arbeit auf die eigentliche Lagersteuerung ausgerichtet ist.

Ziel dieses Abschnittes war, den Begriff der Lagerhaltung vom Begriff der Lagersteuerung abzugrenzen, die wichtigsten funktionalen Komponenten der Lagerhaltung aufzuzeigen und die Begriffe des Ein- und Auslagerungsauftrages einzuführen.

Lagersteuerung

Die Lagersteuerung befasst sich im Gegensatz zur Lagerhaltung mit den internen Transport- und Lagerprozessen eines Lagers. Als Input für die Lagersteuerung werden von der Lagerhaltung die dort generierten Ein- und Auslagerungsaufträge übergeben. In Anlehnung an [Martin, H. (1990)] lässt sich die Lagersteuerung aufteilen in eine drei funktionale Komponenten: Administrative Komponente, dispositive Steuerung und operationelle Steuerung.

Administrative Komponente

Die administrative Komponente umfasst die Verwaltung der Lagereinheiten und Lagerpositionen sowie die Verwaltung der Lagerplatzbelegung. Auf der Ebene der Lagerhaltung werden die Lagergüter in Form von Artikeln und Lagerbeständen verwaltet. Im Gegensatz dazu werden sie innerhalb eines Lagers als einzelne Lagereinheiten behandelt. Je nach Art des Lagers und der lagerinternen Transporteinrichtungen wird eventuell weiter unterschieden zwischen Transporteinheiten sowie Ein- und Auslagerungseinheiten (vgl. Abschnitt 1.2.). Lagereinheiten werden häufig gebildet, indem eine bestimmte Materialmenge eines Artikels mit Hilfe eines Lagergeräts (z.B. einer Palette) zu einer Verwaltungseinheit zusammengefasst wird.

Die Verwaltung der Lagereinheiten umfasst die Identifikation und Registrierung der verschiedenen Lagereinheiten sowie die Speicherung der benötigten Informationen zu den Einheiten (z.B. Artikel, Menge, Gewicht, Volumen, Einlagerungsdatum, etc.).

Die Verwaltung der Lagerpositionen umfasst die Identifikation und Registrierung der verschiedenen Lagerplätze sowie ihrer technischen Eigenschaften (z.B. Ort, Grösse, Typ der möglichen Lagereinheiten, Transportwege, etc.).

Die Verwaltung der Lagerplatzbelegung registriert für jeden Lagerplatz, mit welcher Lagereinheit er belegt ist, bzw. für jede Lagereinheit, an welcher Position sie gelagert ist. Diese Informationen werden laufend aufgrund der ausgeführten Ein-, Aus- und Umlagerungen nachgeführt. Die Lagerplatzbelegung widerspiegelt den aktuellen Zustand des Lagers.

Dispositive Steuerung

Die dispositive Steuerung stellt die logische Steuerungskomponente eines Lagers dar, in welcher die wesentlichen Entscheide in Bezug auf die lagerinternen Transport- und Lagervorgänge festgelegt werden. Die dispositive Steuerung übernimmt die Ein- und Auslagerungsaufträge aus der Lagerhaltung und wandelt sie in konkrete Transportanweisungen um. Dies beinhaltet die Wahl der Lagereinheiten und Lagerpositionen, der Eingangs- und Ausgangspunkte sowie der Transportmittel und Transportrouten. Weiter werden in der dispositiven Steuerung die Aktionen der verschiedenen Transportmittel bestimmt und zeitlich eingeplant. Der daraus resultierende Ablaufplan wird an die operationelle Steuerung zur Ausführung übergeben.

Die Problemstellungen dieser Arbeit beziehen sich vorwiegend auf die Ebene der dispositiven Steuerung, weshalb der Besprechung dieser Komponente besondere Beachtung geschenkt wird. Eine detaillierte Diskussion der Aufgaben und Funktionen der dispositiven Steuerung findet sich im nächsten Abschnitt.

Operationelle Steuerung

Die operationelle Steuerung (bzw. Anlagensteuerung) befasst sich mit der untersten, technischen Ebene der Lagersteuerung. Im Falle von automatisierten Lagern basiert die operationelle Steuerung im allgemeinen auf der Technik der speicherprogrammierten Steuerungen (SPS). Typischerweise ist jedes Transportmittel mit einem eigenen SPS-Modul ausgerüstet, welches die Aktionen des Geräts mit Hilfe von Antrieben, Signalgebern, Sensoren, etc. steuert. Die SPS der verschiedenen Transportmittel sind mit einem Feldbus untereinander verbunden und werden von einem Leitrechner kontrolliert, welcher seinerseits mit dem dispositiven Steuerungssystem in Verbindung steht. Die operationelle Steuerung informiert das dispositive Steuerungssystem laufend über den aktuellen Zustand des Lager- und Transportsystems. Eine weitere Aufgabe der operationellen Steuerung ist die Erfassung von Störungen im Lagerbetrieb und die Uebergabe der entsprechenden Informationen an die dispositive Steuerung.

Die operationelle Steuerung von Lagersystemen liegt ausserhalb der Betrachtungen dieser Arbeit und wird nicht weiter diskutiert. Weiterführende Informationen finden sich beispielsweise in [Martin, H. (1990)], [Bösl, R. (1989)] und diversen Beiträgen in [Kapoun, J. (1993 - 1996)] und [Kapoun, J. (1993 - 1997)].

Uebersicht über die Lagerverwaltung

Die untenstehende Abbildung zeigt eine Uebersicht über die besprochenen Komponenten der Lagerverwaltung und die wichtigsten Informationsflüsse zwischen diesen. Komponenten mit reiner Datenverwaltungsfunktion sind als Ovale eingezeichnet und Komponenten mit Entscheidungsfunktionen als Rechteck. Die Informationsflüsse sind kursiv beschriftet. Als wichtigste Komponente für die Betrachtungen dieser Arbeit ist die dispositive Steuerung hervorgehoben.

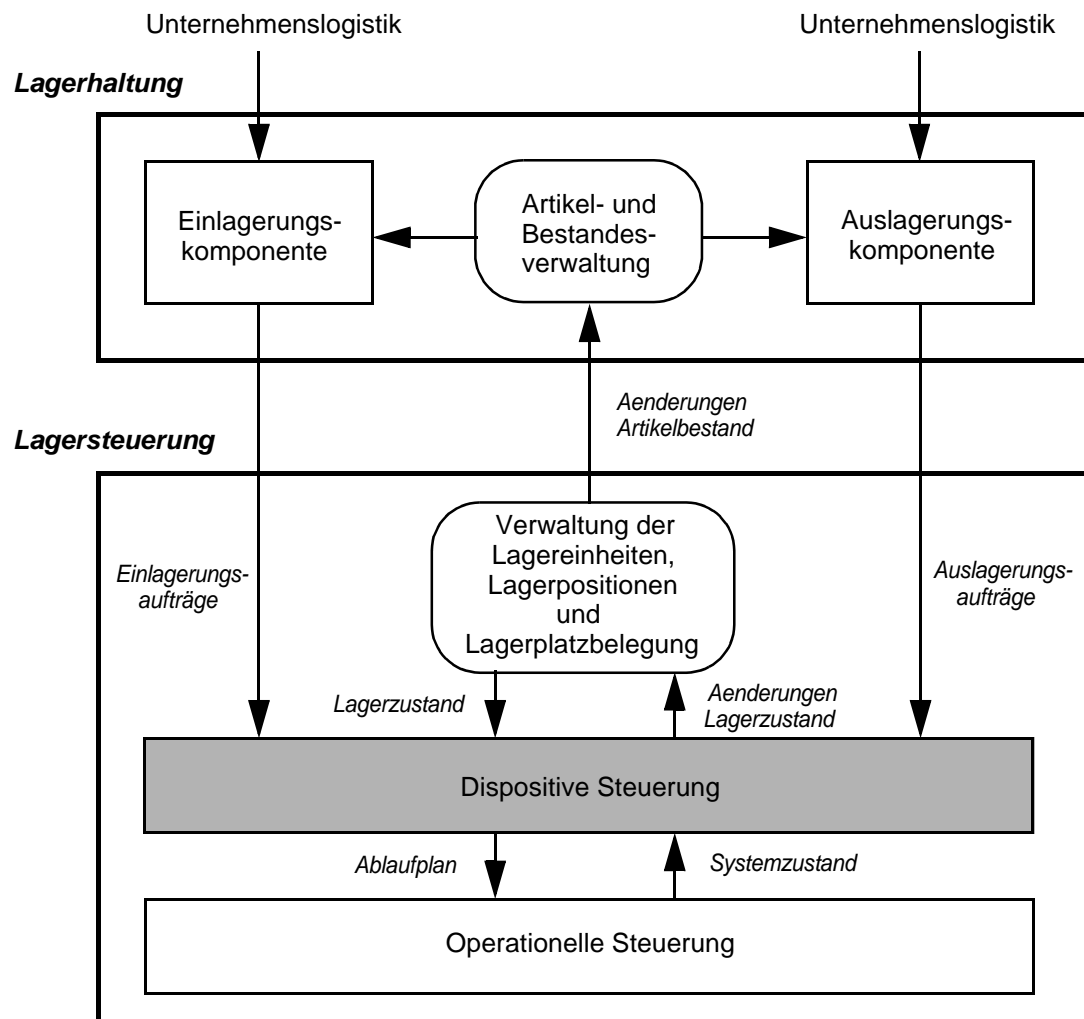


Abbildung 1.6. Komponenten der Lagerverwaltung

Zusammenfassend lässt sich der Betrieb eines Lagersystems wie folgt beschreiben: In den Ein- und Auslagerungskomponenten der Lagerhaltung werden aufgrund der aktuellen Artikelbestände und der externen Beschaffungs-, Produktions- und Absatzinformationen Ein- und Auslagerungsaufträge generiert. Diese werden an die dispositive Steuerung übergeben. Dort werden auf der Basis von internen Entscheidungsmechanismen aus den Ein- und Auslagerungsaufträgen

gen konkrete Transportaufträge generiert und die entsprechenden Transportaktionen zeitlich terminiert. Für diese Aufgaben werden Informationen über den aktuellen Zustand des Lagers und des Transportsystems benötigt. Der resultierende Ablaufplan wird an die operationelle Steuerung übergeben, welche die Ausführung der Transportoperationen kontrolliert. Die operationelle Steuerung informiert die dispositive Ebene laufend über den aktuellen Systemzustand. Nach Ausführung eines Transportauftrags werden die entsprechenden Änderungen des Lagerzustandes in der Lagerplatzverwaltung nachgeführt. Diese informiert anschliessend die Bestandesverwaltung über die Änderungen in den Artikelbeständen.

1.4.2. Dispositive Steuerung

Die dispositive Steuerung bildet die zentrale Komponente einer Lagersteuerung, in welcher die wesentlichen Entscheidungen zur Kontrolle der lagerinternen Transport- und Lagerprozesse gefällt werden.

Als Input übernimmt die dispositive Steuerung die Ein- und Auslagerungsaufträge vom Lagerhaltungsmodul. Aus diesen Aufträgen werden auf der Basis von verschiedenen Entscheidungsmechanismen die konkreten Transportaufträge erzeugt. Die Transportaufträge beschreiben für jedes Transportmittel die auszuführenden Aktionen. Aufgrund der Transportaufträge wird ein Ablaufplan generiert, welcher die Reihenfolge und Ausführungszeitpunkte der verschiedenen Transportaktionen festlegt. Der Ablaufplan wird als Output an die operationelle Steuerung übergeben, welche die vorgegebenen Aktionen ausführt.

Im Falle von Blocklagern, wo nicht alle Lagereinheiten direkt zugreifbar sind, bedingt der Zugriff auf gewisse Lagerplätze unter Umständen die vorangehende Umlagerung anderer Lagereinheiten. Diese erzwungenen Umlagerungen werden von einer speziellen Umlagerungskomponente verwaltet. Manchmal ist es nötig oder zweckmässig, eine Reorganisation der Lagerplatzbelegung vorzunehmen. Die entsprechenden Umlagerungsaufträge werden von einer Reorganisationskomponente geplant.

Neben den Angaben über die Ein- und Auslagerungsaufträge benötigt die dispositive Steuerung als zusätzlichen Input Informationen über den Lagerzustand (d.h. die Lagerplatzbelegung) und den aktuellen Zustand des Transportsystems.

Die untenstehende Abbildung zeigt eine Uebersicht über die verschiedenen Komponenten der dispositiven Steuerung und die wichtigsten Informationsflüsse zwischen diesen. Anschliessend werden die verschiedenen Punkte im einzelnen besprochen.

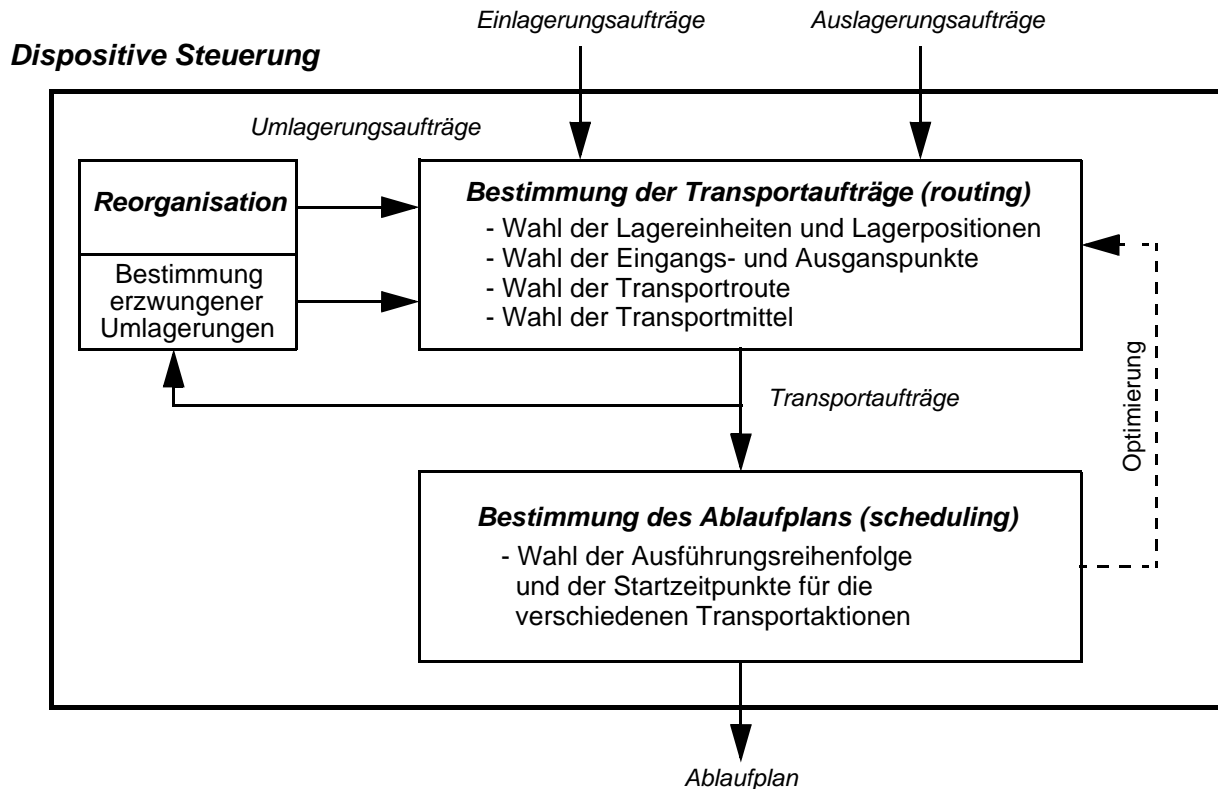


Abbildung 1.7. Dispositive Steuerung eines Lagersystems

Ein- und Auslagerungsaufträge

Die Ein- und Auslagerungsaufträge werden auf der Ebene der Lagerhaltung generiert. Es wird hier angenommen, dass sich Ein- und Auslagerungsaufträge jeweils auf einen Artikel beziehen.

Ein Einlagerungsauftrag spezifiziert typischerweise den Artikel und die Anzahl Lagereinheiten, welche von diesem Artikel eingelagert werden sollen. Falls das einzulagernde Material nicht in Form von Lagereinheiten angeliefert wird, muss es vor der Einlagerung in einzelne Lagereinheiten aufgeteilt werden. Dies geschieht häufig unter Zuhilfenahme von Lagergeräten wie beispielsweise Paletten. Anschliessend müssen die Lagereinheiten identifiziert und zusammen mit den entsprechenden Daten in der administrativen Steuerungskomponente registriert werden. Häufig wird in einem Einlagerungsauftrag auch der Eingangspunkt vorgegeben, an welchem das Material in das Lager aufgenommen werden soll. Es ist auch möglich, dass eine Menge von möglichen Eingangspunkten vorgegeben ist, oder dass der Eingangspunkt nicht spezifiziert ist. In diesem Fall entscheidet die dispositive Steuerung über den Eingangspunkt. Typischerweise enthält ein Einlagerungsauftrag keine Information darüber, an welchen Lagerpositionen die neuen Lagereinheiten eingelagert werden sollen. Die Aufgabe der Lagerplatzzuweisung stellt eine lagerinterne Verwaltungsfunktion dar und wird von der dispositiven Steuerung übernommen.

Ein Auslagerungsauftrag spezifiziert den Artikel und die Anzahl Lagereinheiten, welche von diesem Artikel ausgelagert werden sollen. Eventuell werden noch Angaben gemacht über die möglichen Ausgangspunkte, an denen die Lagereinheiten ausgelagert werden sollen. Bei Wählbarkeit des Ausgangspunktes bestimmt die dispositive Steuerung den Ausgangspunkt. Typischerweise enthält ein Auslagerungsauftrag keine Information darüber, welche konkreten Lagereinheiten, bzw. von welchen Lagerpositionen diese Einheiten ausgelagert werden sollen. Analog zu den Einlagerungen stellt die Wahl der konkreten Lagereinheiten bzw. Lagerpositionen für die Auslagerung eine lagerinterne Verwaltungsfunktion dar, welche von der dispositiven Steuerung übernommen wird.

Es ist möglich, dass zusammen mit den Ein- und Auslagerungsaufträgen gewisse Reihenfolgebedingungen vorgegeben werden, welche Angaben über die Bearbeitungsreihenfolge der Aufträge machen.

Bestimmung der Transportaufträge

Die Komponente, welche die Transportaufträge bestimmt, übernimmt als Input die Ein- und Auslagerungsaufträge sowie die Umlagerungsaufträge aus dem Umlagerungs- und Reorganisationsmodul. Als Output wird eine Menge von Transportaufträgen generiert, anhand welcher die vorgegebenen Ein-, Aus- und Umlagerungsaufträge ausgeführt werden. Es wird hier angenommen, dass die Transporteinheiten den Lagereinheiten entsprechen und dass die Transportmittel Einheitskapazität haben, d.h. dass pro Transportfahrt eine einzige Lagereinheit bewegt wird. In diesem Falle beziehen sich die Transportaufträge auf einzelne Lagereinheiten, d.h. für jede zu transportierende Lagereinheit wird ein Transportauftrag generiert. Ein Transportauftrag entspricht somit der Ein-, Aus- oder Umlagerung einer einzelnen Lagereinheit.

Ein Transportauftrag spezifiziert einerseits die verschiedenen Transportmittel, welche für den Transport einer Lagereinheit eingesetzt werden, und andererseits für jedes beteiligte Transportmittel die Folge der Aktionen (bzw. Operationen), welche von diesem Gerät ausgeführt werden müssen. Mögliche Aktionen sind beispielsweise die Anfahrt des Transportmittels zu einer Lagerposition, das Aufheben oder Absetzen einer Lagereinheit, die Transportfahrt von einer Position zu einer anderen Position und die Uebergabe einer Lagereinheit an ein anderes Transportmittel.

In der Terminologie der Maschinenbelegungs- oder Ablaufplanung entspricht die Generierung der Transportaufträge der Problematik des Routings oder der Ressourcenallokation. Allgemein befasst sich das Routing bzw. die Ressourcenallokation mit der Aufgabe, die Menge der Ressourcen und Operationen zu bestimmen, welche für die Bearbeitung eines Objekts eingesetzt werden, bzw. mit der Zuordnung der für die Bearbeitung eines Objekts benötigten Operationen an die vorhandenen Ressourcen. Im Falle eines Lagersystems entsprechen die Objekte den Lagereinheiten, die Ressourcen den Transportmitteln und die Operationen den Aktionen eines Transportmittels.

Die Bestimmung der Transportaufträge umfasst im allgemeinen Entscheidungsprozesse auf vier Ebenen:

- Wahl der Lagereinheiten und Lagerpositionen
- Wahl der Ein- und Auslagerungspunkte
- Wahl der Transportroute
- Wahl der Transportmittel

Bei der Spezifikation der Transportaufträge muss für jeden Transportauftrag (d.h. jede Ein-, Aus- oder Umlagerung) die Start- und die Zielposition festgelegt werden. Im Falle einer Einlagerung entspricht die Startposition einem Eingangspunkt und die Zielposition einem Lagerplatz, an welchem die Lagereinheit eingelagert wird. Im Falle einer Auslagerung entspricht die Startposition einem Lagerplatz, von welchem die Lagereinheit entnommen wird, und die Zielposition einem Ausgangspunkt. Bei Umlagerungen entspricht die Start- und die Zielposition einem Lagerplatz.

Wahl der Lagereinheiten und Lagerpositionen

Ein Einlagerungsauftrag spezifiziert die Menge der einzulagernden Lagereinheiten eines Artikels. Für jede Lagereinheit (bzw. jede Einlagerung) muss die Zielposition bestimmt werden, an welcher diese Einheit eingelagert wird. Die Bestimmung der Einlagerungspositionen basiert auf einer sogenannten Einlagerungsstrategie. Die Einlagerungsstrategie wird von vielen verschiedenen Faktoren beeinflusst und kann je nach Lager sehr unterschiedlich ausfallen. Ein wichtiger Faktor ist beispielsweise die Unterscheidung zwischen geordneter und chaotischer Lagerung. Bei geordneter Lagerung werden die Lagerplätze aufgeteilt in verschiedene Bereiche, in welchen ausschliesslich Lagereinheiten eines einzigen Artikels gelagert werden. Im Gegensatz dazu werden bei chaotischer Lagerung die Lagereinheiten der verschiedenen Artikel beliebig auf das Lager verteilt. Ein weiterer Faktor ist die Frage der Reihenfolgedisziplin. Meistens wird in Lagern eine FIFO-Strategie (first in first out) angestrebt. Je nach Lagertyp existieren aber auch Lagerbereiche, welche nach dem LIFO-Prinzip (last in first out) funktionieren. Ein weiterer wichtiger Einflussfaktor bei der Einlagerungsstrategie ist die Erreichbarkeit der Lagereinheiten. Häufig kann ein Lagersortiment aufgeteilt werden in verschiedene Artikelklassen, welche sich bezüglich ihrer durchschnittlichen Aufenthaltsdauer im Lager unterscheiden (sogenannte ABC-Klassifikation bzw. Schnelldreher und Langsamdreher). Artikel, welche häufig ein- und ausgelagert werden, sollen möglichst in Lagerbereichen platziert werden, welche gut erreichbar sind und wenig Transportaufwand erzeugen. Artikel, welche selten gebraucht werden, können in weniger gut positionierten Lagerbereichen gelagert werden. Schliesslich kann die Wahl der Einlagerungspositionen auch beeinflusst werden durch Informationen über anstehende Auslagerungsaufträge, welche es ermöglichen, die eingelagerten Einheiten so zu positionieren, dass sie für die nachfolgende Auslagerung günstig platziert sind.

Bei Auslagerungsaufträgen ist die Anzahl der auszulagernden Lagereinheiten eines Artikels vorgegeben. Die Wahl der konkreten Lagereinheiten, welche von diesem Artikel ausgelagert werden,

ist Aufgabe der dispositiven Steuerung. Die Wahl der auszulagernden Einheiten entspricht der Wahl der Lagerpositionen, an welchen die Lagereinheiten entnommen werden sollen. Somit wird bei einer Auslagerung durch die Wahl der Lagereinheit auch die Startposition des entsprechenden Transportauftrags festgelegt. Analog zu den Einlagerungen erfolgt die Bestimmung der auszulagernden Einheiten aufgrund einer bestimmten Auslagerungsstrategie. Diese basiert auf ähnlichen Überlegungen wie die Einlagerungsstrategie und ist mit dieser eng verknüpft. Zusammen bilden die Ein- und Auslagerungsstrategie die sogenannte Lagerstrategie.

Zusammenfassend bestimmt die Wahl der Einlagerungspositionen bei Einlagerung die Zielpositionen der entsprechenden Transportaufträge und die Wahl der Auslagerungspositionen (bzw. der Lagereinheiten) bei Auslagerungen die Startposition der Transportaufträge.

Wahl der Eingangs- und Ausgangspunkte

Die Bestimmung der Start- und Zielpositionen der Transportaufträge beinhaltet auch die Festlegung der Eingangs- bzw. Ausgangspunkte für die verschiedenen Ein- und Auslagerungen. Bei einer Einlagerung entspricht die Startposition einem Eingangspunkt und bei einer Auslagerung entspricht die Zielposition einem Ausgangspunkt. Wie schon früher erwähnt wurde, sind die Eingangs- und Ausgangspunkte in den Ein- und Auslagerungsaufträgen häufig nicht oder nur teilweise spezifiziert. In diesem Fall ist es Aufgabe der dispositiven Steuerung, diese zu bestimmen. Die Wahl der Eingangs- und Ausgangspunkte kann auf verschiedenen Kriterien basieren. Ein Entscheidungsfaktor kann beispielsweise die Minimierung der zeitlichen Dauer der Ein- und Auslagerungsprozesse sein. In diesem Falle kann versucht werden, die Eingangs- und Ausgangspunkte so zu wählen, dass die entsprechenden Transportrouten eine möglichst ausgeglichene Verteilung der Transportaktivitäten auf die verfügbaren Transportmittel bewirken.

Wahl der Transportroute

In vielen Lagersystemen ist die Transportroute für einen Transportauftrag eindeutig festgelegt, wenn die Start- und Zielposition des Auftrags vorgegeben ist. Es gibt jedoch Lagersysteme, bei welchen zwischen einer vorgegebenen Start- und Zielposition mehrere Transportrouten existieren. In diesem Falle ist es Aufgabe der dispositiven Steuerung, eine spezifische Transportroute auszuwählen.

Wahl der Transportmittel

Häufig sind in einem Lager mit der Vorgabe der Transportroute auch die für den Transport verwendeten Fördermittel festgelegt. Dies ist beispielsweise der Fall bei Hochregallagern, welche mit Regalbediengeräten ausgestattet sind. Bei gewissen Lagersystemen ist es jedoch möglich, dass der Transport auf einer bestimmten Route von mehreren verschiedenen Transportmitteln ausgeführt werden kann. Als Beispiel seien Lager erwähnt, welche von mehreren personengesteuerten Staplern bedient werden. Auch wenn die Transportrouten durch die vorhandenen

Lagergassen vorgegeben sind, besteht die Wahlmöglichkeit, welcher Transportauftrag mit welchem Stapler ausgeführt wird.

Bestimmung des Ablaufplans

Die Komponente, welche für die Bestimmung des Ablaufplans zuständig ist, übernimmt als Input eine Menge von Transportaufträgen und generiert daraus als Output einen Ablaufplan, welcher zur Ausführung an die operationelle Steuerung übergeben wird. Durch die Transportaufträge wird für jede Ein-, Aus- oder Umlagerung festgelegt, welche Aktivitäten von den beteiligten Transportmitteln auszuführen sind. Umgekehrt wird durch die Gesamtheit der Transportaufträge für jedes Transportmittel bestimmt, welche Aktivitäten von diesem Transportmittel im Verlaufe des betrachteten Lagerprozesses ausgeführt werden müssen. Die Transportaufträge legen jedoch nicht fest, in welcher Reihenfolge und zu welchem Zeitpunkt diese Aktivitäten stattfinden sollen. Diese Entscheidungen werden bei der Bestimmung des Ablaufplans gefällt.

Ein Ablaufplan definiert für jedes Transportmittel die Reihenfolge, in welcher die Aktivitäten ausgeführt werden sollen und für jede Aktivität den Zeitpunkt der Ausführung. Ein notwendiges Entscheidungskriterium bei der Generierung eines Ablaufplans ist die Zulässigkeit. Damit ein Ablaufplan zulässig ist, müssen die Aktionen zeitlich so eingeplant werden, dass die resultierenden Vorgaben von den Transporteinrichtungen tatsächlich ausgeführt werden können. Ein weiteres Zielkriterium kann beispielsweise die Minimierung der Gesamtdauer des Lagerprozesses sein.

In der Terminologie der Maschinenbelegungs- oder Ablaufplanung entspricht die Generierung eines Ablaufplans der Problematik des Sequencing und Scheduling. Allgemein befasst sich das Sequencing mit der Aufgabe, die Ausführungsreihenfolge von Operationen auf verschiedenen Maschinen so zu planen, dass ein bestimmtes Zielkriterium optimiert wird. Scheduling befasst sich mit der zeitlichen Einplanung von Operationen auf Maschinen unter Berücksichtigung eines Zielkriteriums.

Bestimmung der erzwungenen Umlagerungen

In Blocklagern besteht typischerweise nicht auf alle Lagereinheiten bzw. Lagerpositionen direkter Zugriff. Dadurch wird bei der Ein- oder Auslagerung von Lagereinheiten unter Umständen die Umlagerung weiterer Lagereinheiten erzwungen. Bei Einlagerungen ist dies der Fall, wenn die Zielposition belegt ist oder der Zugriff im Lagerkanal von anderen Lagereinheiten versperrt wird. Bei Auslagerungen werden Umlagerungen erzwungen, wenn der Zugriff auf die Auslagerungsposition im Lagerkanal versperrt ist. Die Menge der erzwungenen Umlagerungen kann erst bestimmt werden, wenn die Start- und Zielpositionen sowie die Transportrouten der Ein- und Auslagerungen festgelegt sind. Erst dann wird ersichtlich, welche Lagereinheiten die Transportwege versperren und folglich umgelagert werden müssen.

Die Komponente, welche die erzwungenen Umlagerungen kontrolliert, generiert auf der Basis einer Umlagerungsstrategie Umlagerungsaufträge, welche von der dispositiven Steuerung weiterverarbeitet werden. Ein Umlagerungsauftrag spezifiziert die umzulagernde Lagereinheit sowie die Start- und die Zielposition der Umlagerung. Die Bestimmung der Transportroute und der Transportmittel erfolgt in den entsprechenden Modulen der dispositiven Steuerung. Die Umlagerungsstrategie ist abhängig von verschiedenen Entscheidungsfaktoren und kann je nach Lager unterschiedlich sein. Eine mögliche Umlagerungsstrategie ist beispielsweise, dass die Lagereinheiten, welche den Zugang zu einer Position in einem Lagerkanal versperren, kurzfristig auf freie Lagerplätze umgelagert werden und nach erfolgter Ein- bzw. Auslagerung sofort wieder an ihre ursprüngliche Position zurücktransportiert werden. In andern Umlagerungsstrategien werden gewisse Lagerkanäle freigehalten und als Abstellplatz für die umgelagerten Paletten verwendet. Bei Lagersystemen mit chaotischer Lagerstrategie kann die Umlagerungsstrategie darin bestehen, dass die umzulagernden Einheiten auf beliebige freien Lagerplätze umgelagert werden.

Reorganisation

Unter Reorganisation wird die lagerinterne Umplazierung von Lagereinheiten zum Zwecke der Herstellung einer bestimmten Lagerordnung verstanden. Reorganisation kann aus verschiedenen Gründen durchgeführt werden. Beim Betrieb von Lagern ist es häufig so, dass in Phasen, wo das System stark belastet ist, die Einlagerung neueintreffender Einheiten auf chaotische Weise erfolgt. Anschliessend wird in Zeiten mit weniger Aktivität die gewünschte Ordnung im Lager wiederhergestellt. Die Verwaltung der entsprechenden Umlagerungen ist Aufgabe der Reorganisationskomponente. Ein weiterer Zweck der Reorganisation kann sein, dass in wenig aktiven Zeiten Vorbereitungsarbeiten für die anstehenden Ein- und Auslagerungsaufträge durchgeführt werden. Diese können beispielsweise darin bestehen, dass die Lagereinheiten für einen bevorstehenden Auslagerungsauftrag im voraus zusammengestellt und in einem Bereich des Lagers platziert werden, wo sie schnell und mit wenig Transportaufwand zu den Ausgangspunkten befördert werden können. Die Entscheidungsmechanismen auf der Ebene der Reorganisation werden unter dem Begriff Reorganisationsstrategie zusammengefasst. Als Output der Reorganisationskomponente wird eine Menge von Umlagerungsaufträgen generiert, welche von der dispositiven Steuerung weiterverarbeitet werden.

Reorganisation spielt speziell im Falle von automatisierten Lagern eine wichtige Rolle. Da die Transporteinrichtungen automatisch gesteuert sind, kann mit wenig finanziellem und personellem Aufwand in inaktiven Zeiten (typischerweise während der Nacht) eine Reorganisation des Lagers vorgenommen werden. Je nach Terminierung der Lageraufträge und der Verfügbarkeit der entsprechenden Informationen können dadurch umfangreiche Vorbereitungsarbeiten durchgeführt werden, welche die anschliessenden Lagerprozesse wesentlich beschleunigen können. In diesem Sinne bildet die Reorganisation eine wichtige leistungsbeeinflussende Komponente bei der Steuerung von automatisierten Lagersystemen.

Ziel dieses Abschnittes war, die wichtigsten Komponenten der dispositiven Lagersteuerung zu besprechen und die verschiedenen Entscheidungsdimensionen im Bereich der dispositiven Steuerung aufzuzeigen. Die Problemstellungen dieser Arbeit beziehen sich grösstenteils auf die Ebene der dispositiven Lagersteuerung, weshalb der Besprechung dieser Thematik besondere Aufmerksamkeit geschenkt wurde.

Automatisierte Kompaktlager

Das Thema der vorliegenden Arbeit ist die Optimierung automatisierter Kompaktlager. Im vorangehenden Kapitel wurden verschiedene Lagertypen vorgestellt, welche unter dem Oberbegriff automatisierte Kompaktlager oder automatisierte Blocklager zusammengefasst werden. Diese Arbeit bezieht sich auf ein konkretes Lagersystem, die sogenannten ACTIW-Lager. Im folgenden werden die Bezeichnungen ACTIW-Lager und automatisiertes Kompaktlager gleichbedeutend verwendet. Dieses Kapitel liefert eine ausführliche Beschreibung und Diskussion dieses Lagertyps.

In Abschnitt 2.1. wird eine detaillierte Systembeschreibung von automatisierten Kompaktlagern gegeben. Abschnitt 2.2. beschäftigt sich mit den speziellen Eigenschaften der Steuerung dieser Systeme. In Abschnitt 2.3. wird ein graphisches Modell zur formalen Darstellung automatisierter Kompaktlager vorgeschlagen. Es handelt sich um den sogenannten System-Graphen, welcher in den folgenden Kapiteln bei der Leistungsabschätzung und Steuerungsoptimierung eine wichtige Rolle spielt.

2.1. Systembeschreibung und Funktionsweise

In der modernen Lagertechnik bilden ACTIW-Lager ein relativ neuartiges, platzsparendes Konzept zur Lagerung von Palettenladungen. ACTIW-Lager sind vollautomatisierte Palettenlager und

gehören zur Klasse der dynamischen Blocklager (vgl. Abschnitt 1.3.). Ursprünglich entwickelt wurden diese Lager vor über dreissig Jahren von der finnischen Firma ACTIW. In der Zwischenzeit hat sich das Lagersystem kontinuierlich entwickelt und verbreitet. Es zählt in der modernen Lagertechnik zu einem wichtigen Lagertyp im Bereich der automatisierten Kompaktlagerung.

ACTIW-Lager und ähnliche Lagersysteme werden in der Literatur auch bezeichnet als automatisierte Kompaktlager, automatisierte Blocklager, dynamische Blocklager, dynamische Kanallager oder Kanalfahrzeuglager. In dieser Arbeit wird zur Bezeichnung von ACTIW-Lagern stets der Begriff automatisierte Kompaktlager verwendet.

2.1.1. Systemübersicht

Ein automatisiertes Kompaktlager (bzw. ACTIW-Lager) besteht aus mehreren Lagerebenen, welche unterteilt sind in verschiedene Sektoren (bzw. Blöcke). Die Sektoren sind gegliedert in parallel nebeneinanderliegende Lagerkorridore (bzw. Lagerkanäle), in welchen die Palettenladungen gelagert werden. Jeder Korridor umfasst eine Reihe von hintereinanderliegenden Lagerplätzen. Zwischen den Sektoren befinden sich Quergänge (bzw. Lagergassen), welche den Zugang zu den Korridoren ermöglichen. Die Korridore und Quergänge sind mit integrierten Förderwagen ausgestattet und die verschiedenen Lagerebenen durch Aufzüge (bzw. Senkrechtförderer) miteinander verbunden.

Die Ein- und Ausgang von Paletten erfolgt an sogenannten Ein- und Ausgangspunkten (bzw. Ein- und Auslagerungspunkten), an welchen sich Transporteinrichtungen in Form von Förderbändern, Rollenbahnen oder ähnlichen Förderinstallationen befinden. Sie ermöglichen die Aufnahme von neuen Paletten in das Lager bzw. die Abgabe von gelagerten Paletten an das umgebende System. Die Ein- und Ausgangspunkte können sich an verschiedenen Stellen im Lager befinden, typischerweise liegen sie auf einer gemeinsamen Grundebene (Kommissionier-Ebene) und werden lagerseitig von den Senkrechtförderern oder Quergangswagen bedient. Häufig erfolgen die Ein- und Ausgänge an denselben Punkten. Die Ein- und Ausgangspunkte bilden die Schnittstelle zwischen dem Lager und dem umgebenden System.

Die folgende Abbildung zeigt schematisch die Struktur eines automatisierten Kompaktlagers.

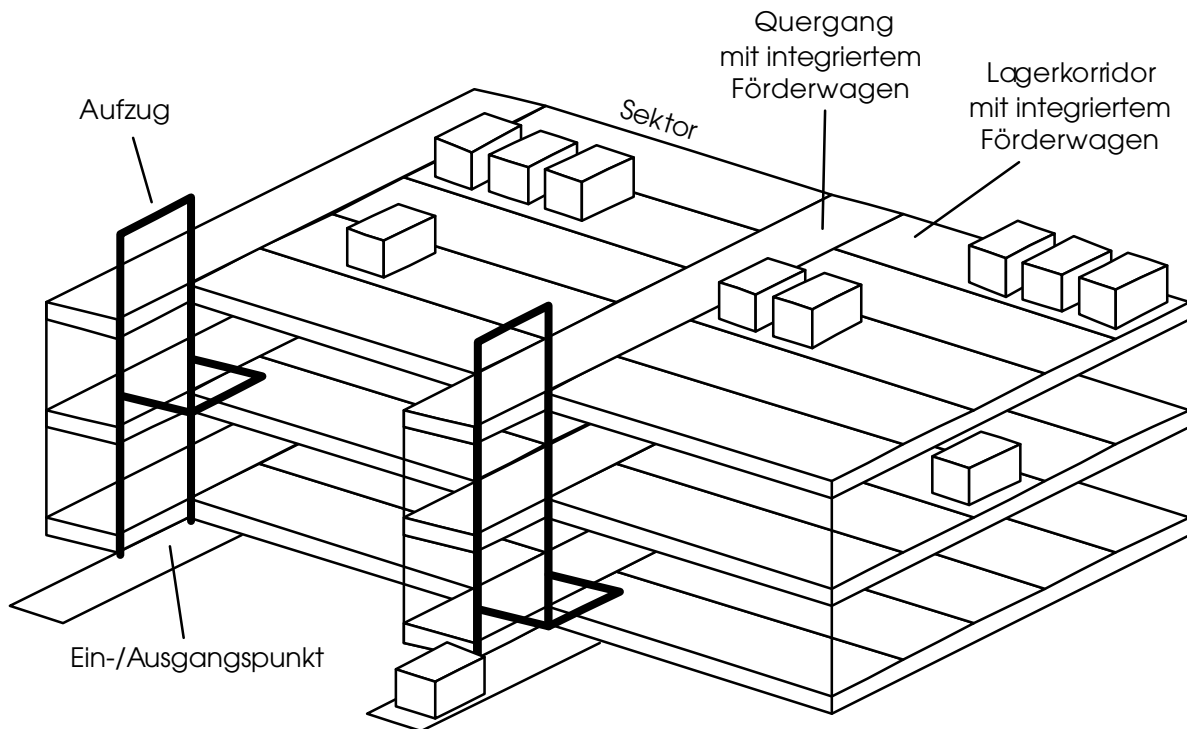


Abbildung 2.1. Automatisiertes Kompaktlager (ACTIW-Lager)

2.1.2. Transporteinrichtungen

Automatisierte Kompaktlager besitzen ein dreistufiges Transportsystem. Die Kanalförderung basiert auf dem Einsatz von integrierten, autonom angetriebenen Kanalfahrzeugen (sogenannten Korridorwagen). Die Vertikal- und Querförderung erfolgt separat durch Aufzüge und integrierte Quergangswagen.

Korridorwagen

Jeder Lagerkorridor ist mit einem integrierten Korridorwagen (bzw. Kanalfahrzeug, Kuli) ausgestattet, welcher den Palettentransport innerhalb der Lagerkanals übernimmt. Die Korridorwagen sind im allgemeinen schienengebunden und werden durch einfache Seilzug-Motoren angetrieben. Entweder verfügt jeder Korridorwagen über einen eigenen, unabhängig gesteuerten Antrieb oder mehrere Korridorwagen zusammen werden vom selben Antrieb gesteuert. Im folgenden wird angenommen, dass die Korridorwagen unabhängig angetrieben und gesteuert werden.

Die Korridorwagen sind so konstruiert, dass sie unter den Paletten durchfahren und sich somit frei in einem Lagerkorridor bewegen können, auch wenn dieser mit Paletten belegt ist. Zur Lastaufnahme positioniert sich ein Korridorwagen unter eine Palette und lädt diese mittels einer

Hebevorrichtung auf. Ein beladener Korridorwagen kann sich nur in denjenigen Kanalbereichen bewegen, welche nicht durch andere Paletten belegt sind.

Für den Transfer (d.h. für die Uebergabe oder Uebernahme) einer Palette zwischen einem Korridor- und einem Quergangswagen muss der Korridorwagen in den Quergang einfahren und sich unter den Quergangswagen positionieren. Voraussetzung dafür ist, dass sich der Quergangswagen an der entsprechenden Position befindet. Der Transfer erfolgt synchron, d.h. die beiden beteiligten Förderwagen müssen sich gleichzeitig für eine bestimmte Zeitspanne an dem Uebergabeort befinden. Falls sich Aufzüge in den Lagerkorridoren befinden, sind die Korridorwagen und Aufzüge so konstruiert, dass ein Lasttransfer möglich ist. Die Palettenübergabe erfolgt auch in diesem Falle synchron.

Quergangswagen

Jeder Quergang ist mit einem integrierten Quergangswagen (bzw. Querversetzwagen, Gassenfahrzeug) ausgestattet. Diese werden im allgemeinen mittels Seilzug-Motoren angetrieben und individuell gesteuert. Die Quergangswagen sind so konstruiert, dass ein Lasttransfer zu den angrenzenden Korridorwagen und Aufzügen möglich ist. Falls sich an einem Quergangende ein Ein- oder Ausgangspunkt befindet, erfolgt ein Lasttransfer zu den entsprechenden Fördereinrichtungen (z.B. Rollenbahnen) des Ein- bzw. Ausgangspunkts. Dieser Transfer ist im allgemeinen asynchron.

Aufzüge

Die Aufzüge dienen dem Palettentransport zwischen den verschiedenen Lagerebenen. Sie sind individuell gesteuert und befinden sich entweder an den Enden der Quergänge oder innerhalb eines Lagerkorridors. Häufig sind die Aufzüge mit ausfahrbaren Ladegabeln ausgestattet, welche einen Lasttransfer zu den angrenzenden Quergang- bzw. Korridorwagen ermöglichen. Dieser Lasttransfer erfolgt in beiden Fällen synchron. Der Lasttransfer zwischen den Aufzügen und den Ein- und Ausgangspunkten erfolgt typischerweise asynchron.

Für die Ausführung einer Ein- bzw. Auslagerung werden im allgemeinen alle drei Typen von Fördereinrichtungen benötigt. Bei einer Einlagerung wird eine Palette beispielsweise von einem Aufzug an einem Einlagerungspunkt übernommen und zur gewünschten Lagerebene transportiert. Anschliessend übernimmt ein Quergangswagen die Palette und transportiert sie zum Zielkorridor. Dort wird die Palette vom Korridorwagen in Empfang genommen und an die vorgegebene Lagerposition transportiert. Es ist auch möglich, dass eine Lagerbewegung (d.h. eine Ein-, Aus- oder Umlagerung) den Einsatz von mehr als drei Transportmitteln umfasst. Eine Einlagerung kann beispielsweise von einem Eingangspunkt via Aufzug, Quergang, Korridor, Quergang und Korridor zur Zielposition erfolgen. Die Beschreibung der verschiedenen möglichen Transportrouten in einem automatisierten Kompaktlager wird weiter unten diskutiert.

2.2. Steuerung

Die Steuerung von Lagersystemen und insbesondere von automatisierten Blocklagern wurde ausführlich besprochen in Abschnitt 1.4. Die dort gemachten Ausführungen lassen sich direkt übertragen auf automatisierte Kompaktlager. Abbildung 1.7. gibt eine Uebersicht über die Funktionsweise der dispositiven Steuerung automatisierter Kompaktlager. Zusammengefasst besteht die dispositive Steuerung aus folgenden funktionalen Komponenten:

- Bestimmung der Transportaufträge (Routing)
 - Wahl der Lagereinheiten und Lagerpositionen
 - Wahl der Ein- und Ausgangspunkte
 - Wahl der Transportroute
 - Wahl der Transportmittel
- Bestimmung des Ablaufplans (Scheduling)
 - Wahl der Ausführungsreihenfolgen der Transportaktionen
 - Wahl der Startzeitpunkte der Transportaktionen
- Bestimmung der Umlagerungen
 - Bestimmung der Umlagerungen zwecks Reorganisation
 - Bestimmung der erzwungenen Umlagerungen

Als Ergänzung zu Abschnitt 1.4. werden im folgenden gewisse Aspekte der dispositiven Steuerung besprochen, welche speziell bei automatisierten Kompaktlagern von Bedeutung sind.

2.2.1. Transportrouten

Die Bestimmung der Transportaufträge umfasst die Festlegung der Transportrouten der Lagerbewegungen (d.h. der Ein-, Aus- und Umlagerungen) sowie die Wahl der beteiligten Transportmittel. Im Falle der automatisierten Kompaktlager sind die Transportmittel eindeutig bestimmt, wenn die Transportroute vorgegeben ist. Somit entspricht die Spezifikation der Transportaufträge der Wahl der Transportrouten.

Die Bestimmung der Transportroute einer Lagerbewegung beinhaltet einerseits die Festlegung der Start- und Zielposition der Route und andererseits die Vorgabe des Transportweges zwischen diesen beiden Positionen. Die Start- und Zielposition wird festgelegt durch die Wahl der Ein- bzw. Auslagerungsposition (bzw. Auslagerungseinheit) und durch die Wahl des Ein- bzw. Ausgangspunktes. In einem automatisierten Kompaktlager ist je nach Lage der Start- und Zielposition die Transportroute eindeutig vorgegeben oder es existieren verschiedene Routen zwischen den beiden Positionen.

Die untenstehende Abbildung illustriert dies an einem Beispiel. Bei gegebenem Ausgangspunkt ist die Auslagerungsroute für Palette A eindeutig bestimmt. Für die Auslagerung von Palette B sind jedoch verschiedene Transportrouten möglich, je nach Wahl des Durchfahrtskorridors.

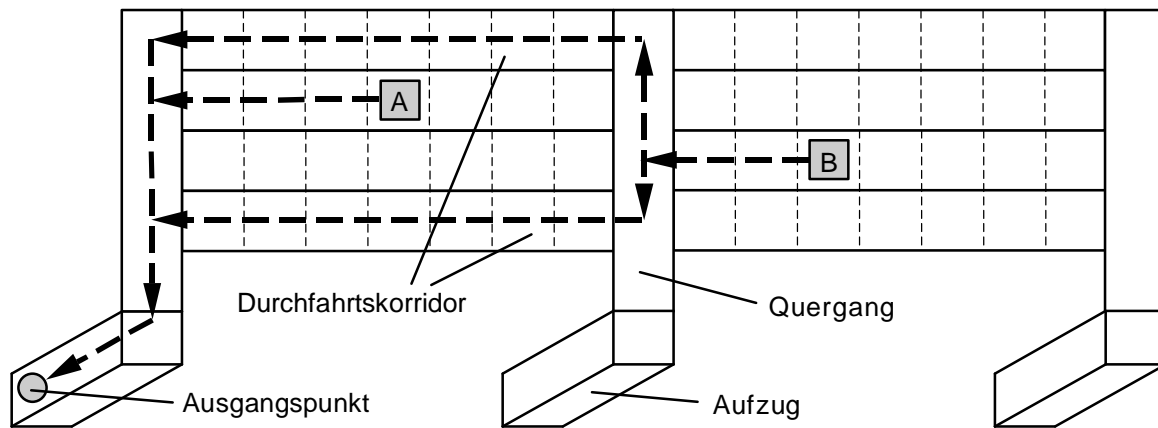


Abbildung 2.2. Transportrouten

Durchfahrtskorridore

Falls die Start- und Zielposition einer Lagerbewegung nicht im gleichen Sektor bzw. in benachbarten Sektoren liegen, müssen die dazwischenliegenden Sektoren traversiert werden. Dies ist nur möglich, wenn in den betreffenden Sektoren ein freier Lagerkanal vorhanden ist. Kanäle, welche zur Traversierung eines Sektors verwendet werden, heißen Durchfahrtskorridore. Es gibt verschiedene Strategien zur Verwaltung der Durchfahrtskorridore. Eine Möglichkeit ist, in jedem Sektor einen bestimmten Kanal freizuhalten und diesen als Durchfahrtskorridor zu benutzen. Eine andere Strategie besteht darin, einen beliebigen freien Kanal für die Durchfahrt zu verwenden. Falls kein freier Korridor existiert, wird durch Umlagerungen ein Kanal freigeschafft.

Zusammenfassend lässt sich sagen, dass Transportrouten von Lagerbewegungen ohne Sektortraversierung eindeutig bestimmt sind bei gegebener Start- und Zielposition. Bei Sektortraversierungen hingegen existieren je nach Wahl der Durchfahrtskorridore im allgemeinen mehrere verschiedene Transportrouten. In Abhängigkeit von der Anzahl der benützten Transportmittel wird eine Transportroute (bzw. die entsprechende Lagerbewegung) als k -stufig bezeichnet, wenn k Transportmittel an der Ausführung beteiligt sind. Transportrouten ohne Sektortraversierung sind im allgemeinen 3-stufig (z.B. Korridorwagen, Quergangswagen und Aufzug). Bei Sektortraversierungen umfassen die Routen entsprechend mehr Stufen.

Die Gesamtheit aller möglichen Transportrouten in einem automatisierten Kompaktlager ist abhängig von der räumlichen Anordnung (dem "Layout") der verschiedenen Fördereinrichtungen. In Abschnitt 2.3. wird ein Netzwerkmodell zur Darstellung von automatisierten Kompaktlagern

gern vorgeschlagen, mit dessen Hilfe die Menge alle möglichen Transportrouten auf einfache Weise beschrieben werden kann.

2.2.2. Transportaktionen

Die Spezifizierung der Transportaufträge auf der Ebene der dispositiven Steuerung beinhaltet die Wahl der Transportrouten für die verschiedenen Lagerbewegungen. Ein Transportauftrag enthält detaillierte Angaben über alle Aktionen, welche von den beteiligten Transportmitteln auszuführen sind. In diesem Abschnitt wird gezeigt, wie sich die Informationen über die notwendigen Transportaktionen aus den Transportrouten ableiten lassen.

Unter einer Transportaktion wird hier eine elementare Aktivität eines Transportmittels verstanden, welche auf der Ebene der dispositiven Steuerung als Prozesseinheit angesehen und nicht weiter in Teilprozesse zerlegt wird. Beispiele von Transportaktionen sind das Aufladen einer Palette oder der Transport einer Palette von einer Position zu einer andern. Auf der Ebene der operativen Steuerung setzt sich eine Transportaktivität im allgemeinen aus mehreren Teiloperationen zusammen (z.B. Beschleunigung und Abbremsung von Motoren, Betätigung hydraulischer Ladevorrichtungen, etc.).

Aktionen der Transportmittel

Nachstehend wird eine Uebersicht über die möglichen Aktionen der verschiedenen Transportmittel gegeben. Für Korridorwagen, Quergangwagen und Aufzüge lassen sich die folgenden gemeinsamen Transportaktionen festlegen:

- Transportfahrt von einer Start- zu einer Zielposition
- Leerfahrt von einer Start- zu einer Zielposition
- Uebergabe einer Palette an ein angrenzendes Transportmittel
- Uebernahme einer Palette von einem angrenzenden Transportmittel

Für Korridorwagen werden zwei zusätzliche Aktionen definiert:

- Aufladen einer Palette an einer Lagerposition
- Abladen einer Palette an einer Lagerposition

Quergangwagen und Aufzüge, welche mit einem Ein- oder Ausgangspunkt verbunden sind, verfügen analog über die folgenden beiden Aktionen:

- Aufladen einer Palette an einem Eingangspunkt
- Abladen einer Palette an einem Ausgangspunkt

Mit Hilfe dieser elementaren Transportaktionen lassen sich alle Lagerbewegungen vollständig beschreiben. Die Zeitdauer der Uebergabe- und Uebernahme- sowie der Lade- und Entladeaktionen ist abhängig von den technischen Geräteparametern und wird hier als konstant angenommen.

Die Dauer der Uebergabe und der entsprechenden Uebernahme bei einem Lasttransfer ist identisch. Die Dauer der Transport- und Leerfahrten hängt einerseits von der Fahrtgeschwindigkeit und Beschleunigung der Fördermittel ab und andererseits von der Distanz zwischen der Start- und Zielposition.

Beispiel: Transportaktionen bei einer Auslagerung

Zur Illustration wird am Beispiel einer dreistufigen Auslagerung gezeigt, welche Transportaktionen auszuführen sind. Der zeitliche Verlauf der verschiedenen Aktionen wird in einem Gantt-Diagramm dargestellt.

Gantt-Diagramme werden allgemein im Bereich der Maschinenbelegungsplanung zur Visualisierung von Ablaufplänen verwendet. Ein Gantt-Diagramm ist ein zweidimensionales Balkendiagramm, bei welchem auf der horizontalen Achse die Zeit und auf der vertikalen Achse die verschiedenen Maschinen (bzw. Prozessoren, Ressourcen) dargestellt werden. Eine Operation (bzw. Aktion, Aktivität) wird als Balken eingezeichnet, welcher vom Startzeitpunkt bis zum Beendigungszeitpunkt der Operation reicht.

Die untenstehende Abbildung zeigt ein Gantt-Diagramm für eine dreistufige Auslagerung, bei welcher eine Palette von einer Auslagerungsposition mittels Korridorwagen, Quergangswagen und Aufzug zu einem Ausgangspunkt transportiert wird (vgl. Auslagerung A in Abbildung 2.2.).

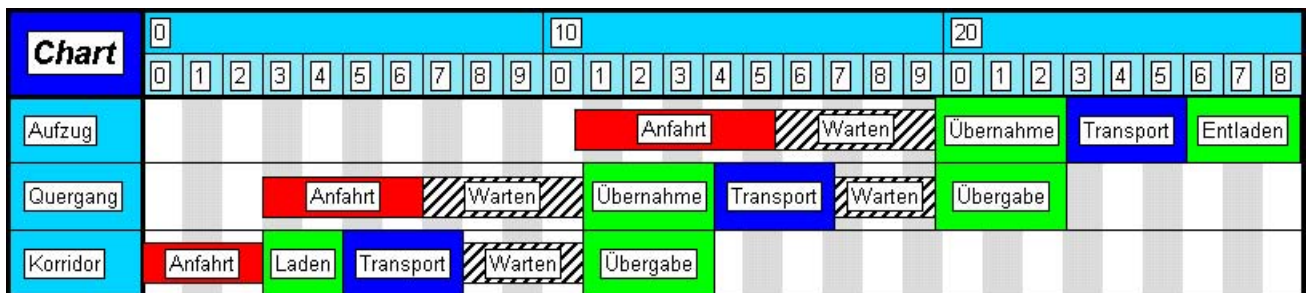


Abbildung 2.3. Transportaktionen bei einer Auslagerung

Der Korridorwagen muss sich als erstes von seiner aktuellen Position zu der Lagerposition bewegen, an welcher die Palette ausgelagert wird (Anfahrt, Leerfahrt). Anschliessend wird die Palette aufgeladen (Laden) und von der Lagerposition zur Uebergabeposition beim Quergang transportiert (Transport). Je nachdem, ob sich der Quergangswagen schon an der Uebergabeposition befindet, kann die Palettenübergabe (Uebergabe) unmittelbar erfolgen, oder der Korridorwagen muss mit der Uebergabe warten (Warten), bis der Quergangswagen ankommt.

Der Quergangswagen bewegt sich von seiner anfänglichen Position zu der Stelle im Quergang, wo die Palette vom Korridorwagen übernommen wird. Falls sich bei seiner Ankunft der Korridorwagen noch nicht an der Uebergabestelle befindet, muss der Quergangswagen warten, bis der Korridorwagen ankommt. Anschliessend erfolgt die Palettenübernahme. Darauf transportiert der Quergangswagen die Palette zum Ende des Quergangs, wo sich der Aufzug befindet. Falls sich der Aufzug noch nicht an der Uebergabestelle befindet, muss der Quergang auf dessen Ankunft warten. Darauf erfolgt die Uebergabe der Palette an den Aufzug.

Der Aufzug fährt von seiner anfänglichen Position zur Lagerebene, wo die Palette ausgelagert wird. Dort wartet er auf den Quergangswagen, falls dieser noch nicht anwesend ist. Nach der Ankunft des Quergangswagens übernimmt der Aufzug die Palette und transportiert sie zum Ausgangspunkt. Dort lädt er die Palette ab und übergibt sie an das umgebende System.

Struktur der Transportaktionen

Bei Betrachtung des obigen Gantt-Diagramms fällt auf, dass die Folge der Aktionen, welche ausgeführt werden müssen, für alle drei Transportmittel eine ähnliche Struktur aufweisen. Ein einheitliches Bild ergibt sich, wenn die Ladeaktionen der Korridorwagen als spezielle Uebernahmeaktionen aufgefasst wird, bei welchen eine Palette von einem Lagerplatz übernommen wird. Analog können die Entladeaktionen der Aufzüge (und eventuell Quergänge) als spezielle Uebergabeaktionen interpretiert werden, bei welchen eine Palette an das Umsystem übergeben wird. Mit dieser Interpretation verfügen alle Transportmittel über dieselbe Menge von Aktionen: (1) Transportfahrt, (2) Leerfahrt, (3) Uebergabe, (4) Uebernahme.

Allgemein lässt sich feststellen, dass die verschiedenen Transportmittel im Verlaufe eines Lagerprozesses (d.h. bei der Ausführung einer Menge von Ein-, Aus- und Umlagerungen) stets dieselbe Teilsequenz von Aktionen wiederholen. Die Struktur dieser Teilsequenz ist unabhängig vom Typ des Transportmittels (Korridorwagen, Quergangswagen oder Aufzug) und von der Art der Lagerbewegung (Ein-, Aus- oder Umlagerung). Diese Teilsequenz umfasst der Reihe nach die folgenden Aktionen:

1. Leerfahrt
2. eventuelle Wartephase, unbeladen
3. Uebernahme
4. Transport
5. eventuelle Wartephase, beladen
6. Uebergabe

Hierbei stellen die Wartephasen keine eigentlichen Transportaktionen dar. In Hinblick auf die Ablaufplanung muss unterschieden werden zwischen beladenen und unbeladenen Wartephasen. Ein Transportmittel, welches in beladenem Zustand wartet, ist nicht verfügbar für die Bearbeitung anderer Paletten. In diesem Sinne kann eine beladene Wartephase als spezielle Aktion aufge-

fasst werden, welche analog zu den andern Aktionen die Ressource (d.h. das Transportmittel) belegt und für andere Aktivitäten nicht verfügbar macht. Im Gegensatz dazu ist bei einer Wartephase in unbeladenem Zustand das Transportmittel prinzipiell verfügbar für andere Aktivitäten. Als Konvention werden im folgenden bei der Darstellung von Ablaufplänen beladene Wartephasen als schmale, schraffierte Balken eingezeichnet, während unbeladene Wartephasen nicht eingezeichnet werden.

Dauer der Transportaktionen

Für zeitliche Betrachtungen muss unterschieden werden zwischen Aktionen mit vorgegebener Dauer und solchen mit kontextabhängiger Dauer.

Bei Aktionen mit vorgegebener Dauer ist der Zeitbedarf im voraus bestimmt durch die Transportaufträge und unabhängig von der Einplanung der Aktionen. Dies trifft zu für die Uebergabe- und Uebernahmeaktionen (bzw. Lade- und Entladeaktionen) sowie für die Transportfahrten.

Bei Aktionen mit kontextabhängiger Dauer ist der Zeitbedarf abhängig von der Einplanung der Aktionen. Dies trifft zu für die Leerfahrten. Die Dauer einer Leerfahrt, d.h. einer Anfahrt von einer ursprünglichen Position zu einer Uebernahmeposition, ist abhängig von der Ausführungsreihenfolge der Aktionen des betreffenden Transportmittels. Die Ausgangsposition einer Leerfahrt entspricht der letzten Position, an welcher eine Palettenübergabe erfolgt ist. Je nachdem, an welcher Stelle diese Uebergabe stattgefunden hat, wird für die Leerfahrt mehr oder weniger Zeit benötigt.

Die Dauer der beladenen Wartephasen ist ebenfalls kontextabhängig. Eine beladene Wartephase kann aus Synchronisationsgründen notwendig sein vor einer Uebergabeaktion. Die Wartezeit ist davon abhängig, zu welchem Zeitpunkt die zugehörige Uebernahmeaktion auf dem entsprechenden Transportmittel eingeplant ist.

Es können zwei Arten von Kontextabhängigkeit unterschieden werden. Leerfahrtzeiten sind reihenfolgeabhängig, aber unabhängig von der konkreten zeitlichen Einplanung der Aktionen. Beladene Wartezeiten sind stärker kontextabhängig und werden durch die zeitliche Einplanung anderer Aktionen bestimmt.

Jobs und Operationen

In diesem Abschnitt werden automatisierte Kompaktlager von einem abstrahierten Standpunkt aus betrachtet. Dadurch wird eine Verbindung hergestellt zu den Konzepten und der Terminologie der allgemeinen Maschinenablaufplanung. Ein automatisiertes Kompaktlager kann aufgefasst werden als ein Verarbeitungssystem bestehend aus verschiedenen Prozessoren (bzw. Maschinen, Ressourcen), auf welchen eine Menge von Operationen ausgeführt wird. Die Prozessoren entsprechen den Transportmitteln, wobei drei Typen von Prozessoren unterschieden werden:

Korridorwagen, Quergangwagen und Aufzüge. Die Objekte, welche von den Prozessoren bearbeitet werden, entsprechen den Paletten.

Die gesamte Bearbeitung eines Objekts wird als Job bezeichnet. Ein Job entspricht somit einem Transportauftrag, d.h. der Ein-, Aus- oder Umlagerung einer einzelnen Palette. Ein Job besteht aus einer Folge von Operationen, wobei jede Operation die Bearbeitung des betreffenden Objekts auf einem Prozessor darstellt. Die Operationen entsprechen somit der "Bearbeitung" der Paletten auf den verschiedenen Transportmitteln. Die dreistufige Auslagerung aus Abbildung 2.3. umfasst beispielsweise drei Operationen, welche der Reihe nach den Transportaktionen des Korridorwagens, des Quergangwagens und des Aufzugs entsprechen.

Die einzelnen Operationen setzen sich aus einer Folge von Transportaktionen zusammen. Weiter oben wurde gezeigt, dass alle Operationen dieselbe Struktur haben, d.h. dieselbe Folge von Aktionen umfassen: (1) Uebernahme, (2) Transport, (3) beladene Wartephase, (4) Uebergabe.

Die Leerfahrten-Aktionen werden nicht in die Operationen miteinbezogen. Sie können in der Terminologie der Maschinenbelegungsplanung aufgefasst werden als sogenannte Setup-Zeiten (bzw. Umrüstzeiten). Setup-Zeiten repräsentieren allgemein den Zeitbedarf für Aktivitäten, welche nach der Bearbeitung einer Operation durchgeführt werden müssen, um den betreffenden Prozessor in einen Zustand zu versetzen, welcher die Bearbeitung der nächsten Operation erlaubt. Es wird unterschieden zwischen konstanten und reihenfolgeabhängigen Setup-Zeiten. Die Leerfahrten stellen reihenfolgeabhängige Setup-Zeiten dar.

Aus Gründen, welche in Kapitel 4 ersichtlich werden, ist es zweckmässig, die Operationen aufzuspalten in zwei Teiloperationen, eine sogenannte Head-Operation und eine Tail-Operation. Die Head-Operation umfasst die Uebernahme- und die Transport-Aktion, während die Tail-Operation der Uebergabe-Aktion entspricht. Zu beachten ist, dass die Dauer dieser Teiloperationen vorgegeben ist durch die Transportaufträge. Im Gegensatz dazu ist die Dauer der gesamten Operation kontextabhängig, da die Operation eine kontextabhängige Wartephase enthält.

Die untenstehende Abbildung zeigt das weiter oben besprochene Auslagerungsbeispiel, wobei die Operationen und deren Teiloperationen graphisch hervorgehoben sind.

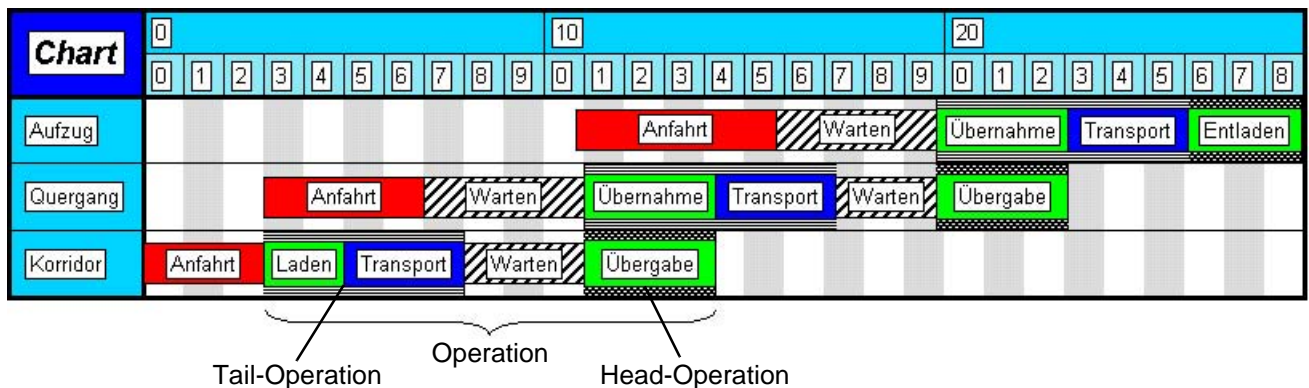


Abbildung 2.4. Operationen und Teiloperationen

Im folgenden werden automatisierte Kompaktlager häufig auf der abstrakten Ebene eines Verarbeitungssystems betrachtet. Dementsprechend werden in der Terminologie die lagerspezifischen Ausdrücke wie Transportmittel, Transportauftrag und Transportaktion zunehmend ersetzt durch allgemeine Begriffe wie Prozessor, Job und Operation.

2.3. Netzwerk-Modell zur Systembeschreibung

In diesem Abschnitt wird ein Modell vorgeschlagen, mit welchem sich automatisierte Kompaktlager auf adäquate Weise formal darstellen lassen. Die Modellierung basiert auf einem gerichteten, gewichteten Graphen, welcher als System-Graph bezeichnet wird.

Der System-Graph repräsentiert die topologische Struktur eines automatisierten Kompaktlagers (bzw. dessen Transportsystems) und umfasst die wesentlichen Systemparameter, welche bei der Steuerung der Transportprozesse von Bedeutung sind. Mit Hilfe des System-Graphen lässt sich die Menge aller möglichen Transportrouten auf einfache Weise beschreiben. Für jede Transportroute spezifiziert der Graph die beteiligten Transportmittel sowie die Folge der Aktionen, welche von diesen ausgeführt werden müssen.

2.3.1. Beschreibung

Im folgenden wird gezeigt, wie der System-Graph eines automatisierten Kompaktlagers konstruiert wird. Die Bedeutung der Knoten und Bogen wird erklärt, und es wird gezeigt, wie die möglichen Transportrouten und die entsprechenden Aktionen mit Hilfe des System-Graphen beschrieben werden können.

Knoten

Positionsknoten

Die Knoten des System-Graphen repräsentieren die verschiedenen Positionen im Lager, an welchen die Prozessoren anhalten, um eine Uebergabe- oder Uebernahme-Aktion (bzw. Lage- oder Entlade-Aktion) auszuführen. Zu beachten ist, dass ein Prozessor prinzipiell beliebige Fahrtbewegungen ausführen kann und an jedem Ort auf seiner Fahrtstrecke anhalten kann. Bei vernünftiger Steuerung kann jedoch angenommen werden, dass ein Prozessor nur Halt macht an einer Stelle, wo eine Uebergabe- oder Uebernahme-Aktion stattfindet.

Für Korridorwagen ist die Menge der zu verwaltenden Positionen gegeben durch die Menge der Lagerpositionen im betreffenden Kanal. Zusätzlich kommen die Positionen, an welchen der Lasttransfer zu den angrenzenden Quergangswagen oder Aufzügen erfolgt. Für Quergangswagen sind diejenigen Positionen von Bedeutung, an welchen ein Lasttransfer zu den angrenzenden Korridorwagen oder Aufzügen erfolgt. Für Aufzüge müssen die Transferpositionen zu den angrenzenden Quergang- oder Korridorwagen verwaltet werden sowie die Ein- und Ausgangspunkte, an welchen ein Lasttransfer zum Umsystem erfolgt.

Künstliche Start- und Endknoten

Zusätzlich zu den Knoten, welche die Prozessorpositionen repräsentieren, werden zwei Paare von künstlichen Knoten eingeführt, je ein fiktiver Start- und Endknoten (Quelle und Senke) für die Ein- bzw. Auslagerungen. Diese Knoten werden benötigt für die Darstellung der Auflade- und Ablade-Aktionen.

Bogen

Die Bogen des System-Graphen entsprechen den Aktionen, welche von den Prozessoren ausgeführt werden.

Fahrtbogen

Die Fahrtbogen repräsentieren die Transport- oder Leerfahrten der Prozessoren. Ein Fahrtbogen verbindet zwei Positionsknoten eines Prozessors und entspricht der Fahrt zwischen diesen beiden Positionen. Da zwischen jedem Paar von Positionsknoten eines Prozessors eine Fahrt in beiden Richtungen möglich ist, bilden die Fahrtbogen zusammen mit den Positionsknoten einen vollständigen Subgraphen für jeden Prozessor.

Transferbogen

Die Transferbogen repräsentieren die Uebergabe- bzw. Uebernahme-Aktionen zwischen zwei Prozessoren. Ein Bogen verbindet zwei Positionsknoten unterschiedlicher Prozessoren, an welchen eine Uebergabe- bzw. Uebernahme erfolgen kann. Da der Transfer im allgemeinen in beiden Rich-

tungen erfolgen kann, sind die entsprechenden Knoten jeweils durch zwei Bogen entgegengesetzter Richtung miteinander verbunden.

Aufnahme- und Abgabebogen

Die Aufnahme- bzw. Abgabebogen repräsentieren die Auflade- bzw. Abladeaktionen an den Lagerplätzen und an den Ein- und Ausgangspunkten. Ein Aufnahmebogen für Einlagerungen verbindet die fiktive Einlagerungsquelle mit einer Eingangsposition eines Prozessors (Aufzug oder Quergangswagen). Analog verbindet ein Aufnahmebogen für Auslagerungen die fiktive Auslagerungsquelle mit einer Lagerposition eines Korridorwagens. Ein Abgabebogen für Einlagerungen führt von einer Lagerposition zur fiktiven Einlagerungssenke und ein Abgabebogen für Auslagerungen von einem Ausgangspunkt zur fiktiven Auslagerungssenke.

Bogengewichte

Die Bogengewichte sind gegeben durch die Dauer der verschiedenen Aktionen, welche von den Bogen repräsentiert werden. Die Zeiten für die Fahrt-, Transfer- und Ladeaktionen sind vorgegeben durch die technischen Parameter der verschiedenen Transportmittel. Zu beachten ist, dass die Fahrzeiten der Förderwagen typischerweise stark von den Beschleunigungs- und Abbremsvorgängen abhängig sind und folglich nicht annäherungsweise als konstant angenommen werden können. Dementsprechend ist die Dauer einer Fahrt mit Unterbrechungen im allgemeinen grösser als die Dauer der entsprechenden Direktfahrt. Im System-Graphen bedeutet dies, dass in den vollständigen Subgraphen, welche die Fahrten der Prozessoren repräsentieren, die Bogengewichte die Dreiecksungleichung erfüllen. Diese besagt, dass die Summe der Bogengewichte einer Kantenfolge mindestens so gross wie das Gewicht des entsprechenden transitiven Bogens ist.

Transportrouten und Aktionen

Ein Weg im System-Graphen stellt eine Bogenfolge dar und entspricht somit einer Folge von Aktionen.

Jede mögliche Lagerbewegung (d.h. Ein-, Aus- oder Umlagerung) lässt sich darstellen als Weg von einer Quelle zu einer Senke. Eine Ein- bzw. Auslagerung entspricht einem Weg von der Ein- bzw. Auslagerungsquelle zur Ein- bzw. Auslagerungssenke. Eine Umlagerung entspricht einem Weg von der Auslagerungsquelle zur Einlagerungssenke. Umgekehrt lässt sich in analoger Weise jeder Weg von einer Quelle zu einer Senke interpretieren als eine Lagerbewegung (angenommen die Wege von der Einlagerungsquelle zur Auslagerungssenke).

Somit lässt sich die Menge aller möglichen Lagerbewegungen mit Hilfe des System-Graphen auf einfache Weise beschreiben. Die Menge aller möglichen Ein- bzw. Auslagerungen ist gegeben durch die Menge aller Wege von der Ein- bzw. Auslagerungsquelle zur Ein- bzw. Auslagerungssenke. Die Menge aller möglichen Umlagerungen entspricht der Menge aller Wege von der Auslagerungsquelle zur Einlagerungssenke.

Hierbei ist zu beachten, dass mit dieser Beschreibung auch Lagerbewegungen erfasst werden, bei welchen die Prozessoren Zwischenhalte einlegen auf den Transportfahrten. Die Menge aller Lagerbewegungen ohne Zwischenhalte ist gegeben durch die Menge aller oben beschriebenen Wege, für welche keine transitiven Bogen existieren. Dies ist so zu verstehen, dass kein Teilweg (von Transportbogen) existiert, welcher durch einen einzelnen (Transport-)Bogen ersetzt werden kann.

Zusammenfassend wurde gezeigt, wie mit Hilfe des System-Graphen die Menge aller möglichen Lagerbewegungen in einem automatisierten Kompaktlager auf formale Weise beschrieben werden kann. In Kapitel 3 wird gezeigt, wie auf der Basis des System-Graphen ein Modell zur Leistungsabschätzung automatisierter Kompaktlager entworfen werden kann.

Der System-Graph spielt auch auf der Ebene der dispositiven Steuerung eine wichtige Rolle. In Abschnitt 2.2. wurde gezeigt, dass die dispositive Steuerung eine Komponente umfasst, welche für die Bestimmung der Transportaufträge (Routing) zuständig ist. Die Aufgabe dieser Komponente besteht darin, aus den vorgegebenen Ein-, Aus- und Umlagerungsaufträgen eine Menge von Transportaufträgen zu generieren. Im System-Graphen entspricht dies der Aufgabe, eine Menge von Ein-, Aus- und Umlagerungswegen zu bestimmen, welche die vorgegebenen Aufträge erfüllt. Eine Menge von Transportaufträgen (bzw. eine Menge von Jobs) lässt sich somit formal spezifizieren als eine Menge von Wegen im System-Graphen. Durch diese Wege werden die Aktionen der Transportaufträge (bzw. die Operationen der Jobs) eindeutig beschrieben. Die formale Spezifikation der Transportaufträge mit Hilfe des System-Graphen wird in Kapitel 4 benötigt beim Entwurf einer optimierten Steuerung für automatisierte Kompaktlager.

2.3.2. Beispiel

In diesem Abschnitt wird die Konstruktion des System-Graphen anhand eines Beispiels illustriert. Eine formale Definition des System-Graphen erfolgt im nächsten Abschnitt. Die untenstehende Abbildung zeigt einen kleinen Lagerausschnitt, für welchen der System-Graph konstruiert wird. Der Ausschnitt umfasst zwei Korridore, zwei Quergänge und einen Aufzug.

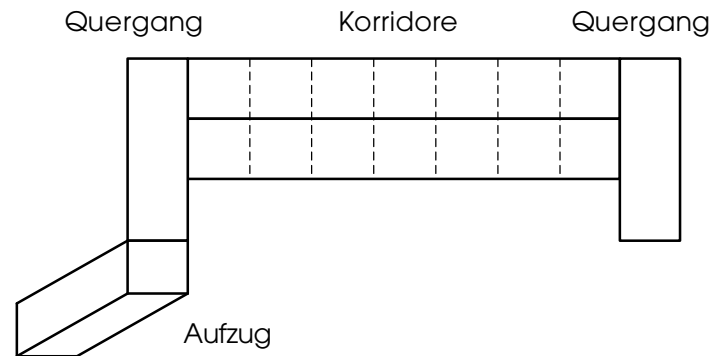


Abbildung 2.5. Lagerausschnitt für Konstruktion des System-Graphen

Die folgende Abbildung zeigt den System-Graphen zu diesem Beispiel. Zur Vereinfachung ist in jedem Korridor nur eine der sieben Lagerpositionen eingezeichnet. Jeder Korridor umfasst somit eine Lagerposition und an jedem Ende eine Uebergabe- bzw. Uebernahmeposition für den Lasttransfer zu den angrenzenden Quergangswagen. Für die Quergangswagen müssen zwei Transferpositionen zu den Korridoren und eine zum Aufzug verwaltet werden. Für den Aufzug ist die Transferposition zum Quergangswagen und eine Ausgangsposition eingezeichnet. Die Quelle und Senke für die Ein- bzw. Auslagerungen werden bezeichnet als s und t bzw. \bar{s} und \bar{t} . An Stelle von zwei Bogen entgegengesetzter Richtung wird eine ungerichtete Kante eingezeichnet. Die zweite Abbildung zeigt die Bedeutung der verschiedenen Bogen im System-Graphen.

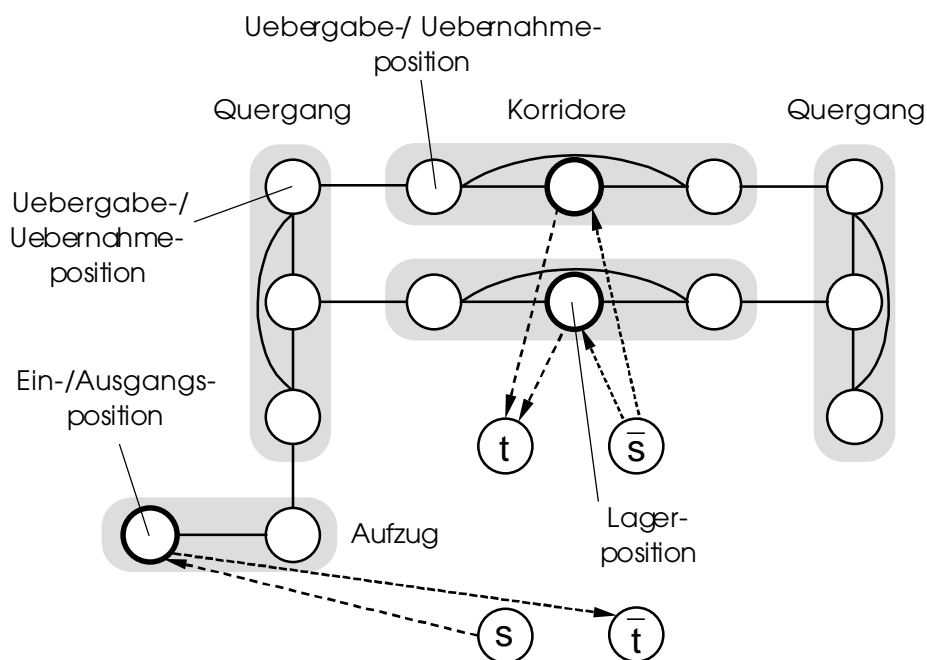


Abbildung 2.6. Bedeutung der Knoten im System-Graphen

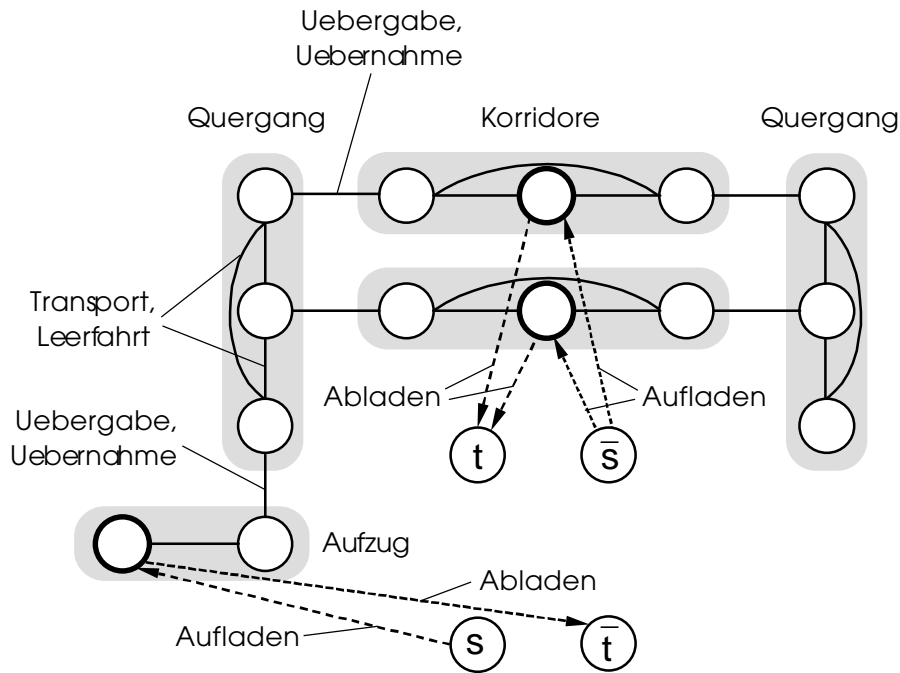


Abbildung 2.7. Bedeutung der Bogen im System-Graphen

2.3.3. Formale Definition des System-Graphen

Nachfolgend wird eine formale Definition des System-Graphen $G = \langle V^0, E^0, c \rangle$ gegeben. G ist ein gerichteter gewichteter Graph mit Knotenmenge V^0 , Bogenmenge E^0 und Bogengewichten $c \in \mathbb{R}^{E^0}$. Zusätzlich zu den oben erwähnten Bogen wird für jeden Positionsknoten v noch eine Schlinge (v, v) eingeführt. Diese Schlingen dienen zur Darstellung von Wartezeiten und werden in Kapitel 3 bei der Leistungsanalyse genauer erläutert.

System-Graph $G = \langle V^0, E^0, c \rangle$

Prozessoren:

- M Gesamtmenge der Förderprozessoren, $M = M^A \cup M^B \cup M^C$
- M^A Menge der Aufzugsprozessoren
- M^B Menge der Quergangprozessoren
- M^C Menge der Korridorprozessoren

Knoten:

- V^0 Gesamtmenge der Knoten, $V^0 = V \cup \{s, t, \bar{s}, \bar{t}\}$
- V Gesamtmenge der Prozessorpositionen, $V = \bigcup_{m \in M} V_m$

V_m	Menge der Positionen von Prozessor m , $m \in M$
s, t	(fiktive) Quelle und Senke für Einlagerungsflüsse
\bar{s}, \bar{t}	(fiktive) Quelle und Senke für Auslagerungsflüsse
V^I	Gesamtmenge der Eingangspositionen, $V^I \subseteq V$, $V^I = \bigcup_{m \in M^A \cup M^B} V_m^I$
V_m^I	Menge der Eingangspositionen von Prozessor m , $m \in M^A \cup M^B$
V^O	Gesamtmenge der Ausgangspositionen, $V^O \subseteq V$, $V^O = \bigcup_{m \in M^A \cup M^B} V_m^O$
V_m^O	Menge der Ausgangspositionen von Prozessor m , $m \in M^A \cup M^B$
V^P	Gesamtmenge der Lagerpositionen in den Korridoren, $V^P \subseteq \bigcup_{m \in M^C} V_m$
V_m^P	Menge der Lagerpositionen in Korridor m , $m \in M^C$

Bogen:

E^0	Gesamtmenge der Bogen, $E^0 = E^S \cup E^W \cup E^* \cup \bar{E}^* \cup E^L$
E^S	Gesamtmenge der Fahrtbogen der Prozessoren, $E^S = \bigcup_{m \in M} E_m^S$
E_m^S	Menge der Fahrstrecken von Prozessor m , $E_m^S = \{(v, w) : v, w \in V_m, v \neq w\}$ $m \in M$
E^T	Gesamtmenge der Transferbogen zwischen den Prozessoren, $E^T \subseteq \{(v, w) : v \in V_m, w \in V_{m'}, m, m' \in M, m \neq m'\}$, $E^T = \bigcup_{m \in M} E_m^T$
E_m^T	Menge der Transferbogen von Prozessor m , $E_m^T = \{(v, w) \in E^T : v \in V_m\} \cup \{(v, w) \in E^T : w \in V_m\}$, $m \in M$
E^*	Gesamtmenge der Aufnahme- und Abgabebogen für Einlagerungen, $E^* = \{(s, w) : w \in V^I\} \cup \{(v, t) : v \in V^P\}$, $E^* = \bigcup_{m \in M} E_m^*$
E_m^*	Menge der Aufnahme- und Abgabebogen für Einlagerungen von Prozessor m , $E_m^* = \{(s, w) \in E^* : w \in V_m\} \cup \{(v, t) \in E^* : v \in V_m\}$, $m \in M$
\bar{E}^*	Gesamtmenge der Aufnahme- und Abgabebogen für Auslagerungen, $\bar{E}^* = \{(\bar{s}, w) : w \in V^P\} \cup \{(v, \bar{t}) : v \in V^O\}$, $\bar{E}^* = \bigcup_{m \in M} \bar{E}_m^*$
\bar{E}_m^*	Menge der Aufnahme- und Abgabebogen für Auslagerungen von Prozessor m , $\bar{E}_m^* = \{(\bar{s}, w) \in \bar{E}^* : w \in V_m\} \cup \{(v, \bar{t}) \in \bar{E}^* : v \in V_m\}$, $m \in M$
E^L	Gesamtmenge der Wartebogen (Schlingen), $E^L = \bigcup_{m \in M} E_m^L$
E_m^L	Menge der Wartebogen (Schlingen) von Prozessor m , $E_m^L = \{(v, v) : v \in V_m\}$ $m \in M$
E	Menge aller Bogen für Einlagerungsflüsse, $E = E^S \cup E^T \cup E^*$
E_m	Menge der Bogen für Einlagerungsflüsse von Prozessor m , $E_m = E_m^S \cup E_m^T \cup E_m^*$ $m \in M$
\bar{E}	Menge aller Bogen für Auslagerungsflüsse, $\bar{E} = E^S \cup E^T \cup \bar{E}^*$
\bar{E}_m	Menge der Bogen für Auslagerungsflüsse von Prozessor m , $\bar{E}_m = \bar{E}_m^S \cup \bar{E}_m^T \cup \bar{E}_m^*$ $m \in M$

Bogengewichte:

c_e	Zeitbedarf für die durch den Bogen e repräsentierte Prozessor-Aktion, $e \in E^0$
	• für $e = (v, w) \in E^S$: Dauer einer Fahrt von Position v nach Position w

- für $e = (v, w) \in E^T$: Dauer einer Palettenübergabe von Position v nach Position w
- für $e = (s, w) \in E^*$: Dauer einer Palettenaufnahme an Eingangsposition w
- für $e = (v, t) \in E^*$: Dauer des Abladens einer Palette an Lagerposition v
- für $e = (\bar{s}, w) \in \bar{E}^*$: Dauer des Aufladens einer Palette an Lagerposition w
- für $e = (v, \bar{t}) \in \bar{E}^*$: Dauer einer Palettenabgabe an Auslagerungsposition v
- für $e = (v, v) \in E^L$: untere Schranke für die Wartezeit eines Prozessors, wenn dieser an Position v eine Palette übergibt (oder ablädt) und anschliessend direkt wieder eine Palette an Position v übernimmt (bzw. auflädt)

Der Subgraph $\langle V^0, E, c \rangle$ bzw. $\langle V^0, \bar{E}, c \rangle$ wird als Ein- bzw. Auslagerungsgraph von \mathcal{G} bezeichnet. Als gemeinsame Bogen umfassen diese beiden Subgraphen die Fahrtbogen E^S und die Transferbogen E^T . Zusätzlich enthält der Einlagerungsgraph die Aufnahme- und Abgabebogen E^* , welche mit der Einlagerungsquelle s bzw. -senke t inzidieren. Analog enthält der Auslagerungsgraph die Aufnahme- und Abgabebogen \bar{E}^* , welche mit der Auslagerungsquelle \bar{s} bzw. -senke \bar{t} inzidieren.

Reduktion des System-Graphen

Bei der Entwicklung eines Modells zur Leistungsabschätzung in Kapitel 3 wird aus Gründen der Modellgrösse von einem reduzierten System-Graphen ausgegangen, bei welchem die Lagerplätze eines Korridors nicht einzeln verwaltet, sondern zu einem einzigen Lagerknoten zusammengefasst werden. Gemäss obiger Definition des System-Graphen wird jeder Lagerplatz durch einen eigenen Knoten repräsentiert. Aufgrund der grossen Anzahl von Lagerplätzen führt dies zu einer umfangreichen Menge von Knoten im System-Graphen. Die Zahl der Fahrtbogen eines Korridorprozessors wächst quadratisch mit der Anzahl der Lagerpositionen des Korridors, so dass eine enorme Anzahl von Korridor-Fahrtbogen resultiert. Die Grösse des System-Graphen kann beträchtlich reduziert werden, wenn die Lagerplätze eines Korridors zu einem einzigen Knoten (oder zu einigen wenigen Knoten) zusammengefasst werden. In diesem Falle können die Lagerknoten in den Korridoren als fiktive Lagerplätze interpretiert werden, an welchen mehrere Paletten gleichzeitig gelagert werden können. Die Kapazität eines Lagerknotens entspricht der Anzahl Lagerplätze, welche durch den Knoten repräsentiert werden. Falls die Lagerplätze eines Korridors durch einen einzigen Lagerknoten dargestellt werden, ist die Kapazität dieses Knotens gegeben durch die Anzahl Lagerplätze im Korridor. Die Fahrzeiten zu diesen fiktiven Lagerknoten entsprechen der durchschnittlichen Fahrzeit zu den repräsentierten Lagerplätzen. Auf die Leistungsabschätzung hat diese Reduktion keinen wesentlichen Einfluss, da die Korridorprozessoren typischerweise wenig ausgelastet sind und keine Engpass-Ressourcen darstellen.

KAPITEL 3

Modell zur Leistungsanalyse

Bei der Konzeption automatisierter Kompaktlager ist die Leistungsanalyse von wichtiger Bedeutung. Die Evaluation verschiedener Entwurfsalternativen bedingt eine Methodik zur Abschätzung der zu erwartenden Lagerleistung. Als Leistungsmass wird die maximale Anzahl Ein- und Auslagerungen betrachtet, welche in einem bestimmten Zeitraum möglich ist. Auf Simulation basierende Ansätze sind aufgrund der Komplexität der Steuerungsprozesse nur schwer realisierbar. In dieser Arbeit wird ein Lineares Optimierungsmodell für die Leistungsanalyse entwickelt, welches steuerungsunabhängig ist und ein Minimum von Eingabedaten benötigt. Das Modell basiert auf Ressourcenallokations-Ueberlegungen und verwendet eine Darstellung der Ein- und Auslagerungen als Mehrgüterflüsse im System-Graphen. Durch Flussoptimierung lässt sich die Belastung der Förderprozessoren bei optimaler Allokationsstrategie abschätzen, woraus eine obere Schranke für die Lagerleistung resultiert. Als spezielle Modellkomponenten werden die reihenfolgeabhängigen Anfahrtszeiten der Förderprozessoren und die Verweilzeiten der Artikel im Lager berücksichtigt.

Abschnitt 3.1. bespricht den Entwurf automatisierter Kompaktlager und diskutiert die Problematik der Leistungsanalyse. Dabei wird auch auf Erfahrungen aus der Praxis eingegangen. In Abschnitt 3.2. wird gezeigt, wie sich Ein- und Auslagerungen als Flüsse im System-Graphen darstellen lassen. Daraus wird ein Lineares Optimierungsmodell zur Leistungsabschätzung entwickelt. In Abschnitt 3.3. werden die Leerfahrten, d.h. die reihenfolgeabhängigen Anfahrtswege der Prozessoren in das Modell integriert. Abschnitt 3.4. zeigt, wie die Verweilzeiten der Artikel im Lager durch das Modell berücksichtigt werden können. Abschnitt 3.5. gibt eine vollständige

Übersicht über das entwickelte Modell zur Leistungsanalyse. In Abschnitt 3.6. werden numerische Beispiele präsentiert, welche die Validität und die Anwendungsmöglichkeiten des Modells illustrieren.

3.1. Problembeschreibung

3.1.1. Entwurf automatisierter Kompaktlager

Automatisierte Kompaktlager sind massgeschneiderte Systeme, welche für jeden Kunden individuell entworfen und an dessen spezifische Bedürfnisse angepasst werden. Der konzeptuelle Entwurf findet sowohl auf der physischen Ebene der Lager- und Transporteinrichtungen als auch auf der logischen Ebene des Steuerungssystems statt.

Der Entwurf der Lager- und Transporteinrichtungen beinhaltet einerseits die Dimensionierung und andererseits die Anordnung der verschiedenen Einrichtungen. Dabei müssen unter anderem die folgenden Entscheidungsgrößen festgelegt werden:

- Dimensionierung:
Anzahl Sektoren, Anzahl Lagerebenen, Anzahl Korridore pro Sektor, Anzahl Lagerplätze pro Korridor
- Anordnung:
Anordnung der Aufzüge (an Quergang-Enden oder innerhalb von Korridoren),
Anordnung der Ein- und Ausgangspunkte (bei Aufzügen oder an Quergang-Enden)

Der Entwurf des Steuerungssystems ist eine komplexe Aufgabe, welche eine Vielzahl von Entscheidungsdimensionen umfasst. In Abschnitt 2.2. wurde die Funktionsweise der Steuerung und insbesondere des dispositiven Steuerungssystems besprochen. Zusammenfassend können die folgenden Hauptkomponenten unterschieden werden:

- Bestimmung der Transportrouten:
Wahl der Ein- und Auslagerungspositionen (Lagerstrategie), Wahl der Ein- und Ausgangspunkte, Wahl der Transportroute
- Bestimmung des Ablaufplans:
Wahl der Ausführungsreihenfolge, Wahl der Startzeitpunkte der Operationen
- Bestimmung der Umlagerungen und Reorganisation:
Bestimmung der erzwungenen Umlagerungen, Bestimmung der Reorganisations-Strategie

Planungsvorgaben

Bei der Konzeption eines automatisierten Kompaktlagers werden vom Kunden Anforderungen spezifiziert, welche vom Lager zu erfüllen sind. Die Vorgaben lassen sich in folgende Bereiche aufteilen:

- Lagerdimensionen und Anordnung der Einrichtungen
- Lagerkapazität
- Lagerleistung
- Steuerstrategien

Lagerdimensionen und Anordnung der Einrichtungen

Häufig sind aufgrund des verfügbaren Grundstücks Vorgaben hinsichtlich der Grösse der Lagergrundfläche gegeben. Dazu kommen eventuelle Vorgaben über maximale Gebäudehöhen, sei es aufgrund von Bauvorschriften oder aufgrund konstruktiver Einschränkungen. Falls das Lager in ein bestehendes Gebäude eingebaut wird oder mit vorhandenen Gebäuden und Einrichtungen verbunden wird, ergeben sich weitere Vorgaben hinsichtlich der Dimensionierung und Anordnung der Lager- und Transporteinrichtungen.

Lagerkapazität

Im allgemeinen wird die Gesamtzahl der Palettenplätze spezifiziert, welche das Lager bereitstellen muss (z.B. 12000 Lagerplätze). Bisweilen existieren zusätzliche Vorgaben, dass gewisse Lagerbereiche eine bestimmte Anzahl von Lagerplätzen umfassen müssen. Häufig sind noch Angaben verfügbar über die Anzahl der gelagerten Artikel und die Anzahl Lagerplätze, welche für jeden Artikel vorgesehen werden muss.

Lagerleistung

Als Vorgaben für die Lagerleistung wird typischerweise die Anzahl Ein- und Auslagerungen angegeben, welche vom Lagersystem in einer bestimmten Zeitperiode (z.B. in einem Tag) bewältigt werden muss. Bisweilen wird unterschieden zwischen Normalbetrieb und Maximalbetrieb. Auf die Problematik der Leistungsabschätzung wird weiter unten noch näher eingegangen

Steuerstrategien

Hinsichtlich der Entscheidungsstrategien auf der Steuerungsebene bestehen in der Entwurfsphase im allgemeinen keine oder nur wenige Vorgaben. Die Konzeption des Steuerungssystems erfolgt typischerweise erst nach dem physischen Entwurf des Lagersystems. Bisweilen werden gewisse grundlegende Vorgaben für die Lagerstrategien gemacht. Es kann beispielsweise die Forderung bestehen, dass die Lagereinheiten artikelrein gelagert werden, d.h. dass ein Lagerkanal oder ein bestimmter Lagerbereich nur mit Paletten desselben Artikels belegt wird. Eine weitere typische Vorgabe ist beispielsweise die Forderung, dass die Lagerstrategie nach dem FIFO-Prinzip (first in first out) erfolgen muss.

3.1.2. Leistungsanalyse

Die Offertenstellung bei einem Kunden beinhaltet die Konzeption, die Spezifikation und die Evaluation verschiedener Entwurfsalternativen. Eine besondere Bedeutung kommt hierbei der Leistungsanalyse zu. Die Abschätzung der zu erwartenden Lagerleistung bildet eine der Hauptschwierigkeiten bei der Evaluation von Design-Alternativen. Die übrigen Planungsvorgaben (Dimension und Anordnung der Einrichtungen, Lagerkapazität) sind statische, betriebsunabhängige Werte, welche sich bei gegebener Planungsalternative leicht überprüfen lassen. Die Lagerleistung jedoch ist ein dynamischer Wert, der neben dem physischen Lagerlayout in starkem Masse vom eingesetzten Steuerungssystem abhängig ist. Die Problematik dabei ist, dass in der Phase des Entwurfs das Steuerungssystem im allgemeinen nicht oder nur in Ansätzen spezifiziert ist.

Die Abschätzung der Lagerleistung bei automatisierten Kompaktlagern ist eine komplexe Aufgabe, für deren Bewältigung wenig methodisches Knowhow vorhanden ist. Es lassen sich zwei methodische Ansätze unterscheiden: Simulation und analytische Methoden.

Simulation

Die Simulation eines automatisierten Kompaktlagers bedingt eine Nachbildung des physischen Lagersystems sowie des gesamten Steuerungssystems. Dies bildet aus zwei Gründen eine aufwendige und anspruchsvolle Aufgabe. Erstens zeichnen sich automatisierte Kompaktlager dadurch aus, dass sie über eine grosse Zahl (Hunderte bis Tausende) von autonomen Förderprozessoren verfügen, welche alle als eigenständige Entitäten im Simulationsmodell verwaltet werden müssen. Zweitens ist die Steuerungslogik im Vergleich zu anderen Lagertypen sehr komplex und hängt von vielen Entscheidungsfaktoren ab. Ein wesentlicher Grund dafür ist, dass im Unterschied zu vielen klassischen Lagersystemen automatisierte Kompaktlager ein mehrstufiges Transportsystem haben, bei welchem mehrere Förderprozessoren an den Lagerbewegungen beteiligt sind. Aufgrund der Synchronisationsbedingungen bei der Lastübergabe müssen die Aktionen der Prozessoren zeitlich aufeinander abgestimmt werden. Dadurch wird die Komplexität des Steuerungssystems wesentlich erhöht.

Eine weitere Problematik bei der Simulation ist die schon erwähnte Tatsache, dass in der Phase des Entwurfs die Steuerungslogik typischerweise noch nicht spezifiziert ist. Für automatisierte Kompaktlager existieren keine Standard-Softwarelösungen für die Lagersteuerung. Das Steuerungssystem wird für jedes Lager individuell entworfen und den Kundenbedürfnissen angepasst. Somit fehlen beim Entwurf eines Simulationsmodells wichtige Systeminformationen, welche einen wesentlichen Einfluss auf die Simulationsergebnisse haben. Zur Lösung dieser Problematik könnte man versuchen, eine optimale Steuerung zu entwickeln und diese im Simulationsmodell zu implementieren. Dadurch könnten Aussagen gemacht werden über die maximale Leistung, welche mit einem Lagersystem erreicht werden kann. Der Versuch, eine global optimale Steuerung zu entwickeln, ist jedoch unrealistisch aufgrund der Komplexität der Steuerungslogik (vgl.

Abschnitt 1.4.). Somit basiert ein Simulationsmodell gezwungenermassen auf einem Steuerungssystem, welches von Annahmen und Vereinfachungen ausgeht, deren Angemessenheit schwer zu überprüfen ist. Dadurch wird die Aussagekraft der Simulationsergebnisse teilweise in Frage gestellt.

Schliesslich sei auf eine weitere Problematik aufmerksam gemacht, welche vielen Simulationsmodellen anhaftet. Aufgrund der Komplexität des Transport- und Steuerungssystems hängt die Simulation von einer Vielzahl von unterschiedlichen Parametern ab. Erfahrungsgemäss sind Simulationsmodelle mit grosser Parameterzahl schwer kontrollierbar, da es schwierig ist, den Einfluss der verschiedenen Parameter auf das System zu verstehen und adäquate Parameterkonstellationen zu bestimmen.

Zusammenfassend wurde in diesem Abschnitt gezeigt, dass der Einsatz von Simulationen zur Leistungsabschätzung automatisierter Kompaktlager aufwendig und in verschiedener Hinsicht problematisch ist. An dieser Stelle sei auch die Arbeit von [Ludwig, S., Malloth, C. (1990)] erwähnt, welche sich mit der Simulation automatisierter Kompaktlager befasst und die beschriebene Problematik in verschiedenen Punkten bestätigt.

Analytische Methoden

Analytische Methoden versuchen im wesentlichen, einen mathematischen Zusammenhang zwischen den Parametern eines Lagersystems und der Lagerleistung herzustellen. Im Bezug auf die Leistungsabschätzung automatisierter Kompaktlager sind dem Autor zwei Beiträge bekannt, welche auf einem analytischen Ansatz beruhen. Beide Arbeiten basieren auf dem Konzept des Lagerspiels. Ein Lagerspiel stellt eine kombinierte Lagerbewegung dar, bei welcher eine Einlagerung und daran anschliessend eine Auslagerung von einem andern Lagerplatz erfolgt. Lagerspiele werden häufig eingesetzt bei der Leistungsabschätzung von traditionellen Lagersystemen.

Die Arbeit von [Schulte-Zurhausen, M. (1982)] beschäftigt sich mit statischen, nicht automatisierten Blocklagern. Zur Leistungsabschätzung wird das Lager in verschiedene Bereiche unterteilt, und anschliessend wird für jeden Bereich der mittlere Zeitbedarf für ein Lagerspiel berechnet. Die Lagerbereiche werden verschiedenen Artikelklassen (Schnelldreher, Langsamdreher) zugeordnet, welche sich bezüglich ihrer Verweilzeit bzw. ihrer Zugriffshäufigkeit unterscheiden. Aufgrund der Zugriffshäufigkeiten werden die Lagerspiele auf die Lagerbereiche verteilt, woraus eine Abschätzung der mittleren Lagerleistung berechnet wird. Der zweite Beitrag von [VDI (1996)] ist eine offizielle Richtlinie des Vereins Deutscher Ingenieure, welche Normen für die Leistungsabschätzung von automatischen Kanallager-Systemen festlegt. Die Arbeit basiert auf ähnlichen Überlegungen wie der vorhergehende Beitrag. Es wird im wesentlichen gezeigt, wie die Lagerspielzeiten in einem mehrstufigen Kompaktlager abgeschätzt werden können.

Diese Beiträge gehen von zwei grundlegenden Konzepten aus: Erstens werden die Ein- und Auslagerungen zu Lagerspielen zusammengefasst. Zweitens werden die Lagerbewegungen nach

bestimmten statistischen Plausibilitätsregeln auf die Lagerplätze verteilt. Als Kritik an diesen Ansätzen kann angeführt werden, dass sie von vereinfachenden Annahmen ausgehen, deren Auswirkung auf die berechneten Leistungszahlen schwer abschätzbar ist. Es besteht keine Information darüber, ob die Abschätzung der Leistung zu hoch oder zu tief ausfällt. Dadurch sind die Aussagen über die Lagerleistung mit einem Unsicherheitsfaktor behaftet, welcher deren Nützlichkeit unter Umständen in Frage stellt.

Als weiterer Beitrag zum Thema sei die Arbeit von [Andenmatten, K. (1992)] erwähnt. Diese Arbeit beschäftigt sich mit der Leistungsabschätzung automatisierter Kompaktlager, wobei nur elementare Konfigurationen bestehend aus einer Ebene und einem Sektor betrachtet werden. Es wird gezeigt, dass in diesem Fall das Problem der Leistungsabschätzung als spezielles Traveling Salesman Problem formuliert werden kann.

Zusammenfassend haben diese Ausführungen gezeigt, dass ein Mangel besteht an geeigneten Methoden zur Leistungsabschätzung automatisierter Kompaktlager. Die praktischen Erfahrungen, welche im nächsten Abschnitt geschildert werden, bestätigen diesen Umstand.

Die vorliegende Arbeit versucht, einen Beitrag zur Behebung dieses Mangels zu leisten. Im folgenden wird ein analytisches Modell zur Leistungsabschätzung entwickelt, welches unabhängig ist von den Entscheidungsmechanismen auf der Ebene der Lagersteuerung und ein Minimum an Eingabedaten benötigt. Es handelt sich um ein Lineares Optimierungsmodell auf der Basis von Mehrgüterflüssen. Die Ein- und Auslagerungen werden als Palettenflüsse im System-Graphen dargestellt. Durch Flussoptimierung lässt sich die Belastung der Förderprozessoren bei optimaler Allokationsstrategie abschätzen, woraus eine obere Schranke für die Lagerleistung resultiert. Diese Leistungsschranke hängt ausschliesslich von der Lagerarchitektur ab und bildet somit eine verlässliche Aussage über die maximal mögliche Leistung eines automatisierten Kompaktlagers. Der Anstoss zur Entwicklung dieses Modells entstammt ursprünglich den praktischen Erfahrungen, welche im folgenden geschildert werden.

Erfahrungen aus der Praxis

Die Problemstellungen dieser Arbeit ergaben sich aus dem Kontakt mit einem Logistik-Unternehmen, welches unter anderem automatisierte Kompaktlager konstruiert. Die praktischen Erfahrungen dieses Unternehmens beim Entwurf und Betrieb automatisierter Kompaktlager zeigten Probleme auf, welche mit den verfügbaren Methoden kaum zu bewältigen waren. Es handelte sich einerseits um die Problematik der Leistungsabschätzung und andererseits um die Entwicklung einer effizienten Steuerung.

Im Rahmen dieses Firmenkontakts wurden wir mit einem Projekt konfrontiert, welches in deutlicher Weise den Bedarf von entscheidungsunterstützenden Methoden beim Entwurf von automatisierten Kompaktlagern zeigt. Das Unternehmen hatte für einen Kunden ein grosses Kompaktlager konstruiert, bei welchem sich nach der Fertigstellung herausstellte, dass es kaum

die Hälfte der geforderten Leistung erbrachte. Die Gründe für diese massive Fehleinschätzung der Lagerleistung waren keineswegs offensichtlich. Die Sachlage wurde dadurch erschwert, dass die Entwicklung der Steuerungs-Software nicht vom Logistik-Unternehmen selbst ausgeführt wurde, sondern vom Kunden an eine Drittfirma übergeben wurde. Bei anderen Kompaktlager-Projekten wurde die Steuerung von der Logistik-Firma selbst entwickelt. Daraus ergab sich der folgende Konflikt: Aus der Sicht der Logistik-Firma lag der Grund für die unzureichende Leistung an einer mangelhaft implementierten Steuerungs-Software. Die Architektur und Dimensionierung des Lagers war nach ihrer Auffassung in Ordnung. Auf der Seite des Kunden wurde die fehlende Leistung einer mangelhaften Lagerkonstruktion zugeschrieben, die Steuerungs-Software jedoch nicht in Frage gestellt. Eine weitere Schwierigkeit bei der Beurteilung des Sachverhalts ergab sich aus der Tatsache, dass der Kunde das Lager mit einer grösseren Artikelzahl (bei artikelreiner Lagerung) betrieb, als in der Offerte vorgesehen war. Welchen Einfluss die veränderten Artikelzahlen auf die Lagerleistung hatten, war schwer abzuschätzen.

Beide Konfliktparteien liessen Gutachten erstellen, welche die Verantwortlichkeit für den Leistungsmangel klären sollten. Wir erhielten Einblick in das Gutachten des Logistik-Unternehmens und wurden angefragt, ob wir dieses näher untersuchen und beurteilen könnten. Das Gutachten machte keine klaren Aussagen und war in seinen Ausführungen teilweise schwer nachvollziehbar. Methodisch wurde eine Leistungsanalyse durchgeführt, welche ausgehend von einer statistischen Verteilung der Lagerbewegungen die Belastung der Prozessoren berechnete.

Inspiziert von diesem Gutachten entschieden wir, ein eigenes Modell zur Leistungsabschätzung zu entwickeln. Daraus resultierte schliesslich das vorliegende Fluss-Modell. Eine Leistungsanalyse des betreffenden Lagersystems mit unserem Modell ergab, dass der Leistungsmangel im wesentlichen durch eine ungünstige System-Architektur verursacht wurde. Alle Aufzüge waren in der Mitte des Lagers innerhalb der Lagerkanäle eines einzigen Sektors positioniert. Als Folge davon musste jede ein- und ausgelagerte Palette auf einem der beiden Quergangswagen transportiert werden, welche an diesen Sektor grenzten. Diese beiden Quergangswagen stellten deutliche Engpass-Ressourcen dar, welche massiv überlastet waren. Durch eine ausgeglichene Verteilung der Aufzüge hätte der Leistungsmangel vermieden werden können.

Ziel dieses Erfahrungsberichtes war, zu zeigen, dass die Leistungsabschätzung automatisierter Kompaktlager von wesentlicher praktischer Bedeutung ist, und dass das Fehlen von entscheidungsunterstützenden Modellen beim Entwurf komplexer Lagersysteme weitreichende Folgen haben kann.

3.2. Ressourcenallokation mittels Mehrgüterflüssen

In diesem Abschnitt wird gezeigt, wie sich Ein- und Auslagerungen in einem automatisierten Kompaktlager als Flüsse im System-Graphen darstellen lassen. Darauf basierend wird ein lineares Optimierungsmodell zur Leistungsabschätzung entwickelt. Dieses berechnet für eine vorgegebene Zeitperiode die maximal mögliche Anzahl der Ein- und Auslagerungen.

Bei den folgenden Betrachtungen werden die Lagergüter unterteilt in verschiedene Artikeltypen (bzw. Güterklassen, Commodities) $i \in I$, welche sich unterscheiden hinsichtlich ihrer mittleren Verweilzeit im Lager. Eine Begründung für diese Unterscheidung wird in Abschnitt 3.4. gegeben. Dort wird ersichtlich, dass eine realistische Leistungsabschätzung voraussetzt, dass die Ein- und Auslagerungsflüsse von Artikeltypen mit verschiedener Verweilzeit getrennt voneinander betrachtet werden.

3.2.1. Ein- und Auslagerungsflüsse

Gegeben sei der System-Graph $G = \langle V^0, E^0, c \rangle$ eines automatisierten Kompaktlagers. Ein Einlagerungsfluss von Artikeltyp $i \in I$ in G ist ein s - t -Fluss $x^i \in \mathbb{R}^E$, welcher die folgenden Flussbilanz-Gleichungen erfüllt:

$$\sum \{x_e^i : e \in \delta^+(v) \cap E\} - \sum \{x_e^i : e \in \delta^-(v) \cap E\} = \begin{cases} \sigma^i & \text{für } v = s \\ 0 & \text{für } v \in V, \\ -\sigma^i & \text{für } v = t \end{cases} \quad i \in I \quad (3.1)$$

Die Variable σ^i bezeichnet die Stärke (bzw. Intensität) des Flusses x^i . Die Bogenmenge E entspricht der Menge der Einlagerungsbogen in G und setzt sich zusammen aus den Fahrt- und Transferbogen sowie den Annahme- und Abgabebogen für Einlagerungen, d.h. $E = E^S \cup E^T \cup E^*$. Die Bezeichnungen $\delta^+(v)$ bzw. $\delta^-(v)$ stellen die Menge der Bogen in G dar, welche aus dem Knoten $v \in V$ hinaus- bzw. in diesen hineinführen. Die linke Summe in den obigen Gleichungen entspricht dem Gesamtfluss, welcher aus v hinausfließt, und die rechte Summe dem Gesamtfluss, welcher in v hineinfließt. Die Differenz dieser beiden Summen stellt die Flussbilanz (bzw. Divergenz) des Knotens v dar. Die obigen Gleichungen besagen, dass die Flussbilanz für alle Zwischenknoten $v \in V$ ausgeglichen, d.h. gleich Null, sein muss. Für die Quelle s bzw. die Senke t entspricht die Divergenz der positiven bzw. negativen Flussstärke σ^i .

Analog wird für jeden Artikeltyp $i \in I$ ein Auslagerungsfluss $\bar{x}^i \in \mathbb{R}^{\bar{E}}$ definiert, welcher die folgenden Flussbilanz-Gleichungen erfüllt:

$$\sum \{x_e^i : e \in \delta^+(v) \cap \bar{E}\} - \sum \{x_e^i : e \in \delta^-(v) \cap \bar{E}\} = \begin{cases} \bar{\sigma}^i & \text{für } v = \bar{s} \\ 0 & \text{für } v \in V, \\ -\bar{\sigma}^i & \text{für } v = \bar{t} \end{cases} \quad i \in I \quad (3.2)$$

Die Bogenmenge \bar{E} entspricht der Menge der Auslagerungsbogen in G und setzt sich zusammen aus den Fahrt- und Transferbogen sowie den Annahme- und Abgabebogen für Auslagerungen, d.h. $\bar{E} = E^S \cup E^T \cup \bar{E}^*$. Die Knoten \bar{s} und \bar{t} stellen die Quelle bzw. Senke für Auslagerungen dar.

Ein- und Auslagerungen als Flüsse

In Abschnitt 2.3. wurde gezeigt, dass sich eine Ein- bzw. Auslagerung im System-Graphen G darstellen lässt als ein Weg von s nach t bzw. \bar{s} nach \bar{t} .

Bezeichne P^0 die Menge aller Wege von s nach t im Einlagerungsgraphen $\langle V, E, c \rangle$ und \bar{P}^0 die Menge aller Wege von \bar{s} nach \bar{t} im Auslagerungsgraphen $\langle V, \bar{E}, c \rangle$. Sei weiter $P^0(e)$ bzw. $\bar{P}^0(e)$ die Menge derjenigen Wege in P^0 bzw. \bar{P}^0 , welche den Bogen $e \in E$ bzw. $e \in \bar{E}$ enthalten.

Betrachte nun eine einzelne Einlagerung $p \in P^0$ von Artikeltyp $i \in I$. Durch diese Einlagerung lässt sich der folgende Fluss definieren:

$$x_e^i = \begin{cases} 1 & \text{falls } e \in p, \\ 0 & \text{sonst} \end{cases}, \quad e \in E$$

Der Vektor x^i entspricht dem Inzidenzvektor der Weges p . x^i stellt offensichtlich einen s - t -Fluss der Stärke $\sigma^i = 1$ dar. x^i erfüllt die Flussbilanz-Gleichungen (3.1), da der Weg p bei jedem Zwischenknoten genau einmal hinein und wieder hinausführt. Analog lässt sich eine Auslagerung als ein Einheitsfluss auf einem Weg $p \in \bar{P}^0$ darstellen.

Gegeben sei nun eine Menge von Einlagerungen. Diese lässt sich darstellen als eine Reihe von Skalaren $(\mu_p^i \in \mathbb{N} : p \in P^0)$, wobei μ_p^i die Anzahl Paletten von Artikeltyp $i \in I$ angibt, welche auf dem Weg p eingelagert werden. Durch diese Einlagerungen wird folgender Fluss definiert:

$$x_e^i = \sum_{p \in P^0(e)} \mu_p^i, \quad e \in E$$

Der auf diese Weise definierte Fluss x^i stellt offensichtlich einen s - t -Fluss dar. Die Flussbilanzgleichungen (3.1) sind erfüllt, da x^i eine Summe von Flüssen auf s - t -Wegen darstellt. Die Flussstärke von x^i ist gegeben durch $\sum_{p \in P^0} \mu_p^i$ und entspricht der Anzahl der eingelagerten Paletten.

Somit wurde gezeigt, wie sich eine Menge von Einlagerungen als (ganzzahliger) Einlagerungsfluss im System-Graphen G darstellen lässt. Analog lässt sich eine Menge von Auslagerungen als (ganzzahliger) Auslagerungsfluss, d.h. als \bar{s} - \bar{t} -Fluss, darstellen.

Flüsse als Ein- und Auslagerungen

Vorangehend wurde gezeigt, dass sich eine Menge von Ein- bzw. Auslagerungen als ganzzahliger Ein- bzw. Auslagerungsfluss im System-Graphen G darstellen lässt. In diesem Abschnitt wird gezeigt, wie sich umgekehrt ein ganzzahliger Ein- oder Auslagerungsfluss als eine Menge von Ein- bzw. Auslagerungen interpretieren lässt.

Bezeichne P die Menge aller s - t -Wege und Zyklen im Einlagerungsgraphen $\langle V, E, c \rangle$ und \bar{P} die Menge aller \bar{s} - \bar{t} -Wege und Zyklen im Auslagerungsgraphen $\langle V, \bar{E}, c \rangle$

Gegeben sei ein ganzzahliger Einlagerungsfluss x^i von Artikeltyp $i \in I$. Aufgrund der Flussdekompositions-Eigenschaft existieren Skalare $(\mu_p^i \in \mathbb{N} : p \in P)$ so dass sich x^i darstellen lässt als:

$$x_e^i = \sum_{p \in P(e)} \mu_p^i, \quad e \in E$$

Somit kann x^i interpretiert werden als eine Menge von Einlagerungen und eine Menge von zyklischen Transportbewegungen. Hierbei entspricht μ_p^i der Anzahl Einlagerungen bzw. zyklischen Transporte auf dem Weg bzw. Zyklus $p \in P$. Die Flussstärke von x^i ist gegeben durch die Summe der Transporte auf den s - t -Wegen $p \in P^0$, d.h. $\sigma^i = \sum_{p \in P^0} \mu_p^i$. Die Flussstärke entspricht der Anzahl der eingelagerten Paletten. Man beachte, dass die Flussdekomposition $(\mu_p^i \in \mathbb{N} : p \in P)$ im allgemeinen nicht eindeutig ist. Das bedeutet, dass ein gegebener Einlagerungsfluss normalerweise auf verschiedene Arten in Einlagerungen und zyklische Transporte zerlegt werden kann.

Die zyklischen Transportbewegungen, welche durch die Zerlegung von x^i definiert werden, sind von einem praktischen Standpunkt aus sinnlos. Weiter unten wird gezeigt, wie mit Hilfe einer geeigneten Zielfunktion zyklische Palettentransporte in einer Optimallösung eliminiert werden können.

Weiter sei darauf hingewiesen, dass es bei der Flussdekomposition möglich ist, dass Einlagerungsrouten $p \in P^0$ benutzt werden (d.h. für welche $\mu_p^i > 0$ ist), bei denen ein Prozessor auf einer Transportfahrt Zwischenhalte einlegt. Dies ist dann der Fall, wenn die Transportfahrt eines Prozessors in p durch mehr als einen Bogen repräsentiert wird. Solche Zwischenhalte sind in Bezug auf die Praxis ebenfalls sinnlos, da eine Direktfahrt (d.h. eine Fahrt auf dem entsprechenden transitiven Bogen) typischerweise weniger Zeit benötigt als eine Fahrt mit Zwischenhalten. Transportfahrten mit Zwischenhalten können ebenfalls mit Hilfe einer geeigneten Zielfunktion eliminiert werden.

Analog zu diesen Ausführungen lässt sich ein ganzzahliger Auslagerungsfluss x^i als eine Menge von Auslagerungen und zyklischen Transporten interpretieren.

Zusammensetzung der Ein- und Auslagerungen

Das Modell zur Leistungsabschätzung basiert auf dem Ansatz, die maximal mögliche Anzahl Ein- und Auslagerungen in einem bestimmten Zeitintervall zu berechnen. Die Anzahl der Ein- und Auslagerungen von Artikeltyp $i \in I$ bei gegebenen Ein- und Auslagerungsflüssen x^i und \bar{x}^i entspricht den Flussstärken σ^i und $\bar{\sigma}^i$. Als Mass für die Lagerleistung wird der Wert $\sum_{i \in I} \sigma^i + \bar{\sigma}^i$ betrachtet. Dieses Leistungsmass ist relativ unspezifisch, da sowohl die Ein- als auch die Auslagerungszahlen der verschiedenen Artikeltypen zu einem einzigen Wert (Anzahl Paletten) zusammengefasst werden.

Bei Verwendung dieses Leistungsmasses hängt die Lagerleistung offensichtlich ab von der Art der betrachteten Lagerprozesse. Dabei müssen zwei wichtige Einflussfaktoren berücksichtigt werden:

- Verhältnis der Ein- und Auslagerungen
- Verteilung der Ein- und Auslagerungen auf die verschiedenen Artikeltypen $i \in I$

Falls beispielsweise Lagerprozesse betrachtet werden, bei welchen nur Ein- oder Auslagerungen erfolgen, ergibt sich eine andere Leistung, als wenn gleichzeitig Ein- und Auslagerungen durchgeführt werden. Ebenso ist die Lagerleistung abhängig davon, wie sich die Ein- und Auslagerungen auf die verschiedenen Artikeltypen verteilen. Wie schon früher erwähnt, unterscheiden sich die Artikeltypen hinsichtlich ihrer Verweildauer im Lager. Lagerprozesse, bei denen vorwiegend Artikeltypen mit grosser Verweildauer (sogenannte Langsamdreher) ein- und ausgelagert werden, ergeben typischerweise eine kleinere Leistung, als wenn Artikeltypen mit kleiner Verweildauer betrachtet werden. Dieser Zusammenhang wird in Abschnitt 3.4. näher diskutiert.

Zur Kontrolle der Lagerprozesse werden die folgenden Parameter eingeführt:

- α^i Verhältnis der Aus- zur Einlagerungsmenge von Artikeltyp i , $i \in I$
 β^i Verhältnis der Einlagerungsmenge von Artikel i zur gesamten Einlagerungsmenge,
 $i \in I$, $\sum_{i \in I} \beta^i = 1$

Die entsprechenden Restriktionen im Modell lauten:

$$\bar{\sigma}^i = \alpha^i \sigma^i, \quad i \in I \quad (3.3)$$

$$\sigma^i = \beta^i \sum \{\sigma^i : i \in I\}, \quad i \in I \quad (3.4)$$

Die Gleichungen (3.3) spezifizieren das Verhältnis zwischen den Aus- und Einlagerungen von Artikeltyp i . Die Restriktionen (3.4) legen fest, wie sich die Einlagerungen auf die verschiedenen Artikeltypen verteilen. Die Summe $\sum \{\sigma^i : i \in I\}$ entspricht der Gesamtzahl der Einlagerungen. Der Parameter β^i gibt den Anteil der gesamten Einlagerungen an, welche sich auf Paletten von Artikeltyp i beziehen.

Beachte, dass durch die obigen Restriktionen auch das Verhältnis der gesamten Aus- und Einlagerungsmenge sowie der Anteil der Auslagerungen der verschiedenen Artikeltypen an der Auslagerungsmenge bestimmt wird. Es ergeben sich die folgenden Beziehungen:

$$\bar{\sigma} = \sum_{i \in I} \alpha^i \beta^i \sigma$$

$$\bar{\sigma}^i = \frac{\alpha^i \beta^i}{\sum_{i \in I} \alpha^i \beta^i} \bar{\sigma}, \quad i \in I$$

Die erste Gleichung spezifiziert das Verhältnis zwischen der gesamten Aus- und Einlagerungsmenge. Die zweiten Gleichungen geben den Anteil der Auslagerungen von Artikeltyp i an der gesamten Auslagerungsmenge an.

3.2.2. Leistungsabschätzung

In diesem Abschnitt wird gezeigt, wie auf der Basis der Ein- und Auslagerungsflüsse ein Modell zur Leistungsabschätzung entworfen werden kann.

Prozessorbelastung

Gegeben seien ganzzahlige Ein- und Auslagerungsflüsse $x = \{x^i : i \in I\}$ und $\bar{x} = \{\bar{x}^i : i \in I\}$. Die Bogen des System-Graphen entsprechen den Aktionen (Transport, Transfer, Auf-/Abladen) der verschiedenen Prozessoren. Der Wert einer einzelnen Variablen x_e^i bzw. \bar{x}_e^i besagt, wie oft die Aktion, welche durch den Bogen e repräsentiert wird, ausgeführt werden muss. Bezeichne $L_m(x, \bar{x})$ die gesamte zeitliche Belastung eines Prozessors $m \in M$, welche aus den vorgegebenen Ein- und Auslagerungsflüssen x und \bar{x} resultiert. $L_m(x, \bar{x})$ entspricht der Summe der Dauer aller Transport-, Transfer- und Auf-/Ablade-Aktionen, welche von Prozessor m ausgeführt werden müssen, d.h.

$$L_m(x, \bar{x}) = \sum \{c_e x_e^i : e \in E_m, i \in I\} + \sum \{c_e \bar{x}_e^i : e \in \bar{E}_m, i \in I\}, \quad m \in M \quad (3.5)$$

Offensichtlich bildet $L_m(x, \bar{x})$ eine untere Schranke für die effektive Zeit, welche ein Prozessor in einem konkreten Ablaufplan benötigt, um die durch die Flüsse vorgegebenen Lagerbewegungen auszuführen. Nicht berücksichtigt sind die Zeiten für die Leerfahrten sowie die Wartezeiten (in beladenem oder unbeladenem Zustand), welche sich ergeben aus den Reihenfolge- und Synchronisationsbedingungen bei der Ausführung der Lagerbewegungen. Ebenso ist der Zeitaufwand für eventuell notwendige Umlagerungen nicht einbezogen.

$L_m(x, \bar{x})$ stellt eine (optimistische) Abschätzung der Prozessorbelastung dar, welche unabhängig ist von der zeitlichen Einplanung der Operationen und nur von der Wahl der Ein- und Auslagerungsrouten (d.h. von den Flüssen x und \bar{x}) abhängt.

Optimierungsmodell

Betrachte nun die Aufgabe, die maximale Anzahl Ein- und Auslagerungen in einer bestimmten Zeitperiode T zu bestimmen. Unter Benützung der abgeschätzten Prozessorenbelastungen $L_m(x, \mathfrak{x})$ lässt sich folgendes (lineare) Optimierungsmodell formulieren:

$$\begin{aligned} \max \quad & \sum \{ \sigma^i + \bar{\sigma}^i : i \in I \} \\ L_m(x, \mathfrak{x}) & \leq T, \quad m \in M \end{aligned}$$

Hierbei gelten als weitere Restriktionen die Flussbilanzgleichungen (3.1) und (3.2) sowie die Bedingungen (3.3) und (3.4). Dieses Modell maximiert die Gesamtzahl der Ein- und Auslagerungen unter der Bedingung, dass die Belastung der einzelnen Prozessoren die vorgegebene Zeitperiode T nicht überschreitet. Diese Restriktionen werden Kapazitätsbedingungen genannt. Da $L_m(x, \mathfrak{x})$ eine untere Schranke für die Prozessorbelastrungen darstellt, bildet der Optimalwert dieses Linearen Programms eine obere Schranke für die Anzahl der möglichen Ein- und Auslagerungen. Mit Hilfe dieses Modells kann somit die Leistung eines automatisierten Kompaktlagers abgeschätzt werden, ohne dass Vorgaben gemacht werden hinsichtlich der Entscheidungsmechanismen auf der Ebene der dispositiven Steuerung.

Gemäss Abschnitt 2.2. umfasst die dispositive Steuerung drei Hauptfunktionen: (1) Bestimmung der Transportaufträge, (2) Bestimmung des Ablaufplans, (3) Bestimmung der Umlagerungen. Die Bestimmung der Transportaufträge umfasst die Wahl der Transportrouten und die Wahl der Ein- und Auslagerungspositionen (Lagerstrategie). In Bezug auf die Transportaufträge werden in dem Modell keine Einschränkungen vorgegeben. Die Bestimmung der Transportaufträge erfolgt durch das Optimierungsverfahren bei der Wahl der Ein- und Auslagerungsflüsse. In Bezug auf den Ablaufplan werden ebenfalls keine Bedingungen vorgegeben. Die Wartezeiten, welche sich aus den Reihenfolge- und Synchronisationsbedingungen ergeben, werden vernachlässigt. Es wird angenommen, dass die Aktionen eines Prozessors ohne zwischenzeitliche Wartephasen ausgeführt werden können. Umlagerungen werden in dem Modell ebenfalls nicht berücksichtigt.

Somit basiert das Modell auf einer optimistischen Abschätzung (d.h. einer unteren Schranke) der Prozessorbelastrung und liefert eine optimistische Abschätzung (d.h. eine obere Schranke) für die Lagerleistung.

Abschliessend sei noch eine Bemerkung zur Ganzzahligkeits-Forderung angeführt. Die bisherigen Ueberlegungen basierten auf der Annahme, dass die Ein- und Auslagerungsflüsse ganzzahlig sind und somit als diskrete Menge von Ein- und Auslagerungen interpretiert werden können. Zum Zwecke der Leistungsabschätzung können diese Ganzzahligkeits-Bedingungen jedoch problemlos ignoriert werden.

Zielfunktion

Weiter oben wurde gezeigt, dass sich die Ein- und Auslagerungsflüsse x und \bar{x} interpretieren lassen als eine Menge von Ein- und Auslagerungen und eine Menge von zyklischen Transportbewegungen. Es wurde darauf hingewiesen, dass die zyklischen Transportbewegungen von einem praktischen Standpunkt aus sinnlos sind und mit Hilfe einer geeigneten Zielfunktion eliminiert werden können. Um dies zu erreichen, wird die gesamte Prozessorenbelastung mit einem marginalen Faktor ε gewichtet und als negativer Term in die Zielfunktion aufgenommen:

$$\max \sum \{\sigma^i + \bar{\sigma}^i : i \in I\} - \varepsilon (\sum \{c_e x_e^i : e \in E, i \in I\} + \sum \{c_e \bar{x}_e^i : e \in \bar{E}, i \in I\}) \quad (3.6)$$

Es ist leicht nachzuvollziehen, dass mit dieser Zielfunktion zyklische Transportbewegungen ausgeschlossen werden. Sei x, \bar{x} eine Optimallösung bezüglich dieser Zielfunktion. Nehme an, dass in der Flusszerlegung $\{\mu_p^i : p \in P\}$ eines Flusses x^i (bzw. \bar{x}^i) ein Zyklus $p \in P$ existiert mit $\mu_p^i > 0$. Bilde den Fluss \hat{x}^i durch $\hat{x}_e^i = x_e^i - \mu_p^i$ falls $e \in p$ und $\hat{x}_e^i = x_e^i$ sonst. \hat{x}^i ist wiederum ein zulässiger Fluss mit unveränderter Flusstärke. Der Zielfunktionswert hat jedoch zugenommen, da der ε -Ausdruck abgenommen hat. Ein Widerspruch zur Optimalität der ursprünglichen Lösung.

Analog lässt sich zeigen, dass auch Transportfahrten mit Zwischenhalten durch diese Zielfunktion eliminiert werden (siehe den vorangehenden Abschnitt). Eine Transportfahrt mit Zwischenhalten (mehrere Bogen) kann durch die entsprechende Fahrt ohne Zwischenhalt (ein Bogen) ersetzt werden, ohne die Zulässigkeit oder die Flusstärke zu beeinflussen.

Lastverteilung und Engpass-Ressourcen

Abschliessend sei darauf hingewiesen, dass das obige Modell nicht nur eine obere Leistungsschranke berechnet, sondern auch Informationen über die Lastverteilung und die Engpass-Prozessoren in einem Lager liefert. Sei x, \bar{x} eine Optimallösung mit Gesamtflusstärke (d.h. Leistung) $\sigma + \bar{\sigma}$. Diese Lösung definiert für jeden Prozessoren den Wert $L_m(x, \bar{x})$ Prozessoren, bei welchen die Kapazitätsbedingung bindend ist, d.h. $L_m(x, \bar{x}) = T$ stellen Engpass-Ressourcen dar, welche die Lagerleistung beschränken.

Für Engpass-Prozessoren entspricht $L_m(x, \bar{x})$ einer unteren Schranke für die Prozessorbelastung, wenn eine Lagerleistung von $\sigma + \bar{\sigma}$ realisiert werden soll, d.h. bei vorgegebener Leistung $\sigma + \bar{\sigma}$ werden diese Prozessoren mit einem Arbeitsaufwand von mindestens $L_m(x, \bar{x})$ belastet. Ohne modifizierte Zielfunktion gilt diese Aussage nicht für Prozessoren, welche keine Engpass-Ressourcen darstellen. Durch den ε -Ausdruck in der Zielfunktion wird jedoch erreicht, dass auch für diese Prozessoren der Wert von $L_m(x, \bar{x})$ minimal ausfällt und somit als Mindestbelastung bei gegebener Leistung $\sigma + \bar{\sigma}$ interpretiert werden kann.

Zusammenfassend lässt sich sagen, dass das entworfene Modell detaillierte Informationen liefert über die Verteilung der Belastung auf die verschiedenen Prozessoren in einem automatisierten

Lager. Dadurch lassen sich Engpass-Prozessoren identifizieren, welche die Lagerleistung beschränken. Gerade in der Entwurfsphase bei der Dimensionierung eines Lagers bilden diese Informationen einen wertvollen Beitrag zur Evaluation verschiedener Layout-Varianten.

Darüber hinaus liefert eine Optimallösung auch Hinweise darüber, wie die Ein- und Auslagerungspositionen und die Transportrouten günstig gewählt werden, damit eine möglichst grosse Leistung erzielt wird. Diese Informationen lassen sich verwenden bei der Bestimmung der Transportaufträge auf der Ebene der dispositiven Steuerung.

3.3. Leerfahrten

Im vorangehenden Abschnitt wurde die zeitliche Belastung der Prozessoren bei der Ausführung der Ein- und Auslagerungen berechnet auf der Basis der Transport-, Transfer- und Auf-/Abladeaktionen. Zusätzlich zu diesen Aktionen müssen die Prozessoren im Verlaufe eines Lagerprozesses auch Leerfahrten ausführen. Die Belastung der Prozessoren durch Leerfahrten wurde im obigen Modell nicht berücksichtigt. Typischerweise ist der Zeitaufwand für die Leerfahrten in einer ähnlichen Grössenordnung wie der Aufwand für die Transportfahrten. Eine realistische Leistungsabschätzung kann somit nur erfolgen, wenn die Leerfahrten in das Modell miteinbezogen werden.

Die Schwierigkeit bei der Betrachtung der Leerfahrten liegt darin, dass deren Dauer im Gegensatz zu den andern Aktionen nicht durch die Transportrouten (d.h. die Ein- und Auslagerungsflüsse) vorgegeben ist, sondern von der Ausführungsreihenfolge der Transportaktionen abhängig ist. Die Festlegung der Operationsreihenfolgen geschieht auf der Ebene der dispositiven Steuerung im Rahmen der Bestimmung des Ablaufplans. Da das vorgeschlagene Modell steuerungsunabhängig ist, enthält es keine Vorgaben bezüglich der Ausführungsreihenfolgen der Operationen und folglich keine Angaben über die Dauer der Leerfahrten.

In diesem Abschnitt wird gezeigt, wie die Leerfahrten trotz dieser Schwierigkeiten in das Modell miteinbezogen werden können. Der Ansatz dabei ist, dass mit Hilfe eines Transportproblems, welches in das Modell integriert wird, eine untere Schranke für die Leerfahrtenbelastung der Prozessoren bestimmt werden kann.

Transport- und Leerfahrten

In Abschnitt 2.2. wurde gezeigt, dass die Prozessoren im Verlauf eines Lagerprozesses eine sich stets wiederholende Sequenz von Aktionen ausführen: (1) Uebernahme (bzw. Aufladen) einer Palette, (2) Transportfahrt von der Uebernahme- zur Uebergabeposition, (3) Uebergabe bzw. Abladen der Palette, (3) Leerfahrt von der Uebergabe- zur nächsten Uebernahmeposition. Somit erfolgt

zwischen zwei aufeinanderfolgenden Transportfahrten jeweils eine Leerfahrt von der Endposition der ersten Transportfahrt zur Startposition der nachfolgenden Transportfahrt. Die Fahrtbewegungen bilden folglich eine alternierende Sequenz von Transport- und Leerfahrten.

O-Leerfahrten

Ein Spezialfall liegt vor, wenn bei zwei aufeinanderfolgenden Transportfahrten die Zielposition der ersten mit der Startposition der nächsten übereinstimmt. Dies bedeutet, dass der Prozessor eine Palette übergibt und anschliessend an derselben Stelle eine Palette übernimmt. In diesem Fall wird keine Leerfahrt zwischen den Transportfahrten ausgeführt. Es kann jedoch angenommen werden, dass der Prozessor nicht dieselbe Palette wieder übernimmt, die soeben übergeben wurde. Folglich muss der übernehmende Prozessor zwischenzeitlich die Palette an einen andern Prozessor abgeben und eine neue Palette von einem angrenzenden Prozessor übernehmen. Diese neue Palette wird anschliessend vom ursprünglichen Prozessor übernommen. Daraus geht hervor, dass ein Prozessor zwischen einer Uebergabe und Uebernahme an derselben Position eine gewisse Zeit warten muss, bis die neue Palette in Empfang genommen werden kann. Diese Wartezeit bedeutet eine zeitliche Belastung des Prozessors und kann in diesem Sinne als eine spezielle sogenannte O-Leerfahrt betrachtet werden. Aufgrund der notwendigen zwischenzeitlichen Prozessorbewegungen kann eine untere Schranke berechnet werden für die Wartezeit zwischen einer Uebergabe und einer Uebernahme an derselben Position. Die Dauer einer O-Leerfahrt wird dieser Wartezeit gleichgesetzt.

Für die folgenden Betrachtungen wird angenommen, dass sich ein Prozessor zu Beginn eines Lagerprozesses an einer Übergabe- oder Übernahmeposition befindet und am Ende wieder an diese Position zurückkehrt. In diesem Fall bilden die Fahrtbewegungen eines Prozessors eine geschlossene Route, welche an einer bestimmten Position startet und am selben Ort endet. Diese Route stellt eine alternierende Folge von Transport- und Leerfahrten dar.

Prozessor-Subgraph

Als Prozessor-Subgraph G_m eines Prozessors $m \in M$ wird der (vollständige) Subgraph im System-Graphen G bezeichnet, welcher durch die zum Prozessor gehörenden Knoten V_m induziert wird. Dieser umfasst die Fahrtbogen E_m^S und Schlingen E_m^L (für O-Leerfahrten) und ist gegeben durch $G_m = \langle V_m, E_m^S \cup E_m^L, c \rangle$

Im G_m bilden die Fahrtbewegungen eines Prozessors m eine geschlossene Bogenfolge, welche sich aus Teilwegen zusammensetzt, die alternierend Transportfahrten und Leerfahrten repräsentieren. Eine Transportfahrt ohne Zwischenhalt entspricht einem einzelnen Bogen, während eine Transportfahrt mit Zwischenhalten einen Teilweg darstellt, welcher aus mehreren Bogen besteht. Prinzipiell sind auch bei Leerfahrten Zwischenhalte möglich. Weiter unten wird jedoch gezeigt, dass aufgrund der Modellierung der Leerfahrten angenommen werden kann, dass diese ohne Zwischenhalte ausgeführt werden und somit einem einzelnen Bogen entsprechen.

Die folgende Abbildung zeigt schematisch die Fahrtroute eines Prozessors im Graphen G_m . Transportfahrten sind im Gegensatz zu Leerfahrten mit dicken Pfeilen eingezeichnet. Durch die eingezeichneten Zwischenknoten bei den Transportfahrten wird angedeutet, dass diese (bei Zwischenhalten) aus mehreren Bogen bestehen können. Die Uebergabebogen (bzw. Auf-/Ablade-Bogen) sind gestrichelt eingezeichnet. Sie führen aus dem Prozessor-Subgraphen G_m hinaus oder in diesen hinein.

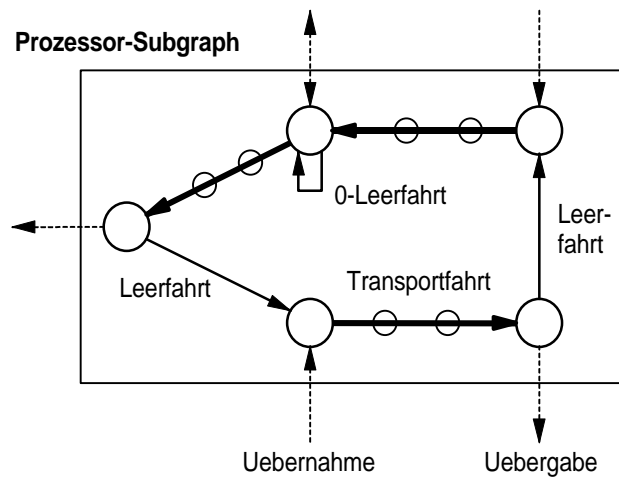


Abbildung 3.1. Fahrtroute eines Prozessors im Subgraphen G_m

Start- und Zielpositionen der Leerfahrten

Im vorangehenden Abschnitt wurde gezeigt, dass auf jede Transportfahrt eine Leerfahrt folgt, welche an der Zielposition (d.h. der Uebergabeposition) der Transportfahrt startet. Umgekehrt erfolgt vor jeder Transportfahrt eine Leerfahrt, welche an der Startposition (d.h. der Uebernahmeposition) der Transportfahrt endet.

Betrachte nun eine einzelne Prozessorposition (d.h. eine Uebergabe- bzw. Uebernahmeposition). Die Anzahl der Leerfahrten, welche an dieser Position starten, entspricht der Anzahl Transportfahrten, welche an dieser Position enden. Analog entspricht die Anzahl der Leerfahrten, welche an dieser Position enden, der Anzahl Transportfahrten, welche an dieser Position starten.

Die Anzahl der Transportfahrten, welche an einer Position enden, entspricht der Anzahl Uebergaben bzw. Ablade-Aktionen, welche an dieser Position stattfinden. Analog entspricht die Anzahl der Transportfahrten, welche an einer Position starten, der Anzahl der Uebernahmen bzw. Auflade-Aktionen, welche an dieser Position erfolgen.

Bei gegebenen Ein- und Auslagerungsflüssen x^i und π^i , $i \in I$ lässt sich die Anzahl der Uebergabe- und Ablade-Aktionen an einer Position bestimmen. Sie entspricht der Summe der Ein- und Auslagerungsflüsse auf den Transfer- und Abgabebogen, welche aus dem entsprechenden Knoten

im System-Graphen herausführen. Analog entspricht die Anzahl der Uebernahme- bzw. Auflade-Aktionen der Summe der Ein- und Auslagerungsflüsse auf den Transfer- und Aufnahmebogen, welche in den entsprechenden Knoten hineinführen. Bezeichne die Anzahl Uebergabe- und Ablade-Aktionen in einem Knoten $v \in V$ als a_v und die Anzahl Uebernahme- und Auflade-Aktionen als b_v . Es gilt:

$$a_v = \sum \{x_e^i : e \in \delta^+(v) \cap (E^T \cup E^*), i \in I\} + \sum \{x_e^i : e \in \delta^+(v) \cap (E^T \cup \bar{E}^*), i \in I\}, \quad v \in V \quad (3.7)$$

$$b_v = \sum \{x_e^i : e \in \delta^-(v) \cap (E^T \cup E^*), i \in I\} + \sum \{x_e^i : e \in \delta^-(v) \cap (E^T \cup \bar{E}^*), i \in I\}, \quad v \in V \quad (3.8)$$

Hierbei bezeichnet $\delta^+(v)$ bzw. $\delta^-(v)$ die Menge aller Bogen im System-Graphen, welche aus einem Knoten $v \in V$ herausführen bzw. in diesen hineinführen. Die Menge E^T entspricht den Transferbogen und die Mengen E^* bzw. \bar{E}^* den Aufnahme und Abgabebogen für Ein- bzw. Auslagerungsflüsse.

Gemäss obigen Ausführungen entspricht a_v bzw. b_v gleichzeitig der Anzahl Transportaktionen, welche in v enden bzw. starten.

Leerfahrten-Flüsse

Zur Modellierung der Leerfahrten werden zusätzliche Flussvariablen $\{y_e : e \in E^S \cup E^L\}$ auf den Fahrtbogen E^S und den Schlingen E^L (für O-Leerfahrten) eingeführt. Diese entsprechen der Anzahl Leerfahrten, welche von Start- zum Endknoten des betreffenden Bogens ausgeführt werden.

Die Anzahl der Leerfahrten, welche in einem Knoten $v \in V$ starten, ist gegeben durch die Summe der Leerfahrtenflüsse auf den Fahrtbogen und Schlingen, welche aus v herausführen. Analog ist die Anzahl der Leerfahrten, welche in v enden, gegeben durch die Summe der Leerfahrtenflüsse auf den Fahrtbogen und Schlingen, welche in v hineinführen.

Vorangehend wurde gezeigt, dass die Anzahl der Transportfahrten, welche in einem Knoten $v \in V$ enden bzw. starten, gegeben ist durch a_v bzw. b_v . Weiter wurde bemerkt, dass die Anzahl der Leerfahrten, welche in v starten bzw. enden, der Anzahl der Transportfahrten entspricht, welche in v enden bzw. starten. Somit lassen sich die folgenden Gleichungen formulieren:

$$a_v = \sum \{y_e : e \in \delta^+(v) \cap (E^S \cup E^L)\} \quad (3.9)$$

$$b_v = \sum \{y_e : e \in \delta^-(v) \cap (E^S \cup E^L)\} \quad (3.10)$$

Hierbei entspricht die rechte Seite der Anzahl Leerfahrten, welche in v starten bzw. enden. E^S ist die Menge der Fahrtbogen und E^L die Menge der Schlingen, welche die O-Fahrten repräsentieren.

Transportproblem

Gegeben seien ganzzahlige Ein- und Auslagerungsflüsse x^i und \bar{x}^i , $i \in I$. Gemäss (3.7) und (3.8) werden durch diese Flüsse für jeden Knoten $v \in V_m$, $m \in M$ die Grössen a_v und b_v bestimmt. a_v entspricht der Anzahl der in v endenden und b_v der in v startenden Transportfahrten von Prozessor m .

Betrachte nun die Gleichungen (3.9) und (3.10) für einen einzelnen Prozessor $m \in M$. Da der Prozessor-Subgraph $G_m = \langle V_m, E_m^S \cup E_m^L, c \rangle$ vollständig ist, lassen sich die rechten Seiten dieser Gleichungen schreiben als:

$$\sum \{y_e : e \in \delta^+(v) \cap (E^S \cup E^L)\} = \sum_{w \in V_m} y_{vw}, \quad v \in V_m$$

$$\sum \{y_e : e \in \delta^-(v) \cap (E^S \cup E^L)\} = \sum_{w \in V_m} y_{vw}, \quad w \in V_m$$

Daraus wird ersichtlich, dass Gleichungen (3.9) und (3.10) für jeden Prozessor m die Restriktionen eines Transport-Problems darstellen. Dieses Transportproblem modelliert die Zuordnung der Leerfahrten zwischen den Prozessorpositionen V_m . Auf der einen Seite stehen die Knoten V_m als "Angebotsknoten", wobei jeder Knoten $v \in V_m$ ein "Angebot" von a_v Leerfahrten hat, welche in v starten. Auf der anderen Seite stehen die Knoten V_m als "Bedarfsknoten", wobei jeder Knoten $w \in V_m$ einen Bedarf von b_w Leerfahrten hat, welche in w enden.

Eine untere Schranke für die Dauer einer Leerfahrt von v nach w ist gegeben durch das Bogen-gewicht c_{vw} im System-Graphen G . Dieses entspricht der Dauer einer Direktfahrt von v nach w . Falls bei einer Leerfahrt Zwischenhalte eingeschaltet werden, ist ihre Dauer auf Grund der Dreiecksungleichung, welche für die Bogen-gewichte von G gilt, mindestens so gross wie bei einer Direktfahrt. Für eine Abschätzung der Leerfahrten-Dauer kann somit angenommen werden, dass die Leerfahrten direkt erfolgen und ihre Dauer durch c_{vw} gegeben ist.

Zur Abschätzung des Leerfahrten-Aufwandes lässt sich nun für jeden Prozessor $m \in M$ das folgende Transport-Problem $\Pi(m)$ formulieren.

Transport-Problem $\Pi(m)$:

$$\min \sum_{v, w \in V_m} c_{vw} y_{vw} \tag{3.11}$$

$$\sum_{w \in V_m} y_{vw} = a_v, \quad v \in V_m \tag{3.12}$$

$$\sum_{v \in V_m} y_{vw} = b_w, \quad w \in V_m \tag{3.13}$$

$$y_{vw} \geq 0, \quad v, w \in V_m \tag{3.14}$$

Dieses Transport-Problem ist eingebettet in das weiter oben beschriebene Flussproblem in dem Sinne, dass die "Angebote" a_v und "Bedarfe" b_v keine Konstanten darstellen, sondern gemäss (3.7) und (3.8) von den Werten der Ein- und Auslagerungsflüsse abhängig sind.

Im nächsten Satz wird gezeigt, dass das Minimum von $\Pi(m)$ eine untere Schranke darstellt für die Gesamtdauer der Leerfahrten von Prozessor $m \in M$. In der Beweisführung wird gezeigt, dass sich das Problem der minimalen Leerfahrten-Dauer eines Prozessors m darstellen lässt als Traveling Salesman Problem in einem vollständigen gerichteten Graphen. Die Knoten dieses Graphen entsprechen den einzelnen Transportfahrten des Prozessors. Die Bogengewichte sind gegeben als die Dauer einer Leerfahrt (ohne Zwischenhalte) von der Endposition einer Transportfahrt zur Startposition einer andern Transportfahrt. Das Traveling Salesman Problem löst die Aufgabe, die Transportfahrten so aneinanderzureihen, dass die Gesamtdauer der Leerfahrten minimal ist. Anschliessend wird gezeigt, dass das Transport-Problem $\Pi(m)$ eine Relaxation dieses Traveling Salesman Problems darstellt und äquivalent ist zu dessen Assignment-Relaxation (bei welcher die Teiltoureneliminations-Ungleichungen weggelassen werden).

Satz 3.1. Seien x^i und x^i , $i \in I$, ganzzahlige Ein- und Auslagerungsflüsse. Dann ist das Optimum von $\Pi(m)$ eine untere Schranke für die Dauer der Leerfahrten und Wartezeiten von Prozessor $m \in M$.

Beweis:

(1) Zuerst wird gezeigt, wie aufgrund der Ein- und Auslagerungsflüsse die Menge $Q(m)$ der Transportfahrten auf Prozessor m bestimmt werden kann. Sei P die Menge aller s - t -Wege und Zyklen im Einlagerungsgraphen $\langle V, E, c \rangle$ und \bar{P} die Menge aller \bar{s} - \bar{t} -Wege und Zyklen im Auslagerungsgraphen $\langle V, \bar{E}, c \rangle$. Bezeichne $P(e)$ bzw. $\bar{P}(e)$ die Menge der Wege und Zyklen, welche den Bogen $e \in E$ bzw. $e \in \bar{E}$ enthalten.

Aufgrund der Flussdekompositions-Eigenschaft existieren Skalare $(\mu_p^i \in \mathbb{N} : p \in P)$ und $(\bar{\mu}_p^i \in \mathbb{N} : p \in \bar{P})$, so dass für alle $i \in I$ gilt:

$$x_e^i = \sum_{p \in P(e)} \mu_p^i, \quad e \in E \quad \text{und} \quad x_e^i = \sum_{p \in \bar{P}(e)} \bar{\mu}_p^i, \quad e \in \bar{E}$$

Die Werte von μ_p^i bzw. $\bar{\mu}_p^i$ entsprechen der Anzahl Paletten von Artikeltyp i , welche auf der Route p ein- bzw. ausgelagert werden. Beachte, dass diese Flusszerlegung im allgemeinen nicht eindeutig ist, d.h. dass verschiedene Möglichkeiten existieren können, die Ein- und Auslagerungsflüsse als Palettentransporte zu interpretieren.

Betrachte nun einen einzelnen Prozessor $m \in M$. Sei $Q^0(m)$ die Menge aller (elementaren) Pfade im Graphen $\langle V_m, E_m^L, c \rangle$. $Q^0(m)$ entspricht der Menge aller möglichen azyklischen Transportrouten von m . Bezeichne $P[q]$ die Menge der Routen in P , welche einen Palettentransport auf Fahrtroute $q \in Q^0(m)$ umfassen, d.h. welche q als Teilweg enthalten:

$$P[q] = \{(v_0, e_1, v_1, \dots, v_l) \in P : \text{es existieren } 0 < j \leq k < l, \text{ so dass} \\ (v_j, e_{j+1}, v_{j+1}, \dots, v_k) = q \text{ und } v_{j-1}, v_{k+1} \notin V_m\}$$

Sei $\bar{P}[q]$ analog definiert für \bar{P} . Für eine gegebene Flusszerlegung ist die Anzahl Palettentransporte λ_q auf Route $q \in Q^0(m)$ gegeben durch

$$\lambda_q = \sum_{i \in I, p \in P[q]} \mu_p^i + \sum_{i \in I, p \in \bar{P}[q]} \bar{\mu}_p^i$$

Die Menge $Q(m)$ aller (azyklischen) Transporte auf Prozessor m lässt sich somit darstellen durch das Multiset

$$Q(m) = \{q^{(r)} : q \in Q^0(m), r = 1, \dots, \lambda_q\}$$

Beachte, dass $Q(m)$ durch die Ein- und Auslagerungsflüsse nicht eindeutig definiert ist, sondern von der gewählten Flusszerlegung abhängig ist.

(2) Die Berechnung der minimalen Gesamtdauer der Leerfahrten von Prozessor m bei gegebener Menge $Q(m)$ von Transportfahrten kann als Traveling Salesman Problem formuliert werden. Für $q \in Q(m)$ bezeichne $t(q)$ die Start- und $h(q)$ die Endposition der Transportfahrt. Zwischen zwei aufeinanderfolgenden Transportfahrten $q, q' \in Q(m)$ muss eine Leerfahrt von $h(q)$ nach $t(q')$ ausgeführt werden. Die Dauer dieser Leerfahrt kann gemäss obigen Ausführungen angenommen werden als $c_{h(q), t(q')}$. Die minimale Gesamtdauer der Leerfahrten von Prozessor m ist somit gegeben durch die Länge einer minimalen Traveling Salesman Tour im vollständigen gerichteten Graphen mit Knoten $Q(m)$ und Distanzen $c_{h(q), t(q')}$ für $q, q' \in Q(m), q \neq q'$. Bezeichne dieses Traveling Salesman Problem als $\Pi^0(m)$.

(3) Seien $\{z_{q, q'} \in \{0, 1\} : q, q' \in Q(m), q \neq q'\}$ binäre Variablen mit $z_{q, q'} = 1$ falls Transportfahrt q' unmittelbar nach q ausgeführt wird, und $z_{q, q'} = 0$ sonst. Das folgende Zuordnungsproblem $\Pi^1(m)$ bildet eine Relaxation (Weglassen der Teiltoureneliminations-Ungleichungen) des Traveling Salesman Problems $\Pi^0(m)$:

Zuordnungsproblem $\Pi^1(m)$:

$$\min \sum_{q, q' \in Q(m), q \neq q'} c_{h(q), t(q')} \cdot z_{q, q'} \quad (3.15)$$

$$\sum_{q' \in Q(m), q' \neq q} z_{q, q'} = 1, \quad q \in Q(m) \quad (3.16)$$

$$\sum_{q \in Q(m), q \neq q'} z_{q, q'} = 1, \quad q' \in Q(m) \quad (3.17)$$

$$z_{q, q'} \in \{0, 1\}, \quad q, q' \in Q(m), q \neq q' \quad (3.18)$$

Bezeichne Q_v die Menge der Transportfahrten in $Q(m)$, welche an Position $v \in V_m$ enden und Q^w die Menge der Transportfahrten, welche in $w \in V_m$ starten. Es gilt:

$$Q(m) = \bigcup_{v \in V_m} Q_v = \bigcup_{w \in V_m} Q^w \quad (3.19)$$

Durch Addition der Restriktionen (3.16) über $q \in Q_v$ und (3.17) über $q \in Q^w$ erhält man folgende als Relaxation von $\Pi^1(m)$ das folgende Problem $\Pi^2(m)$

Problem $\Pi^2(m)$:

$$\min \sum_{q, q' \in Q(m), q \neq q'} c_{h(q), t(q')} \cdot z_{q, q'} \quad (3.20)$$

$$\sum_{q \in Q_v} \sum_{q' \in Q(m), q' \neq q} z_{q, q'} = |Q_v|, \quad v \in V_m \quad (3.21)$$

$$\sum_{q' \in Q^w} \sum_{q \in Q(m), q \neq q'} z_{q, q'} = |Q^w|, \quad w \in V_m \quad (3.22)$$

$$z_{q, q'} \in \{0, 1\}, \quad q, q' \in Q(m), q \neq q' \quad (3.23)$$

Durch Zerlegung von $Q(m)$ gemäss (3.19) und unter Berücksichtigung, dass $c_{h(q), t(q')} = c_{vw}$ für alle $q \in Q_v$, $q' \in Q^w$ lässt sich $\Pi^2(m)$ schreiben als:

Problem $\Pi^3(m)$:

$$\min \sum_{v \in V_m} \sum_{w \in V_m} c_{vw} \sum_{q \in Q_v} \sum_{q' \in Q^w, q' \neq q} z_{q, q'} \quad (3.24)$$

$$\text{unter den Bedingungen} \quad (3.25)$$

$$\sum_{w \in V_m} \sum_{q \in Q_v} \sum_{q' \in Q^w, q' \neq q} z_{q, q'} = |Q_v|, \quad v \in V_m \quad (3.26)$$

$$\sum_{v \in V_m} \sum_{q' \in Q^w} \sum_{q \in Q_v, q \neq q'} z_{q, q'} = |Q^w|, \quad w \in V_m \quad (3.27)$$

$$z_{q, q'} \in \{0, 1\}, \quad q, q' \in Q(m), q \neq q' \quad (3.28)$$

Durch die Substitution $y_{vw} = \sum_{q \in Q_v} \sum_{q' \in Q^w, q' \neq q} z_{q, q'}$ erhält man schliesslich das folgende Transportproblem $\Pi^4(m)$:

Transportproblem $\Pi^4(m)$:

$$\min \sum_{v, w \in V_m} c_{vw} y_{vw} \quad (3.29)$$

$$\sum_{w \in V_m} y_{vw} = |Q_v|, \quad v \in V_m \quad (3.30)$$

$$\sum_{v \in V_m} y_{vw} = |Q^w|, \quad w \in V_m \quad (3.31)$$

$$y_{vw} \geq 0, \quad v, w \in V_m \quad (3.32)$$

Dieses Problem entspricht dem anfänglich definierten Transportproblem $\Pi(m)$, mit dem Unterschied, dass bei letzterem die rechten Seiten gegeben sind durch a_v bzw. b_v .

(4) Zeige, dass die beiden Transportprobleme $\Pi^4(m)$ und $\Pi(m)$ identisch sind. Dazu muss gezeigt werden, dass unabhängig von der gewählten Flusszerlegung $|Q_v| = a_v$ und $|Q^w| = b_w$ ist für alle $v, w \in V_m$. Gemäss Definition von Q_v und λ_q gilt

$$|Q_v| = \sum_{q \in Q^0(m) : h(q)=v} \lambda_q = \sum_{q \in Q^0(m) : h(q)=v} \left(\sum_{i \in I, p \in P[q]} \mu_p^i + \sum_{i \in I, p \in \bar{P}[q]} \bar{\mu}_p^i \right)$$

Sei $P_v = \bigcup_{q \in Q^0(m) : h(q)=v} P[q]$ bzw. $\bar{P}_v = \bigcup_{q \in Q^0(m) : h(q)=v} \bar{P}[q]$ Menge der Routen in P bzw. \bar{P} , welche eine Transportfahrt umfassen, die in an Position $v \in V_m$ endet. Damit lässt sich $|Q_v|$ schreiben als

$$|Q_v| = \sum_{i \in I, p \in P_v} \mu_p^i + \sum_{i \in I, p \in \bar{P}_v} \bar{\mu}_p^i$$

Aufgrund der Definition des System-Graphen G ist $P_v = \bigcup_{e \in \delta^+(v) \cap (E^T \cup E^*)} P(e)$ und $\bar{P}_v = \bigcup_{e \in \delta^+(v) \cap (E^T \cup \bar{E}^*)} \bar{P}(e)$. Daraus folgt:

$$|Q_v| = \sum \left\{ \sum_{p \in P(e)} \mu_p^i : e \in \delta^+(v) \cap (E^T \cup E^*), i \in I \right\} + \sum \left\{ \sum_{p \in \bar{P}(e)} \bar{\mu}_p^i : e \in \delta^+(v) \cap (E^T \cup \bar{E}^*), i \in I \right\}$$

und weiter:

$$|Q_v| = \sum \{x_e^i : e \in \delta^+(v) \cap (E^T \cup E^*), i \in I\} + \sum \{x_e^i : e \in \delta^+(v) \cap (E^T \cup \bar{E}^*), i \in I\}$$

Gemäss (3.7) gilt somit $|Q_v| = a_v$ für alle $v \in V_m$. Analog kann gezeigt werden, dass $|Q^w| = b_w$ ist für alle $w \in V_m$.

(5) Schliesslich wird noch gezeigt, dass die Optimalwerte von $\Pi^1(m)$ und $\Pi(m)$ identisch sind. Da die Optimalwerte von $\Pi^2(m)$, $\Pi^3(m)$ und $\Pi(m)$ offensichtlich identisch sind, muss nur gezeigt werden, dass $\Pi^1(m)$ und $\Pi^2(m)$ denselben Optimalwert haben. Da $\Pi^2(m)$ eine Relaxation von $\Pi^1(m)$ darstellt, ist das Optimum von $\Pi^2(m)$ kleiner gleich dem Optimum von $\Pi^1(m)$. Somit muss nur gezeigt werden, dass das Optimum von $\Pi^1(m)$ kleiner gleich dem Optimum von $\Pi^2(m)$ ist. Dazu wird gezeigt, dass aus jeder Lösung von $\Pi^2(m)$ eine Lösung von $\Pi^1(m)$ mit identischem Zielfunktionswert konstruiert werden kann.

Sei $\{z_{q,q'} : q, q' \in Q(m), q \neq q'\}$ eine Lösung von $\Pi^2(m)$. Verändere diese Lösung sukzessive für alle $v \in V_m$ durch wiederholtes Ausführen der folgenden Schritte: (1) Finde ein $q \in Q_v$ mit $\sum_{q' \in Q(m), q' \neq q} z_{q,q'} = 0$. (2) Falls ein solches q existiert, dann existiert gemäss (3.21) auch ein $r \in Q_v$, $r \neq q$, mit $\sum_{q' \in Q(m), q' \neq r} z_{r,q'} > 0$. (3) Wähle ein $r' \in Q(m)$ mit $z_{r,r'} > 0$. (4) Setze $z_{r,r'} = 0$ und $z_{q,r'} = 1$.

Die nach einer solchen Variablenänderung erhaltenen Werte bilden offensichtlich wiederum eine Lösung von $\Pi^2(m)$. Der Zielfunktionswert der neuen Lösung ist identisch mit demjenigen der vorangehenden Lösung, da $c_{h(r), t(r')} = c_{h(q), t(r')} = c_{v, t(r')}$. Wenn für alle $v \in V$ ein $q \in Q_v$ mit der obigen Eigenschaft mehr gefunden wird, liegt eine Lösung von $\Pi^2(m)$ vor, welche alle Gleichungen (3.16) erfüllt. Auf analoge Weise kann diese Lösung weiter verändert werden, so dass sie schliesslich auch alle Gleichungen (3.17) erfüllt. Damit ist die Behauptung bewiesen.

Zusammenfassend wurde zunächst gezeigt, dass eine untere Schranke für die Leerfahrten-Belastung von Prozessor m gegeben ist durch das Optimum des Traveling Salesman Problems $\Pi^0(m)$. Da das Zuordnungsproblem $\Pi^1(m)$ eine Relaxation von $\Pi^0(m)$ bildet, ist auch das Optimum von $\Pi^1(m)$ eine (schlechtere) untere Schranke für die Leerfahrten-Belastung. Schliesslich wurde gezeigt, dass das Optimum des Transport-Problems $\Pi(m)$ identisch ist mit dem Optimum von $\Pi^1(m)$ und folglich eine untere Schranke für die Leerfahrten-Belastung darstellt. ♦

Betrachte nun eine optimale Lösung $\{x^i, \bar{x}^i, y : i \in I\}$ des Flussproblems von Abschnitt 3.2. mit den zusätzlichen Leerfahrten-Gleichungen (3.7)...(3.10). Die Werte von y stellen nicht unbedingt Optimallösungen der Transport-Probleme $\Pi(m)$, $m \in M$ dar. Das heisst, dass die Leerfahrten-Belastungen möglicherweise überschätzt sind in dieser Lösung. Diese Ueberschätzung hat jedoch keinen Einfluss auf das Optimum des Problems. Dies ist folgendermassen einzusehen: Aus der Optimallösung $\{x^i, \bar{x}^i, y : i \in I\}$ lässt sich eine zulässige Lösung $\{x^i, \bar{x}^i, y' : i \in I\}$ konstruieren, so dass y' eine Optimallösung für alle $\Pi(m)$, $m \in M$ darstellt hinsichtlich der gegebenen Flüsse x^i, \bar{x}^i , $i \in I$. In dieser Lösung sind die Leerfahrten korrekt durch eine untere Schranke abgeschätzt. Der Zielfunktionswert dieser Lösung kann jedoch nicht grösser sein als derjenige der ursprünglichen Optimallösung, da diese sonst widersprüchlicherweise nicht optimal wäre.

Prozessorbelastung

Im bisherigen Modell wurde eine untere Grenze für die zeitliche Belastung der Prozessoren auf der Basis der Transport-, Transfer- und Auf-/Abladeaktionen berechnet. Aufgrund der obigen Ausführungen lässt sich nun zusätzlich eine Abschätzung der Leerfahrtenbelastung in die Berechnung miteinbeziehen. Unter Einbezug der Leerfahrtenflüsse y erhält man folgende Abschätzung für die Belastung eines Prozessors $m \in M$:

$$L(m) = \sum \{c_e x_e^i : e \in E_m, i \in I\} + \sum \{c_e \bar{x}_e^i : e \in \bar{E}_m, i \in I\} + \sum \{c_e y_e : e \in E_m^S \cup E_m^L\} \quad (3.33)$$

Die neu hinzugefügte Komponente entspricht der Summe der Leerfahrtenzeiten auf den Fahrtbögen E_m^S und Schlingen E_m^L von Prozessor m .

Zielfunktion

In Abschnitt 3.2. wurde gezeigt, wie mit Hilfe eines negativen ε -Ausdrucks in der Zielfunktion unnötige Zwischenhalte und zyklische Leerfahrten in einer Optimallösung eliminiert werden

können. Analog kann eine Ueberschätzung der Leerfahrten-Belastung vermieden werden, indem die Leerfahrten ebenfalls in den ε -Ausdruck aufgenommen werden. Die neue Zielfunktion lautet dementsprechend:

$$\begin{aligned} \min \sum \{ \sigma^i + \bar{\sigma}^i : i \in I \} \\ - \varepsilon (\sum \{ c_e x_e^i : e \in E, i \in I \} + \sum \{ c_e \bar{x}_e^i : e \in \bar{E}, i \in I \} + \sum \{ c_e y_e : e \in E^S \cup E^L \}) \end{aligned} \quad (3.34)$$

Zusammenfassend wurde in diesem Abschnitt gezeigt, wie die Belastung der Prozessoren durch Leerfahrten im Modell berücksichtigt werden kann. Dazu wurde für jeden Prozessor ein Transport-Problem definiert, welches eine untere Schranke für die Leerfahrtenbelastung berechnet. Diese Transport-Probleme sind in das Flussproblem eingebettet und hängen ab von den Werten der Ein- und Auslagerungsflüsse.

3.4. Rotation und Verweilzeiten

In den vorangehenden Abschnitten wurde ein Optimierungsmodell entwickelt, welches eine Menge von Ein- und Auslagerungen generiert, so dass die Anzahl der ein- und ausgelagerten Paletten maximal ist in der betrachteten Zeitperiode. Für die Wahl der Transportrouten der Ein- und Auslagerungen werden in dem Modell keine Vorgaben gemacht. Der Optimierungsalgorithmus entscheidet über die Start- und Zielpositionen der Routen sowie über die Fahrtwege zwischen diesen Positionen. Die Wahl der Lagerplätze, an denen eingelagert wird, oder von denen ausgelagert wird, ist somit Aufgabe des Optimierungsverfahrens und unterliegt keinen Einschränkungen.

Daraus resultiert ein unerwünschter Effekt: Da die Anzahl der Ein- und Auslagerungen maximiert werden soll, erfolgen die Ein- und Auslagerungen vorzugsweise an Lagerpositionen, welche sich nahe bei den Ein- und Ausgangspunkten befinden. Dadurch kann der Arbeitsaufwand der Prozessoren verringert und entsprechend die Zahl der Ein- und Auslagerungen erhöht werden. Eine mögliche Konsequenz ist, dass in einer Optimallösung nur einige wenige Lagerpositionen aktiv sind, an denen eine grosse Anzahl von Paletten ein- und ausgelagert wird. Dies ist für den konkreten Lagerbetrieb aus zwei Gründen unrealistisch: Erstens haben die Lagerpositionen eine beschränkte Lagerkapazität. Zweitens verbleibt eine Palette normalerweise für einen gewisse Zeit im Lager und wird nicht sofort nach der Einlagerung wieder ausgelagert.

An dieser Stelle sei noch einmal darauf hingewiesen, dass aus Gründen der Modellgrösse ein reduzierter System-Graph betrachtet wird, bei welchem nicht alle Lagerplätze einzeln als Knoten verwaltet werden (vgl. Abschnitt 3.1.). Die Lagerplätze eines Korridors werden zu einem einzigen

(oder einigen wenigen) fiktiven Lagerknoten zusammengefasst, dessen Kapazität der Anzahl der Lagerplätze entspricht, welche durch diesen Knoten repräsentiert werden.

Falls ein Lagerprozess betrachtet wird, bei welchem nur Einlagerungen durchgeführt werden, ist die maximale Anzahl Einlagerungen an einem Lagerknoten gegeben durch die Anzahl der Lagerplätze, welche durch diesen Knoten repräsentiert werden. Wenn nur Auslagerungen betrachtet werden, ist die Anzahl der Auslagerungen ebenfalls durch die Lagerkapazität des Lagerknotens beschränkt. Falls in einem Lagerprozess gleichzeitig Ein- und Auslagerungen erfolgen, kann zwischen drei Arten von Paletten unterschieden werden: (1) Paletten, die eingelagert werden und im Lager bleiben, (2) Paletten, die anfänglich im Lager sind und ausgelagert werden, (3) Paletten, welche eingelagert und im Verlaufe des Lagerprozesses wieder ausgelagert werden. Diese Paletten werden rotierende Paletten genannt (bezüglich des betrachteten Lagerprozesses).

Rotierende Paletten verbleiben normalerweise für eine bestimmte Zeitdauer im Lager. Häufig wird bei der Lagerhaltung unterschieden zwischen verschiedenen Klassen von Artikeln, welche sich bezüglich ihrer durchschnittlichen Aufenthaltszeit im Lager unterscheiden. Ein bekannter Ansatz ist die sogenannte ABC-Klassifizierung, bei welcher die Artikel in Schnelldreher, Normaldreher und Langsamdreher eingeteilt werden. Schnelldreher haben eine relativ kurze mittlere Verweilzeit, während sich Langsamdreher durchschnittlich viel länger im Lager aufhalten.

In diesem Abschnitt wird gezeigt, wie die Informationen über die Lagerkapazität der Lagerknoten sowie die mittlere Verweilzeit der verschiedenen Artikelklassen in das Modell integriert werden können. Ziel dabei ist, den oben erwähnten, unerwünschten Effekt zu eliminieren und eine realistische Verteilung der Ein- und Auslagerungen auf die Lagerknoten zu erreichen.

An dieser Stelle wird auch ersichtlich, weshalb bei der obigen Formulierung des Modells verschiedene Typen $i \in I$ von Artikeln (bzw. verschiedene Typen x^i von Güterflüssen) eingeführt wurden. Diese Unterscheidung erlaubt es, Artikelklassen mit unterschiedlicher Verweildauer getrennt voneinander im Modell zu behandeln.

Zusätzliche Modellgrößen

Es werden die folgenden zusätzlichen Größen in das bestehende Modell eingeführt:

Parameter:

τ^i	Mittlere Verweilzeit (in Std.) einer Palette von Artikeltyp i , $i \in I$
n_v	Lagerkapazität (Anzahl repräsentierte Lagerplätze) von Lagerknoten v , $v \in V^P$
U_v^i	Untere Schranke (Anzahl Paletten) für den Anfangslagerbestand von Artikeltyp i im Lagerknoten v , $v \in V^P$. Es gilt $U_v^i \geq 0$
\hat{U}_v^i	Obere Schranke (Anzahl Paletten) für den Anfangslagerbestand von Artikeltyp i im Lagerknoten v , $v \in V^P$. Es gilt $\hat{U}_v^i \leq n_v$

Variablen:

- u_v^i Anfangslagerbestand (Anzahl Paletten) von Artikeltyp i im Lagerknoten v , $v \in V^P$,
 $i \in I$
- s_v^i Anzahl stationärer Paletten von Artikeltyp i im Lagerknoten v , $v \in V^P$, $i \in I$

Ein- und Auslagerungen an den Lagerknoten

Die Gesamtzahl der Einlagerungen eines Artikeltyps i an einem Lagerknoten $v \in V^P$ ist gegeben durch den Einlagerungsfluss $x_{v,t}^i$ auf dem Abgabebogen (v, t) , welcher vom Lagerknoten v zur Einlagerungssenke t führt. Analog ist die Gesamtzahl der Auslagerungen an einem Lagerknoten $v \in V^P$ gegeben durch den Auslagerungsfluss $x_{\bar{s},v}^i$ auf dem Aufnahmebogen (\bar{s}, v) , welcher von der Auslagerungsquelle \bar{s} zum Knoten v führt.

Lagerbestände

Die beabsichtigte Modellerweiterung erfordert die Verwaltung der anfänglichen Lagerbestände in den Lagerknoten. Dazu werden die Variablen u_v^i eingeführt, welche für jeden Lagerknoten $v \in V^P$ angeben, wieviel Paletten von jedem Artikeltyp i vor Prozessbeginn in dem Knoten v gelagert sind. Da die Grössen u_v^i als Variablen modelliert werden, entscheidet das Modell über den anfänglichen Lagerzustand und ist somit unabhängig von der Vorgabe eines bestimmten Anfangszustandes. Um trotzdem eine Einflussmöglichkeit auf den Anfangszustand zu haben, werden die unteren und oberen Schranken U_v^i bzw. \hat{U}_v^i als Parameter eingeführt. Für die Anfangslagerbestände der verschiedenen Artikeltypen in den Lagerknoten wird gefordert, dass sie sich innerhalb dieser Schranken befinden. Für den gesamten Anfangslagerbestand (aufsummiert über alle Artikeltypen) in einem Lagerknoten muss gelten, dass dieser die Lagerkapazität des Knotens nicht überschreitet. Daraus resultieren die folgenden Restriktionen:

$$U_v^i \leq u_v^i \leq \hat{U}_v^i, \quad v \in V^P, i \in I \quad (3.35)$$

$$\sum \{u_v^i : i \in I\} \leq n_v, \quad v \in V^P \quad (3.36)$$

Die Endlagerbestände der Artikeltypen in einem Lagerknoten $v \in V^P$ sind gegeben durch die Summe der Anfangsbestände und der Einlagerungen abzüglich der Auslagerungen, d.h. $u_v^i + x_{v,t}^i - x_{\bar{s},v}^i$. Es wird gefordert, dass diese Endlagerbestände nicht negativ sind. Für den gesamten Endlagerbestand (aufsummiert über alle Artikeltypen) in einem Lagerknoten muss gelten, dass dieser die Lagerkapazität des Knotens nicht überschreitet. Daraus resultieren die folgenden Restriktionen:

$$u_v^i + x_{v,t}^i - x_{\bar{s},v}^i \geq 0, \quad v \in V^P, i \in I \quad (3.37)$$

$$\sum \{u_v^i + x_{v,t}^i - x_{\bar{s},v}^i : i \in I\} \leq n_v, \quad v \in V^P \quad (3.38)$$

Rotierende Paletten

Neben den Anfangslagerbeständen u_v^i werden als zusätzliche Variablen die Anzahl s_v^i der stationären Paletten von Artikeltyp i im Lagerknoten $v \in V^P$ eingeführt. Ein Palette heisst stationär, wenn sie sich im Lager befindet, aber während des gesamten betrachteten Lagerprozesses nicht bewegt wird. Die Gesamtmenge $x_{\bar{s},v}^i$ der ausgelagerten Paletten von Artikeltyp i im Knoten v lässt sich unterteilen in nicht-rotierende und rotierende Paletten. Ein nicht-rotierende Auslagerungspalette befindet sich zu Beginn im Lager und wird während des Lagerprozesses ausgelagert. Eine rotierende Auslagerungspalette wird während des Lagerprozesses ein- und wieder ausgelagert. Die Anzahl der nicht-rotierenden Auslagerungspaletten von Artikeltyp i im Lagerknoten v ist gegeben durch $u_v^i - s_v^i$. Die Anzahl der rotierenden Auslagerungspaletten entspricht $x_{\bar{s},v}^i - (u_v^i - s_v^i)$. Da die Anzahl der nicht-rotierenden bzw. rotierenden Paletten nicht negativ sein kann, werden die folgenden Restriktionen aufgestellt:

$$u_v^i - s_v^i \geq 0, \quad v \in V^P, i \in I \quad (3.39)$$

$$x_{\bar{s},v}^i - (u_v^i - s_v^i) \geq 0, \quad v \in V^P, i \in I \quad (3.40)$$

Eine rotierende Palette von Artikeltyp i hält sich im Durchschnitt τ^i Zeiteinheiten im Lager auf. Die Summe der mittleren Verweilzeiten aller rotierenden Paletten in einem Lagerknoten ist gegeben durch $\sum \{ (x_{\bar{s},v}^i - (u_v^i - s_v^i)) \tau^i : i \in I \}$. Diese Summe entspricht dem mittleren Zeitbedarf für Aufenthalte von rotierenden Paletten im Lagerknoten $v \in V^P$. Stationäre Paletten halten sich während der ganzen Zeitperiode T im Lagerknoten auf. Der gesamte Zeitbedarf für Aufenthalte von stationären Paletten in einem Lagerknoten v ist somit gegeben durch $\sum \{ s_v^i T : i \in I \}$.

In einem Lagerknoten v können maximal n_v Paletten während maximal T Zeiteinheiten gelagert werden. Die gesamte verfügbare Zeit für Aufenthalte in einem Lagerknoten ist somit gegeben durch $n_v T$. Als Bedingung wird nun gefordert, dass die Summe der Aufenthaltszeiten der rotierenden und der stationären Paletten die verfügbare Aufenthaltszeit in einem Lagerknoten nicht überschreiten darf. Die entsprechende Restriktion lautet:

$$\sum \{ (x_{\bar{s},v}^i - (u_v^i - s_v^i)) \tau^i : i \in I \} + \sum \{ s_v^i T : i \in I \} \leq n_v T, \quad v \in V^P$$

Durch diese Bedingung wird die Zahl der rotierenden Paletten in einem Lagerknoten in Abhängigkeit von der mittleren Aufenthaltsdauer eingeschränkt. Damit wird der oben erwähnte, unerwünschte Effekt beseitigt, dass in einigen wenigen Lagerknoten sehr viele Paletten rotieren, d.h. ein- und wieder ausgelagert werden. Durch Umformung lässt sich die obige Restriktion schreiben als:

$$\sum \{ \tau^i \frac{1}{T} x_{\bar{s},v}^i : i \in I \} - \sum \{ \tau^i \frac{1}{T} (u_v^i - s_v^i) : i \in I \} \leq n_v - \sum_{i \in I} s_v^i, \quad v \in V^P \quad (3.41)$$

Bemerkung

Falls keine unteren Schranken als Vorbedingungen für die Anfangslagerbestände u_v^i festgelegt sind, d.h. falls $U_v^i = 0$ für alle $i \in I$ und $v \in V^P$, können alle $s_v^i = 0$ gesetzt werden, d.h. die Variablen s_v^i können eliminiert werden und die obigen Restriktionen vereinfachen sich dementsprechend. Dies entspricht der Forderung, dass stets alle am Anfang im Lager vorhandenen Paletten ausgelagert werden. Daraus resultiert keine Beschränkung für $x_{\bar{s},v}^i$, da u_v^i beliebig nach unten angepasst werden kann.

Zusammenhang mit Formel von Little

In diesem Abschnitt wird gezeigt, dass sich bei Betrachtung über eine lange Zeitperiode, d.h. $T \rightarrow \infty$, die Restriktionen (3.41) als Anwendung des bekannten Little'schen Gesetzes interpretieren lassen.

Die Formel von Little bezieht sich allgemein auf Bearbeitungssysteme, bei welchen Objekte in das System eintreten, sich eine Zeit lang im System aufhalten, und anschliessend wieder austreten. Sie gilt im wesentlichen dann, wenn sich ein System im Gleichgewicht befindet, d.h. wenn die Zahl der Ein- und Austritte, über lange Zeit betrachtet, identisch ist. Genauere Angaben über die Gültigkeit der Formel von Little finden sich beispielsweise in [Little, J. D. C. (1960)], [Jewell, W. S. (1967)] und [Stidham Jr., S. (1972)].

Das Little'sche Gesetz stellt einen Zusammenhang her zwischen der mittleren Eintrittsrate λ , der mittleren Aufenthaltsdauer τ der Objekte im System und der mittleren Anzahl \bar{u} von Objekten, welche sich im System aufhalten. Sie besagt, dass die mittlere Anzahl Objekte im System dem Produkt der mittleren Eintrittsrate und der mittleren Aufenthaltsdauer entspricht:

$$\bar{u} = \lambda \tau$$

Betrachte nun die Restriktionen (3.41) für $T \rightarrow \infty$. Zunächst kann in diesem Fall angenommen werden, dass $s_v^i \rightarrow 0$ ist, da in einem Lager sinnvollerweise jede Lagereinheit irgendwann einmal ausgelagert wird. Somit reduzieren sich die Restriktionen zu $\sum \{ \tau \frac{1}{T} x_{\bar{s},v}^i : i \in I \} - \sum \{ \tau \frac{1}{T} u_v^i : i \in I \} \leq n_v$. Aufgrund von (3.36) gilt $u_v^i \leq n_v$. Folglich geht auch die zweite Summe gegen Null.

Da über sehr lange Zeit betrachtet die Ein- und Auslagerungsbilanz in einem Lagerknoten annähernd ausgeglichen sein muss, kann für $T \rightarrow \infty$ angenommen werden, dass $x_{v,t}^i \cong x_{\bar{s},v}^i$ gilt, d.h. dass die Zahl der Ein- und Auslagerungen identisch ist. Somit entsprechen die Restriktionen (3.41) für $T \rightarrow \infty$ den folgenden Bedingungen:

$$\sum \{ \tau \frac{1}{T} x_{v,t}^i : i \in I \} \leq n_v, \quad v \in V^P$$

Der Ausdruck $\frac{1}{T} x_{v,t}^i$ entspricht der mittleren Einlagerungsrate λ_v^i von Artikeltyp i im Lagerknoten v . Der Parameter τ^i entspricht der mittleren Verweilzeit einer Palette von Artikeltyp i in

einem Lagerknoten. Folglich kann der Ausdruck $\tau^i \frac{1}{T} x_{v,t}^i = \tau^i \lambda_v^i$ aufgrund des Little'schen Gesetzes interpretiert werden als der mittlere Lagerbestand von Artikeltyp i im Lagerknoten v . Die obige Bedingung besagt somit, dass die Summe der mittleren Lagerbestände der verschiedenen Artikel die Kapazität n_v des Lagerknotens nicht überschreiten darf.

3.5. Modellübersicht

In diesem Abschnitt werden die in den vorangehenden Abschnitten entwickelten Modellkomponenten zusammengefasst und in einer vollständigen Modellübersicht präsentiert. Das Modell stellt ein Lineares Programm dar und basiert auf dem Konzept der Mehrgüterflüssen (multi-commodity flows). Darin eingebettet sind Transportprobleme zur Abschätzung der Leerfahrtenbelastung. Als weitere Elemente kommen die Restriktionen im Zusammenhang mit den Kapazitäten der Lagerknoten und den Verweilzeiten der Paletten hinzu.

Modell zur Leistungsanalyse automatisierter Kompaktlager

Parameter:

Allgemeine Parameter:

G	System-Graph, $G = \langle V^0, E^0, c \rangle$
I	Menge der verschiedenen Artikeltypen (Güterklassen, Commodities)
T	Dauer der betrachteten Zeitperiode
ε	Marginaler positiver Gewichtungsfaktor in der Zielfunktion

Ein- und Auslagerungsflüsse:

α^i	Verhältnis der Aus- zur Einlagerungsmenge von Artikel i , $i \in I$
β^i	Verhältnis der Einlagerungsmenge von Artikel i zur gesamten Einlagerungsmenge, $i \in I$, $\sum_{i \in I} \beta^i = 1$

Rotation und Verweilzeiten:

τ^i	Mittlere Verweilzeit (in Std.) einer Palette von Artikeltyp i , $i \in I$
n_v	Lagerkapazität (Anzahl repräsentierte Lagerplätze) von Lagerknoten v , $v \in V^P$
U_v^i	Untere Schranke (Anzahl Paletten) für den Anfangslagerbestand von Artikeltyp i im Lagerknoten v , $v \in V^P$. Es gilt $U_v^i \geq 0$
\hat{U}_v^i	Obere Schranke (Anzahl Paletten) für den Anfangslagerbestand von Artikeltyp i im Lagerknoten v , $v \in V^P$. Es gilt $\hat{U}_v^i \leq n_v$

Variablen:*Ein- und Auslagerungsflüsse:*

x_e^i	Einlagerungsfluss (Anzahl Paletten) von Artikeltyp i auf Bogen e , $e \in E$ (x_{vw}^i entspricht der Anzahl Einlagerungspaletten von Artikeltyp i , welche von Position v nach w transportiert bzw. übergeben werden)
\bar{x}_e^i	Auslagerungsfluss (Anzahl Paletten) von Artikeltyp i auf Bogen e , $e \in \bar{E}$ (\bar{x}_{vw}^i entspricht der Anzahl Auslagerungspaletten von Artikeltyp i , welche von Position v nach w transportiert bzw. übergeben werden)
σ^i	Anzahl eingelagerter Paletten von Artikeltyp i , $i \in I$
$\bar{\sigma}^i$	Anzahl ausgelagerter Paletten von Artikeltyp i , $i \in I$

Leerfahrten:

y_e	Anzahl Leertourfahrten auf dem Bogen e , $e \in E^S \cup E^L$
-------	---

Lagerbestände und stationäre Paletten:

u_v^i	Anfangslagerbestand (Anzahl Paletten) von Artikeltyp i im Lagerknoten v , $v \in V^P$, $i \in I$
s_v^i	Anzahl stationärer Paletten von Artikeltyp i im Lagerknoten v , $v \in V^P$, $i \in I$

Weitere Bezeichnungen:

$\delta^+(v)$	Menge der aus Knoten v herausgehenden Bogen in G , $v \in V^0$, $\delta^+(v) = \{(v, w) \in E^0\}$
$\delta^-(v)$	Menge der in Knoten v hineingehenden Bogen in G , $v \in V^0$, $\delta^-(v) = \{(w, v) \in E^0\}$

Zielfunktion:

$$\begin{aligned} & \max \\ & \sum \{\sigma^i + \bar{\sigma}^i : i \in I\} \\ & - \varepsilon (\sum \{c_e x_e^i : e \in E, i \in I\} + \sum \{c_e \bar{x}_e^i : e \in \bar{E}, i \in I\} + \sum \{c_e y_e : e \in E^S \cup E^L\}) \end{aligned} \quad (3.42)$$

Restriktionen:**Ein- und Auslagerungsflüsse:***Flussbilanzen für Ein- und Auslagerungen:*

$$\sum \{x_e^i : e \in \delta^+(v) \cap E\} - \sum \{\bar{x}_e^i : e \in \delta^-(v) \cap \bar{E}\} = \begin{cases} \sigma^i & \text{für } v = s \\ 0 & \text{für } v \in V, \\ -\sigma^i & \text{für } v = t \end{cases} \quad i \in I \quad (3.43)$$

$$\sum \{x_e^i : e \in \delta^+(v) \cap \bar{E}\} - \sum \{x_e^i : e \in \delta^-(v) \cap \bar{E}\} = \begin{cases} \bar{\sigma}^i & \text{für } v = \bar{s} \\ 0 & \text{für } v \in V, \\ -\bar{\sigma}^i & \text{für } v = \bar{t} \end{cases} \quad i \in I \quad (3.44)$$

Kapazitätsrestriktionen:

$$\sum \{c_e x_e^i : e \in E_m, i \in I\} + \sum \{c_e x_e^i : e \in \bar{E}_m, i \in I\} + \sum \{c_e y_e : e \in E_m^S \cup E_m^L\} \leq T, \quad m \in M \quad (3.45)$$

Verhältnis Aus-/Einlagerungen:

$$\bar{\sigma}^i = \alpha^i \sigma^i, \quad i \in I \quad (3.46)$$

Commodity-Mix:

$$\sigma^i = \beta^i \sum \{\sigma^i : i \in I\}, \quad i \in I \quad (3.47)$$

Leerfahrten:

Transportprobleme für die Prozessoren:

$$\begin{aligned} \sum \{y_e : e \in \delta^+(v) \cap (E^S \cup E^L)\} &= \sum \{x_e^i : e \in \delta^+(v) \cap (E^T \cup E^*), i \in I\} + \sum \{x_e^i : e \in \delta^+(v) \cap (E^T \cup \bar{E}^*), i \in I\} \\ v &\in V_m, m \in M \end{aligned} \quad (3.48)$$

$$\begin{aligned} \sum \{y_e : e \in \delta^-(v) \cap (E^S \cup E^L)\} &= \sum \{x_e^i : e \in \delta^-(v) \cap (E^T \cup E^*), i \in I\} + \sum \{x_e^i : e \in \delta^-(v) \cap (E^T \cup \bar{E}^*), i \in I\} \\ v &\in V_m, m \in M \end{aligned} \quad (3.49)$$

Rotation und Verweilzeiten:

Anfangslagerbestände in den Lagerknoten:

$$U_v^i \leq u_v^i \leq \hat{U}_v^i, \quad v \in V^P, i \in I \quad (3.50)$$

$$\sum \{u_v^i : i \in I\} \leq n_v, \quad v \in V^P \quad (3.51)$$

Endlagerbestände in den Lagerknoten:

$$u_v^i + x_{v,t}^i - x_{\bar{s},v}^i \geq 0, \quad v \in V^P, i \in I \quad (3.52)$$

$$\sum \{u_v^i + x_{v,t}^i - x_{\bar{s},v}^i : i \in I\} \leq n_v, \quad v \in V^P \quad (3.53)$$

Nicht-rotierende und rotierende Paletten:

$$u_v^i - s_v^i \geq 0, \quad v \in V^P, i \in I \quad (3.54)$$

$$x_{\bar{s},v}^i - (u_v^i - s_v^i) \geq 0, \quad v \in V^P, i \in I \quad (3.55)$$

Rotationsbeschränkung:

$$\sum \left\{ \tau \frac{1}{T} x_{\bar{s},v}^i : i \in I \right\} - \sum \left\{ \tau \frac{1}{T} (u_v^i - s_v^i) : i \in I \right\} \leq n_v - \sum_{i \in I} s_v^i, \quad v \in V^P \quad (3.56)$$

Nichtnegativität:

$$x_e^i \geq 0, \quad e \in E, i \in I \quad (3.57)$$

$$\bar{x}_e^i \geq 0, \quad e \in \bar{E}, i \in I \quad (3.58)$$

$$\sigma^i, \bar{\sigma}^i \geq 0, \quad i \in I \quad (3.59)$$

$$y_e \geq 0, \quad e \in E^S \cup E^L \quad (3.60)$$

$$u_v^i, s_v^i \geq 0, \quad v \in V^P, i \in I \quad (3.61)$$

3.6. Numerische Resultate

In diesem Abschnitt wird anhand von zwei konkreten Beispielen die Funktion und Anwendung des entworfenen Modells zur Leistungsanalyse gezeigt.

In den folgenden Beispielen wird, falls nicht anders vermerkt, bei der Betrachtung automatisierter Kompaktlager von einer Standardkonfiguration ausgegangen. Es wird angenommen, dass jeder Sektor beidseitig an einen Quergang angrenzt. Weiter wird vorausgesetzt, dass sich auf beiden Seiten eines Quergangs ein Aufzug befindet, und dass die Ein- und Ausgangspunkte auf der untersten Lagerebene bei den Aufzügen liegen. Schliesslich wird angenommen, dass alle Sektoren die gleiche Anzahl Korridore umfassen und jeder Korridor gleich viele Lagerplätze besitzt. Die Lagerplätze eines Korridors werden im System-Graphen durch einen einzelnen Lagerknoten repräsentiert.

Problemgrösse

Bei der Modellierung von Lagersystemen mit realistischen Problemdaten resultieren Lineare Programme von beträchtlicher Grössenordnung. Da die Variablen und Restriktionen des Modells den Knoten und Bogen des System-Graphen zugeordnet sind, hängt die Modellgrösse direkt von der Grösse des System-Graphen ab.

Betrachte ein automatisiertes Kompaktlager mit s Sektoren, f Ebenen, k Korridoren pro Sektor und p Positionen pro Korridor. Die Anzahl Aufzüge ist gegeben durch $|M^A| = 2s$ die Anzahl

Quergangswagen durch $|M^B| = (s+1)f$ und die Anzahl Korridorwagen durch $|M^C| = sfk$. Die Anzahl Lagerpositionen beträgt $|V^P| = sfkp$.

Die Grösse der Prozessor-Subgraphen $G_m = \langle V_m, E_m^S \cup E_m^L, c \rangle$ $m \in M$ kann wie folgt berechnet werden: Es gilt $|V_m| = f$ für $m \in M^A$, $|V_m| = k+1$ für $m \in M^B$ und $|V_m| = p$ für $m \in M^C$. Da die Prozessor-Subgraphen vollständig sind, beträgt die Bogenzahl $|E_m^S \cup E_m^L| = |V_m|^2$. Da zwischen jedem Quergangswagen und Aufzug sowie zwischen jedem Quergangswagen und Korridorwagen ein Transferbogen in beiden Richtungen existiert, ist die Gesamtzahl der Transferbogen gegeben durch $E^T = 2 \cdot 2 \cdot |M^B| + 2 \cdot 2 \cdot |M^C|$. Die Anzahl Aufnahme-/Abgabebogen für Einlagerungen bzw. Auslagerungen entspricht $E^* = \bar{E}^* = |M^A| + |V^P|$.

Zusammenfassend ist die Grösse des Ein- bzw. Auslagerungsgraphen $\langle V \cup \{s, t\}, E, c \rangle$ bzw. $\langle V \cup \{\bar{s}, \bar{t}\}, \bar{E}, c \rangle$ wie folgt gegeben:

$$\begin{aligned} |V| &= \sum_{m \in M} |V_m| = [2sf] + [(s+1)f(k+2)] + [sfk(p+2)] \\ |E| = \bar{E} &= \sum_{m \in M} |E_m^S \cup E_m^L| + E^T + E^* \\ &= [2sf^2 + (s+1)f(k+2)^2 + sfk(p+2)^2] + [4(s+1)f + 4sfk] + [2s + sfkp] \end{aligned}$$

Betrachte nun als Beispiel ein Lager mit $s = 5$ Sektoren, $f = 6$ Ebenen, $k = 20$ Korridoren pro Sektor. Da die Lagerplätze eines Korridors durch einen einzelnen Knoten repräsentiert werden, gilt $p = 1$. Mit diesen Lagerdimensionen resultiert ein System-Graph der folgenden Grössenordnung:

$$\begin{aligned} |V| &= 60 + 792 + 1800 = 2652 \\ |E| = \bar{E} &= [360 + 17424 + 5400] + [144 + 2400] + [10 + 600] = 26338 \end{aligned}$$

Für jeden Artikeltyp $i \in I$ enthält das Modell einen Einlagerungsfluss $x^i \in \mathbb{R}^E$ und einen Auslagerungsfluss $\bar{x}^i \in \mathbb{R}^{\bar{E}}$. Bei drei Artikeltypen ergibt sich daraus für die Flussvariablen eine Gesamtzahl in der Grössenordnung von 150'000 Variablen. Dazu kommen noch die Variablen für die Leerfahrtenflüsse sowie für die Lagerbestände und die stationären Paletten.

Der Hauptanteil der Restriktionen wird gebildet durch die Bilanzgleichungen der Ein- und Auslagerungsflüsse (für jeden Artikeltyp $i \in I$) sowie durch die Transportgleichungen für die Leerfahrten. Die Anzahl dieser Restriktionen ist im wesentlichen gegeben durch die Anzahl Knoten des System-Graphen. Für das betrachtete Beispiel resultiert daraus eine Anzahl von Restriktionen in der Grössenordnung von 18000 Restriktionen.

Dieses Beispiel zeigt, dass bei realistischen Lagerdimensionen die Linearen Programme sehr gross werden können. Die berechneten Zahlen beziehen sich auf einen reduzierten System-Graphen, bei welchem die Lagerplätze eines Korridors durch einen einzelnen Knoten repräsentiert werden. Ohne diese Reduktion würde die Linearen Programme noch weit umfangreicher ausfallen und in einem Grössenbereich liegen, welcher kaum mehr zu bewältigen wäre.

Abschliessend sei darauf hingewiesen, dass die Modellgrösse wesentlich verringert werden kann, wenn die Prozessorgraphen G_m nicht als vollständige Graphen, sondern nur als beidseitig gerichtete Wege modelliert werden. Dies bedeutet, dass durch Weglassen der transitiven Bogen in G_m auf die Darstellung der direkten Transportfahrten verzichtet wird und stattdessen Fahrten mit Zwischenhalten betrachtet werden. Falls die Prozessoren sich mit konstanter Geschwindigkeit bewegen würden, wäre dieser Ansatz äquivalent zum Modell mit transitiven Bogen, da die Fahrzeit einer Direktfahrt in diesem Falle der Summe der Fahrzeiten der Teilfahrten entsprechen würden. Weiter oben wurde jedoch schon darauf hingewiesen, dass die Beschleunigungs- und Abbremsvorgänge einen wesentlichen Einfluss auf die Fahrzeiten der Prozessoren haben. Deshalb werden die transitiven Bogen im Modell belassen und auf diese Vereinfachung verzichtet.

Implementation

Zur Implementation des Modells wurde die an der Universität Freiburg entwickelte Modelliersprache LPL verwendet (vgl. [Hürlimann, T. (2001)]). LPL ermöglicht eine parametrisierte Formulierung von Linearen Programmen und generiert als Output sogenannte MPS-Dateien, welche die Linearen Programme in einer standardisierten Form darstellen. Die MPS-Dateien können von einem externen Optimierungsprogramm (LP-Solver) weiterverarbeitet werden. Zur Optimierung wurde der bekannte Solver CPLEX von der Firma ILOG benützt.

Die numerischen Tests wurden auf einem Rechner mit Prozessor AMD K6-2, 233 MHz, und 64 MB Speicher unter Windows NT 4.0 durchgeführt. Der Rechenaufwand für Beispiele mit realistischen Lagerdimensionen erwies sich als relativ hoch, aber bewältigbar. Für Beispiele in der Gröszenordnung des nachfolgenden Beispiels 2 ergab sich ein Zeitaufwand von bis zu 10 Stunden für die Kompilation der LPL-Programme und die Generierung der MPS-Dateien. Die Optimierung der Linearen Programme mit dem CPLEX-Solver benötigte wesentlich weniger Zeit und lag im Bereich von 0.5 - 2 Stunden. Der Grund für die hohen Kompilierzeiten liegt in der Generierung der grossen Matrizen, welche für die Darstellung der Lager im System-Graphen benötigt werden. Eine wesentliche Reduktion der Kompilierzeiten könnte erreicht werden, wenn die Lagerdaten von einem separaten, effizienten Programm generiert würden und als Input für die LPL-Modellbeschreibung verwendet würden.

Prozessoren-Parameter

Die Prozessoren-Parameter für die folgenden Beispiele basieren auf den Zahlen eines real existierenden automatisierten Kompaktlagers. Durch die Prozessoren-Parameter wird die Dauer der verschiedenen Prozessoraktionen (Transport, Uebergabe/Uebernahme, Auf-/Abladen) bestimmt, woraus sich die Bogengewichte im System-Graphen ergeben. Die Dauer der Transportfahrten wurde ermittelt aufgrund der Distanzen zwischen den Fahrtpositionen sowie der Geschwindigkeits- und Beschleunigungsparameter der verschiedenen Prozessoren. Die Dauer der Uebergabe- und Uebernahme- sowie der Auf- und Ablade-Aktionen basiert auf den Zeitparametern der ver-

schiedenen Teilaktionen (Positionierung des Transportwagens, Heben- und Senken der Ladevorrichtungen, Ein- und Ausfahren der Ladegabeln, etc.).

3.6.1. Beispiel 1

Ziel dieses Beispiels ist, zu zeigen, wie in einer optimalen Lösung die Ein- und Auslagerungen der verschiedenen Artikeltypen auf die Lagerkorridore verteilt werden. Daraus lassen sich unter anderem wertvolle Informationen für den Entwurf von Lagerstrategien ableiten.

Betrachtet wird ein kleiner Lagerausschnitt, welcher nur einen Sektor und eine Ebene umfasst. Der Sektor wird von einem einzigen Quergangswagen bedient, an dessen beiden Enden sich je ein Aufzug befindet. Der Sektor umfasst 16 Korridore und in jedem Korridor stehen 20 Lagerplätze zur Verfügung. Die betrachtete Zeitperiode beträgt $T = 8 \text{ Std.}$ Für die anfänglichen Lagerbestände in den Lagerknoten (d.h. Korridoren) werden keine Bedingungen vorgegeben, d.h. $U_v^i = 0$ und $\hat{U}_v^i = n_v$ für $i \in I$ $v \in V^P$.

Die Artikel werden unterteilt in vier Artikeltypen $i \in I = \{1, 2, 3, 4\}$ mit unterschiedlicher Verweilzeit τ_i . Die Verweilzeiten werden in Abhängigkeit von der Zeitperiode T spezifiziert. Die Problemparameter für die verschiedenen Artikeltypen i sind der folgenden Uebersichtstabelle zu entnehmen. Der Parameter α_i entspricht dem Verhältnis der Aus- zu den Einlagerungen von Artikeltyp i und β_i dem Anteil von Typ i am gesamten Einlagerungsfluss.

Als Resultat der Optimierung zeigt die Tabelle für jeden Artikeltyp i eine obere Schranke σ^i bzw. $\bar{\sigma}^i$ für die Anzahl Ein- bzw. Auslagerungen, welche im Zeitraum $T = 8 \text{ Std.}$ möglich sind. Die gesamte Ein- bzw. Auslagerungsleistung ist gegeben durch $\sum_{i \in I} \sigma^i = \sum_{i \in I} \bar{\sigma}^i = 44$ (Die totale Lagerleistung beträgt somit 880 Paletten).

Lagerdimensionen: 1 Sektor, 1 Ebene, 16 Korridore/Sektor, 20 Lagerplätze/Korridor Zeitperiode: T = 8h				
Artikeltyp i :	1	2	3	4
Verhältnis Aus-/Einlagerungen α_i :	1	1	1	1
Anteil an Einlagerungen β_i :	0.4	0.3	0.2	0.1
Mittlere Verweilzeit τ_i :	0.25T	0.5T	1T	2T
Einlagerungsleistung σ^i :	176	132	88	44
Auslagerungsleistung $\bar{\sigma}^i$:	176	132	88	44

Abbildung 3.2. Uebersicht über Modell-Parameter und Lagerleistung

Verteilung der Ein- und Auslagerungen

Die folgende Abbildung zeigt die Verteilung der Ein- und Auslagerungen der verschieden Artikeltypen $i \in I$ auf die Korridore. Die Wahl der Ein- und Auslagerungspositionen ist abhängig von der mittleren Verweilzeit τ_i der Artikeltypen und wird im Modell kontrolliert durch die Restriktionen (3.50) ... (3.56).

Die Abbildung zeigt, dass die Paletten mit der kleinsten Verweilzeit (Artikeltyp 1) in denjenigen Korridoren ein- und ausgelagert werden, welche am nächsten bei den Aufzügen liegen (d.h. Korridore C_1, C_2 und C_{15}, C_{16}). Je grösser die Verweildauer eines Artikeltyps ist, desto weiter von den Aufzügen entfernt werden die Paletten ein- und ausgelagert. Gleichzeitig nimmt die Zahl der Ein- und Auslagerungen in einem Korridor ab, je weiter der Korridor vom Aufzug entfernt ist.

Dieser Effekt lässt sich folgendermassen erklären: (1) Je näher sich ein Korridor bei einem Aufzug befindet, desto kürzer ist die Transportstrecke des Quergangwagens zwischen dem Korridor und dem Aufzug. Folglich belasten Ein- und Auslagerungen in randständigen Korridoren den Quergangswagen weniger als solche in weiter entfernt liegenden Korridoren. (2) Je kürzer die Verweildauer eines Artikeltyps ist, desto mehr Paletten können in einem Korridor rotieren, d.h. ein- und wieder ausgelagert werden. Somit werden in einer Optimallösung die randständigen Korridore für Paletten mit kleiner Verweilzeit verwendet, welche eine hohe Rotation im Korridor erlauben.

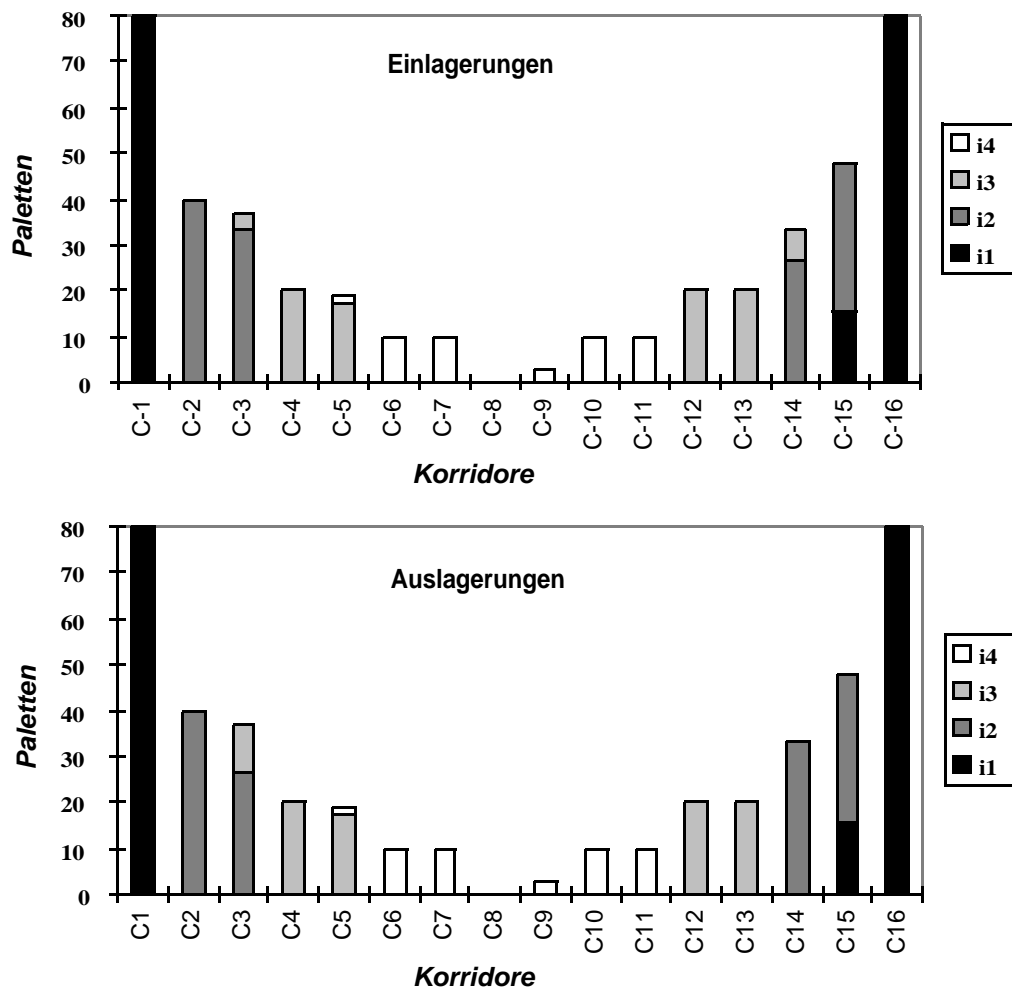


Abbildung 3.3. Verteilung der Ein- und Auslagerungen auf die Korridore

Dieses Beispiel zeigt, dass mit Hilfe der Restriktionen (3.50) ... (3.56) eine Lagerstrategie modelliert wird, die auf intuitiv nachvollziehbaren Entscheidungsmechanismen basiert. Artikeltypen, welche häufig ein- und ausgelagert werden (sogenannte Schnelldreher), werden an Lagerplätzen gelagert, welche sich nahe an den Ein- und Ausgangspunkten befinden und mit wenig Transportaufwand bedient werden können. Artikeltypen mit grosser Verweilzeit (sogenannte Langsamdreher), werden in weniger günstig gelegenen Korridoren gelagert.

In der Praxis ist es häufig so, dass die Lagerplätze in verschiedene Lagerbereiche unterteilt werden, welche den Artikeltypen (z.B. ABC-Klassifizierung) zugeordnet werden. Schnelldreher werden in gut erreichbaren Lagerbereichen positioniert und Langsamdreher in Lagerbereichen, welche mehr Transportaufwand erfordern. Im obigen Beispiel, wo nur ein Sektor und eine Ebene betrachtet wird, ist sofort ersichtlich, welche Lagerbereiche günstig gelegen sind. Ein reales Kompaktlager umfasst jedoch eine Vielzahl von Aufzügen, Quergangswagen, Durchfahrtskorri-

doren sowie Ein- und Ausgangspositionen, welche auf verschiedene Art angeordnet sein können. In einem solchen Lager ist die Bestimmung der günstig gelegenen Lagerbereiche nicht mehr offensichtlich und stellt ein komplexes Problem dar.

Eine wichtige Rolle spielt dabei auch die Frage nach den Engpass-Prozessoren des Lagersystems. Ein Lagerbereich kann günstig gelegen sein in dem Sinne, dass er distanzmässig nahe bei einem Ein- bzw. Auslagerungspunkt liegt und mit wenig Transportaufwand bedient werden kann. Wenn jedoch ein zur Bedienung benötigter Prozessor eine Engpass-Ressource darstellt, ist es durchaus möglich, dass ein anderer Lagerbereich, welcher weiter entfernt ist und mehr Transportaufwand erfordert, insgesamt günstiger gelgen ist für die Positionierung von Schnelldrehern. Dies ist möglich, wenn zur Bedienung dieses Lagerbereichs andere Prozessoren benützt werden als die Engpass-Ressource und diese dadurch entlastet wird.

Diese Diskussion zeigt, dass die Lagerstrategie (d.h. die Verteilung der Ein- und Auslagerungen der Artikeltypen auf die Lagerpositionen) in automatisierten Kompaktlagern ein komplexes Entscheidungsproblem darstellt. Das Modell zur Leistungsanalyse kann beim Entwurf einer Lagerstrategie wertvolle Unterstützung bieten. Es liefert Informationen darüber, wie die verschiedenen Artikeltypen in Abhängigkeit von ihrer Verweilzeit optimal auf die Lagerbereiche verteilt werden können. Das Modell gibt Hinweise, welche Lagerbereiche günstig gelegen sind, welche Artikeltypen in welchen Bereichen gelagert werden sollen und wie gross die Lagerbereiche für die verschiedenen Artikeltypen eingeplant werden müssen.

Abschliessend sei noch darauf hingewiesen, dass die Verweilzeiten in diesem Beispiel relativ klein gewählt wurden ($0.25T \leq \tau_i \leq 2T$), damit der beschriebene Effekt bei Verteilung der Artikeltypen in Erscheinung tritt. Wenn die Verweilzeiten grösser sind als die betrachtete Zeitperiode T , findet in den Korridoren keine Palettenrotation statt, und die Wahl der Ein- und Auslagerungspositionen erfolgt willkürlich.

Belastung der Prozessoren

In der folgenden Abbildung wird gezeigt, wie sich der Arbeitsaufwand bei maximaler Leistung auf die verschiedenen Prozessoren verteilt. Die aufgeführten zeitlichen Belastungen stellen eine untere Schranke dar für den Arbeitsaufwand der Prozessoren, wenn ein Lagerprozess mit der maximal möglichen Lagerleistung von 880 Paletten in 8 Stunden realisiert werden soll. Gemäss Modell ist diese Aufwandabschätzungen unabhängig von den Entscheidungsmechanismen auf der Ebene der dispositiven Steuerung (Lagerstrategie, Routenwahl, Ausführungsreihenfolge, etc.) und gilt für beliebige Lagersteuerungen.

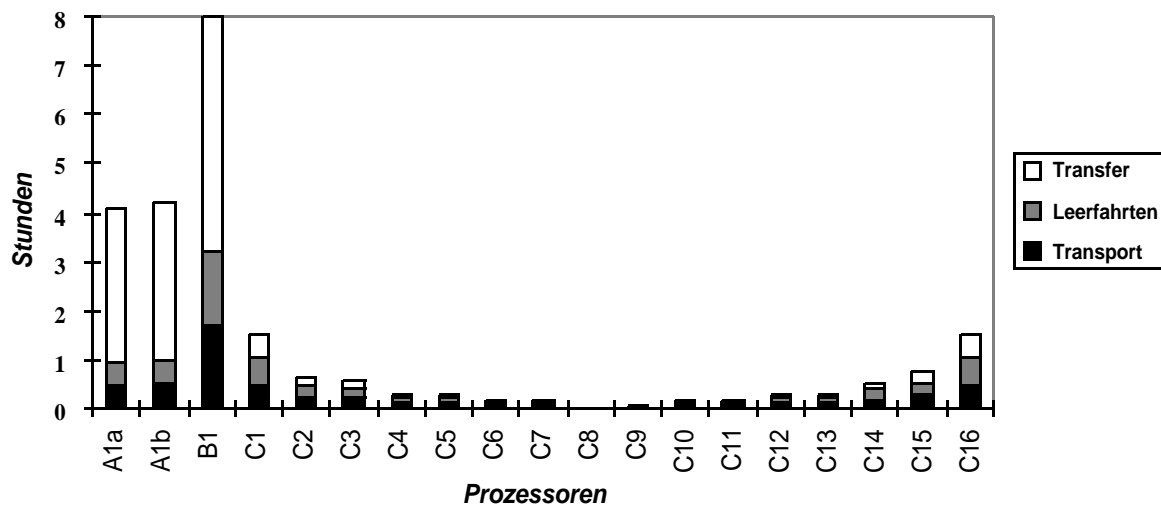


Abbildung 3.4. Belastung der Prozessoren

Die Aufzüge an den beiden Quergangenden sind bezeichnet als A_{1a} und A_{1b} der Quergangswagen als B_1 und die Korridorwagen als C_1, \dots, C_{16} . Die zeitliche Belastung der Prozessoren ist aufgeteilt in Transportfahrten, Leerfahrten und Transferaktionen (Uebergabe/Uebernahme und Auf-/Abladen).

Als erstes wird aus dem Diagramm ersichtlich, dass der Quergangswagen die Engpass-Ressource darstellt. Dies lässt sich folgendermassen erklären: Jede Ein- und Auslagerung umfasst eine Transportfahrt auf dem Quergangswagen. Da zwei Aufzüge vorhanden sind, verteilen sich die Lagerbewegungen auf beide Aufzüge, was (in diesem Beispiel) zu einer Belastung führt, welche etwa der Hälfte des Aufwandes des Quergangwagens entspricht. Die Korridorprozessoren sind im Vergleich zu den Quergang- und Aufzugsprozessoren wenig belastet, da sie nur an den Ein- und Auslagerungen im betreffenden Korridor beteiligt sind. Allgemein gilt für automatisierte Kompaktlager, dass die Korridorwagen im Vergleich zu den andern Prozessoren nur wenig ausgelastet sind und selten Engpass-Ressourcen darstellen.

Als zweites fällt auf, dass die Prozessorbelastung durch Transferaktionen sehr hoch ist. Vergleich zu der Transport- und Leerfahrten-Belastung. Daraus wird ersichtlich, dass eine Optimierung auf der Ebene der Lagersteuerung nur bis zu einem gewissen Grad die Lagerleistung erhöhen kann. Offensichtlich ist die Leistung in wesentlichem Mass abhängig von der Anzahl Transferaktionen und der Dauer dieser Aktionen. Der Arbeitsaufwand für die Transferaktionen ist abhängig von der technologischen Ausführung der Transportprozessoren. Durch technologische Entwicklungen, welche eine Beschleunigung der Transferaktionen ermöglichen würden, könnte eine wesentliche Leistungssteigerung erzielt werden.

Tatsächlich stellen die Palettenübergaben und -übernahmen einen Schwachpunkt in der Technologie der automatisierten Kompaktlager dar, was von Fachleuten aus der Praxis bestätigt wird. Vom einem technischen Standpunkt aus könnten durchaus Transfervorrichtungen konstruiert werden, welche effizienter sind als die bestehenden Technologien. Das Hauptproblem liegt in der Kostenfrage. Ein automatisiertes Kompaktlager umfasst eine grosse Anzahl (typischerweise mehrere Hunderte) von Korridoren, von denen jeder mit einem eigenen Kanalfahrzeug ausgestattet ist. Aufgrund dieser grossen Zahl muss die technische Konstruktion der Korridorwagen möglichst einfach und kostengünstig ausgelegt sein, damit die Lagersysteme in Bezug auf die Erstellungskosten konkurrenzfähig bleiben. Neuere Entwicklungen in Bereich der Kompaktlager zielen unter anderem darauf hin, die Transferzeiten zu vermindern, wobei zwei Tendenzen erkennbar sind: (1) Einsatz von passiven Stetigförderern wie Rollenbahnen in den Lagerkanälen, welche kostengünstig sind und relative niedrige Transferzeiten haben (2) Verzicht auf integrierte Fördereinrichtungen in den Kanälen und Einsatz von technologisch hoch entwickelten, versetzbaren Kanal-Satelliten.

Schliesslich stellt man als bei der Betrachtung der obigen Abbildung fest, dass der Arbeitsaufwand für Leerfahrten in einem ähnlichen Bereich liegt wie der Aufwand für Transportfahrten. Daraus folgt, dass die Berücksichtigung der Leerfahrten eine wichtige Modellkomponente darstellt, welche nicht vernachlässigt werden darf.

3.6.2. Beispiel 2

In diesem Beispiel wird gezeigt, wie das Modell für die Leistungsanalyse beim Entwurf von automatisierten Kompaktlagern zur Evaluation verschiedener Konfigurationen verwendet werden kann. Einerseits können Leistungsgrenzen berechnet, deren Vergleich Hinweise darauf gibt, mit welchen Konfigurationen die beste Leistung erzielt werden kann. Andererseits geben die Belastungsdiagramme Auskunft darüber, welche Prozessoren Engpass-Ressourcen darstellen, woraus Rückschlüsse auf die Dimensionierung der Lager gemacht werden können.

Als Vorgabe für den Entwurf sei festgelegt, dass das Lager 3600 Lagerplätze umfassen muss. Ein Lager dieser Grösse lässt sich in ganz unterschiedlichen Dimensionen realisieren. Für die Evaluation werden vier verschiedene Konfigurationen vorgeschlagen, welche sich in der Anzahl der Sektoren, der Lagerebenen, der Korridore pro Sektor und der Lagerplätze pro Korridor unterscheiden.

Es werden wiederum Standardkonfigurationen betrachtet, wie sie zu Beginn dieses Abschnitts beschrieben wurden. Für die anfänglichen Lagerbestände werden keine Vorbedingungen gefordert. Die Zeitperiode beträgt $T = 8 \text{ Std.}$. Es werden zwei Artikeltypen $i \in I = \{1, 2\}$ unterschieden mit den Verweilzeiten $\tau_1 = 0.5T$ und $\tau_2 = 1.5T$. Die Ein- und Auslagerungsbilanzen sind ausgeglichen, d.h. $\alpha_1 = \alpha_2 = 1$. Die Anteile der beiden Artikeltypen betragen je 50% d.h. $\beta_1 = \beta_2 = 0.5$.

Die untenstehende Tabelle zeigt die Dimensionen der verschiedenen Lagerkonfigurationen sowie die berechneten Leistungsschranken. Die Leistungsangaben entsprechen der gesamten Ein- und Auslagerungsleistung.

Lagerdimensionen:	Sektoren	Ebenen	Korridore pro Sektor	Lagerplätze pro Korridor	Leistung
Konfiguration 1	3	3	10	40	2533
Konfiguration 2	3	6	5	40	2442
Konfiguration 3	4	3	10	30	3321
Konfiguration 4	3	2	30	20	2409

Abbildung 3.5. Uebersicht über die Lagerkonfigurationen

In der anschliessenden Abbildung werden die Konfigurationen graphisch veranschaulicht.

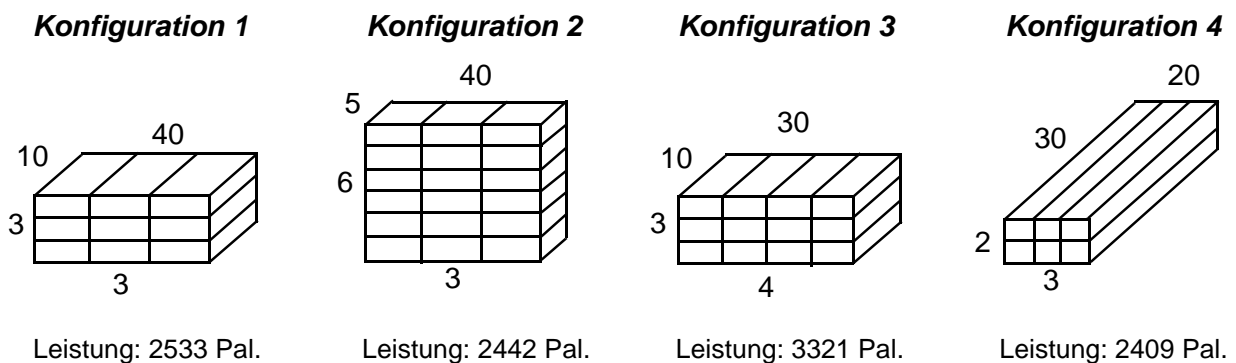


Abbildung 3.6. Schematische Darstellung der Lagerkonfigurationen

Die Evaluation der verschiedenen Konfigurationen zeigt, dass Konfiguration 3 die höchste Leistungsschranke von 3321 Paletten in 8 Stunden aufweist. Die übrigen drei Konfigurationen haben eine deutlich tiefere Leistungsschranke im Bereich von 2400 – 2500 Paletten.

Diese Leistungsschranken können als Hinweis darauf interpretiert werden, welche der betrachteten Lagerkonfiguration die grösste Leistung erbringt. Es kann mit grosser Wahrscheinlichkeit angenommen werden, dass die effektiv realisierbare Leistung eines Lagersystems annähernd proportional ist zu den mit diesem Modell berechneten Leistungsschranken. Eine absolute Gewissheit für diese Aussage besteht jedoch nicht, da die effektiv realisierbare Maximalleistung von der konkreten Steuerung des Lagers abhängt.

Der Grund, weshalb eine Konfiguration leistungsstärker ist als eine andere, liegt in der unterschiedlichen Anzahl der Prozessoren und in den variierenden Dimensionen der Ebenen, Sektoren,

Quergänge und Korridore. Ein genaueres Verständnis dieser Zusammenhänge ergibt sich bei der Betrachtung der Belastungsprofile der Prozessoren.

Die folgende Abbildung zeigt die Belastung der Aufzugs- und Quergangprozessoren für die verschiedenen Konfigurationen. Die Zahlen bilden eine untere Schranke für die Belastung, wenn im Zeitraum $T = 8 \text{ Std.}$ die maximale Ein- und Auslagerungsleistung erbracht wird. Die Aufzüge sind so bezeichnet, dass beispielsweise $A1a$ und $A1b$ die Aufzüge bezeichnen, welche auf den beiden Seiten des ersten Sektors liegen. Für die Quergangwagen gilt, dass beispielsweise $B23$ den Quergang auf der zweiten Ebene im dritten Sektor bezeichnet. Die Korridorwagen sind weggelassen, da sie nur minimal belastet werden und keine Engpass-Ressourcen darstellen. Die Belastungen sind wiederum aufgeteilt in Transportfahrten, Leerfahrten und Transferaktionen.

In Konfiguration 1 sind sowohl die Aufzüge als auch die Quergangwagen praktisch vollständig ausgelastet. Eine Ausnahme bilden die Quergangwagen auf der dritten Ebene, welche kaum benützt werden. Daraus lässt sich schliessen, dass die Aufzüge die eigentlichen Engpass-Ressourcen darstellen. Andernfalls könnte durch die Benützung der dritten Lagerebene die Leistung erhöht werden. Allgemein fällt bei allen Konfigurationen wieder auf, dass die Transferbelastungen sehr hoch sind im Vergleich zu den Transport- und Leerfahrtenbelastungen.

In Konfiguration 2 sind die Aufzüge eindeutig die Engpass-Prozessoren. Die Quergänge werden knapp halb so stark belastet wie die Aufzüge. Die obersten beiden Lagerebenen werden wie in der vorangehenden Konfiguration kaum benützt, da die Fahrten zu den oberen Ebenen zu aufwendig sind für die Aufzüge. Dies ist ein Hinweis darauf, dass das Lager zu hoch gebaut ist, was zu einer Ueberbelastung der Aufzüge führt.

Konfiguration 3 ist die leistungsfähigste Variante. Die Aufzüge bilden wiederum die Engpass-Ressourcen und die Quergangwagen sind zu etwa zwei Dritteln ausgelastet. Auffällig ist, dass im Gegensatz zu Konfiguration 1 und 2 auch die oberste Lagerebene benützt wird. Dies deutet darauf hin, dass Lagerdimensionen besser aufeinander abgestimmt sind als in den vorangehenden Varianten. Ein wesentlicher Grund für die grössere Leistungsfähigkeit von Konfiguration 3 ist die Tatsache, dass gegenüber Variante 1 und 2 ein weiterer Sektor hinzugefügt wurde. Durch diesen Sektor wurde die Zahl der Aufzüge von sechs auf acht erhöht. Da die Aufzüge in den vorangehenden Konfigurationen die Engpass-Ressourcen darstellten, ist es naheliegend, dass eine Erhöhung der Anzahl Aufzüge zu einer Leistungssteigerung führt.

Die letzte Konfiguration 4 stellt ein Lager mit sehr langen Quergängen dar. Durch die langen Transportfahrten werden die Quergangwagen so stark belastet, dass sie nun Engpass-Ressourcen darstellen. Die Aufzüge sind jedoch auch fast maximal ausgelastet.

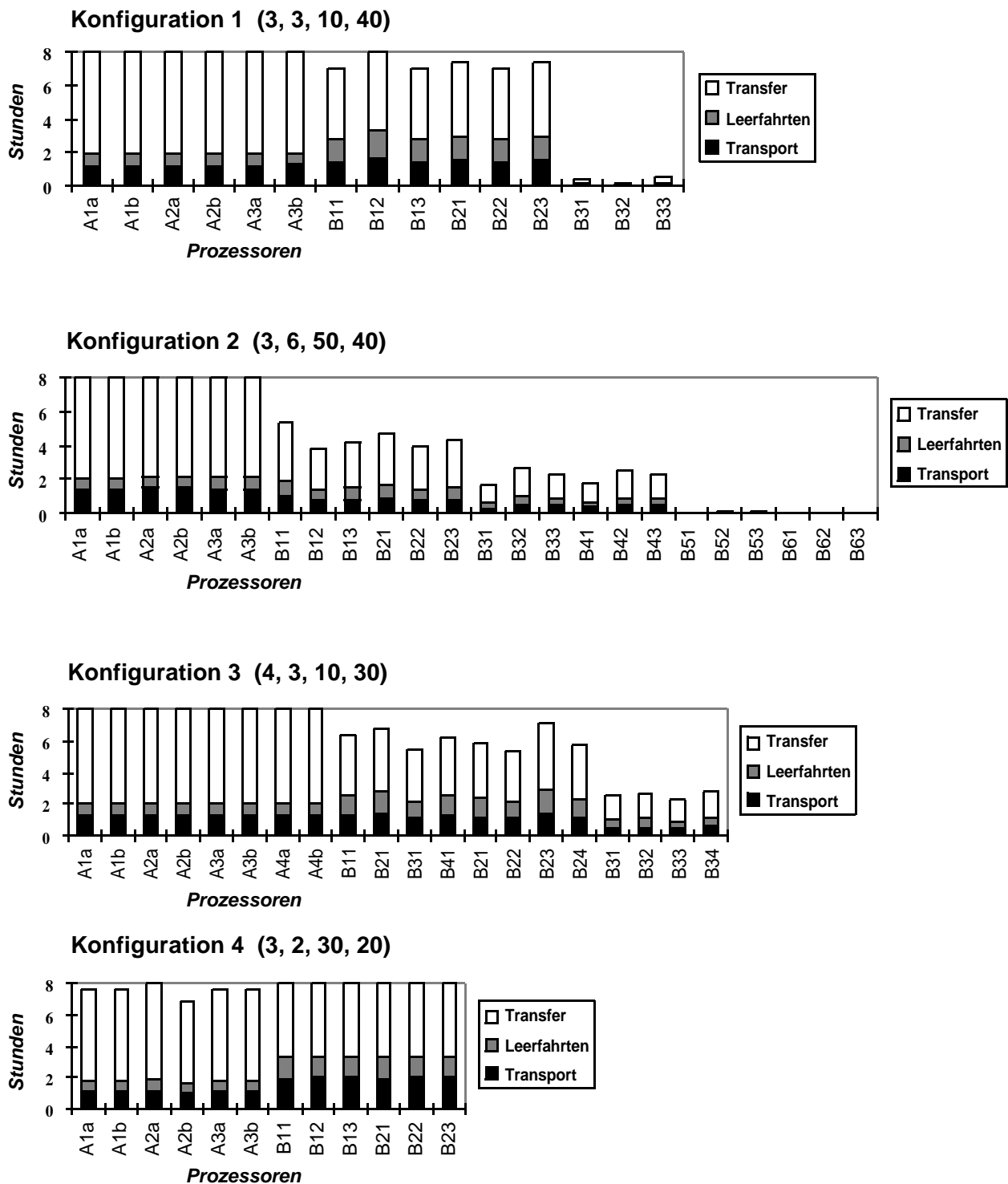


Abbildung 3.7. Prozessor-Belastungen für die verschiedenen Konfigurationen

3.6.3. Zusammenfassung

Anhand von zwei numerischen Beispielen wurde das Modell für die Leistungsanalyse automatisierter Kompaktlager illustriert. Dabei wurden die Funktionsweise und die Anwendungsmöglichkeiten des Modells unter verschiedenen Aspekten erläutert. Das Modell bietet Unterstützung in zwei Bereichen: Beim Entwurf und bei der Dimensionierung eines Lagers sowie bei der Entwicklung einer Lagersteuerung. Die wichtigsten Punkte der Entscheidungsunterstützung werden nachstehend zusammengefasst:

Entwurf

- Berechnung einer oberen Leistungsschranke
- Evaluation und Vergleich verschiedener Konfigurationen und Bestimmung der leistungsfähigsten Variante
- Berechnung von Belastungsprofilen (untere Schranke)
- Erkennen von Engpass-Prozessoren
- Rückschlüsse auf Dimensionierung des Lagers

Steuerung

- Unterstützung beim Entwurf von Lagerstrategien
- Hinweise für die Routenwahl
- Informationen über günstige Ausgangs-Konfigurationen (bei Reorganisation)

Diese Auflistung zeigt, dass das Modell substantielle Beiträge zur Entscheidungsunterstützung leisten kann, sowohl bei der Konzeption als auch bei der Steuerung automatisierter Kompaktlager.

KAPITEL 4

Modell zur Steuerungsoptimierung

Die Leistung eines automatisierten Kompaktlagers hängt in wesentlichem Masse von der Effizienz der eingesetzten Lagersteuerung ab. Durch die Optimierung der Steuerungsprozesse kann eine Erhöhung der Lagerleistung erzielt werden. Die Steuerung eines Lagersystems bildet eine komplexe Aufgabe, welche eine Vielzahl von Entscheidungsdimensionen umfasst. Gemäss Abschnitt 1.4. lassen sich auf der Stufe der dispositiven Steuerung drei Hauptfunktionen unterscheiden: Bestimmung der Transportaufträge aufgrund der Ein- und Auslagerungsaufträge (Lagerstrategie, Routenwahl), Bestimmung der Umlagerungen (erzwungene Umlagerungen, Reorganisation) und Bestimmung des Ablaufplans (Reihenfolge und Terminierung der Transportaktionen). Jede dieser Komponenten bietet ein Optimierungspotential.

In dieser Arbeit wird die Optimierung der Steuerungsprozesse auf der Ebene der Ablaufpläne betrachtet. Es wird ein Optimierungsmodell entwickelt, welches ausgehend von einer Menge von Transportaufträgen einen optimierten Ablaufplan generiert. Das Modell basiert auf dem Konzept der disjunktiven Präzedenzbedingungen. Es wird gezeigt, dass sich die Menge aller zulässigen Ablaufpläne vollständig beschreiben lässt mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen. In Kapitel 5 wird anschliessend erläutert, wie sich dieses Modell als Optimierungsproblem in verallgemeinerten disjunktiven Graphen darstellen lässt. In Kapitel 6 wird ein heuristisches Optimierungsverfahren für das disjunktive Graphenproblem entwickelt.

Das vorliegenden Kapitel erläutert in Abschnitt 4.1. zunächst die Problemstellung. In Abschnitt 4.2. wird der Begriff der disjunktiven Präzedenzbedingung erklärt und gezeigt, wie sich Reihenfolgeprobleme mit Hilfe von disjunktiven Präzedenzbedingungen modellieren lassen. Dabei wird

auch auf das bekannte Job-Shop-Problem eingegangen. In Abschnitt 4.3. wird gezeigt, wie sich die zulässigen Ablaufpläne in einem automatisierten Kompaktlager mit konjunktiven und disjunktiven Präzedenzbedingungen beschreiben lassen. Daraus wird ein Modell zur Optimierung der Steuerung entwickelt. Abschnitt 4.4. beschäftigt sich mit der Modellierung der räumlichen Nebenbedingungen, welche sich daraus ergeben, dass Palettentransporte in den Korridoren nur möglich sind, wenn die betreffenden Lagerplätze nicht mit Paletten belegt sind.

4.1. Problembeschreibung

Als Problemstellung wird die Optimierung der Steuerungsprozesse auf der Ebene der Ablaufpläne betrachtet. Gemäss Abschnitt 1.4. bildet die Bestimmung der Ablaufpläne (scheduling) eine der funktionalen Hauptkomponenten der dispositiven Steuerung. Die Aufgabe dieser Komponente besteht darin, ausgehend von einer Menge von Transportaufträgen einen ausführbaren Ablaufplan zu generieren. Es wird ein Optimierungsmodell entwickelt, welches auf der Basis der Transportaufträge einen optimierten Ablaufplan berechnet, welcher die gesamte Ausführungsdauer des Lagerprozesses (makespan) minimiert. Dieses Optimierungsproblem wird im folgenden als Lager-Problem bezeichnet.

Input

Die Eingabedaten des Modells bestehen aus einer Menge von Transportaufträgen, welche von der Routing-Komponente der dispositiven Steuerung generiert werden (vgl. Abbildung 1.7.). Jeder Transportauftrag entspricht einer elementaren Lagerbewegung, d.h. der Ein-, Aus- oder Umlagerungen einer einzelnen Palette. Gemäss Abschnitt 2.2. spezifiziert jeder Transportauftrag die Folge der Aktionen, welche von den beteiligten Prozessoren ausgeführt werden müssen. Die Gesamtheit der Transportaufträge definiert für jeden Prozessor die Menge der auszuführenden Operationen. Gemäss Abschnitt 2.3. lassen sich die Transportaufträge als Wege im System-Graphen darstellen.

In Abschnitt 2.2. wurde gezeigt, dass die Transportaufträge in der Terminologie der Maschinen-Ablaufplanung als Jobs aufgefasst werden können, wobei jeder Job einer elementaren Lagerbewegung entspricht. Ein Job besteht aus einer Folge von Operationen, welche auf einem vorgegebenen Prozessor ausgeführt werden. Eine Operation umfasst alle Aktionen, welche für die Bearbeitung einer Palette auf einem einzelnen Prozessor notwendig sind. Alle Operationen haben dieselbe Struktur und bestehen aus einer Uebernahmeaktion, einer anschliessenden Transportaktion, einer eventuellen Wartephase und einer Uebergabeaktion. Hierbei werden die Lade- und Entladeaktionen als spezielle Uebernahme- bzw. Uebergabeaktionen aufgefasst.

Zusammenfassend entspricht der Input des Modells einer Menge von Jobs, welche die auszuführenden Lagerbewegungen definieren. Durch diese Jobs wird für jeden Prozessor die Menge der auszuführenden Operationen (bzw. Aktionen) festgelegt.

Output

Als Output generiert das Modell einen zulässigen, d.h. ausführbaren Ablaufplan, welcher die Gesamtdauer des Lagerprozesses (makespan) minimiert. Der Ablaufplan spezifiziert für jeden Prozessor die Reihenfolge, in welcher die Operationen ausgeführt werden. Weiter wird für jede Aktion der Startzeitpunkt der Ausführung festgelegt. Die Dauer der Aktionen ist durch die Transportaufträge und die Prozessorparameter vorgegeben. Der Ablaufplan stellt eine vollständige Beschreibung des Lagerprozesses dar und wird von der operationellen Steuerung als Vorgabe für die Steuerung der Transportprozessoren übernommen.

4.2. Modellierungsansatz: Disjunktive Präzedenzbedingungen

In diesem Abschnitt wird das Konzept der konjunktiven und disjunktiven Präzedenzbedingungen erläutert. Es wird gezeigt, wie Reihenfolgeprobleme in der Ablaufplanung mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen modelliert werden können. Als klassisches Reihenfolgeproblem wird das bekannte Job-Shop-Problem besprochen. In Kapitel 5 wird gezeigt, wie sich Modelle mit disjunktiven Präzedenzbedingungen als Optimierungsprobleme in disjunktiven Graphen darstellen lassen.

4.2.1. Konjunktive und disjunktive Präzedenzbedingungen

Unter einer Präzedenzbedingung wird hier allgemein eine Ungleichung der Form $x_j - x_i \geq c_{ij}$ verstanden. Präzedenzbedingungen spielen eine wichtige Rolle bei der Modellierung von Projektplanungsproblemen (project scheduling, resource-constraint project scheduling) und Maschinenbelegungsproblemen (machine scheduling). In diesem Zusammenhang stellen i und j Aktivitäten dar und x_i bzw. x_j die Startzeitpunkte dieser Aktivitäten. Eine Präzedenzbedingung der obigen Form besagt, dass der Startzeitpunkt von Aktivität j frühestens c_{ij} Zeiteinheiten nach dem Startzeitpunkt von Aktivität i erfolgen kann.

Projektplanung

Projektplanungsprobleme befassen sich (im einfachsten Fall) mit der Aufgabe, eine Menge von Aktivitäten I mit gegebener Ausführungsdauer d_i , $i \in I$ zeitlich so einzuplanen, dass die

gesamte Ausführungsdauer (makespan) minimal ist. Die Restriktionen werden gebildet durch die Forderung, dass bestimmte Aktivitäten erst ausgeführt werden können, wenn gewisse andere Aktivitäten abgeschlossen sind. Gegeben ist also eine partielle Ordnung $A \subseteq I \times I$ mit der Bedeutung, dass für alle $(i, j) \in A$ gelten muss, dass Aktivität i vor Aktivität j ausgeführt wird. Die Bedingung, dass eine Aktivität i vor einer Aktivität j ausgeführt werden muss, bedeutet, dass der Startzeitpunkt von j nach dem Beendigungszeitpunkt von i stattfinden muss. Somit lässt sich ein Projektplanungsproblem mit Hilfe von Präzedenzbedingungen wie folgt formulieren:

$$\begin{aligned} \min x_\tau \\ x_j - x_i &\geq d_i, & (i, j) \in A \\ x_\tau - x_i &\geq d_i, & i \in I \\ x_i &\geq 0, & i \in I \cup \tau \end{aligned}$$

Hierbei stellt $\tau \in I$ eine fiktive Abschlussaktivität dar, welche erst nach Beendigung aller andern Aktivitäten ausgeführt wird.

Projektplanungsprobleme dieser Art können als gerichtete, gewichtete Graphen dargestellt werden, indem die Aktivitäten als Knoten und die Präzedenzbedingungen als Bogen dargestellt werden. Das oben beschriebene Problem hat als Graph die Form $G = \langle V, A', c \rangle$ wobei die Knotenmenge gegeben ist durch $V = I \cup \tau$ und die Bogenmenge durch $A' = A \cup A_\tau$ mit $A_\tau = \{(i, \tau) : i \in I\}$. Die Bogengewichte betragen $c_{ij} = d_i$ für $(i, j) \in A'$. Es ist einfach zu zeigen (und hinlänglich bekannt), dass das Optimum des obigen Problems der Länge eines längsten in τ endenden Weges in G entspricht (siehe Abschnitt 5.1.). Weiter gilt, dass der frühestmögliche Startzeitpunkt einer Aktivität $i \in I$ gegeben ist durch die Länge eines längsten in i endenden Weges in G . Aus diesem Umstand heraus hat sich der Ausdruck CPM (critical path method) etabliert im Zusammenhang mit Projektplanungsproblemen dieser Art.

Das Gebiet der Projektplanungsprobleme liegt nicht im Fokus dieser Arbeit und wird deshalb nicht weiter diskutiert. Es existiert eine grosse Menge von Literatur zu diesem Thema, wovon zur Einführung das klassische Buch von [Baker, K. R. (1974)] erwähnt sei. Ziel dieses Absatzes war es, den Begriff der Präzedenzbedingung zu erläutern und zu zeigen, wie Präzedenzbedingungen als gewichtete Bogen in einem Graphen dargestellt werden können.

Projektplanung mit beschränkten Ressourcen

Ein klassisches Problem im Bereich der Projektplanung mit beschränkten Ressourcen (resource-constraint project scheduling) ist das folgende: Gegeben ist ein Projektplanungsproblem gemäss vorangehendem Absatz. Zusätzlich sind nun Ressourcen (Maschinen, Material, Arbeitskraft, etc.) mit beschränkten Kapazitäten zu berücksichtigen, welche von gewissen Aktivitäten zur Ausführung benötigt werden. Hierbei wird der Spezialfall betrachtet, wo die Ressourcen Einheitskapazität haben, d.h. eine Ressource kann zu jedem Zeitpunkt von höchster einer Aktivität benützt

werden. Weiter gilt, dass die Benützung einer Ressource durch eine Aktivität nicht unterbrochen werden kann durch eine andere Aktivität (no preemption).

Sei M die Menge der Ressourcen und $I_m \subseteq I$ die Menge der Aktivitäten, welche Ressource $m \in M$ benützen. Da die Ressourcen Einheitskapazität haben, kann eine Ressource nicht gleichzeitig von zwei Aktivitäten benützt werden. Folglich gilt für alle Paare von Operationen $\{i, j\} \subseteq I_m$ dass entweder Operation i vor Operation j oder j vor i ausgeführt werden muss. Somit können die aus der Ressourcenbenützung resultierenden Restriktionen wie folgt mit Hilfe von disjunktiven Präzedenzbedingungen ausgedrückt werden:

$$x_j - x_i \geq d_i \quad \text{OR} \quad x_i - x_j \geq d_j, \quad \{i, j\} \subseteq I_m, \quad m \in M$$

Die disjunktive Bedingung $x_j - x_i \geq d_i \quad \text{OR} \quad x_i - x_j \geq d_j$ ist hierbei so zu verstehen, dass mindestens eine der beiden Präzedenzbedingungen zutreffen muss.

Zusammen mit den Restriktionen aus dem vorangehenden Absatz lässt sich das beschriebene Projektplanungsproblem mit eingeschränkten Ressourcen somit anhand von konjunktiven und disjunktiven Präzedenzbedingungen vollständig beschreiben. Hierbei wird eine Präzedenzbedingung als konjunktiv bezeichnet, wenn sie auf jeden Fall eingehalten werden muss. Ein Paar von Präzedenzbedingungen wird als disjunktiv bezeichnet, wenn mindestens eine der beiden Bedingungen eingehalten werden muss.

Analog zum vorangehenden Problem lässt sich das Projektplanungsproblem mit beschränkten Ressourcen in Form eines sogenannten disjunktiven Graphen darstellen: Die Aktivitäten werden als Knoten und die Präzedenzbedingungen als gewichtete Bogen dargestellt. Eine konjunktive Präzedenzbedingung entspricht einem konjunktiven Bogen und eine disjunktive Präzedenzbedingung entspricht einem Paar von disjunktiven Bogen, von welchen mindestens ein Bogen ausgewählt werden muss. In Abschnitt 5.1. wird das Konzept der disjunktiven Graphen ausführlich diskutiert, so dass an dieser Stelle nicht weiter darauf eingegangen wird.

Ziel dieses Absatzes war es, den Begriff der disjunktiven Präzedenzbedingung zu erläutern und zu zeigen, wie disjunktive Präzedenzbedingungen im Zusammenhang mit Projektplanungsproblemen mit beschränkten Ressourcen entstehen. Es wurde gezeigt, dass disjunktive Präzedenzbedingungen eine Möglichkeit bilden, die Menge der zulässigen Startzeitpunkte zu beschreiben, wenn mehrere Aktivitäten eine Ressource mit Einheitskapazität untereinander teilen.

Probleme, bei welchen Aktivitäten eine gemeinsame Ressource mit Einheitskapazität teilen, werden auch als Reihenfolgeprobleme (sequencing problems) bezeichnet. Dies ist daher begründet, dass jeder zulässige Ablaufplan (schedule), d.h. jede Wahl von Startzeitpunkten, welche die angegebenen konjunktiven und disjunktiven Restriktionen erfüllt, eine eindeutige Bearbeitungsreihenfolge der Aktivitäten auf jeder Ressource definiert. Umgekehrt wird durch eine zulässige Vorgabe der Bearbeitungsreihenfolgen auf den Ressourcen ein eindeutiger Ablaufplan definiert, welcher optimal ist bezüglich dieser Reihenfolgen. Es handelt sich dabei um den sogenannten

semi-aktiven Ablaufplan bezüglich dieser Reihenfolgen, welcher so definiert ist, dass jede Operation zum frühestmöglichen Zeitpunkt ausgeführt wird, unter Einhaltung der gegebenen Reihenfolgen. Die Menge der zu betrachtenden Ablaufpläne kann demnach reduziert werden auf die Menge der semi-aktiven Ablaufpläne. Somit lässt sich ein Reihenfolgeproblem lösen, indem für jede Ressource eine Bearbeitungsreihenfolge bestimmt wird, so dass der zugehörige semi-aktive Ablaufplan optimal ist. Eine detaillierte Darstellung dieser fundamentalen Zusammenhänge findet sich beispielsweise in den Standardwerken von [Baker, K. R. (1974)] und [Conway, R. W., Maxwell, W. L., Miller L. W. (1967)]. Ziel dieser Ausführungen war, den Begriff der Reihenfolgeprobleme zu erläutern und darauf hinzuweisen, dass Reihenfolgeprobleme mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen modelliert werden können, wobei die disjunktiven Präzedenzbedingungen aus der Modellierung der möglichen Reihenfolgen (Permutationen) auf den Einheitsressourcen resultieren.

Abschliessend sei darauf hingewiesen, dass diverse klassische Maschinenbelegungsprobleme als Spezialfall des beschriebenen Projektplanungsproblems mit Einheitsressourcen aufgefasst werden können. Insbesondere gilt dies für das bekannte Job-Shop-Problem, welches im folgenden Abschnitt diskutiert wird. Im Falle von Maschinenbelegungsproblemen entsprechen die Ressourcen den Maschinen und die Aktivitäten den auszuführenden Operationen.

4.2.2. Job-Shop-Problem

Das Job-Shop-Problem entspricht folgendem Maschinenbelegungsproblem: Gegeben ist eine Menge von Jobs, welche auf einer Menge von Maschinen ausgeführt werden. Jeder Job besteht aus einer Menge von Operationen, welche in einer vorgegebenen Reihenfolge bearbeitet werden müssen. Für jede Operation ist die Maschine, auf welcher die Operation ausgeführt wird, sowie die Bearbeitungsdauer auf dieser Maschine gegeben. Weiter gelten die folgenden Einschränkungen: (1) Einheitskapazität (unit capacity): Eine Maschine kann höchstens eine Operation zu jedem Zeitpunkt bearbeiten. (2) Kein Unterbruch (no preemption): Die Ausführung einer Operation auf einer Maschine kann nicht durch eine andere Operation unterbrochen werden. Gesucht ist eine Einplanung der Operationen, so dass die gesamte Ausführungsdauer (makespan) minimal ist.

Beispiel:

Es wird ein Job-Shop-Problem mit 4 Prozessoren $M = \{P_1, P_2, P_3, P_4\}$ und 4 Jobs $\mathcal{J} = \{J_1, J_2, J_3, J_4\}$ gezeigt. Die Operationen der verschiedenen Jobs werden der Reihe nach auf folgenden Prozessoren ausgeführt:

Job:	Operation 1:	Operation 2:	Operation 3:	Operation 4:
J_1	P_1	P_2	P_3	P_4
J_2	P_1	P_2	P_4	
J_3	P_3	P_4		
J_4	P_1	P_2		

Abbildung 4.1. Prozessorreihenfolge der verschiedenen Jobs

Nachfolgend werden die Gantt-Diagramme von zwei verschiedenen zulässigen Ablaufplänen für dieses Job-Shop-Problem gezeigt. Hierbei wird die i -te Operation von Job J_n in den Abbildungen mit $n.i$ bezeichnet. Beachte, dass die aufgeführten Pläne den semi-aktiven Ablaufplänen bezüglich der angegebenen Ausführungsreihenfolgen entsprechen, d.h. die Operationen werden jeweils zum frühestmöglichen Zeitpunkt gestartet. ♦

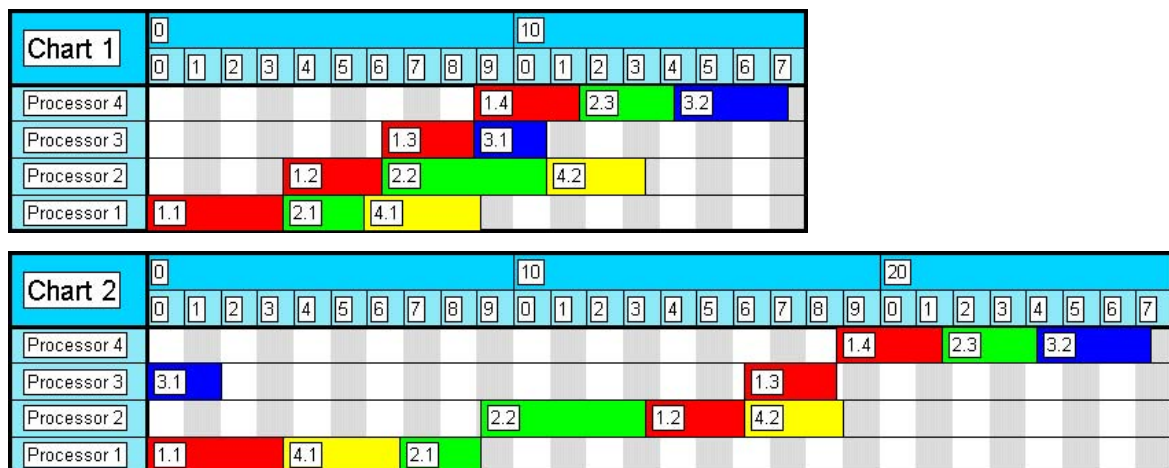


Abbildung 4.2. Gantt-Diagramme zum Job-Shop-Problem

Das Job-Shop-Problem ist eines der bekanntesten Probleme im Bereich der Maschinen-Ablaufplanung. Seit seiner Formulierung in der Mitte des vergangenen Jahrhunderts bildet es eine der grossen Herausforderungen auf dem Gebiet der kombinatorischen Optimierung. Das Job-Shop-Problem bzw. dessen Entscheidungsversion ist NP-hart (siehe beispielsweise [Lenstra, J. K., Rinnooy Kan, A. H. G., Brucker, P. (1977)]) und wird zu den schwierigsten Problemen in der kombinatorischen Optimierung gezählt. Als Hinweis auf seine Schwierigkeit sei die Tatsache erwähnt, dass ein relativ kleines Problem mit 10 Jobs und 10 Maschinen, welches in [Fischer, J. F., Thompson, G. L. (1963)] formuliert wurde, während über 20 Jahren ungelöst blieb. Mittlerweile existiert eine Vielzahl von Modellen und Methoden zur Lösung des Job-Shop-Problems. Ein grosser Teil

der bestehenden Ansätze sind enumerative Verfahren (Branch & Bound), welche häufig direkt oder indirekt auf einer Problemdarstellung mittels disjunktiver Graphen basieren (vgl. [Pinson, E. (1995)]).

Die Modellierung des Job-Shop-Problems mit Hilfe von disjunktiven Graphen wurde erstmals vorgeschlagen von [Roy, B., Sussman, B. (1964)]. Wie schon erwähnt, steht das Konzept der disjunktiven Graphen in engem Zusammenhang mit der Problemdarstellung mittels disjunktiver Präzedenzbedingungen. In Abschnitt 4.2. wurde gezeigt, wie sich Projektplanungsprobleme mit beschränkten Ressourcen mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen darstellen lassen. Das Job-Shop-Problem bildet einen Spezialfall der dort beschriebenen Problemklasse, bei welchem jede Operation (Aktivität) genau eine Maschine (Ressource) benützt. Die konjunktiven Präzedenzbedingungen ergeben sich aus den vorgegebenen Ausführungsreihenfolgen für die Operationen eines Jobs. Nachfolgend wird eine Formulierung des Job-Shop-Problems mittels disjunktiver Präzedenzbedingungen gegeben. Der Zusammenhang zwischen dieser Darstellung und dem entsprechenden disjunktiven Graphen wird in Abschnitt 5.1. und Abschnitt 5.3. diskutiert.

Das Job-Shop-Problem $\Pi = (M, I, \mathcal{J}, \mu, d)$

Parameter:

M	Menge der Prozessoren (Maschinen)
I	Menge der Operationen
I_m	Menge der Operationen, welche auf Prozessor m ausgeführt werden, $m \in M$
σ, τ	Fiktive Start- bzw. Abschlussoperation, welche vor bzw. nach allen andern Operationen ausgeführt wird
\mathcal{J}	Menge der Jobs. Es gilt $\mathcal{J} \subseteq \text{Pot}(I)$ und \mathcal{J} bildet eine Partition von I Die Operationen von Job $J \in \mathcal{J}$ werden bezeichnet als $J = \{J_1, \dots, J_{ J }\}$ wobei J_r $r \in \{1, \dots, J \}$, die r -te Operation bezeichnet in der Ausführungsreihenfolge von J .
μ_i	Prozessor, auf welchem Operation i ausgeführt wird, $i \in I$
d_i	Bearbeitungsdauer von Operation i , $i \in I$

Variablen:

x_i	Startzeitpunkt von Operation i , $i \in I \cup \{\sigma, \tau\}$
-------	--

Zielfunktion:

$$\min x_\tau \tag{4.1}$$

Restriktionen:*Konjunktive Präzedenzen für Jobs:*

$$x_j - x_i \geq d_i, \quad (i, j) \in \{(J_r, J_{r+1}) : J \in \mathcal{J}, r = 1, \dots, |J-1|\} \quad (4.2)$$

Konjunktive Präzedenzen für σ, τ :

$$x_i - x_\sigma \geq 0, \quad i \in I \quad (4.3)$$

$$x_\tau - x_i \geq d_i, \quad i \in I \quad (4.4)$$

Disjunktive Präzedenzen für Prozessoren:

$$x_j - x_i \geq d_i \quad \text{OR} \quad x_i - x_j \geq d_j, \quad \{i, j\} \subseteq I_m, m \in M \quad (4.5)$$

Nichtnegativitäts-Bedingungen:

$$x_i \geq 0, \quad i \in I \cup \{\sigma, \tau\} \quad (4.6)$$

4.3. Modell für das Lager-Problem

In diesem Abschnitt wird ein Optimierungsmodell für das Lager-Problem, d.h. das Problem der optimierten Steuerung auf der Ebene der Ablaufpläne, entworfen, welches auf dem Konzept der Präzedenzbedingungen basiert. Es wird gezeigt, wie sich die Menge aller zulässigen Ablaufpläne (d.h. der Lösungsraum) mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen vollständig beschreiben lässt. In Kapitel 5 wird gezeigt, dass sich dieses Modell als Optimierungsproblem in einem verallgemeinerten disjunktiven Graphen darstellen lässt. Eine Heuristik zur Lösung des disjunktiven Graphenproblems wird in Kapitel 6 entworfen.

4.3.1. Synchronisation und kontextabhängige Operationsdauer

Gegeben sei eine Menge von Jobs, d.h. eine Menge von Ein-, Aus- und Umlagerungen, für deren Ausführung ein Ablaufplan bestimmt werden soll. Ein Job besteht aus einer Folge von Operationen, wobei jede Operation die Bearbeitung der betreffenden Palette auf einem Prozessor darstellt. Gemäss Abschnitt 2.2. setzen sich die Operationen zusammen aus einer Folge von Prozessoraktionen, wobei alle Operationen dieselbe Struktur haben: (1) Uebernahme, (2) Transport, (3) beladene Wartephase, (4) Uebergabe.

Die Uebergabe einer Palette von einem Prozessor auf einen andern erfolgt synchron, d.h. beide Prozessoren müssen sich für den Transfer zur gleichen Zeit am Uebergabeort befinden. Daraus ergeben sich Synchronisationsbedingungen zwischen den Uebergabe- und Uebernahmeaktionen.

σ, τ	Fiktive Start- bzw. Abschlussoperation, welche vor bzw. nach allen andern Operationen ausgeführt wird.
\mathcal{J}	Menge der Jobs. Es gilt $\mathcal{J} \subseteq \text{Pot}(I)$ und \mathcal{J} bildet eine Partition von I . Die Operationen von Job $J \in \mathcal{J}$ werden bezeichnet als $J = \{J_1, \dots, J_{ J }\}$ wobei J_r , $r \in \{1, \dots, J \}$ die r -te Operation bezeichnet in der Ausführungsreihenfolge von J .
μ_i	Prozessor, auf welchem Operation i ausgeführt wird, $i \in I$

Zeiten:

d_i	Dauer der Transportfahrt von Operation i , $i \in I$. Die Transportfahrt führt von der Startposition von i zur Zielposition von i .
d_i^t	Dauer der Palettenübernahme ("take over") von Operation i , $i \in I$. Falls i die erste Operation eines Jobs ist, entspricht d_i^t der Aufladezeit, andernfalls der Uebergabezeit.
d_i^h	Dauer der Palettenübergabe ("hand over") von Operation i , $i \in I$. Falls i die letzte Operation eines Jobs ist, entspricht d_i^h der Abladezeit, andernfalls der Uebergabezeit. Es wird vorausgesetzt, dass die Uebergabe- und Uebernahmezeit bei einem Palettentransfer identisch sind, d.h. $d_{J_r}^h = d_{J_{r+1}}^t$ für alle $J \in \mathcal{J}$ und $r = 1, \dots, J - 1$.
d_{ij}^s	Dauer der Leerfahrt ("setup") zwischen den Operationen i und j , $\{i, j\} \subseteq I_m$, $m \in M$. Die Leerfahrt führt von der Zielposition von i zur Startposition von j .
$d_{\sigma i}^s$	Dauer der Leerfahrt ("setup") von der anfänglichen Position von Prozessor μ_i zur Startposition von Operation i , $i \in I$.
$d_{i\tau}^s$	Dauer der Leerfahrt ("setup") von der Zielposition von Operation i zur abschliessenden Position von Prozessor μ_i , $i \in I$.

Weitere Bezeichnungen:

$t(i), t_i$	Tail-Operation von Operation i , $i \in I$
$h(i), h_i$	Head-Operation von Operation i , $i \in I$

Variablen:

$x_{t(i)}$	Startzeitpunkt der Tail-Operation von Operation i , $i \in I$
$x_{h(i)}$	Startzeitpunkt der Head-Operation von Operation i , $i \in I$
x_σ, x_τ	Startzeitpunkt der (fiktiven) Startoperation σ bzw. Endoperation τ

Konjunktive Präzedenzen zur Synchronisation

Als Synchronisationsbedingung bei einem Palettentransfer muss gelten, dass die Startzeitpunkte der Uebergabe- und Uebernahmeaktion identisch sind. Seien J_r und J_{r+1} zwei aufeinanderfolgende Operationen eines Jobs $J \in \mathcal{J}$. Am Ende von Operation J_r erfolgt eine Palettenübergabe zu Operation J_{r+1} , d.h. von Prozessor $\mu(J_r)$ zu Prozessor $\mu(J_{r+1})$. Der Startzeitpunkt der Uebergabeaktion ist gegeben durch den Startzeitpunkt $x_{h(J_r)}$ der Head-Operation von J_r . Der Startzeitpunkt der Uebernahmeoperation entspricht dem Startzeitpunkt $x_{t(J_{r+1})}$ der Tail-Operation von

J_{r+1} . Es muss gelten $x_{h(J_r)} = x_{t(J_{r+1})}$. Diese Gleichung lässt sich in Form von zwei Ungleichungen ausdrücken, nämlich $x_{t(J_{r+1})} - x_{h(J_r)} \geq 0$ und $x_{h(J_r)} - x_{t(J_{r+1})} \geq 0$. Somit lassen sich die Synchronisationsbedingungen durch folgende Paare von konjunktiven Präzedenzbedingungen modellieren:

$$x_{t(j)} - x_{h(i)} \geq 0, x_{t(i)} - x_{h(j)} \geq 0, \quad (i, j) \in \{(J_r, J_{r+1}) : J \in \mathcal{J}, r = 1, \dots, |J| - 1\} \quad (4.7)$$

Konjunktive Präzedenzen für Tail- und Head-Operationen

Jede Operation $i \in I$ besteht aus einer Tail-Operation $t(i)$ und einer Head-Operation $h(i)$. Es gilt die Bedingung, dass die Head-Operation nach der Tail-Operation ausgeführt werden muss. Dies bedeutet, dass der Startzeitpunkt der Head-Operation frühestens nach der Beendigung der Tail-Operation stattfinden kann. Die Dauer der Tail-Operation $t(i)$ ist gegeben durch die Dauer der Uebernahmeaktion und der Transportfahrt, d.h. $d_i^t + d_i$. Daraus ergeben sich die folgenden konjunktiven Präzedenzbedingungen:

$$x_{h(i)} - x_{t(i)} \geq d_i^t + d_i, \quad i \in I \quad (4.8)$$

Konjunktive Präzedenzen für σ und τ

Analog zum Job-Shop-Modell wird eine fiktive Startoperation σ und eine fiktive Abschlussoperation τ eingeführt. Die Startoperation muss vor allen andern Operationen ausgeführt werden, d.h. für jede Operation $i \in I$ muss der Startzeitpunkt $x_{t(i)}$ der Tail-Operation von i nach dem Startzeitpunkt x_σ von σ erfolgen. Zu Beginn des Lagerprozesses befindet sich ein Prozessor an einer bestimmten Anfangsposition. Bevor die erste Operation ausgeführt werden kann, muss sich der Prozessor an die Stelle bewegen, wo die erste Palette übernommen wird. Die Fahrzeit eines Prozessors von der Anfangsposition zur Uebernahmeposition einer Operation $i \in I$ wird als anfängliche Setup-Zeit bezeichnet und ist gegeben durch $d_{\sigma i}^s$. Somit kann der Startzeitpunkt $x_{t(i)}$ der Tail-Operation von i frühestens $d_{\sigma i}^s$ Zeiteinheiten nach Beginn des Lagerprozesses, d.h. nach dem Startzeitpunkt x_σ erfolgen. Daraus ergeben sich die folgenden konjunktiven Präzedenzbedingungen:

$$x_{t(i)} - x_\sigma \geq d_{\sigma i}^s, \quad i \in I \quad (4.9)$$

Die Abschlussoperation τ muss nach der Beendigung aller andern Operationen ausgeführt werden. Falls für die Prozessoren eine Zielposition vorgegeben ist, an welcher sie sich am Ende des Lagerprozesses befinden sollen, müssen sie nach Abschluss der letzten Operation von der Uebergabeposition zur Zielposition fahren. Die Dauer dieser Leerfahrt wird als abschliessende Setup-Zeit bezeichnet und ist gegeben durch $d_{i\tau}^s$, falls Operation $i \in I$ die letzte Operation eines Prozessors darstellt. Der Startzeitpunkt x_τ von τ kann somit frühestens $d_i^h + d_{i\tau}^s$ Zeiteinheiten nach dem Startzeitpunkt $x_{h(i)}$ der Head-Operation von $i \in I$ erfolgen. Daraus ergeben sich die folgenden konjunktiven Präzedenzbedingungen:

$$x_{\tau} - x_{h(i)} \geq d_i^h + d_{i\tau}^s, \quad i \in I \quad (4.10)$$

Disjunktive Präzedenzen für Operationen-Reihenfolgen

Für jeden Prozessor gilt, dass höchstens eine Operation gleichzeitig ausgeführt werden kann, d.h. die Operationen müssen nacheinander in einer bestimmten Reihenfolge abgearbeitet werden. In Bezug auf die Teiloperationen gilt, dass Tail- und Head-Operation einer Operation unmittelbar nacheinander ausgeführt werden müssen, ohne dass eine andere Operation dazwischenliegt. Für jedes Paar von Operationen $\{i, j\} \in I_m$ welche auf einem Prozessor $m \in M$ ausgeführt werden, muss somit gelten, dass entweder die Head-Operation von i vor der Tail-Operation von j ausgeführt wird oder umgekehrt. Falls i vor j ausgeführt wird, muss sich der Prozessor von der Uebergabeposition von i zur Uebernahmeposition von j bewegen. Dies entspricht einer Leerfahrt des Prozessors. Die Dauer dieser Leerfahrt wird als Setup-Zeit zwischen i und j bezeichnet und ist gegeben durch d_{ij}^s . Der Startzeitpunkt $x_{t(j)}$ der Tail-Operation von j kann somit frühestens $d_i^h + d_{ij}^s$ Zeiteinheiten nach dem Startzeitpunkt $x_{h(i)}$ der Head-Operation von i erfolgen. Hierbei bezeichnet d_i^h die Dauer der Head-Operation, d.h. der Uebergabeaktion, von Operation i . Somit können die Operationen-Reihenfolgen auf den Prozessoren durch die folgenden disjunktiven Restriktionen modelliert werden:

$$x_{t(j)} - x_{h(i)} \geq d_i^h + d_{ij}^s \quad \text{OR} \quad x_{t(i)} - x_{h(j)} \geq d_j^h + d_{ji}^s, \quad \{i, j\} \subseteq I_m, m \in M \quad (4.11)$$

Es sei darauf hingewiesen, dass im Vergleich zum Job-Shop-Modell (vgl. Abschnitt 4.2.) die disjunktiven Präzedenzbedingungen (4.11) auf unterschiedliche Paare von Teil-Operationen Bezug nehmen. Die eine Präzedenzbedingung bezieht sich auf die Teiloperationen $h(i), t(j)$ und die andere auf $h(j), t(i)$. In Kapitel 5 wird gezeigt, dass dieser Umstand dazu führt, dass das Konzept der disjunktiven Graphen verallgemeinert werden muss.

Abschliessend sei bemerkt, dass die Modellierung der Operationen-Reihenfolgen mit disjunktiven Präzedenzbedingungen bei sequenzabhängigen Setup-Zeiten nur möglich ist, wenn die Setup-Zeiten die Dreiecksungleichung erfüllen. Darunter wird die Bedingung verstanden, dass für alle Operationen $i, j, k \in I$ gelten muss $d_{ij}^s + d_{jk}^s \geq d_{ik}^s$. Andernfalls ist es möglich, dass eine Präzedenzbedingung $x_{t(j)} - x_{h(i)} \geq d_i^h + d_{ij}^s$ aus (4.11) zwischen zwei Operationen $i, j \in I$ aktiv ist, obwohl diese Operationen nicht unmittelbar hintereinander ausgeführt werden. Im Falle der automatisierten Kompaktlager gilt die Dreiecksungleichung, da die Dauer einer direkten Leerfahrt von der Endposition von i zur Startposition von k kürzer ist als die Summe der Dauer der Leerfahrten von i nach j und von j nach k .

Optimierungsmodell

Ein Ablaufplan für die Jobmenge \mathcal{J} wird definiert durch die Variablen $(x_{t(i)}, x_{h(i)} \in \mathbb{R} : i \in I \cup \{\sigma, \tau\})$. Diese Variablen legen den Startzeitpunkt für alle Teiloperationen (bzw. Aktionen) fest. Die Gesamtheit der Restriktionen (4.7)...(4.11) beschreibt die Menge

aller zulässigen Ablaufpläne, d.h. den Lösungsraum des Modells. Zielsetzung des Modells ist, einen zulässigen Ablaufplan zu bestimmen, welcher die Gesamtdauer des Lagerprozesses (makespan) minimiert. Der Startzeitpunkt x_τ der Abschlussoperation τ entspricht dem Zeitpunkt, an welchem alle Operationen abgeschlossen sind, wenn die Operationen frühestens zum Zeitpunkt 0 gestartet werden (Nichtnegativitäts-Bedingungen). Somit entspricht x_τ dem Makespan des Lagerprozesses und die Zielfunktion des Modells ist gegeben durch:

$$\min x_\tau \quad (4.12)$$

Zusammenfassend wurde in diesem Abschnitt gezeigt, wie sich das Lager-Problem, d.h. das Steuerungsproblem auf der Ebene der Ablaufpläne, als Optimierungsmodell mit konjunktiven und disjunktiven Präzedenzbedingungen formulieren lässt. Der nachfolgende Abschnitt gibt einen vollständigen Ueberblick über das Modell. In Kapitel 5 wird gezeigt, wie sich dieses Modell als Optimierungsproblem in einem verallgemeinerten disjunktiven Graphen darstellen lässt.

4.3.2. Modellübersicht

Das Lager-Problem $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^s)$

Parameter:

M	Menge der Prozessoren
I	Menge der Operationen
I_m	Menge der Operationen, welche auf Prozessor m ausgeführt werden, $m \in M$
σ, τ	Fiktive Start- bzw. Abschlussoperation, welche vor bzw. nach allen andern Operationen ausgeführt wird.
\mathcal{J}	Menge der Jobs. Es gilt $\mathcal{J} \subseteq \text{Pot}(I)$ und \mathcal{J} bildet eine Partition von I . Die Operationen von Job $J \in \mathcal{J}$ werden bezeichnet als $J = \{J_1, \dots, J_{ J }\}$ wobei J_r $r \in \{1, \dots, J \}$ die r -te Operation bezeichnet in der Ausführungsreihenfolge von J .
μ_i	Prozessor, auf welchem Operation i ausgeführt wird, $i \in I$

Zeiten:

d_i	Dauer der Transportfahrt von Operation i , $i \in I$. Die Transportfahrt führt von der Startposition von i zur Zielposition von i .
d_i^t	Dauer der Palettenübernahme ("take over") von Operation i , $i \in I$. Falls i die erste Operation eines Jobs ist, entspricht d_i^t der Aufladezeit, andernfalls der Uebergabezeit.
d_i^h	Dauer der Palettenübergabe ("hand over") von Operation i , $i \in I$. Falls i die letzte Operation eines Jobs ist, entspricht d_i^h der Abladezeit, andernfalls der Uebergabezeit. Es wird vorausgesetzt, dass die Uebergabe- und Uebernahmezeit bei einem Palettentransfer identisch sind, d.h. $d_{J_r}^h = d_{J_{r+1}}^t$ für alle $J \in \mathcal{J}$ und $r = 1, \dots, J - 1$.

d_{ij}^s	Dauer der Leerfahrt ("setup") zwischen den Operationen i und j , $\{i, j\} \subseteq I_m$ $m \in M$ Die Leerfahrt führt von der Zielposition von i zur Startposition von j .
$d_{\sigma i}^s$	Dauer der Leerfahrt ("setup") von der anfänglichen Position von Prozessor m_i zur Startposition von Operation i , $i \in I$
$d_{i\tau}^s$	Dauer der Leerfahrt ("setup") von der Zielposition von Operation i zur abschliessenden Position von Prozessor m_i , $i \in I$

Weitere Bezeichnungen:

$t(i)$, t_i Tail-Operation von Operation i , $i \in I$

$h(i)$, h_i Head-Operation von Operation i , $i \in I$

Variablen:

$x_{t(i)}$ Startzeitpunkt der Tail-Operation von Operation i , $i \in I$

$x_{h(i)}$ Startzeitpunkt der Head-Operation von Operation i , $i \in I$

x_{σ} , x_{τ} Startzeitpunkt der (fiktiven) Startoperation σ bzw. Endoperation τ

Zielfunktion:

$$\min x_{\tau} \quad (4.13)$$

Restriktionen:

Konjunktive Präzedenzen zur Synchronisation:

$$x_{t(j)} - x_{h(i)} \geq 0, x_{t(i)} - x_{h(j)} \geq 0, \quad (i, j) \in \{(J_r, J_{r+1}) : J \in \mathcal{J}, r = 1, \dots, |J| - 1\} \quad (4.14)$$

Konjunktive Präzedenzen für tail- und head-Operationen:

$$x_{h(i)} - x_{t(i)} \geq d_i^t + d_i, \quad i \in I \quad (4.15)$$

Konjunktive Präzedenzen für σ , τ :

$$x_{t(i)} - x_{\sigma} \geq d_{\sigma i}^s, \quad i \in I \quad (4.16)$$

$$x_{\tau} - x_{h(i)} \geq d_i^h + d_{i\tau}^s, \quad i \in I \quad (4.17)$$

Disjunktive Präzedenzen für Prozessoren:

$$x_{t(j)} - x_{h(i)} \geq d_i^h + d_{ij}^s \text{ OR } x_{t(i)} - x_{h(j)} \geq d_j^h + d_{ji}^s, \quad \{i, j\} \subseteq I_m, m \in M \quad (4.18)$$

Nichtnegativitäts-Bedingungen:

$$x_{t(i)}, x_{h(i)} \geq 0, \quad i \in I \quad (4.19)$$

$$x_{\sigma}, x_{\tau} \geq 0 \quad (4.20)$$

4.4. Räumliche Nebenbedingungen

Im vorangehenden Abschnitt wurde ein Modell für das Lager-Problem präsentiert. Bei genauerer Betrachtung stellt man fest, dass das Modell nicht vollständig ist, da gewisse räumliche Nebenbedingungen, welche sich aus der Lagerung der Paletten in den Korridoren ergeben, noch nicht erfasst sind. Die räumlichen Nebenbedingungen ergeben sich aus folgender Tatsache: Falls eine Palette an einer bestimmten Position in einem Lagerkorridor abgestellt ist, kann der Korridorwagen diese Position zwar überfahren, wenn er unbeladen ist, nicht jedoch, wenn er mit einer Palette beladen ist. Daraus lassen sich folgende Nebenbedingungen herleiten: (1) Falls eine Palette mit Operation i an einer Zielposition p in einem Korridor eingelagert wird, dann müssen alle andern Operationen in diesem Korridor, welche diese Position überfahren, vor der Operation i ausgeführt werden. (2) Falls eine Palette, welche mit einer Operation i von einer Startposition p in einem Korridor ausgelagert wird, dann müssen alle andern Operationen in diesem Korridor, welche diese Position überfahren, nach der Operation i ausgeführt werden. Hierbei wird, wie schon früher erwähnt, immer davon ausgegangen, dass eine Palette nicht mehrmals bewegt wird während des betrachteten Lagerprozesses. Dies bedeutet, dass eine Palette, welche durch einen Ein- oder Umlagerungsjob an einem Lagerplatz abgestellt wird, nicht weiter verschoben wird durch einen andern Job.

In diesem Abschnitt wird gezeigt, dass diese räumlichen Nebenbedingungen mit Hilfe von zusätzlichen konjunktiven Präzedenzbedingungen modelliert werden können. Es wird sich zeigen, dass diese Nebenbedingungen zu einer Vorauswahl von bestimmten disjunktiven Präzedenzbedingungen führen. Dies bedeutet, dass für bestimmte Paare von Operationen auf den Korridorprozessoren die Ausführungsreihenfolge im voraus festgelegt ist. Für ein solches Operationenpaar $i, j \in I_m$, $m \in M^C$, wird die disjunktive Bedingung $x_{t(j)} - x_{h(i)} \geq d_i^h + d_{ij}^s$ OR $x_{t(i)} - x_{h(j)} \geq d_j^h + d_{ji}^s$ aus (4.18) ersetzt, indem eine der beiden disjunktiven Restriktionen als konjunktive Restriktion ausgewählt wird. Eine solche Vorauswahl von disjunktiven Präzedenzen wird als Präselektion bezeichnet.

Für die weiteren Ausführungen sei daran erinnert, dass die Operationen, welche auf einem Korridorprozessor $m \in M^C$ ausgeführt werden, in drei Typen eingeteilt werden können: Einlagerungsoperationen, Auslagerungsoperationen und Durchfahrtsoperationen. Eine Einlagerungsoperation übernimmt eine Palette von einem angrenzenden Quergangprozessor und stellt diese Palette an einer Lagerposition im Korridor ab. Eine Einlagerungsoperation stellt die letzte Operation eines Einlagerungs- oder Umlagerungsjobs dar. Eine Auslagerungsoperation hebt eine Palette von einer Lagerposition auf und übergibt sie einem angrenzenden Quergangprozessor. Eine Auslagerungsoperation stellt die erste Operation eines Auslagerungs- oder Umlagerungsjobs dar. Eine Durchfahrtsoperation übernimmt eine Palette von einem angrenzenden Quergangprozessor und übergibt sie ohne zwischenzeitliches Abstellen an den anderen angren-

zenden Quergangprozessor. Eine Durchfahrtsoperation stellt eine "innere" Operation eines Ein-, Aus- oder Umlagerungsjobs dar.

Im weiteren werden die folgenden Bezeichnungen verwendet:

I_m	Menge der Operationen auf Korridorprozessor m , $m \in M^C$
I_m^{in}	Menge der Einlagerungsoperationen auf Korridorprozessor m , $m \in M^C$
I_m^{out}	Menge der Auslagerungsoperationen auf Korridorprozessor m , $m \in M^C$
p_i	Zielposition (d.h. Einlagerungsposition) von Einlagerungsoperation i , $i \in I_m^{in}$, $m \in M^C$, bzw. Startposition (d.h. Auslagerungsposition) von Auslagerungsoperation i , $i \in I_m^{out}$, $m \in M^C$
P_i	Menge der überfahrenen Positionen von Operation i , $i \in I_m$, $m \in M^C$

Die räumlichen Nebenbedingungen in den Korridoren führen gemäss obigen Ausführungen zu folgenden konjunktiven Präzedenzen:

$$Q_m^{in} = \{(i, j) : i \in I_m - j, j \in I_m^{in}, p_j \in P_i\}, \quad m \in M^C$$

$$Q_m^{out} = \{(i, j) : i \in I_m^{out}, j \in I_m - i, p_i \in P_j\}, \quad m \in M^C$$

Die Präzedenzen Q_m^{in} sind folgendermassen zu interpretieren: Falls eine Einlagerungsoperation j eine Palette an Position p_j einlagert, müssen alle andern Operationen i , welche diese Position überfahren (d.h. $p_j \in P_i$), vor Operation j ausgeführt werden. Die Präzedenzen Q_m^{out} haben folgende Interpretation: Falls eine Auslagerungsoperation i eine Palette von Position p_i auslagert, müssen alle andern Operationen j , welche diese Position überfahren (d.h. $p_i \in P_j$), vor Operation i ausgeführt werden. Zusammengefasst ergibt sich die folgende Präselektion Q von disjunktiven Präzedenzen:

$$Q = \bigcup_{m \in M^C} (Q_m^{in} \cup Q_m^{out})$$

Das Lager-Modell (4.13)...(4.20) muss entsprechend um folgende konjunktiven Restriktionen erweitert werden:

$$x_{t(j)} - x_{h(i)} \geq d_i^h + d_{ij}^s, \quad (i, j) \in Q \quad (4.21)$$

Gleichzeitig können alle disjunktiven Bedingungen aus (4.18) entfernt werden, welche eine zu Q gehörende disjunktive Restriktion umfassen.

Die nachfolgende Tabelle gibt eine Uebersicht über die verschiedenen Fälle, die bei der Betrachtung der räumlichen Nebenbedingungen in einem Korridor auftreten. Anhand dieser Tabelle kann überprüft werden, dass durch die Präselektion Q tatsächlich alle zu betrachtenden räumlichen Nebenbedingungen erfasst sind. Jede Zeile der Tabelle zeigt schematisch einen unterschiedlichen Fall, in welchem zwei Operationen i und j zu einer Präzedenzbedingung (i, j) bzw. (j, i)

in Q führen. Der Typ von i und j ist angegeben als *in*, *out* und *cross* für Einlagerungs-, Auslagerungs- und Durchfahrtsoperation.

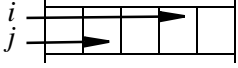
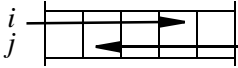
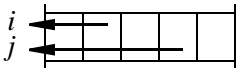


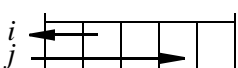

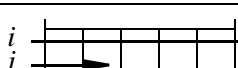
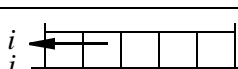
Fall:	Operation:		Abbildung:	Relation:	Bogen:
	i	j			
1	in	in		$p_j \in P_i$	$(i, j) \in Q_m^{in}$
2	in	in		$p_j \in P_i, p_i \in P_j$	$(i, j) \in Q_m^{in}, (j, i) \in Q_m^{in}$
3	out	out		$p_i \in P_j$	$(i, j) \in Q_m^{out}$
4	out	out		$p_i \in P_j, p_j \in P_i$	$(i, j) \in Q_m^{out}, (j, i) \in Q_m^{out}$
5	out	in		$p_j \in P_i$	$(i, j) \in Q_m^{in}$
6	out	in		$p_i \in P_j$	$(i, j) \in Q_m^{out}$
7	out	in		$p_j \in P_i, p_i \in P_j$	$(i, j) \in Q_m^{in}, (i, j) \in Q_m^{out}$
8	cross	in		$p_j \in P_i$	$(i, j) \in Q_m^{in}$
9	out	cross		$p_i \in P_j$	$(i, j) \in Q_m^{out}$

Abbildung 4.4. Präzedenzbedingungen in Lagerkorridoren

Die Fälle 2 und 4 stellen eine unzulässige Situation dar, d.h. es liegt ein Lagerauftrag vor, der nicht realisierbar ist. Zwei Ein- bzw. Auslagerungsoperationen, welche sich gegenseitig räumlich "kreuzen", können nicht beide ausgeführt werden. Die Unzulässigkeit zeigt sich in den entsprechenden Präzedenzen dadurch, dass zwei entgegengesetzte Präzedenzen (i, j) und (j, i) vorhanden sind. Dies entspricht der Forderung, dass einerseits Operation i vor Operation j und andererseits j vor i ausgeführt werden muss. In Abschnitt 6.6. wird gezeigt, dass die Absenz solcher "Zyklen" in der Präselektion Q eine notwendige und unter gewissen Umständen auch hinreichende Bedingung dafür ist, dass das Lagerproblem eine zulässige Lösung hat.

Es sei noch darauf hingewiesen, dass auch in Fall 7 zwei Präzedenzen vorhanden sind. Diese sind jedoch identisch und somit konsistent.

Stationäre Paletten

Vorangehend wurde gezeigt, wie die räumlichen Nebenbedingungen in einem Korridor zu Präzedenzbedingungen zwischen den Korridoroperationen führen. Die dabei betrachteten räumlichen Nebenbedingungen beziehen sich ausschliesslich auf Paletten, welche im Verlauf des Lagerprozesses bewegt werden. Typischerweise befinden sich in einem Lager aber noch viele Paletten, die an einem Ort gelagert sind und während des Lagerprozesses nicht bewegt werden. Solche Paletten werden als stationäre Paletten bezeichnet. Für stationäre Paletten ergeben sich folgende räumlichen Nebenbedingungen: Falls sich eine stationäre Palette an einer Lagerposition p befindet, dann kann diese Position vom entsprechenden Korridorprozessor $m \in M^C$ nicht in beladenem Zustand überfahren werden. Folglich ist ein Lagerauftrag nur dann ausführbar, wenn keine Operation $i \in I_m$ in diesem Korridor die Position p überfährt. Eine notwendige Bedingung dafür, dass ein Lagerauftrag ausführbar ist, ist somit gegeben durch:

$$p \notin P_i, \quad \text{für alle } i \in I_m, m \in M^C$$

Diese Bedingungen lassen sich bei gegebenem Lagerauftrag und gegebener Menge von stationären Paletten einfach überprüfen. Falls eine stationäre Palette existiert, welche diese Bedingung nicht erfüllt, ist der Lagerauftrag nicht ausführbar und folglich das entsprechende Steuerungsproblem unlösbar. Im folgenden wird deshalb stets davon ausgegangen, dass der dem Lagersteuerungs-Problem zugrunde liegende Lagerauftrag diese Bedingungen erfüllt. In diesem Fall haben die stationären Paletten keinen Einfluss für das Steuerungsproblem und können vernachlässigt werden.

Verallgemeinerte disjunktive Graphen

Disjunktive Graphen bilden ein bekanntes und bewährtes Konzept zur Modellierung des klassischen Job-Shop-Problems und ähnlicher Maschinenbelegungsprobleme. Die Idee der disjunktiven Graphen wurde erstmals vorgeschlagen in [Roy, B., Sussman, B. (1964)]. Seither haben sich disjunktive Graphen als eine Standardmethode zur Darstellung des Job-Shop-Problems etabliert. Ein beträchtlicher Teil der Optimierungsverfahren für das Job-Shop-Problem basiert direkt oder indirekt auf der Problemdarstellung in einem disjunktiven Graphen (vgl. [Pinson, E. (1995)]).

Disjunktive Graphen gelangen vorwiegend zur Anwendung in Zusammenhang mit Reihenfolgeproblemen in der Ablaufplanung. In Abschnitt 4.2. wurde gezeigt, wie sich Reihenfolgeprobleme mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen modellieren lassen. Weiter wurde erwähnt, dass sich solche Modelle in Form eines disjunktiven Graphen darstellen lassen. Der disjunktive Graph G zu einem Modell mit konjunktiven und disjunktiven Präzedenzbedingungen wird folgendermassen konstruiert: Die Operationen (Aktivitäten) werden als Knoten und die Präzedenzbedingungen als gewichtete Bogen dargestellt. Eine konjunktive Präzedenzbedingung entspricht einem konjunktiven Bogen und eine disjunktive Präzedenzbedingung einem Paar von disjunktiven Bogen, von welchen mindestens ein Bogen ausgewählt werden muss. Dies bedeutet, dass in G die Menge derjenigen Subgraphen betrachtet wird, welche aus jeder disjunktiven Menge mindestens einen Bogen enthalten. Solche Subgraphen werden vollständig genannt. Modelle mit disjunktiven Präzedenzbedingungen und minimalem Makespan als Zielfunktion lassen sich im entsprechenden disjunktiven Graphen lösen, indem ein vollständiger Subgraph bestimmt wird, welcher die Länge eines längsten Weges minimiert.

In Abschnitt 4.2. wurde eine Formulierung des Job-Shop-Problems präsentiert, welche auf disjunktiven Präzedenzbedingungen basiert. In Abschnitt 4.3. wurde gezeigt, wie sich das Lager-Problem, d.h. das Steuerungsproblem für automatisierte Kompaktlager, ebenfalls als Optimierungsmodell mit disjunktiven Präzedenzbedingungen formulieren lässt. Disjunktive Graphen, welche aus dem Job-Shop- und ähnlichen Problemen resultieren, haben die spezielle Eigenschaft, dass jede disjunktive Menge zwei Bogen mit denselben Endpunkten und entgegengesetzter Richtung umfasst. Solche Graphen werden im weiteren als klassische disjunktive Graphen bezeichnet. Bis anhin wurden in der Literatur im Zusammenhang mit disjunktiven Graphen praktisch ausschliesslich Reihenfolgeprobleme betrachtet, welche sich mittels klassischer disjunktiver Graphen modellieren lassen. Bei Betrachtung der Formulierung des Lager-Problems stellt man fest, dass die disjunktiven Präzedenzbedingungen zu zwei Bogen führen, welche unterschiedliche Endpunkte haben. Dies rührt aus der Tatsache, dass eine Operation im Lager-Problem aufgrund der kontextabhängigen Ausführungsdauer durch zwei Teiloperationen (genannt Tail- und Head-Operation) dargestellt wird. Folglich entspricht die Bedingung, dass eine Operation i vor einer Operation j ausgeführt werden muss, der Forderung, dass die Head-Operation von i vor der Tail-Operation von j ausgeführt wird.

Aus diesem Umstand drängt sich eine Verallgemeinerung des Konzepts der klassischen disjunktiven Graphen auf: Statt nur disjunktive Mengen zu betrachten, welche zwei Bogen entgegengesetzter Richtung umfassen, wird zugelassen, dass eine disjunktive Menge zwei beliebige Bogen enthalten kann. Graphen mit dieser Eigenschaft werden als verallgemeinerte disjunktive Graphen (oder kurz als disjunktive Graphen) bezeichnet. Das Lager-Problem lässt sich somit in einem verallgemeinerten disjunktiven Graphen darstellen.

Es wird sich zeigen, dass durch diese Verallgemeinerung ein beträchtliches Mass an Struktur verloren geht gegenüber klassischen disjunktiven Graphen. Dieser Strukturverlust hat weitgehende Konsequenzen beim Entwurf von Optimierungsalgorithmen für verallgemeinerte disjunktive Graphen. Insbesondere lässt sich ein Grossteil der aus der intensiven Erforschung des Job-Shop-Problems gewonnenen Erkenntnisse in verallgemeinerten disjunktiven Graphen nicht umsetzen. Ein wesentlicher Grund dafür ist die Tatsache, dass das Zulässigkeitsproblem in verallgemeinerten disjunktiven Graphen im Gegensatz zu klassischen disjunktiven Graphen NP-hart ist (siehe Abschnitt 5.5.). Dieser Umstand deutet darauf hin, dass die Optimierung in verallgemeinerten disjunktiven Graphen wesentlich schwieriger ist als die Lösung beispielsweise des Job-Shop-Problems. Da das Job-Shop-Problem zu den schwierigsten Problemen der kombinatorischen Optimierung gezählt wird, lässt sich erahnen, dass die Optimierung in verallgemeinerten disjunktiven Graphen eine anspruchsvolle Aufgabe darstellt.

Im vorliegenden Kapitel wird in Abschnitt 5.1. zunächst gezeigt, wie sich allgemeine Präzedenzmodelle, d.h. Modelle mit konjunktiven und disjunktiven Präzedenzbedingungen, als Optimierungsprobleme in verallgemeinerten disjunktiven Graphen darstellen lassen. In Abschnitt 5.2. wird der Begriff des verallgemeinerten disjunktiven Graphen im Detail diskutiert, und es werden

einige grundlegende Eigenschaften festgehalten. In Abschnitt 5.3. werden verschiedene Klassen von disjunktiven Graphen, welche im Rahmen dieser Arbeit von Interesse sind, beschrieben und an einem Beispiel illustriert: Klassische disjunktive Graphen, Job-Shop-Graphen, Lager-Graphen und 1-Job-Lager-Graphen. In Abschnitt 5.4. wird das Konzept der konjunktiven Erweiterung diskutiert. Konjunktive Erweiterung befasst sich mit der Frage, welche disjunktiven Bogen in einem disjunktiven Graphen verboten bzw. erzwungen sind, d.h. welche Bogen in keiner bzw. allen zulässigen Selektionen vorhanden sind. Erzwungene und verbotene Bogen spielen eine wichtige Rolle beim Entwurf eines Optimierungsverfahrens für das Lager-Problem. Das Konzept der konjunktiven Erweiterung steht in engem Zusammenhang mit dem Zulässigkeitsproblem in disjunktiven Graphen. Das Zulässigkeitsproblem entspricht der Fragestellung, ob in einem disjunktiven Graphen eine zulässige Auswahl von disjunktiven Bogen existiert. Das Zulässigkeitsproblem wird in Abschnitt 5.5. diskutiert. Es werden diverse komplexitätstheoretische und strukturelle Resultate erarbeitet für die verschiedenen, in Abschnitt 5.3. definierten Graphenklassen. Ein wichtiges Resultat bildet hierbei die Erkenntnis, dass das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen und auch für Lager-Graphen NP-hart ist.

Zusammenfassend werden in diesem Kapitel die wesentlichen Grundlagen erarbeitet, welche für den Entwurf eines Optimierungsverfahrens für das Lager-Problem in Kapitel 6 benötigt werden.

5.1. Disjunktive Präzedenzbedingungen und disjunktive Graphen

In diesem Abschnitt wird gezeigt, wie sich disjunktive Präzedenzmodelle, d.h. Modelle mit konjunktiven und disjunktiven Präzedenzbedingungen, in Form von (verallgemeinerten) disjunktiven Graphen darstellen lassen. Zunächst wird der Begriff des disjunktiven Präzedenzmodells definiert und dessen Interpretation im Zusammenhang mit Maschinenbelegungsproblemen diskutiert. Anschliessend wird der Begriff der Selektion eingeführt, worunter eine Auswahl von disjunktiven Präzedenzbedingungen verstanden wird. Es wird gezeigt, dass sich ein disjunktives Präzedenzmodell als zweistufiges Optimierungsproblem formulieren lässt, bei welchem das Minimum eines linearen Programms (bestehend aus konjunktiven Präzedenzbedingungen) über alle zulässigen Selektionen minimiert werden muss. Eine duale Betrachtung dieses Linearen Programms zeigt (die bekannte Tatsache), dass sich dieses Programm als Problem eines längsten Weges in einem zugeordneten Graphen interpretieren lässt. Daraus resultiert das Konzept der disjunktiven Graphen: Ein disjunktives Präzedenzmodell lässt sich lösen, indem in einem zugeordneten disjunktiven Graphen eine zulässige Auswahl von Bogen gesucht wird, welche die Länge eines längsten Weges minimiert. Dieses Problem wird disjunktives Graphenproblem

genannt. Abschliessend wird eine kurze Literaturübersicht zum Thema der disjunktiven Graphen gegeben.

5.1.1. Disjunktive Präzedenzmodelle

Definition 5.1. Sei V eine endliche Grundmenge und $\tau \in V$ ein ausgezeichnetes Element von V .

Sei $A \cup E \subseteq \{(v, w) : v, w \in V, v \neq w\}$, wobei $A \cap E = \emptyset$ gelten soll. Sei weiter $\mathcal{E} \subseteq \text{Pot}(E)$ mit

$E = \bigcup_{D \in \mathcal{E}} D$. Sei $c : A \cup E \rightarrow \mathbb{R}$ eine Gewichtung in A und E .

Ein disjunktives Präzedenzmodell $\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ ist ein Optimierungsproblem der folgenden Form:

$$\min x_\tau \quad (5.1)$$

$$x_w - x_v \geq c_{vw}, \quad (v, w) \in A \quad (5.2)$$

$$\text{OR}_{(v, w) \in D} x_w - x_v \geq c_{vw}, \quad D \in \mathcal{E} \quad (5.3)$$

$$x_v \geq 0, \quad v \in V \quad (5.4)$$

A wird als die Menge der konjunktiven und E als die Menge der disjunktiven Präzedenzen bezeichnet. Die zu A bzw. \mathcal{E} gehörenden Restriktionen heissen konjunktive bzw. disjunktive Restriktionen. Die Mengen $D \in \mathcal{E}$ werden disjunktive Mengen oder Disjunktionen genannt.

Die disjunktiven Bedingungen (5.3) sind folgendermassen zu verstehen: Für jede disjunktive Menge $D \in \mathcal{E}$ der Form $D = \{(v_1, w_1), (v_2, w_2), \dots, (v_r, w_r)\}$ muss gelten $x_{w_1} - x_{v_1} \geq c_{v_1 w_1}$ OR $x_{w_2} - x_{v_2} \geq c_{v_2 w_2}$ OR ... OR $x_{w_r} - x_{v_r} \geq c_{v_r w_r}$, d.h. mindestens eine der disjunktiven Restriktionen von D muss zutreffen.

Interpretation für Maschinenbelegungsprobleme

Falls Π eine Maschinenbelegungsproblem darstellt, werden die Problemdata wie folgt interpretiert: V entspricht der Menge der auszuführenden Operationen und die Variablen x_v stellen die Startzeitpunkte der Operationen $v \in V$ dar. Eine Präzedenzbedingung der Form $x_w - x_v \geq c_{vw}$ entspricht somit der Forderung, dass der Startzeitpunkt von Operation w frühestens c_{vw} Zeiteinheiten nach dem Startzeitpunkt von v erfolgen kann.

Die spezielle Operation $\tau \in V$ stellt eine (fiktive) Abschlussoperation dar, welche erst nach Beendigung aller andern Operation ausgeführt werden soll. Um diese Bedingung zu realisieren, enthält Π normalerweise für jede Operation $v \in V - \tau$ eine konjunktive Präzedenzbedingung der Form $x_\tau - x_v \geq c_{v\tau}$ mit $c_{v\tau} \geq d_v$ wobei d_v die Ausführungsdauer von Operation darstellt. Die entsprechende Menge von konjunktiven Präzedenzen wird bezeichnet als $A_\tau = \{(v, \tau) : v \in V - \tau\}$. Häufig wird neben τ noch ein weiterer spezieller Knoten $\sigma \in V$ eingeführt. σ entspricht einer

(fiktiven) Startoperation, die vor Beginn aller andern Operationen ausgeführt werden soll. Um diese Bedingung zu realisieren, wird analog für jede Operation $v \in V - \sigma$ eine konjunktive Präzedenzbedingung der Form $x_v - x_\sigma \geq c_{\sigma v}$ eingeführt, wobei $c_{\sigma v} \geq 0$. Die entsprechende Menge von konjunktiven Präzedenzen wird bezeichnet als $A_\sigma = \{(\sigma, v) : v \in V - \sigma\}$.

Die Zielfunktion $\min x_\tau$ minimiert den Startzeitpunkt der fiktiven Abschlussoperation τ . Folglich entspricht der optimale Zielfunktionswert dem frühestmöglichen Zeitpunkt, an welchem alle Operationen abgeschlossen sind, wobei die Operationen frühestens zum Zeitpunkt 0 gestartet werden können (Nichtnegativitäts-Bedingungen). Beachte, dass diese Interpretation jedoch nur gilt, falls $A_\tau \subseteq A$. Die Zeitdauer zwischen dem Startzeitpunkt der frühesten Operation und dem Beendigungszeitpunkt der spätesten Operation wird als Makespan bezeichnet. Der Optimalwert von Π entspricht folglich dem minimalen Makespan des Problems. Falls für die Zielfunktion eine andere Operation als τ gewählt wird, d.h. falls die Zielfunktion die Form $\min x_v$ hat, entspricht der Optimalwert von Π dem frühestmöglichen Startzeitpunkt von Operation $v \in V$.

Selektionen

Eine Selektion in Π ist eine Auswahl $S \subseteq E$ von disjunktiven Präzedenzen. Eine disjunktive Menge $D \in \mathcal{E}$ ist geschnitten von S , falls $S \cap D \neq \emptyset$. Die Familie aller von S geschnittenen disjunktiven Mengen wird mit $\mathcal{E}(S) = \{D \in \mathcal{E} : S \cap D \neq \emptyset\}$ bezeichnet.

Für eine Selektion $S \subseteq E$ bezeichne:

$$P(S) = \{x \in \mathbb{R}^V : x_w - x_v \geq c_{vw} \text{ für } (v, w) \in A \cup S, x_v \geq 0 \text{ für } v \in V\}$$

$P(S)$ entspricht dem Polyeder, welches definiert wird durch die konjunktiven Restriktionen aus A , die disjunktiven Restriktionen aus S und die Nichtnegativitäts-Bedingungen. Offensichtlich ist $P(S)$ ein Lösungsteilraum von Π , d.h. es gilt $P(S) \subseteq P(\Pi)$ wobei $P(\Pi)$ den Lösungsraum von Π bezeichnet.

Eine Selektion $S \subseteq E$ heisst vollständig, falls sie alle disjunktiven Mengen schneidet, d.h. falls $\mathcal{E}(S) = \mathcal{E}$. S heisst zulässig, falls S vollständig ist und $P(S)$ eine zulässige Lösung besitzt, d.h. $P(S) \neq \emptyset$. Die Menge aller zulässigen Selektionen wird mit \mathcal{S} bezeichnet.

Es ist leicht nachzuvollziehen, dass der Lösungsraum von Π gegeben ist durch $P(\Pi) = \bigcup_{S \in \mathcal{S}} P(S)$. Er entspricht somit einer endlichen Vereinigung von Polyedern und ist folglich im allgemeinen nicht konvex.

Das Problem Π kann nun alternativ als ein zweistufiges Lineares Programm beschrieben werden:

$$\min \{ \min \{ x_\tau : x \in P(S) \} : S \in \mathcal{S} \} \tag{5.5}$$

Bezeichne das Lineare Programm $\min \{ x_\tau : x \in P(S) \}$ als $\Pi(S)$

Duales Problem

Betrachte für eine zulässige Selektion $S \in \mathcal{S}$ das duale Problem $\Pi^D(S)$ von $\Pi(S)$. Es ist gegeben durch:

$$\max \sum_{(v,w) \in A \cup S} c_{vw} y_{vw} \quad (5.6)$$

$$\sum_{u:(u,v) \in A \cup S} y_{uv} - \sum_{w:(v,w) \in A \cup S} y_{vw} \leq 0, \quad v \in V - \tau \quad (5.7)$$

$$\sum_{u:(u,v) \in A \cup S} y_{uv} - \sum_{w:(v,w) \in A \cup S} y_{vw} \leq 1, \quad v = \tau \quad (5.8)$$

$$y_{vw} \geq 0, \quad (v,w) \in A \cup S \quad (5.9)$$

Da S zulässig ist, hat $\Pi(S)$ eine zulässige Lösung. Da $x_\tau \geq 0$ ist der Optimalwert von $\Pi(S)$ beschränkt. Daraus folgt aus der Dualitätstheorie, dass auch $\Pi^D(S)$ eine zulässige Lösung und ein endliches Optimum besitzt, und dass der primale und der duale Optimalwert identisch sind. Somit kann das Problem Π auch geschrieben werden als

$$\min \{ \max \{ \sum_{(v,w) \in A \cup S} c_{vw} y_{vw} : y \in P^D(S) \} : S \in \mathcal{S} \} \quad (5.10)$$

wobei $P^D(S)$ den Lösungsraum des dualen Problems $\Pi^D(S)$ bezeichnet.

5.1.2. Disjunktive Graphen

Disjunktiver Graph

Sei $\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktives Präzedenzmodell. Π kann auf folgende Weise ein gerichteter Graph zugeordnet werden: V entspricht der Knotenmenge, $A \cup E$ der Bogenmenge und c den Bogengewichten. A wird als die Menge der konjunktiven und E als die Menge der disjunktiven Bogen bezeichnet. Zusammen mit der Familie der disjunktiven Mengen \mathcal{E} bildet dieser Graph einen sogenannten disjunktiven Graphen. Der zu Π gehörende disjunktive Graph wird mit $G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ bezeichnet.

Selektionen

Eine Selektion in G^Π ist eine Auswahl $S \subseteq E$ von disjunktiven Bogen. Eine Selektion heisst vollständig, falls $\mathcal{E}(S) = \mathcal{E}$, d.h. falls aus jeder disjunktiven Menge $D \in \mathcal{E}$ mindestens ein Bogen ausgewählt ist. Für eine Selektion $S \subseteq E$ bezeichne:

$$G^\Pi(S) = \langle V, A \cup S, c \rangle$$

$G^\Pi(S)$ ist ein normaler "konjunktiver" Graph mit Knotenmenge V , Bogenmenge $A \cup S$ und Bogengewichten c .

Zulässige Selektionen

Die Menge der zulässigen Selektionen in Π ist definiert als:

$$\mathcal{S} = \{S \subseteq E : S \text{ ist vollständig und } P(S) \neq \emptyset\}$$

Aufgrund der Dualitätstheorie gilt für alle $S \subseteq E$, dass $\Pi(S)$ genau dann eine zulässige Lösung und ein endliches Optimum besitzt, wenn das duale Problem $\Pi^D(S)$ eine zulässige Lösung und ein endliches Optimum besitzt. Für das primale Problem $\Pi(S)$ gilt aufgrund der Nichtnegativitäts-Bedingung $x_\tau \geq 0$, dass der Optimalwert beschränkt ist. Für das duale Problem $\Pi^D(S)$ gilt, dass immer eine zulässige Lösung existiert, nämlich $y = 0$. Somit hat $\Pi(S)$ genau dann eine zulässige Lösung, d.h. $P(S) \neq \emptyset$, wenn der Optimalwert von $\Pi^D(S)$ beschränkt ist.

Das duale Problem $\Pi^D(S)$ stellt offensichtlich ein Flussproblem dar, wobei die Restriktionen besagen, dass die Divergenz in einem Knoten höchstens gleich 0 bzw. gleich 1 (für $v = \tau$) sein darf. Der Optimalwert von $\Pi^D(S)$ ist genau dann beschränkt, wenn der Graph $G^\Pi(S)$ keinen Zyklus positiven Gewichts enthält. Dies lässt sich folgendermassen einsehen: Sei y eine zulässige Lösung von $\Pi^D(S)$ und Z ein positiver Zyklus in $G^\Pi(S)$. Der Fluss auf dem Zyklus lässt sich beliebig erhöhen, ohne dass die Divergenzen der Knoten verändert werden. Da aber das Gewicht von Z positiv ist, wächst die Zielfunktion bei einer Erhöhung des zyklischen Flusses auf Z .

Somit kann die Menge \mathcal{S} der zulässigen Selektionen beschrieben werden als:

$$\mathcal{S} = \{S \subseteq E : S \text{ ist vollständig und } G^\Pi(S) \text{ ist positiv azyklisch}\}$$

Längste Wege

Betrachte für eine zulässige Selektion $S \subseteq E$ das duale Problem $\Pi^D(S)$. Da S zulässig ist, enthält $G^\Pi(S)$ keine positiven Zyklen. Es ist leicht nachzuvollziehen (und hinlänglich bekannt), dass in diesem Fall das Problem $\Pi^D(S)$ als Problem des längsten τ -Weges in $G^\Pi(S)$ interpretiert werden kann. Das bedeutet: Der Optimalwert von $\Pi^D(S)$ entspricht der Länge eines längsten, in τ endenden Weges in $G^\Pi(S)$. Weiter existiert immer eine ganzzahlige 0-1-Optimallösung, welche den Inzidenzvektor eines längsten, in τ endenden Weges darstellt.

Falls Π eine künstliche Startoperation $\sigma \in V$ enthält mit den entsprechenden Bedingungen $x_v - x_\sigma \geq 0$ (vgl. die Bemerkungen weiter oben), dann existiert immer ein Weg von σ nach τ , welcher einen längsten in τ endenden Weg darstellt. In diesem Fall entspricht $\Pi^D(S)$ dem Problem, einen längsten Weg von σ nach τ in $G^\Pi(S)$ zu bestimmen.

Disjunktives Graphenproblem

Aus den vorangehenden Ausführungen folgt, dass das Problem Π auf folgende Weise in einem disjunktiven Graphen gelöst werden kann:

Bestimme eine vollständige, positiv azyklische Selektion S in G^Π , so dass die Länge eines längsten τ -Weges (bzw. σ, τ -Weges) in $G^\Pi(S)$ minimal ist. (5.11)

Dieses Problem wird als disjunktives Graphenproblem im Graphen G^Π bezeichnet oder kurz als disjunktives Graphenproblem G^Π . Anders ausgedrückt lässt sich ein disjunktives Graphenproblem wie folgt formulieren: Gegeben ist ein (gewöhnlicher) Graph und eine Familie von Bogenteilmengen. Gesucht ist ein zulässiger Subgraph, welcher die Länge eines längsten τ -Weges minimiert. Hierbei ist ein Subgraph zulässig, falls er positiv azyklisch ist und aus jeder Bogenteilmenge mindestens einen Bogen enthält.

Abschliessend sei noch kurz auf die Komplexität des disjunktiven Graphenproblems eingegangen. Wie schon gezeigt wurde und allgemein bekannt ist, lässt sich das Job-Shop-Problem als disjunktives Präzedenzmodell und folglich als disjunktives Graphenproblem formulieren. Da das Job-Shop-Problem (bzw. seine Version als Entscheidungsproblem) bekanntlich ein NP-hartes Problem darstellt, ist auch das allgemeinere disjunktive Graphenproblem NP-hart.

5.1.3. Literatur

In diesem Abschnitt wird eine kurze Uebersicht über Literaturbeiträge zu den Themen disjunktive Präzedenzbedingungen, disjunktive Graphen, verallgemeinerte disjunktive Graphen und disjunktive Programmierung gegeben.

Disjunktive Graphen wurden erstmals erwähnt im viel zitierten Artikel von [Roy, B., Sussman, B. (1964)], in welchem dieser Ansatz zur Modellierung von Reihenfolge-Problemen in der Maschinenablauf-Planung verwendet wurde. Das Konzept der disjunktiven Graphen wurde anschliessend in den Beiträgen von [Balas, E. (1966)] und [Balas, E. (1969)] aufgegriffen und ausführlich diskutiert. Im letzteren der beiden Artikel wird gezeigt, wie sich das Job-Shop-Problem als Optimierungsproblem in einem disjunktiven Graphen darstellen lässt. Davon ausgehend wird ein implizites Enumerationsverfahren entworfen, welches auf dem Austausch von kritischen disjunktiven Bogen basiert. Insbesondere wird in dem Artikel gezeigt, dass der Austausch eines kritischen Bogens in einem Job-Shop-Graphen keine Zyklen erzeugt und somit zu einer zulässigen Nachbarselektion führt. In den vergangenen Jahrzehnten wurde eine Vielzahl von Optimierungsmethoden für das Job-Shop-Problem entwickelt, von denen ein beträchtlicher Teil direkt oder indirekt auf einer Problemdarstellung mittels disjunktiver Graphen beruht (vgl. [Pinson, E. (1995)]).

In dem neueren Artikel von [White Jr., K. P., Rogers, R. V. (1990)] wird ein erweitertes Job-Shop-Problem (extended job-shop problem) mit zusätzlichen Nebenbedingungen betrachtet. Es wird gezeigt, wie verschiedene Typen von Nebenbedingungen mit Hilfe von zusätzlichen Knoten sowie konjunktiven und disjunktiven Präzedenzbedingungen modelliert werden können. Die resultierenden Graphen stellen klassische disjunktive Graphen dar, d.h. es werden nur disjunktive Bogenpaare mit identischen Extremitäten betrachtet. Als Nebenbedingungen werden unter ande-

rem frühestmögliche Starttermine (ready-dates) und spätestmögliche Abschlusstermine (due-dates) betrachtet, wobei erstere durch konjunktive Bogen und letztere durch Hinzufügen von fiktiven Abschlussoperationen mit negativer Dauer modelliert werden. Weiter werden Unterhalt-Arbeiten (scheduled maintenance) durch zusätzliche Operationen modelliert und es wird gezeigt, dass Setup-Zeiten unter gewissen Umständen durch das Hinzufügen von fiktiven Operationen in das Modell integriert werden können. Auf die Optimierungsmethodik wird in dem Artikel nicht eingegangen. In dem Beitrag von [Gillies, D. W., Liu, J. W.-S. (1995)] werden disjunktive Maschinenbelegungsprobleme mit zwei Typen von Operationen (tasks) betrachtet. AND-Tasks sind Operationen, welche erst ausgeführt werden können, wenn alle Vorgänger-Operationen beendet sind. Dies entspricht den klassischen konjunktiven Präzedenzbedingungen. OR-Tasks können gestartet werden, wenn mindestens einer ihrer Vorgänger beendet ist. Für diese Art von disjunktiven Problemen werden zwei Prioritätsregeln-Verfahren zur Minimierung des Makespan entwickelt. Vom Standpunkt der vorliegenden Arbeit aus können diese Modelle als verallgemeinerte disjunktive Graphen aufgefasst werden, bei denen die Bogen, welche in einem OR-Task enden, eine disjunktive Menge bilden. In diesem Sinne führen diese Probleme zu einer (sehr spezifischen) Verallgemeinerung der klassischen disjunktiven Graphen. In dem Artikel von [Blazewicz, J. Pesch, E., Sterna, M. (2000)] wird eine Erweiterung der disjunktiven Job-Shop-Graphen betrachtet, welche darin besteht, dass ein zusätzlicher konjunktiver Bogen hinzugefügt wird, welcher vom Endknoten (Senke) zum Startknoten (Quelle) zurückführt. Das Gewicht dieses Bogens entspricht dem negativen Wert einer oberen Schranke des Job-Shop-Problems. Durch diese Erweiterung lassen sich zeitliche Distanzen und Abhängigkeiten zwischen den Operationen herleiten, welche unter anderem in Branch-and-Bound-Verfahren ausgenützt werden können.

In den Beiträgen von [Esquirol, P., Huguet, M. J., Lopez, P. (1992)] und [Huguet, M.-J. (1994)] werden sogenannte TBON-Graphen (Time-Bound-On-Node Graphen) zur Darstellung von Maschinenbelegungsproblemen verwendet. Operationen werden durch zwei Knoten repräsentiert, wobei der eine dem Startzeitpunkt und der andere dem Endzeitpunkt der Operation entspricht. Durch diese Darstellung ist es möglich, Operationen mit variabler Dauer zu modellieren. Dieser Ansatz zeigt Ähnlichkeiten zu den Tail- und Head-Operationen, welche zur Darstellung des Lager-Problems verwendet wurden. TBON-Graphen stellen verallgemeinerte disjunktive Graphen dar, da die Bedingung, dass disjunktive Bogenpaare identische Extremitäten haben müssen, fallengelassen wird. Es wird gezeigt, wie durch Betrachtung von längsten Wegen (constraint propagation) Zeitintervalle (time bounds) für die Knoten (d.h. die Start- bzw. Endzeitpunkte der Operationen) abgeleitet werden können, innerhalb welcher diese liegen müssen. Auf die Optimierung in TBON-Graphen wird in diesen Arbeiten nicht eingegangen.

Schliesslich seien zwei anwendungsbezogene Arbeiten erwähnt, welche am Institut für Operations Research der ETH Zürich unter der Leitung von H. Gröflin durchgeführt wurden. In der Arbeit von [Schneider, S. (1988)] wird die Kompaktierung von VLSI-Chips betrachtet. Die Problemstellung besteht darin, die Schaltelemente und Zellen so anzuordnen, dass sie auf einer mög-

lichst kleinen Chipfläche vollständig verdrahtet werden können. Für dieses Problem wird ein Optimierungsmodell in Form eines verallgemeinerten disjunktiven Graphen entworfen. Die disjunktiven Präzedenzbedingungen werden dazu benützt, die gegenseitige Lage der verschiedenen Elemente zu beschreiben. Für die Optimierung wird ein Verfahren entwickelt, welches auf impliziter Enumeration und Austausch von kritischen Bogen basiert. Die Arbeit von [Ingold, Th. (1992)] befasst sich mit der Steuerung eines Verarbeitungssystems mit Verschiebungsbrücken. Das System besteht aus zwei stationären Maschinengruppen sowie einer Menge von dazwischenliegenden, verfahrbaren Transferbrücken, welche dazu dienen, die Bearbeitungsobjekte von einer Maschinengruppe zur anderen überzuführen. Die Transferbrücken sind schienengebunden und benutzen alle denselben Schienenstrang. Dadurch werden die Bewegungen der Brücken gegenseitig voneinander beeinflusst. Es wird gezeigt, wie sich dieses Problem als Optimierungsproblem in einem verallgemeinerten disjunktiven Graphen darstellen lässt. Das entworfene Optimierungsverfahren basiert wiederum auf dem Austausch von kritischen Bogen.

Abschliessend sei darauf hingewiesen, dass disjunktive Präzedenzmodelle (bzw. disjunktive Graphen) einen Spezialfall der Disjunktiven Programmierung darstellen. Disjunktive Präzedenzmodelle weisen im Gegensatz zu allgemeinen Disjunktiven Programmen die spezielle Eigenschaft auf, dass alle linearen Ungleichungen genau zwei Variablen umfassen. Dieser Umstand erlaubt es, die Modelle in Form eines disjunktiven Graphen darzustellen. Eine kurze Einführung in die Disjunktive Programmierung ist gegeben in [Nemhauser, G. L., Wolsey, L. A. (1988)]. Eine fundamentale Arbeit zur Thema bildet der Artikel von [Balas, E. (1979)]. Als weitere Beiträge seien die Artikel von [Balas, E. (1974)] und [Helbig, S. (1994)] erwähnt.

5.2. Definitionen und Eigenschaften

In diesem Abschnitt wird das Konzept der verallgemeinerten disjunktiven Graphen formal definiert, und es werden verschiedene damit verbundene Begriffe eingeführt. Danach werden grundlegende Eigenschaften im Zusammenhang mit disjunktiven Mengen und disjunktiven Subgraphen diskutiert, welche später häufig wiederverwendet werden.

5.2.1. Definitionen

Definition 5.2. Ein (verallgemeinerter) disjunktiver Graph G ist definiert durch

$G = \langle V, A, E, \mathcal{E}, c \rangle$. V ist die Menge der Knoten, A die Menge der konjunktiven Bogen und E die Menge der disjunktiven Bogen, wobei $A \cup E \subseteq \{(v, w) : v, w \in V, v \neq w\}$ und $A \cap E = \emptyset$.

$\mathcal{E} \subseteq \text{Pot}(E)$ ist die Familie der disjunktiven Mengen, wobei $E = \bigcup_{D \in \mathcal{E}} D$ und $|D| \geq 2$ für alle disjunktiven Mengen $D \in \mathcal{E}$. Die Funktion $c : A \cup E \rightarrow \mathbb{R}$ definiert die Bogengewichte.

Für einen disjunktiven Graphen G mit $G = \langle V, A, E, \mathcal{E}, c \rangle$ werden die einzelnen Komponenten auch bezeichnet als $V_G = V$, $A_G = A$, $E_G = E$, $\mathcal{E}_G = \mathcal{E}$ und $c_G = c$.

Ein disjunktiver Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ kann aufgefasst werden als ein gewöhnlicher ("konjunktiver") gewichteter, gerichteter Graph $\langle V, A \cup E, c \rangle$ mit zwei Typen A und E von Bogen, für welchen zusätzlich eine Familie $\mathcal{E} \subseteq \text{Pot}(E)$ von disjunktiven Mengen definiert ist. Für disjunktive Graphen gelten dementsprechend dieselben Begriffe und graphentheoretischen Konzepte wie für gewöhnliche Graphen.

Beachte, dass in der obigen Definition durch die Bedingung $|D| \geq 2$, $D \in \mathcal{E}$ nur gefordert wird, dass eine disjunktive Menge mindestens zwei, aber nicht genau zwei Bogen umfassen muss. Somit werden auch disjunktive Graphen zugelassen, deren disjunktive Mengen mehr als zwei Bogen enthalten. Dies stellt eine zusätzliche Verallgemeinerung des Konzepts der disjunktiven Graphen dar, welche im Rahmen dieser Arbeit aber nicht weiter untersucht wurde. Es ist jedoch so, dass ein Teil der in diesem Kapitel erarbeiteten Theorie auch gültig ist für diesen verallgemeinerten Fall, weshalb er in die Definition miteinbezogen wurde.

Definition 5.3. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Eine Selektion in G ist eine Teilmenge $S \subseteq E$ von disjunktiven Bogen.

Eine disjunktive Menge $D \in \mathcal{E}$ ist geschnitten von S , falls $S \cap D \neq \emptyset$. Die Familie aller von S geschnittenen disjunktiven Mengen wird mit $\mathcal{E}(S) = \{D \in \mathcal{E} : S \cap D \neq \emptyset\}$ bezeichnet.

Eine Selektion $S \subseteq E$ heisst vollständig in G , falls sie alle disjunktiven Mengen schneidet, d.h. falls $\mathcal{E}(S) = \mathcal{E}$, andernfalls partiell. Eine Selektion $S \subseteq E$ heisst (positiv) azyklisch in G , falls der Graph $\langle V, A \cup S, c \rangle$ keine (positiven) Zyklen hat. Eine vollständige, positiv azyklische Selektion in G heisst zulässig in G .

Definition 5.4. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $\mathcal{F} \subseteq \mathcal{E}$ eine Teilfamilie von disjunktiven Mengen. Die Menge aller disjunktiven Bogen, die in einer disjunktiven Menge aus \mathcal{F} enthalten sind, wird mit $\mathcal{F}^\circ = \bigcup_{D \in \mathcal{F}} D$ bezeichnet. $G_{\mathcal{F}}$ bezeichnet den Teilgraphen von G , welcher nur disjunktive Mengen aus \mathcal{F} hat, d.h. $G_{\mathcal{F}} = \langle V, A, \mathcal{F}^\circ, \mathcal{F}, c \rangle$.

Die zu einer Selektion $S \subseteq E$ komplementäre Selektion in G ist definiert als $\tilde{S} = \mathcal{E}(S)^\circ - S$.

Definition 5.5. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$ eine Selektion in G . Der durch S definierte konjunktive Subgraph von G ist gegeben als $G(S) = \langle V, A \cup S, c \rangle$. Der Graph $G(\emptyset) = \langle V, A, c \rangle$ wird der konjunktive Teil von G genannt.

Der durch S definierte disjunktive Subgraph von G ist gegeben als

$$G[S] = \langle V, A \cup S, (\mathcal{E} - \mathcal{E}(S))^\circ, \mathcal{E} - \mathcal{E}(S), c \rangle.$$

Der disjunktive Graph G heisst (positiv) azyklisch, falls sein konjunktiver Teil $G(\emptyset)$ keine (positiven) Zyklen hat, andernfalls heisst G (positiv) zyklisch. Z wird ein Zyklus von G genannt, falls Z ein Zyklus von $G(\emptyset)$ ist.

Definition 5.6. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Falls $|\mathcal{E}(e)| = 1$ für alle $e \in E$, existiert genau eine disjunktive Menge $D \in \mathcal{E}$ mit $e \in D$, welche bezeichnet wird als $D(e)$. Falls zusätzlich $|D| = 2$ für alle $D \in \mathcal{E}$, wird der zweite Bogen $e' \in D(e)$, $e \neq e'$, als der Partner von e bezeichnet.

5.2.2. Eigenschaften

Disjunktive Mengen

Lemma 5.7. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt $S \subseteq \mathcal{E}(S)^\circ$.

Beweis:

Sei $e \in S$. Zeige, dass $e \in \mathcal{E}(S)^\circ$. Da $e \in E$ und $E = \bigcup_{D \in \mathcal{E}} D$, existiert ein $D \in \mathcal{E}$ mit $e \in D$. Da $e \in S$, gilt $D \cap S \neq \emptyset$ und folglich $D \in \mathcal{E}(S)$. Da $\mathcal{E}(S)^\circ = \bigcup_{D' \in \mathcal{E}(S)} D'$, gilt $D \subseteq \mathcal{E}(S)^\circ$ und folglich $e \in \mathcal{E}(S)^\circ$. ♦

Lemma 5.8. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $R, S \subseteq E$. Es gilt:

- (i) $\mathcal{E}(R) \cup \mathcal{E}(S) = \mathcal{E}(R \cup S)$
- (ii) $\mathcal{E}(R)^\circ \cup \mathcal{E}(S)^\circ = \mathcal{E}(R \cup S)^\circ$

Beweis:

Beweis von (i): Zeige einerseits, dass $\mathcal{E}(R) \cup \mathcal{E}(S) \subseteq \mathcal{E}(R \cup S)$. Sei $D \in \mathcal{E}(R) \cup \mathcal{E}(S)$. Dann gilt $D \cap R \neq \emptyset$ oder $D \cap S \neq \emptyset$. Folglich ist $D \cap (R \cup S) \neq \emptyset$, d.h. $D \in \mathcal{E}(R \cup S)$. Zeige andererseits, dass $\mathcal{E}(R \cup S) \subseteq \mathcal{E}(R) \cup \mathcal{E}(S)$. Sei $D \in \mathcal{E}(R \cup S)$. Dann gilt $D \cap (R \cup S) \neq \emptyset$. Folglich ist $D \cap R \neq \emptyset$ oder $D \cap S \neq \emptyset$, d.h. $D \in \mathcal{E}(R) \cup \mathcal{E}(S)$.

Beweis von (ii): Zeige einerseits, dass $\mathcal{E}(R)^\circ \cup \mathcal{E}(S)^\circ \subseteq \mathcal{E}(R \cup S)^\circ$. Sei $e \in \mathcal{E}(R)^\circ \cup \mathcal{E}(S)^\circ$. Fall 1: Sei $e \in \mathcal{E}(R)^\circ$. Dann existiert ein $D \in \mathcal{E}$ mit $e \in D$ und $D \cap R \neq \emptyset$. Da $D \cap (R \cup S) \neq \emptyset$, ist $D \in \mathcal{E}(R \cup S)$. Wegen $e \in D$ gilt folglich $e \in \mathcal{E}(R \cup S)^\circ$. Fall 2: Sei $e \in \mathcal{E}(S)^\circ$. Beweis analog. Zeige andererseits, dass $\mathcal{E}(R \cup S)^\circ \subseteq \mathcal{E}(R)^\circ \cup \mathcal{E}(S)^\circ$. Sei $e \in \mathcal{E}(R \cup S)^\circ$. Dann existiert ein $D \in \mathcal{E}$

mit $e \in D$ und $D \cap (R \cup S) \neq \emptyset$. Somit gilt $D \cap R \neq \emptyset$ oder $D \cap S \neq \emptyset$. Fall 1: Sei $D \cap R \neq \emptyset$. Dann ist $D \in \mathcal{E}(R)$ und wegen $e \in D$ gilt $e \in \mathcal{E}(R)^\circ$. Fall 2: Sei $D \cap S \neq \emptyset$. Beweis analog. ♦

Lemma 5.9. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph mit $|\mathcal{E}(e)| = 1$ für alle $e \in E$. Sei $R, S \subseteq E$. Es gilt:

- (i) $\mathcal{E}(R) - \mathcal{E}(S) = \mathcal{E}(R - \mathcal{E}(S)^\circ)$
- (ii) $\mathcal{E}(R)^\circ - \mathcal{E}(S)^\circ = \mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ$

Beweis:

Beweis von (i): Zeige einerseits, dass $\mathcal{E}(R) - \mathcal{E}(S) \subseteq \mathcal{E}(R - \mathcal{E}(S)^\circ)$. Sei $D \in \mathcal{E}(R) - \mathcal{E}(S)$. Zeige, dass $D \in \mathcal{E}(R - \mathcal{E}(S)^\circ)$. Da $D \in \mathcal{E}(R)$ gilt $D \cap R \neq \emptyset$. Da $D \notin \mathcal{E}(S)$ gilt $D \cap S = \emptyset$. $D \cap R \neq \emptyset$, existiert ein $e \in D$ mit $e \in R$. Zeige, dass $e \notin \mathcal{E}(S)^\circ$. Nehme an, dass $e \in \mathcal{E}(S)^\circ$. Dann existiert ein $D' \in \mathcal{E}$ mit $e \in D'$ und $D' \cap S \neq \emptyset$. Da $e \in D$ und $e \in D'$ und gemäss Voraussetzung $|\mathcal{E}(e)| = 1$, gilt $D = D'$. Da $D' \cap S \neq \emptyset$ gilt $D \cap S \neq \emptyset$. Ein Widerspruch. Somit ist gezeigt, dass $e \notin \mathcal{E}(S)^\circ$. Da $e \in R$ gilt $e \in R - \mathcal{E}(S)^\circ$. Da $e \in D$ gilt $D \cap (R - \mathcal{E}(S)^\circ) \neq \emptyset$. Folglich ist $D \in \mathcal{E}(R - \mathcal{E}(S)^\circ)$.

Zeige andererseits, dass $\mathcal{E}(R - \mathcal{E}(S)^\circ) \subseteq \mathcal{E}(R) - \mathcal{E}(S)$. Sei $D \in \mathcal{E}(R - \mathcal{E}(S)^\circ)$. Zeige, dass $D \in \mathcal{E}(R) - \mathcal{E}(S)$. Da $D \in \mathcal{E}(R - \mathcal{E}(S)^\circ)$ gilt $D \cap (R - \mathcal{E}(S)^\circ) \neq \emptyset$. Also existiert ein $e \in D$ mit $e \in R - \mathcal{E}(S)^\circ$. Es ist $e \in R$ und $e \notin \mathcal{E}(S)^\circ$. Zeige erstens, dass $D \in \mathcal{E}(R)$. Da $e \in D$ und $e \in R$ ist $D \cap R \neq \emptyset$ und somit $D \in \mathcal{E}(R)$. Zeige zweitens, dass $D \notin \mathcal{E}(S)$. Da $e \notin \mathcal{E}(S)^\circ$ existiert kein $D' \in \mathcal{E}$ mit $e \in D'$ und $D' \cap S \neq \emptyset$. Wegen $e \in D$ gilt somit $D \cap S = \emptyset$. Folglich ist $D \notin \mathcal{E}(S)$. Somit ist gezeigt, dass $D \in \mathcal{E}(R) - \mathcal{E}(S)$.

Beweis von (ii): Zeige einerseits, dass $\mathcal{E}(R)^\circ - \mathcal{E}(S)^\circ \subseteq \mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ$. Sei $e \in \mathcal{E}(R)^\circ - \mathcal{E}(S)^\circ$. Zeige, dass $e \in \mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ$. Da $e \in \mathcal{E}(R)^\circ$ existiert ein $D \in \mathcal{E}$ mit $e \in D$ und $D \cap R \neq \emptyset$. Da $e \notin \mathcal{E}(S)^\circ$, existiert kein $D' \in \mathcal{E}$ mit $e \in D'$ und $D' \cap \mathcal{E}(S)^\circ \neq \emptyset$. Daraus folgt wegen $e \in D$ dass $D \cap \mathcal{E}(S)^\circ = \emptyset$. Da $D \cap R \neq \emptyset$ existiert ein $f \in D$ mit $f \in R$. Da $D \cap \mathcal{E}(S)^\circ = \emptyset$ ist $f \notin \mathcal{E}(S)^\circ$. Folglich gilt $f \in R - \mathcal{E}(S)^\circ$. Wegen $f \in D$ folgt daraus, dass $D \cap (R - \mathcal{E}(S)^\circ) \neq \emptyset$. Somit ist $D \in \mathcal{E}(R - \mathcal{E}(S)^\circ)$. Da $e \in D$ gilt folglich $e \in \mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ$.

Zeige andererseits, dass $\mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ \subseteq \mathcal{E}(R)^\circ - \mathcal{E}(S)^\circ$. Sei $e \in \mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ$. Zeige, dass $e \in \mathcal{E}(R)^\circ - \mathcal{E}(S)^\circ$. Da $e \in \mathcal{E}(R - \mathcal{E}(S)^\circ)^\circ$ existiert ein $D \in \mathcal{E}$ mit $e \in D$ und $D \cap (R - \mathcal{E}(S)^\circ) \neq \emptyset$. Also existiert ein $f \in D$ mit $f \in R - \mathcal{E}(S)^\circ$. Es ist $f \in R$ und $f \notin \mathcal{E}(S)^\circ$. Zeige erstens, dass $e \in \mathcal{E}(R)^\circ$. Da $f \in D$ und $f \in R$ gilt $D \cap R \neq \emptyset$. Folglich ist $D \in \mathcal{E}(R)$. Daraus folgt wegen $e \in D$, dass $e \in \mathcal{E}(R)^\circ$. Zeige zweitens, dass $e \notin \mathcal{E}(S)^\circ$. Da $f \notin \mathcal{E}(S)^\circ$ existiert kein $D' \in \mathcal{E}$ mit $f \in D'$ und $D' \cap \mathcal{E}(S)^\circ \neq \emptyset$. Da $f \in D$ gilt somit $D \cap \mathcal{E}(S)^\circ = \emptyset$. Daraus folgt wegen $e \in D$ dass $e \notin \mathcal{E}(S)^\circ$. Somit ist gezeigt, dass $e \in \mathcal{E}(R)^\circ - \mathcal{E}(S)^\circ$.

Beachte, dass für den Beweis von (ii) die Bedingung $|\mathcal{E}(e)| = 1$ nicht benötigt wurde. ♦

Disjunktive Subgraphen

Lemma 5.10. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $R \subseteq E$ eine Selektion in G und $S \subseteq E_{G[R]}$ eine Selektion in $G[R]$. Es gilt:

$$(i) \quad G[R][S] = G[R \cup S]$$

$$(ii) \quad G[R](S) = G(R \cup S)$$

Beweis:

Beweis von (i): Sei $G[R] = \langle V, A^0, E^0, \mathcal{E}^0, c \rangle$ $G[R][S] = \langle V, A^1, E^1, \mathcal{E}^1, c \rangle$ und $G[R \cup S] = \langle V, A^2, E^2, \mathcal{E}^2, c \rangle$. Zeige, dass $G[R][S] = G[R \cup S]$. Gemäss Definition ist $A^0 = A \cup R$ und $\mathcal{E}^0 = \mathcal{E} - \mathcal{E}(R)$. Folglich ist $A^1 = A^0 \cup S = A \cup R \cup S$ und $\mathcal{E}^1 = \mathcal{E}^0 - \mathcal{E}^0(S) = (\mathcal{E} - \mathcal{E}(R)) - \mathcal{E}^0(S)$. Nun gilt:

$$\mathcal{E}^0(S) = \{D \in \mathcal{E} - \mathcal{E}(R) : D \cap S \neq \emptyset\} = \{D \in \mathcal{E} : D \cap S \neq \emptyset\} - \mathcal{E}(R) = \mathcal{E}(S) - \mathcal{E}(R)$$

Folglich ist $\mathcal{E}^1 = (\mathcal{E} - \mathcal{E}(R)) - (\mathcal{E}(S) - \mathcal{E}(R)) = \mathcal{E} - \mathcal{E}(R) - \mathcal{E}(S) = \mathcal{E} - (\mathcal{E}(R) \cup \mathcal{E}(S))$. Gemäss Lemma 5.8. ist $\mathcal{E}(R) \cup \mathcal{E}(S) = \mathcal{E}(R \cup S)$. Somit ist $\mathcal{E}^1 = \mathcal{E} - \mathcal{E}(R \cup S)$. Gemäss Definition ist $A^2 = A \cup (R \cup S)$ und $\mathcal{E}^2 = \mathcal{E} - \mathcal{E}(R \cup S)$. Folglich ist $A^1 = A^2$ und $\mathcal{E}^1 = \mathcal{E}^2$. Da $E^1 = (\mathcal{E}^1)^\circ$ und $E^2 = (\mathcal{E}^2)^\circ$, gilt $E^1 = E^2$. Somit ist gezeigt, dass $G[R][S] = G[R \cup S]$.

Beweis von (ii): Gemäss Definition ist $G[R] = \langle V, A \cup R, (\mathcal{E} - \mathcal{E}(R))^\circ, \mathcal{E} - \mathcal{E}(R), c \rangle$ und folglich $(G[R])(S) = \langle V, A \cup R \cup S, c \rangle = G(R \cup S)$. ♦

Lemma 5.11. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $R \subseteq E$.

- (i) Falls S eine zulässige Selektion in G ist mit $R \subseteq S$, dann ist $S \cap E_{G[R]}$ eine zulässige Selektion in $G[R]$.
- (ii) Falls $S \subseteq E_{G[R]}$ eine zulässige Selektion in $G[R]$ ist, dann ist $R \cup S$ eine zulässige Selektion in G .

Beweis:

Beweis von (i): Sei S eine zulässige Selektion in G . Zeige, dass $S' = S \cap E_{G[R]}$ eine zulässige Selektion in $G[R]$ ist. Da $S' \subseteq E_{G[R]}$ ist S' eine Selektion in $G[R]$. Zeige erstens, dass S' positiv azyklisch ist in $G[R]$. Da S zulässig ist in G , ist $G(S)$ positiv azyklisch. Da $R \cup S' \subseteq S$, ist auch $G(R \cup S')$ positiv azyklisch. Gemäss Lemma 5.10. ist $G(R \cup S') = G[R](S')$. Folglich ist $G[R](S')$ positiv azyklisch, d.h. S' ist positiv azyklisch in $G[R]$. Zeige zweitens, dass S' voll-

ständig ist in $G[R]$. Nehme an, dass ein $D \in \mathcal{E}_{G[R]}$ existiert mit $D \cap S' = \emptyset$. Da G vollständig ist in G , gilt $D \cap S \neq \emptyset$. Sei $e \in D \cap S$. Da $e \in D$ und $D \in \mathcal{E}_{G[R]}$ gilt $e \in E_{G[R]}$. Folglich gilt $e \in S'$ und somit $D \cap S' \neq \emptyset$. Ein Widerspruch.

Beweis von (ii): Sei $S \subseteq E_{G[R]}$ eine zulässige Selektion in $G[R]$. Zeige, dass $S' = R \cup S$ eine zulässige Selektion in G ist. Da $S' \subseteq E$ ist S' eine Selektion in G . Zeige erstens, dass S' positiv azyklisch ist in G . Da S positiv azyklisch ist in $G[R]$ ist $G[R](S)$ positiv azyklisch. Gemäss Lemma 5.10. ist $G[R](S) = G(R \cup S)$. Also ist $G(R \cup S)$ positiv azyklisch, d.h. S' ist positiv azyklisch in G . Zeige zweitens, dass S' vollständig ist in G . Nehme an, dass ein $D \in \mathcal{E}$ existiert mit $D \cap S' = \emptyset$. Fall 1: Sei $D \in \mathcal{E}(R)$. Dann gilt $D \cap R \neq \emptyset$. Da $R \subseteq S'$ gilt auch $D \cap S' \neq \emptyset$. Ein Widerspruch. Fall 2: Sei $D \in \mathcal{E} - \mathcal{E}(R)$. Da $\mathcal{E}_{G[R]} = \mathcal{E} - \mathcal{E}(R)$ und S vollständig ist in $G[R]$ gilt $D \cap S \neq \emptyset$. Da $S \subseteq S'$, gilt auch $D \cap S' \neq \emptyset$. Ein Widerspruch. ♦

Lemma 5.12. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph mit $|\mathcal{E}(e)| = 1$ für alle $e \in E$. Sei $S \subseteq E$ und $G[S]$ der entsprechende disjunktive Subgraph von G . Es gilt:

- (i) $\mathcal{E}_{G[S]} = \mathcal{E} - \mathcal{E}(S)$
- (ii) $E_{G[S]} = E - \mathcal{E}(S)^\circ$

Beweis:

Beweis von (i): Gemäss Definition gilt $\mathcal{E}_{G[S]} = \mathcal{E} - \mathcal{E}(S)$.

Beweis von (ii): Da $\mathcal{E} = \mathcal{E}(E)$ ist $\mathcal{E}_{G[S]} = \mathcal{E}(E) - \mathcal{E}(S)$. Gemäss Lemma 5.9. gilt $\mathcal{E}_{G[S]} = \mathcal{E}(E - \mathcal{E}(S)^\circ)$. Gemäss Definition ist $E_{G[S]} = (\mathcal{E}_{G[S]})^\circ$ und folglich $E_{G[S]} = \mathcal{E}(E - \mathcal{E}(S)^\circ)^\circ$. Aufgrund von Lemma 5.9. gilt $\mathcal{E}(E - \mathcal{E}(S)^\circ)^\circ = \mathcal{E}(E)^\circ - \mathcal{E}(S)^\circ$. Da $\mathcal{E}(E)^\circ = \mathcal{E}^\circ = E$, gilt somit $E_{G[S]} = E - \mathcal{E}(S)^\circ$. ♦

5.3. Spezielle Klassen von disjunktiven Graphen

In diesem Abschnitt werden verschiedene Klassen von disjunktiven Graphen beschrieben, welche im Rahmen dieser Arbeit von Interesse sind: Klassische disjunktive Graphen, Job-Shop-Graphen, Lager-Graphen und 1-Job-Lager-Graphen. Die verschiedenen Klassen werden anhand eines Beispiels illustriert.

Klassische disjunktive Graphen sind disjunktive Graphen, bei denen jede disjunktive Menge zwei Bogen mit denselben Endpunkten, aber entgegengesetzter Richtung umfasst.

Job-Shop-Graphen sind disjunktive Graphen, welche aus der Modellierung des Job-Shop-Problems mittels disjunktiver Präzedenzbedingungen resultieren. Job-Shop-Graphen bilden einen Spezialfall der klassischen disjunktiven Graphen. Bis anhin wurden in der Literatur im Zusammenhang mit disjunktiven Graphen praktisch ausschliesslich Job-Shop-Graphen diskutiert.

Lager-Graphen sind disjunktive Graphen, welche aus der Modellierung des Lager-Problems mittels disjunktiver Präzedenzbedingungen resultieren. Lager-Graphen bilden die Basis für die Modellierung und Optimierung des Lagerproblems, d.h. des Steuerungsproblems in automatisierten Kompaktlagern (siehe Kapitel 6).

1-Job-Lager-Graphen werden ausgehend von einem Lager-Graphen (mit Präselektion) konstruiert, indem ein einzelner Job ausgewählt wird und alle disjunktiven Bogen, welche nicht mit einem Knoten dieses Jobs inzidieren, aus dem Graphen entfernt werden. 1-Job-Lager-Graphen bilden eine Relaxation der Lager-Graphen (mit Präselektion) in dem Sinne, dass nur die disjunktiven Präzedenzbedingungen bezüglich eines einzelnen Jobs berücksichtigt werden und die restlichen disjunktiven Bedingungen weggelassen werden. 1-Job-Lager-Graphen sind von wichtiger Bedeutung beim Entwurf eines Optimierungsverfahrens für Lager-Graphen.

Zu den folgenden Abbildungen von disjunktiven Graphen ist zu bemerken: Konjunktive Bogen werden im allgemeinen mit einem ausgezogenen Pfeil und disjunktive Bogen mit einem gestrichelten Pfeil dargestellt. Häufig wird zur Vereinfachung nur einer der beiden Bogen einer disjunktiven Menge eingezeichnet. Manchmal werden die disjunktiven Bogen der Uebersichtlichkeit halber auch ganz weggelassen. Der Partner eines eingezeichneten disjunktiven Bogens wird im allgemeinen mit dessen Namen und einem Apostroph gekennzeichnet (z.B. a_1 und a_1').

Bei der Darstellung von Job-Shop- und Lager-Graphen werden die Start- und Endknoten σ und τ sowie die damit inzidierenden Start- und Endbogen zur Vereinfachung meistens weggelassen. Die Operationen werden so bezeichnet, dass für die i -te Operation von Job J_n , d.h. für J_{ni} geschrieben wird $n.i$. In Lager-Graphen wird das einer Operation $n.i$ zugeordnete th -Knotenpaar bezeichnet als $n.i^t$ und $n.i^h$. Weiter werden in Lager-Graphen zwei Synchronisationsbogen entgegengesetzter Richtung meistens als einzelne ungerichtete Linie eingezeichnet und die th -Bogen mit einer doppelten Pfeilspitze gekennzeichnet.

In Analogie zu den Gantt-Diagrammen werden die Knoten, welche sich auf Operationen beziehen, die auf demselben Prozessor ausgeführt werden, horizontal auf derselben Linie gezeichnet. In gleicher Analogie werden die Graphen so dargestellt, dass alle konjunktiven Bogen von links nach rechts weisen. Dadurch kann die Horizontale als zeitliche Dimension aufgefasst werden, wobei die durch die Knoten repräsentierten Ereignisse um so später stattfinden, je weiter rechts sich der Knoten befindet.

5.3.1. Klassische disjunktive Graphen

Definition 5.13. Ein klassischer disjunktiver Graph ist ein disjunktiver Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$, dessen disjunktive Mengen genau zwei Bogen mit denselben Endpunkten, aber umgekehrter Richtung umfassen. Es gilt also für alle $D \in \mathcal{E}$ mit $D = \{e, e'\}$, dass $\text{tail}(e) = \text{head}(e')$ und $\text{head}(e) = \text{tail}(e')$.

Beachte noch, dass in klassischen disjunktiven Graphen jeder disjunktive Bogen in genau einer disjunktiven Menge vorkommt, d.h. dass $|\mathcal{E}(e)| = 1$ für alle $e \in E$. Dies folgt aus der Definition, da alle disjunktiven Mengen verschieden sind und jede disjunktive Menge zwei Bogen entgegengesetzter Richtung umfasst.

Beispiel:

Die folgende Abbildung zeigt einen klassischen disjunktiven Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$. Die Knotenmenge entspricht $V = \{1, 2, 3, 4\}$. Die Menge der konjunktiven Bogen ist gegeben durch $A = \{(1, 3), (1, 4), (4, 2)\}$ und die Menge der disjunktiven Bogen durch $E = \{(1, 2), (2, 1), (3, 4), (4, 3)\}$. Die Familie der disjunktiven Mengen ist gegeben durch $\mathcal{E} = \{\{(1, 2), (2, 1)\}, \{(3, 4), (4, 3)\}\}$. ♦

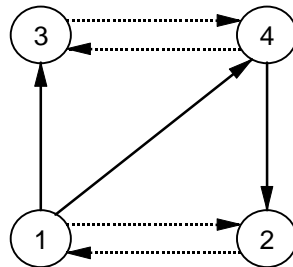


Abbildung 5.1. Klassischer disjunktiver Graph

5.3.2. Job-Shop-Graphen

In Abschnitt 4.2. wurde gezeigt, wie sich das Job-Shop-Problem mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen modellieren lässt. Es resultierte das folgende disjunktive Präzedenzmodell:

Job-Shop-Problem $\Pi = (M, I, \mathcal{J}, \mu, d)$:

$$\min x_{\tau}$$

$$\begin{aligned}
x_j - x_i &\geq d_i, & (i, j) \in \{(J_r, J_{r+1}) : J \in \mathcal{J}, r = 1, \dots, |J| - 1\} \\
x_i - x_\sigma &\geq 0, & i \in I \\
x_\tau - x_i &\geq d_i, & i \in I \\
x_j - x_i &\geq d_i \quad \text{OR} \quad x_i - x_j \geq d_j, & \{i, j\} \subseteq I_m, m \in M \\
x_i &\geq 0, & i \in I \cup \{\sigma, \tau\}
\end{aligned}$$

Die untenstehende Definition zeigt, wie der disjunktive Job-Shop-Graph zu diesem Modell konstruiert wird.

Definition 5.14. Sei $\Pi = (M, I, \mathcal{J}, \mu, d)$ ein Job-Shop-Problem. Der zu Π gehörende Job-Shop-Graph entspricht dem folgenden disjunktiven Graphen $G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$:

- (i) Die Knotenmenge ist gegeben durch $V = V_I \cup \{\sigma, \tau\}$. Die Menge V_I entspricht der Menge der Operationen, d.h. $V_I = I$. σ ist ein künstlicher Start- und τ ein künstlicher Endknoten.
- (ii) Die Menge der konjunktiven Bogen ist gegeben durch $A = A_0 \cup A_\sigma \cup A_\tau$. Die Menge A_0 enthält für jedes Paar von aufeinanderfolgenden Operationen eines Jobs einen Bogen von der vorausgehenden zur nachfolgenden Operation, d.h. $A_0 = \{(J_r, J_{r+1}) : J \in \mathcal{J}, r = 1, \dots, |J| - 1\}$. Die Bogen in A_0 werden als Job-Präzedenzbogen bezeichnet. Die Menge A_σ enthält für jede Operation einen Bogen vom Startknoten zur Operation, d.h. $A_\sigma = \{(\sigma, i) : i \in I\}$. Die Bogen in A_σ werden als Startbogen bezeichnet. Die Menge A_τ enthält für jede Operation einen Bogen von der Operation zum Endknoten, d.h. $A_\tau = \{(i, \tau) : i \in I\}$. Die Bogen in A_τ werden als Endbogen bezeichnet.
- (iii) Die Menge der disjunktiven Bogen ist gegeben durch $E = \bigcup_{m \in M} E_m$. Die Menge E_m , $m \in M$, enthält für jedes Paar von unterschiedlichen Operationen auf Prozessor m zwei Bogen, welche die beiden Operationen gegenseitig verbinden, d.h. $E_m = \{(i, j), (j, i) : \{i, j\} \subseteq I_m\}$. Die Bogen in E_m werden als Prozessor-Präzedenzbogen von m bezeichnet.
- (iv) Die Familie der disjunktiven Mengen ist gegeben durch $\mathcal{E} = \bigcup_{m \in M} \mathcal{E}_m$. Die Familie \mathcal{E}_m , $m \in M$, enthält für jede Auswahl von zwei unterschiedlichen Operationen auf Prozessor m eine disjunktive Menge mit zwei Bogen, welche die beiden Operationen gegenseitig verbinden, d.h. $\mathcal{E}_m = \{\{(i, j), (j, i)\} : \{i, j\} \subseteq I_m\}$.
- (v) Die Bogengewichte sind folgendermassen definiert: Für $e = (i, j) \in A_0$ ist $c_e = d_i$. Für $e = (\sigma, i) \in A_\sigma$ ist $c_e = 0$, und für $e = (i, \tau) \in A_\tau$ ist $c_e = d_i$. Für $e = (i, j) \in E$ ist $c_i = d_i$.

Beispiel:

Es wird ein Job-Shop-Problem $\Pi = (M, I, \mathcal{J}, \mu, d)$ mit 3 Prozessoren $M = \{P_1, P_2, P_3\}$ und 3 Jobs $\mathcal{J} = \{J_1, J_2, J_3\}$ gezeigt. Die Operationen der verschiedenen Jobs werden der Reihe nach auf folgenden Prozessoren ausgeführt:

Job:	Operation 1:	Operation 2:	Operation 3:
J_1	P_1	P_2	P_3
J_2	P_1	P_2	
J_3	P_1	P_3	

Abbildung 5.2. Prozessorreihenfolge der verschiedenen Jobs

Der Job-Shop-Graph $G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ hat folgende Form:

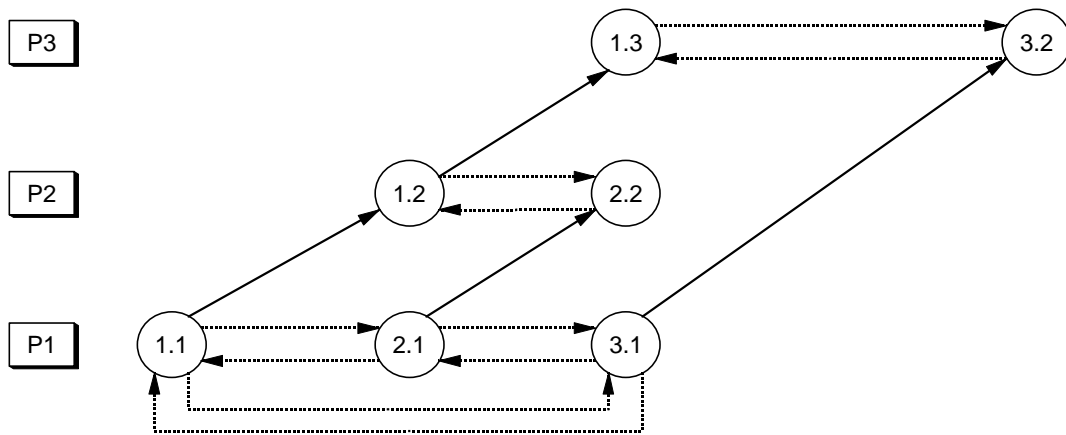


Abbildung 5.3. Beispiel eines Job-Shop-Graphen

Die Menge der konjunktiven und disjunktiven Bogen sowie die Familie der disjunktiven Mengen sind wie folgt gegeben:

- $A_0 = \{(1.1, 1.2), (1.2, 1.3), (2.1, 2.2), (3.1, 3.2)\}$
- $A_\sigma = \{(\sigma, 1.1), (\sigma, 1.2), (\sigma, 1.3), (\sigma, 2.1), (\sigma, 2.2), (\sigma, 3.1), (\sigma, 3.2)\}$
- $A_\tau = \{(1.1, \tau), (1.2, \tau), (1.3, \tau), (2.1, \tau), (2.2, \tau), (3.1, \tau), (3.2, \tau)\}$
- $A = A_0 \cup A_\sigma \cup A_\tau$
- $\mathcal{E}_1 = \{\{(1.1, 2.1), (2.1, 1.1)\}, \{(2.1, 3.1), (3.1, 2.1)\}, \{(1.1, 3.1), (3.1, 1.1)\}\}$
- $\mathcal{E}_2 = \{\{(1.2, 2.2), (2.2, 1.2)\}\}$
- $\mathcal{E}_3 = \{\{(1.3, 3.2), (3.2, 1.3)\}\}$
- $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3$
- $E = \mathcal{E}^\circ$ ♦

5.3.3. Lager-Graphen

In Abschnitt 4.3. wurde gezeigt, wie sich das Lager-Problem mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen modellieren lässt. Daraus resultierte das folgende disjunktive Präzedenzmodell:

Lager-Problem $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^s)$:

$$\begin{aligned}
 & \max x_\tau \\
 & x_{t(j)} - x_{h(i)} \geq 0, x_{t(i)} - x_{h(j)} \geq 0, \quad (i, j) \in \{(J_r, J_{r+1}) : J \in \mathcal{J}, r = 1, \dots, |J| - 1\} \\
 & x_{h(i)} - x_{t(i)} \geq d_i^t + d_i, \quad i \in I \\
 & x_{t(i)} - x_\sigma \geq d_{\sigma i}^s, \quad i \in I \\
 & x_\tau - x_{h(i)} \geq d_j^h + d_{j\tau}^s, \quad i \in I \\
 & x_{t(j)} - x_{h(i)} \geq d_i^h + d_{ij}^s \text{ OR } x_{t(i)} - x_{h(j)} \geq d_j^h + d_{ji}^s, \quad \{i, j\} \subseteq I_m, m \in M \\
 & x_i \geq 0, \quad i \in I \cup \{\sigma, \tau\}
 \end{aligned}$$

Definition 5.15. Sei $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^s)$ ein Lager-Problem. Der zu Π gehörende Lager-Graph entspricht dem folgenden disjunktiven Graphen $G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$:

(i) Die Knotenmenge ist gegeben durch $V = V_I \cup \{\sigma, \tau\}$.

Die Menge V_I enthält zu jeder Operation $i \in I$ ein Knotenpaar t_i, h_i , d.h. $V_I = \{t_i, h_i : i \in I\}$. Hierbei wird t_i als t -Knoten und h_i als h -Knoten von i bezeichnet. σ ist ein künstlicher Start- und τ ein künstlicher Endknoten.

(ii) Die Menge der konjunktiven Bogen ist gegeben durch $A = A_0 \cup A_1 \cup A_\sigma \cup A_\tau$.

Die Menge A_0 enthält für jedes Paar von aufeinanderfolgenden Operationen eines Jobs zwei entgegengesetzte Bogen, welche den h -Knoten der vorausgehenden mit dem t -Knoten der nachfolgenden Operation verbinden, d.h.

$A_0 = \{(h_{J(r)}, t_{J(r+1)}), (t_{J(r+1)}, h_{J(r)}) : J \in \mathcal{J}, r = 1, \dots, |J| - 1\}$. Die Bogen in A_0 werden als Synchronisationsbogen bezeichnet.

Die Menge A_1 enthält für jedes th -Knotenpaar einer Operation einen Bogen, welcher den t - mit dem h -Knoten verbindet, d.h. $A_1 = \{(t_i, h_i) : i \in I\}$. Die Bogen in A_1 werden als th -Bogen bezeichnet.

Die Menge A_σ enthält für jede Operation einen Bogen vom Startknoten zum t -Knoten der Operation, d.h. $A_\sigma = \{(\sigma, t_i) : i \in I\}$. Die Bogen in A_σ werden als Startbogen bezeichnet.

Die Menge A_τ enthält für jede Operation einen Bogen vom h -Knoten der Operation zum Endknoten, d.h. $A_\tau = \{(h_i, \tau) : i \in I\}$. Die Bogen in A_τ werden als Endbogen bezeichnet.

(iii) Die Menge der disjunktiven Bogen ist gegeben durch $E = \bigcup_{m \in M} E_m$.

Die Menge E_m , $m \in M$, enthält für jedes Paar von unterschiedlichen Operationen auf Prozessor

m zwei Bogen, welche den h -Knoten der einen mit dem t -Knoten der andern Operation verbindet, d.h. $E_m = \{(h_i, t_j), (h_j, t_i) : \{i, j\} \subseteq I_m\}$. Die Bogen in E_m werden als Prozessor-Präzedenzbogen von m bezeichnet.

(iv) Die Familie der disjunktiven Mengen ist gegeben durch $\mathcal{E} = \bigcup_{m \in M} \mathcal{E}_m$.

Die Familie \mathcal{E}_m , $m \in M$, enthält für jede Auswahl von zwei unterschiedlichen Operationen auf Prozessor m eine disjunktive Menge mit zwei Bogen, welche je den h - mit dem t -Knoten der beiden Operationen verbinden, d.h. $\mathcal{E}_m = \{\{(h_i, t_j), (h_j, t_i)\} : \{i, j\} \subseteq I_m\}$.

(v) Die Bogengewichte sind folgendermassen definiert: Für $e \in A_0$ ist $c_e = 0$. Für $e = (t_i, h_i) \in A_1$ ist $c_e = d_i^t + d_i$. Für $e = (\sigma, t_i) \in A_\sigma$ ist $c_e = d_{\sigma i}^s$ und für $e = (h_i, \tau) \in A_\tau$ ist $c_e = d_i^h + d_{i\tau}^s$. Für $e = (h_i, t_j) \in E$ ist $c_e = d_i^h + d_{ij}^s$.

In der untenstehenden Definition werden einige weitere Bezeichnungen eingeführt, welche im Zusammenhang mit Lager-Graphen verwendet werden.

Definition 5.16. Sei $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^s)$ ein Lager-Problem. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ der zugehörige Lager-Graph mit Präselektion Q .

- (i) Sei $i \in I$ eine Operation von Π . Die Menge der (disjunktiven) t - bzw. h -Bogen von i ist definiert als $E_i^t = \{e \in E : \text{head}(e) = t_i\}$ bzw. $E_i^h = \{e \in E : \text{tail}(e) = h_i\}$. Die Menge aller (disjunktiven) Bogen von i ist gegeben durch $E_i = E_i^t \cup E_i^h$. Die Familie der disjunktiven Mengen von i ist definiert als $\mathcal{E}_i = \{D \in \mathcal{E} : D \subseteq E_i\}$.
- (ii) Sei $J \in \mathcal{J}$ ein Job von Π . Die Menge der (disjunktiven) t - bzw. h -Bogen von J ist definiert als $E_J^t = \bigcup_{i \in J} E_i^t$ bzw. $E_J^h = \bigcup_{i \in J} E_i^h$. Die Menge aller (disjunktiven) Bogen von J ist gegeben durch $E_J = E_J^t \cup E_J^h$. Die Familie der disjunktiven Mengen von J ist definiert als $\mathcal{E}_J = \bigcup_{i \in J} \mathcal{E}_i$.
- (iii) Sei $v \in V - \{\sigma, \tau\}$ ein Knoten von G . Die zu v gehörende Operation von Π wird bezeichnet als i_v , d.h. $i_v = i$, falls $v = t_i$ oder $v = h_i$ ist. Der zu v gehörende Job von Π wird bezeichnet als J_v , d.h. $J_v = J$, falls $i_v \in J$.

Für die oben definierten Mengen gelten unter anderem die folgenden Zusammenhänge:

$$\begin{aligned} E_i^t &= \{(h_j, t_i) \in E : j \in I_{\mu(i)}, j \neq i\} \text{ und } E_i^h = \{(h_i, t_j) \in E : j \in I_{\mu(i)}, j \neq i\} \\ E_i &= \{(h_j, t_i), (h_i, t_j) \in E : j \in I_{\mu(i)}, j \neq i\} \\ \mathcal{E}_i &= \{\{(h_j, t_i), (h_i, t_j)\} \in \mathcal{E} : j \in I_{\mu(i)}, j \neq i\} \end{aligned}$$

Beispiel:

Es wird ein Lager-Problem $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^S)$ mit 3 Prozessoren $M = \{P_1, P_2, P_3\}$ und 3 Jobs $\mathcal{J} = \{J_1, J_2, J_3\}$ gezeigt. Die Operationen der verschiedenen Jobs werden der Reihe nach auf folgenden Prozessoren ausgeführt:

Job:	Operation 1:	Operation 2:	Operation 3:
J_1	P_1	P_2	P_3
J_2	P_1	P_2	
J_3	P_1	P_3	

Abbildung 5.4. Prozessorreihenfolge der verschiedenen Jobs

Der Lager-Graph $G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ hat folgende Form:

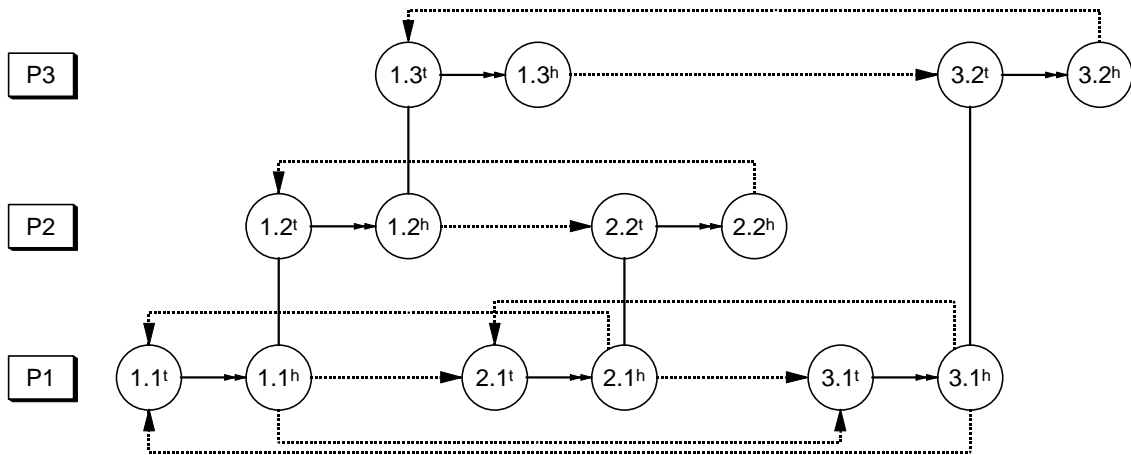


Abbildung 5.5. Beispiel eines Lager-Graphen

Die Menge der konjunktiven und disjunktiven Bogen sowie die Familie der disjunktiven Mengen sind wie folgt gegeben:

- $A_0 = \{(1.1^h, 1.2^t), (1.2^t, 1.1^h), (1.2^h, 1.3^t), (1.3^t, 1.2^h), (2.1^h, 2.2^t), (2.2^t, 2.1^h), (3.1^h, 3.2^t), (3.2^t, 3.1^h)\}$
- $A_1 = \{(1.1^t, 1.1^h), (1.2^t, 1.2^h), (1.3^t, 1.3^h), (2.1^t, 2.1^h), (2.2^t, 2.2^h), (3.1^t, 3.1^h)\}$
- $A_\sigma = \{(\sigma, 1.1^t), (\sigma, 1.2^t), (\sigma, 1.3^t), (\sigma, 2.1^t), (\sigma, 2.2^t), (\sigma, 3.1^t), (\sigma, 3.2^t)\}$
- $A_\tau = \{(1.1^h, \tau), (1.2^h, \tau), (1.3^h, \tau), (2.1^h, \tau), (2.2^h, \tau), (3.1^h, \tau), (3.2^h, \tau)\}$
- $A = A_0 \cup A_1 \cup A_\sigma \cup A_\tau$
- $\mathcal{E}_1 = \{\{(1.1^h, 2.1^t), (2.1^h, 1.1^t)\}, \{(2.1^h, 3.1^t), (3.1^h, 2.1^t)\}, \{(1.1^h, 3.1^t), (3.1^h, 1.1^t)\}\}$

- $\mathcal{E}_2 = \{ \{ (1.2^h, 2.2^t), (2.2^h, 1.2^t) \} \}$
- $\mathcal{E}_3 = \{ \{ (1.3^h, 3.2^t), (3.2^h, 1.3^t) \} \}$
- $\mathcal{E} = \mathcal{E}_1 \cup \mathcal{E}_2 \cup \mathcal{E}_3$
- $E = \mathcal{E}^\circ \quad \blacklozenge$

5.3.4. 1-Job-Lager-Graphen

1-Job-Lager-Graphen werden ausgehend von einem Lager-Graphen (mit Präselektion) konstruiert, indem ein einzelner Job J ausgewählt wird und alle disjunktiven Bogen, welche nicht mit diesem Job inzidieren, aus dem Lager-Graphen entfernt werden. In einem 1-Job-Lager-Graphen inzidieren somit alle disjunktiven Bogen mit einem bestimmten Job J . 1-Job-Lager-Graphen bilden eine Relaxation der Lager-Graphen (mit Präselektion) in dem Sinne, dass nur die disjunktiven Präzedenzbedingungen bezüglich eines einzelnen Jobs J berücksichtigt werden und die restlichen disjunktiven Bedingungen weggelassen werden.

Definition 5.17. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei J ein Job von Π . Der durch J definierte 1-Job-Lager-Graph von G ist gegeben durch $G_J = \langle V, A, \mathcal{E}_J^\circ, \mathcal{E}_J, c \rangle$.

Beispiel:

Betrachte den Lager-Graphen G^Π aus dem vorangehenden Beispiel. Sei $Q = \{ (1.1^h, 2.1^t), (1.3^h, 3.2^t) \}$ eine vorgegebene Präselektion und $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ der entsprechende disjunktive Subgraph. Sei $J = J_2$. Der durch J definierte 1-Job-Lager-Graph $G_J = \langle V, A, \mathcal{E}_J^\circ, \mathcal{E}_J, c \rangle$ hat die folgende Form:

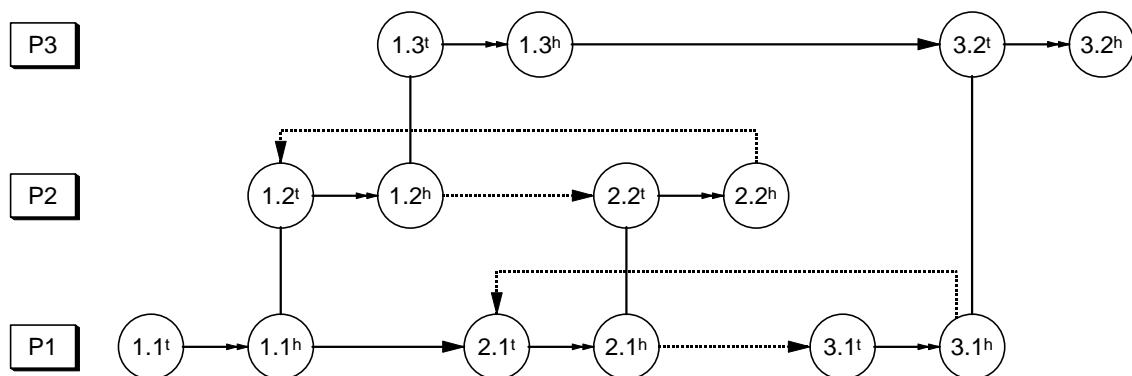


Abbildung 5.6. Beispiel eines 1-Job-Lager-Graphen

Die konjunktiven Bogenmengen A_0 , A_1 , A_σ und A_τ sind gleich wie im vorangehenden Beispiel. Die Menge der konjunktiven Bogen ist gegeben durch $A = A_0 \cup A_1 \cup A_\sigma \cup A_\tau$. Die Menge der disjunktiven Bogen entspricht \mathcal{E}_j° und die Familie der disjunktiven Mengen ist gegeben durch $\mathcal{E}_j = \{ \{ (1.2^h, 2.2^t), (2.2^h, 1.2^t) \}, \{ (2.1^h, 3.1^t), (3.1^h, 2.1^t) \} \}$ ♦

5.4. Konjunktive Erweiterung

5.4.1. Einführung

Erzwungene und verbotene Bogen

Bei der Betrachtung disjunktiver Graphen stellt man fest, dass die Auswahl gewisser disjunktiver Bogen die Wahl bzw. Nichtwahl gewisser anderer disjunktiver Bogen implizieren kann. Dies ist folgendermassen zu verstehen: Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$ eine Auswahl von disjunktiven Bogen. S impliziert die Wahl eines Bogens $e \in E$ in G wenn alle zulässigen Selektionen S' , welche S enthalten, auch den Bogen e enthalten. S impliziert die Nichtwahl eines Bogens $e \in E$, wenn alle zulässigen Selektionen, welche S enthalten, den Bogen e nicht enthalten.

Beispiel:

Gegeben sei der disjunktive Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ in Abbildung 5.7., wobei $A = \{a, b, c, d\}$ $\mathcal{E} = \{ \{e, e'\}, \{f, f'\}, \{g, g'\} \}$ und $c > 0$.

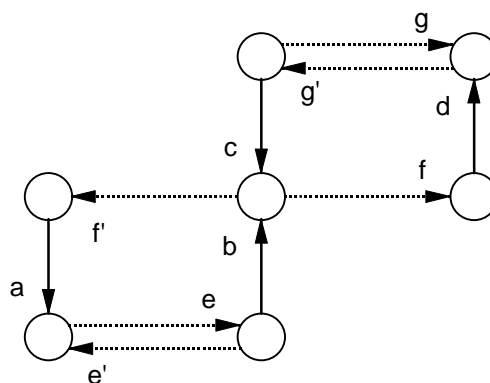


Abbildung 5.7. Verbotene und erzwungene Bogen

Betrachte die Auswahl $S = \{e\}$. Da e in jeder zulässigen Selektion ist, welche S enthält, impliziert S trivialerweise die Wahl von e . Da e' zusammen mit e den positiven Zyklus (e, e') bildet, kann e' in keiner zulässigen Selektion sein, welche S enthält. Somit impliziert S die Nichtwahl von e' . Da f' zusammen mit e den positiven Zyklus (a, e, b, f') bildet, impliziert S die Nichtwahl von f' . Da jede zulässige Selektion in G mindestens einen Bogen der disjunktiven Menge $\{f, f'\}$ enthält, muss folglich f in jeder zulässigen Selektion in G enthalten sein, d.h. S impliziert die Wahl von f . Da g' zusammen mit f den Zyklus (c, f, d, g') bildet, impliziert S auch die Nichtwahl von g' . Da jede zulässige Selektion mindestens einen Bogen der disjunktiven Menge $\{g, g'\}$ enthält, impliziert S weiter die Wahl von g . Zusammengefasst impliziert S somit die Nichtwahl von $\{e', f', g'\}$ und die Wahl von $\{e, f, g\}$.

Die Menge aller zulässigen Selektionen in G ist gegeben durch $\{\{e, f, g\}, \{e', f, g\}, \{e', f', g\}, \{e', f', g'\}\}$. Somit gilt tatsächlich für alle zulässigen Selektionen S' mit $S \subseteq S'$, dass $S' \cap \{e', f', g'\} = \emptyset$ und $\{e, f, g\} \subseteq S'$ ♦

Im vorangehenden Beispiel wurde gezeigt, wie anhand verschiedener Ueberlegungen eine Menge von Bogen bestimmt werden kann, deren Nichtwahl bzw. Wahl durch S impliziert wird. Diese Ueberlegungen lassen sich zu folgenden drei Punkten zusammenfassen: Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. (1) S impliziert die Nichtwahl eines Bogens $e \in E$ falls $G(S \cup e)$ positiv zyklisch ist. Ein Bogen, für welchen dies zutrifft, heisst direkt verboten durch S . (2) S impliziert die Wahl eines Bogens $e \in E$, falls eine disjunktive Menge $D \in \mathcal{E}$ existiert, so dass alle Bogen $e' \in D - e$ direkt verboten sind durch S . Ein Bogen, für welchen dies zutrifft, heisst direkt erzwungen durch S . (3) Falls $e \in E$ ein durch S direkt erzwungener Bogen ist, dann impliziert S auch die Nichtwahl aller durch $S' = S \cup e$ direkt verbotenen Bogen und die Wahl aller durch S' direkt erzwungenen Bogen. Aus diesen drei Feststellungen lässt sich das folgende iterative Verfahren zur Bestimmung von verbotenen und erzwungenen Bogen ableiten: Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ und $S \subseteq E$.

1. Setze $Q = S$ und $\bar{Q} = \emptyset$
2. Bestimme alle durch Q direkt verbotenen Bogen \bar{R} in G . Setze $\bar{Q} = \bar{Q} \cup \bar{R}$
3. Bestimme mit Hilfe von \bar{Q} alle durch Q direkt erzwungenen Bogen R in G . Setze $Q = Q \cup R$. Gehe zu Schritt 2.

Gemäss obigen Ausführungen ist \bar{Q} eine Menge von Bogen, deren Nichtwahl durch S impliziert wird, und Q eine Menge von Bogen, deren Wahl durch S impliziert wird. \bar{Q} wird bezeichnet als die Menge der durch S (lokal) verbotenen Bogen und Q als die Menge der durch S (lokal) erzwungenen Bogen in G .

Global erzwungene und verbotene Bogen

Im vorangehenden Absatz wurde ein Verfahren beschrieben, mit welchem in einem disjunktiven Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$ Bogen bestimmt werden können, deren Nichtwahl bzw. Wahl durch $S \subseteq E$ impliziert wird. Die daraus resultierenden Mengen \bar{Q} und Q wurden als die durch S (lokal) verbotenen bzw. erzwungenen Bogen bezeichnet. Es stellt sich nun die Frage, ob mit diesem Verfahren alle Bogen gefunden werden, deren Nichtwahl bzw. Wahl durch S impliziert wird.

Bezeichne einen Bogen, dessen Nichtwahl durch S impliziert wird, als global verboten durch S , und einen Bogen, dessen Wahl durch S impliziert wird, als global erzwungen durch S in G . Die Frage lautet somit, ob - bei gegebenem S - die Menge der (lokal) verbotenen bzw. erzwungenen Bogen identisch ist mit der Menge der global verbotenen bzw. erzwungenen Bogen. Wie sich schon errahnen lässt, trifft dies im allgemeinen nicht zu. Betrachte dazu das folgende Gegenbeispiel.

Beispiel:

Gegeben sei der disjunktive Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ in Abbildung 5.8., wobei $A = \{a\}$, $\mathcal{E} = \{\{e, e'\}, \{f, f'\}\}$ und $c > 0$. Sei $S = \emptyset$

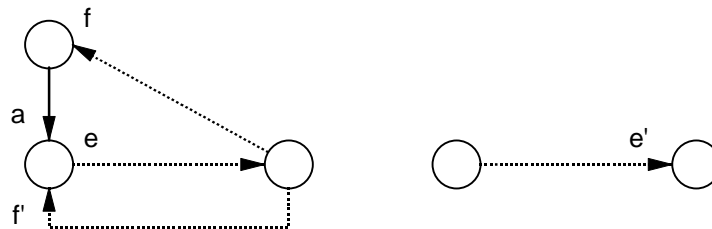


Abbildung 5.8. Global verbotene und erzwungene Bogen.

Offensichtlich existieren keine Bogen, welche durch S (lokal) erzwungen bzw. verboten werden. Die Menge der zulässigen Selektionen S' in G mit $S \subseteq S'$ entspricht der Menge aller zulässigen Selektionen in G und ist gegeben durch $\{\{e', f\}, \{e', f'\}, \{e, f, f'\}\}$. Somit gilt $e \notin S'$ und $e' \in S'$ für alle zulässigen Selektionen S' in G mit $S \subseteq S'$. Folglich ist e global verboten und e' global erzwungen durch S in G . ♦

Dieses Beispiel zeigt, dass die Konzepte, welche bei der Bestimmung der (lokal) verbotenen bzw. erzwungenen Bogen zur Anwendung kommen, im allgemeinen nicht ausreichend sind, um alle global verbotenen bzw. erzwungenen Bogen zu berechnen. In Abschnitt 5.4.4. wird der Zusammenhang zwischen lokal und global verbotenen und erzwungenen Bogen diskutiert. In Abschnitt

5.4.5. wird gezeigt, dass die zur Bestimmung der lokal verbotenen bzw. erzwungenen Bogen angewandten Konzepte eine lokale Sichtweise darstellen, welche sich weiter verallgemeinern lässt.

In Abschnitt 5.5. wird gezeigt, dass ein enger Zusammenhang besteht zwischen dem Zulässigkeitsproblem in einem disjunktiven Graphen und der Bestimmung der global verbotenen bzw. erzwungenen Bogen. Unter dem Zulässigkeitsproblem wird hierbei die Fragestellung verstanden, ob ein disjunktiver Graph eine zulässige Selektion besitzt. Als ein Resultat dieser Betrachtungen wird sich herausstellen, dass sowohl das Zulässigkeitsproblem als auch die Bestimmung der global verbotenen bzw. erzwungenen Bogen in verallgemeinerten disjunktiven Graphen von der Komplexität her NP-hart ist. Für bestimmte Klassen von disjunktiven Graphen ist dieses Problem jedoch polynomial lösbar. Insbesondere wird gezeigt, dass im Falle der klassischen disjunktiven Graphen die Menge der lokal verbotenen bzw. erzwungenen Bogen identisch ist mit der Menge der global verbotenen bzw. erzwungenen Bogen. Daraus folgt, dass das Problem in diesem Fall polynomial lösbar ist, da die Bestimmung der lokal verbotenen bzw. erzwungenen Bogen gemäss Abschnitt 5.4.8. polynomial lösbar ist.

Konjunktive Erweiterung

In den vorangehenden Absätzen wurde gezeigt, wie für einen disjunktiven Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$ und $S \subseteq E$ zusätzliche Informationen über die Struktur der zulässigen Selektionen, welche S enthalten, abgeleitet werden kann. Einerseits können Bogen bestimmt werden, welche in keiner zulässigen Selektion vorhanden sind, und andererseits Bogen, welche in jeder zulässigen Selektion vorhanden sind. In diesem Absatz wird gezeigt, wie sich diese Informationen dazu verwenden lassen, ein disjunktives Graphenproblem zu vereinfachen.

Im folgenden werden global verbotene und erzwungene Bogen betrachtet für den Spezialfall, wo $S = \emptyset$ ist. Diese Bogen werden als global verboten bzw. erzwungen in G bezeichnet. Es handelt sich um Bogen, welche in keiner zulässigen Selektion von G vorhanden sein können, bzw. welche in jeder zulässigen Selektion von G vorhanden sein müssen.

Definition 5.18. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. Sei $Q \subseteq E$ eine Menge von global erzwungenen Bogen in G . Der disjunktive Subgraph $G[Q]$ wird als konjunktive Erweiterung von G bezeichnet.

Gemäss Definition ist $G[Q] = \langle V, A', E', \mathcal{E}', c \rangle$ wobei $A' = A \cup Q$, $\mathcal{E}' = \mathcal{E} - \mathcal{E}(Q)$ und $E' = (E - \mathcal{E}(Q))^\circ$. Die konjunktive Erweiterung $G[Q]$ wird also folgendermassen gebildet: Der "konjunktive Teil" von G wird um Q erweitert, d.h. $A' = A \cup Q$ und der "disjunktive Teil" von G wird um die disjunktiven Mengen, welche von Q geschnitten werden, vermindert, d.h. $\mathcal{E}' = \mathcal{E} - \mathcal{E}(Q)$, bzw. $E' = (E - \mathcal{E}(Q))^\circ$. $G[Q]$ stellt einen vereinfachten disjunktiven Graphen dar,

da die Zahl der disjunktiven Mengen vermindert wurde. Das folgende Lemma zeigt, dass die disjunktiven Graphenprobleme in G und in der konjunktiven Erweiterung $G[Q]$ äquivalent sind.

Lemma 5.19. *Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $Q \subseteq E$ eine Menge von global erzwungenen Bogen in G . Dann sind die disjunktiven Graphenprobleme G und $G[Q]$ äquivalent.*

Beweis:

Bezeichne $\lambda(H)$ die Länge eines längsten $\sigma\tau$ -Weges in einem Graphen H . Zeige erstens: G besitzt genau dann eine zulässige Selektion, wenn $G[Q]$ eine zulässige Selektion besitzt. Sei einerseits S eine zulässige Selektion in G . Da S zulässig ist und Q eine Menge von global erzwungenen Bogen darstellt, gilt $Q \subseteq S$. Folglich gilt gemäss Lemma 5.11., dass $R = S \cap E_{G[Q]}$ eine zulässige Selektion in $G[Q]$ ist. Sei andererseits R eine zulässige Selektion in $G[Q]$. Gemäss Lemma 5.11. ist $S = Q \cup R$ eine zulässige Selektion in G . Zeige zweitens: Für eine optimale Selektion S in G und eine optimale Selektion R in $G[Q]$ gilt $\lambda(G(S)) = \lambda(G[Q](R))$. Zeige einerseits, dass $\lambda(G(S)) \leq \lambda(G[Q](R))$. Da R zulässig ist in $G[Q]$, gilt gemäss Lemma 5.11., dass $S' = Q \cup R$ eine zulässige Selektion in G ist. Da S optimal ist in G , gilt $\lambda(G(S)) \leq \lambda(G(S'))$. Gemäss Lemma 5.10. ist $G(S') = G(Q \cup R) = G[Q](R)$. Folglich gilt $\lambda(G(S)) \leq \lambda(G[Q](R))$. Zeige andererseits, dass $\lambda(G[Q](R)) \leq \lambda(G(S))$. Da S zulässig ist in G und Q eine Menge von global erzwungenen Bogen in G darstellt, ist $Q \subseteq S$. Folglich gilt gemäss Lemma 5.11., dass $R' = S \cap E_{G[Q]}$ eine zulässige Selektion in $G[Q]$ ist. Da R optimal ist in $G[Q]$, gilt $\lambda(G[Q](R)) \leq \lambda(G[Q](R'))$. Da $Q \cup R' \subseteq S$ und durch das Entfernen von Bogen in einem Graphen die Länge eines längsten Weges nicht zunimmt, gilt $\lambda(G(Q \cup R')) \leq \lambda(G(S))$. Gemäss Lemma 5.10. ist $G(Q \cup R') = G[Q](R')$. Folglich ist $\lambda(G[Q](R')) \leq \lambda(G(S))$. Wegen $\lambda(G[Q](R)) \leq \lambda(G[Q](R'))$ folgt daraus $\lambda(G[Q](R)) \leq \lambda(G(S))$. ♦

Es wurde gezeigt, wie Information über die global erzwungenen Bogen in einem disjunktiven Graphen G dazu verwendet werden kann, den Graphen zu vereinfachen. Auf ähnliche Weise ist es auch möglich, Information über die global verbotenen Bogen zur Vereinfachung von G zu verwenden. Falls ein Bogen $e \in E$ global verboten ist in G , dann kann er in keiner zulässigen Selektion vorkommen. Folglich kann e aus allen disjunktiven Mengen entfernt werden. Diese Art der Vereinfachung ist jedoch nur von Interesse, wenn disjunktive Graphen betrachtet werden, welche disjunktive Mengen mit mehr als zwei Bogen besitzen. Für den Fall, wo $|D| = 2$ für alle $D \in \mathcal{E}$ wird auf diese Weise keine weitere Vereinfachung erzielt. Dies lässt sich folgendermassen einsehen: Nehme an, dass ein global verbotener Bogen e in G gegeben ist. Sei $D = \{e, e'\}$ eine disjunktive Menge, welche e enthält. Da $|D| = 2$ und e global verboten ist, folgt unmittelbar, dass e' global erzwungen ist. Somit kann die disjunktive Menge D durch konjunktive Erweiterung von G durch e' eliminiert werden.

Zusammenfassend war das Ziel dieses Abschnittes, eine Einführung in die Idee der verbotenen und erzwungenen Bogen in einem disjunktiven Graphen zu geben. Es wird sich zeigen, dass dieses Konzept eine wichtige Bedeutung hat bei der Entwicklung eines Optimierungsverfahrens für disjunktive Lager-Graphen. In den folgenden Abschnitten wird diese Theorie weiter formalisiert und ausführlich diskutiert.

5.4.2. Definitionen

In diesem Abschnitt werden die Definitionen zur Formalisierung des Konzepts der verbotenen und erzwungenen Bogen gegeben. Zunächst werden zwei Mengenfunktionen $\chi_G(S)$ und $\phi_G(S)$ eingeführt, welche in Abhängigkeit von einer vorgegebenen Selektion S die Menge der direkt verbotenen bzw. erzwungenen Bogen in einem disjunktiven Graphen G darstellen. Durch Iteration dieser Mengenfunktionen wird daraus der Begriff der r -stufig verbotenen bzw. erzwungenen Bogen abgeleitet, bezeichnet als $\chi_G^r(S)$ bzw. $\phi_G^r(S)$. Die Menge $\chi_G^*(S)$ bzw. $\phi_G^*(S)$ aller verbotenen bzw. erzwungenen Bogen wird erhalten, indem diese Iteration bis zu ihrem Abschluss durchgeführt wird, d.h. solange bis keine weiteren verbotenen bzw. erzwungenen Bogen mehr gefunden werden. Der disjunktive Subgraph $G[\phi_G^*(S)]$ welcher resultiert, wenn alle durch S erzwungenen Bogen in G eingefügt werden, wird als konjunktiver Abschluss von G bezüglich S bezeichnet. Ein disjunktiver Graph wird als konjunktiv abgeschlossen bezeichnet, wenn keine Bogen in ihm erzwungen werden.

Wie schon früher erwähnt, bilden die auf diese Weise gefundenen verbotenen bzw. erzwungenen Bogen eine Teilmenge der global verbotenen bzw. erzwungenen Bogen in einem disjunktiven Graphen, wobei im allgemeinen jedoch nicht alle global verbotenen bzw. erzwungenen Bogen erfasst werden. Dieser Zusammenhang wird ausführlich diskutiert in Abschnitt 5.4.4.

Definition 5.20. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$.

(i) Die Menge der direkt verbotenen disjunktiven Bogen in G bei Wahl von S ist definiert durch:

$$\chi_G(S) = \{e \in E : G(S \cup e) \text{ ist positiv zyklisch}\}$$

(ii) Die Menge der direkt erzwungenen disjunktiven Bogen in G bei Wahl von S ist definiert durch:

$$\phi_G(S) = S \cup \{e \in E : \text{es existiert ein } D \in \mathcal{E}(e), \text{ so dass } e' \in \chi_G(S) \text{ für alle } e' \in D - e\}$$

Für den Spezialfall $S = \emptyset$ gelten die Bezeichnungen $\chi_G(\emptyset) = \chi(G)$ und $\phi_G(\emptyset) = \phi(G)$. Weiter bezeichnet $\Phi_G(R) = \{e \in E : \text{es existiert ein } D \in \mathcal{E}(e), \text{ so dass } e' \in R \text{ für alle } e' \in D - e\}$ für $R \subseteq E$. Mit dieser Bezeichnung gilt $\phi_G(S) = S \cup \Phi_G(\chi_G(S))$.

Definition 5.21. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$.

(i) Die Menge der r -stufig erzwungenen disjunktiven Bogen in G bei Wahl von S ist definiert durch:

$$\begin{aligned}\varphi_G^r(S) &= S, & r &= 0 \\ \varphi_G^r(S) &= \varphi_G(\varphi_G^{r-1}(S)), & r &\geq 1\end{aligned}$$

(ii) Die Menge der r -stufig verbotenen disjunktiven Bogen in G bei Wahl von S ist definiert durch:

$$\begin{aligned}\chi_G^r(S) &= \emptyset, & r &= 0 \\ \chi_G^r(S) &= \chi_G(\varphi_G^{r-1}(S)), & r &\geq 1\end{aligned}$$

Für den Spezialfall $S = \emptyset$ gelten die Bezeichnungen $\varphi_G^r(\emptyset) = \varphi^r(G)$ und $\chi_G^r(\emptyset) = \chi^r(G)$ für $r \geq 0$. Beachte, dass $\varphi_G^1(S) = \varphi_G(S)$ und $\chi_G^1(S) = \chi_G(S)$.

Definition 5.22. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$.

(i) Die Menge aller verbotenen disjunktiven Bogen in G bei Wahl von S ist definiert als:

$$\chi_G^*(S) = \chi_G^\infty(S)$$

(ii) Die Menge aller erzwungenen disjunktiven Bogen in G bei Wahl von S ist definiert als:

$$\varphi_G^*(S) = \varphi_G^\infty(S)$$

$\varphi_G^*(S)$ heisst das konjunktive Erzeugnis von S in G , und $\varphi^*(G) = \varphi_G^*(\emptyset)$ das konjunktive Erzeugnis von G . $G[\varphi_G^*(S)]$ heisst der konjunktive Abschluss von G bezüglich S , und $G[\varphi^*(G)]$ der konjunktive Abschluss von G .

Definition 5.23. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. G heisst konjunktiv abgeschlossen, falls $\varphi(G) = \emptyset$ ist.

5.4.3. Eigenschaften

In diesem Abschnitt werden verschiedene Eigenschaften im Zusammenhang mit dem Konzept der verbotenen bzw. erzwungenen Bogen diskutiert. Zur Uebersicht sind die Resultate thematisch in Absätze gegliedert.

Monotonie und Iterationskette

Lemma 5.24. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Für alle $r \geq 0$ sind die Funktionen $\chi_G^r: \text{Pot}(E) \rightarrow \text{Pot}(E)$ und $\phi_G^r: \text{Pot}(E) \rightarrow \text{Pot}(E)$ monoton. D.h. für alle $r \geq 0$ und $S, S' \subseteq E$ gilt:

- (i) $S \subseteq S' \Rightarrow \chi_G^r(S) \subseteq \chi_G^r(S')$
- (ii) $S \subseteq S' \Rightarrow \phi_G^r(S) \subseteq \phi_G^r(S')$

Beweis:

Der Beweis erfolgt mittels Induktion über r . Die Induktion wird simultan für (i) und (ii) durchgeführt. Sei $S, S' \subseteq E$ mit $S \subseteq S'$ (1) Induktionsbehauptung: Für alle $r \geq 0$ gilt $\mathcal{P}(r)$ (i) $\chi_G^r(S) \subseteq \chi_G^r(S')$ und (ii) $\phi_G^r(S) \subseteq \phi_G^r(S')$ (2) Induktionsverankerung: Zeige, dass $\mathcal{P}(0)$ gilt. Zu (i): Gemäss Definition ist $\chi_G^0(S) = \chi_G^0(S') = \emptyset$ und somit $\chi_G^0(S) \subseteq \chi_G^0(S')$. Zu (ii): Gemäss Definition ist $\phi_G^0(S) = S$ und $\phi_G^0(S') = S'$. Wegen $S \subseteq S'$ gilt somit $\phi_G^0(S) \subseteq \phi_G^0(S')$. (3) Induktionsschritt: Zeige, dass für alle $r \geq 0$ gilt: $\mathcal{P}(r) \Rightarrow \mathcal{P}(r+1)$. Sei $r \geq 0$ und gelte $\mathcal{P}(r)$. Zeige, dass $\mathcal{P}(r+1)$ gilt.

Zu (i): Sei $e \in \chi_G^{r+1}(S)$. Zeige, dass $e \in \chi_G^{r+1}(S')$. Gemäss Definition ist $\chi_G^{r+1}(S) = \chi_G(\phi_G^r(S))$. Da $e \in \chi_G^{r+1}(S)$ ist $G(\phi_G^r(S) \cup e)$ positiv zyklisch. Gemäss $\mathcal{P}(r)$ gilt $\phi_G^r(S) \subseteq \phi_G^r(S')$. Somit ist auch $G(\phi_G^r(S') \cup e)$ positiv zyklisch. Daraus folgt $e \in \chi_G(\phi_G^r(S')) = \chi_G^{r+1}(S')$.

Zu (ii): Sei $e \in \phi_G^{r+1}(S)$. Zeige, dass $e \in \phi_G^{r+1}(S')$. Gemäss Definition ist $\phi_G^{r+1}(S) = \phi_G(\phi_G^r(S)) = \phi_G^r(S) \cup \Phi_G(\chi_G(\phi_G^r(S)))$ und $\phi_G^{r+1}(S') = \phi_G(\phi_G^r(S')) = \phi_G^r(S') \cup \Phi_G(\chi_G(\phi_G^r(S')))$. Fall 1: Sei $e \in \phi_G^r(S)$. Gemäss $\mathcal{P}(r)$ gilt $\phi_G^r(S) \subseteq \phi_G^r(S')$. Folglich ist $e \in \phi_G^{r+1}(S')$. Fall 2: Sei $e \in \Phi_G(\chi_G(\phi_G^r(S)))$. Dann existiert ein $D \in \mathcal{E}(e)$, so dass $e' \in \chi_G(\phi_G^r(S)) = \chi_G^{r+1}(S)$. Für alle $e' \in D$ – Wie schon gezeigt wurde, gilt $\chi_G^{r+1}(S) \subseteq \chi_G^{r+1}(S')$. Somit ist $e' \in \chi_G^{r+1}(S') = \chi_G(\phi_G^r(S'))$ für alle $e' \in D$. Daraus folgt $e \in \Phi_G(\chi_G(\phi_G^r(S')))$ und somit $e \in \phi_G^{r+1}(S')$ ♦

Lemma 5.25. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Für alle $r \geq 0$ gilt:

- (i) $\phi_G^r(S) \subseteq \phi_G^{r+1}(S)$
- (ii) $\chi_G^r(S) \subseteq \chi_G^{r+1}(S)$

Beweis:

Beweis von (i): Sei $r \geq 0$. Gemäss Definition ist $\phi_G^{r+1}(S) = \phi_G(\phi_G^r(S)) = \phi_G^r(S) \cup \Phi_G(\chi_G(\phi_G^r(S)))$ und somit $\phi_G^r(S) \subseteq \phi_G^{r+1}(S)$.

Beweis von (ii): Fall 1: Sei $r = 0$. Gemäss Definition ist $\chi_G^0(S) = \emptyset \subseteq \chi_G^1(S)$. Fall 2: Sei $r \geq 1$. Sei $e \in \chi_G^r(S)$. Zeige, dass $e \in \chi_G^{r+1}(S)$. Gemäss Definition ist $\chi_G^r(S) = \chi_G(\phi_G^{r-1}(S))$. Aus

$e \in \chi_G^r(S)$ folgt somit, dass $G(\phi_G^{r-1}(S) \cup e)$ positiv zyklisch ist. Gemäss Beweis von (i) gilt $\phi_G^{r-1}(S) \subseteq \phi_G^r(S)$. Somit ist auch $G(\phi_G^r(S) \cup e)$ positiv zyklisch. Daraus folgt $e \in \chi_G(\phi_G^r(S)) = \chi_G^{r+1}(S)$. ♦

Die Mengen $\chi_G^r(S)$ und $\phi_G^r(S)$ $r \geq 0$ bilden somit eine Kette, d.h. $\chi_G^0(S) \subseteq \chi_G^1(S) \subseteq \chi_G^2(S) \subseteq \dots$ und $\phi_G^0(S) \subseteq \phi_G^1(S) \subseteq \phi_G^2(S) \subseteq \dots$. Für ϕ_G und $S = E$ handelt es sich hierbei um die sogenannte Iterationskette, welche für alle monotonen Mengenfunktionen existiert.

Iteration

Lemma 5.26. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt:

- (i) Es existiert eine Zahl $0 \leq r \leq |E|$, so dass $\phi_G^{r+1}(S) = \phi_G^r(S)$.
- (ii) Es existiert eine Zahl $0 \leq s \leq |E|$, so dass $\chi_G^{s+1}(S) = \chi_G^s(S)$.

Beweis:

Beweis von (i): Nehme an, dass kein solches r existiert. Dann gilt für alle $0 \leq r \leq |E|$ dass $\phi_G^{r+1}(S) \neq \phi_G^r(S)$. Gemäss Lemma 5.25. ist $\phi_G^r(S) \subseteq \phi_G^{r+1}(S)$ für alle $r \geq 0$. Somit gilt $\phi_G^r(S) \subset \phi_G^{r+1}(S)$ für alle $0 \leq r \leq |E|$. Sei $k = |E| + 1$. Es gilt $\phi_G^0(S) \subset \phi_G^1(S) \subset \dots \subset \phi_G^k(S)$. Folglich ist $|\phi_G^k(S)| \geq |\phi_G^0(S)| + k$. Da $k = |E| + 1$, gilt somit $|\phi_G^k(S)| \geq |E| + 1$. Da aber gemäss Definition $\phi_G^k(S) \subseteq E$ gilt $|\phi_G^k(S)| \leq |E|$. Ein Widerspruch.

Beweis von (ii): Analog. ♦

Lemma 5.27. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt:

- (i) Falls $\phi_G^{r+1}(S) = \phi_G^r(S)$ für ein $r \geq 0$, dann ist $\phi_G^{r+n}(S) = \phi_G^r(S)$ für alle $n \geq 1$.
- (ii) Falls $\chi_G^{s+1}(S) = \chi_G^s(S)$ für ein $s \geq 0$, dann ist $\chi_G^{s+n}(S) = \chi_G^s(S)$ für alle $n \geq 1$.

Beweis:

Beweis von (i): Der Beweis erfolgt mittels Induktion über n . Sei $\phi_G^{r+1}(S) = \phi_G^r(S)$ für ein $r \geq 0$.
 (1) Induktionsbehauptung: Für alle $n \geq 1$ gilt die Aussage $\mathcal{P}(n)$ $\phi_G^{r+n}(S) = \phi_G^r(S)$.
 (2) Induktionsverankerung: Zeige, dass $\mathcal{P}(1)$ gilt. Gemäss Voraussetzung ist $\phi_G^{r+1}(S) = \phi_G^r(S)$.
 (3) Induktionsschritt: Zeige, dass für alle $n \geq 1$ gilt: $\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1)$. Sei $n \geq 1$ und gelte $\mathcal{P}(n)$. Zeige, dass $\mathcal{P}(n+1)$ gilt. Gemäss Definition ist $\phi_G^{r+n+1}(S) = \phi_G(\phi_G^{r+n}(S))$. Gemäss $\mathcal{P}(n)$ gilt $\phi_G^{r+n}(S) = \phi_G^r(S)$. Somit ist $\phi_G^{r+n+1}(S) = \phi_G(\phi_G^r(S)) = \phi_G^{r+1}(S)$. Gemäss Voraussetzung gilt $\phi_G^{r+1}(S) = \phi_G^r(S)$. Daraus folgt $\phi_G^{r+n+1}(S) = \phi_G^r(S)$.

Beweis von (ii): Sei $\chi_G^{s+1}(S) = \chi_G^s(S)$ für ein $s \geq 0$. Zeige zunächst, dass $\phi_G^{s+1}(S) = \phi_G^s(S)$

Fall 1: Sei $s = 0$. Dann gilt $\chi_G^1(S) = \chi_G^0(S) = \emptyset$. Gemäss Definition ist $\phi_G^0(S) = S$ und $\phi_G^1(S) = S \cup \Phi_G(\chi_G^1(S))$. Da $\Phi_G(\chi_G^1(S)) = \Phi_G(\emptyset) = \emptyset$ gilt $\phi_G^1(S) = \phi_G^0(S) = S$.
Fall 2: Sei $s \geq 1$. Gemäss Definition ist $\phi_G^{s+1}(S) = \phi_G^s(\phi_G^s(S))$ d.h. $\phi_G^{s+1}(S) = \phi_G^s(S) \cup \Phi_G(\chi_G^s(\phi_G^s(S)))$.
Da $\chi_G(\phi_G^s(S)) = \chi_G^{s+1}(S)$ gilt somit $\phi_G^{s+1}(S) = \phi_G^s(S) \cup \Phi_G(\chi_G^{s+1}(S))$. Analog ist $\phi_G^s(S) = \phi_G^{s-1}(S) \cup \Phi_G(\chi_G^s(S))$. Durch Einsetzen erhält man $\phi_G^{s+1}(S) = \phi_G^{s-1}(S) \cup \Phi_G(\chi_G^s(S)) \cup \Phi_G(\chi_G^{s+1}(S))$. Da $\chi_G^{s+1}(S) = \chi_G^s(S)$ gilt somit $\phi_G^{s+1}(S) = \phi_G^{s-1}(S) \cup \Phi_G(\chi_G^s(S))$, d.h. $\phi_G^{s+1}(S) = \phi_G^s(S)$.

Da $\phi_G^{s+1}(S) = \phi_G^s(S)$ folgt gemäss Beweis von (i), dass $\phi_G^{s+n}(S) = \phi_G^s(S)$ für alle $n \geq 1$.
Da $\chi_G^{s+n}(S) = \chi_G(\phi_G^{s+n-1}(S))$ und $\phi_G^{s+n-1}(S) = \phi_G^s(S)$ für alle $n \geq 1$ gilt somit $\chi_G^{s+n}(S) = \chi_G(\phi_G^s(S)) = \chi_G^{s+1}(S)$ für alle $n \geq 1$. Da gemäss Voraussetzung $\chi_G^{s+1}(S) = \chi_G^s(S)$ folgt daraus $\chi_G^{s+n}(S) = \chi_G^s(S)$ für alle $n \geq 1$. ♦

Lemma 5.28. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Sei $r \geq 0$ die kleinste Zahl mit $\phi_G^{r+1}(S) = \phi_G^r(S)$ und $s \geq 0$ die kleinste Zahl mit $\chi_G^{s+1}(S) = \chi_G^s(S)$. Es gilt:

(i) $\phi_G^*(S) = \phi_G^r(S)$ und $\chi_G^*(S) = \chi_G^s(S)$

(ii) $r \leq s \leq r+1$ und $0 \leq r, s \leq |E|$

Beweis:

Beachte zunächst, dass die Zahlen r und s gemäss Lemma 5.26. tatsächlich existieren und dass gilt $0 \leq r, s \leq |E|$

Beweis von (i): Zeige erstens, dass $\phi_G^*(S) = \phi_G^r(S)$. Gemäss Lemma 5.27. gilt $\phi_G^{r+n}(S) = \phi_G^r(S)$ für alle $n \geq 1$. Folglich ist $\phi_G^*(S) = \phi_G^\infty(S) = \phi_G^r(S)$.
Zeige zweitens, dass $\chi_G^*(S) = \chi_G^s(S)$. Gemäss Lemma 5.27. gilt $\chi_G^{s+n}(S) = \chi_G^s(S)$ für alle $n \geq 1$. Folglich ist $\chi_G^*(S) = \chi_G^\infty(S) = \chi_G^s(S)$.

Beweis von (ii): Zeige erstens, dass $r \leq s$. Fall 1: Sei $s = 0$. Dann gilt $\chi_G^1(S) = \chi_G^0(S) = \emptyset$. Gemäss Definition ist $\phi_G^1(S) = S \cup \Phi_G(\chi_G^1(S))$ und $\phi_G^0(S) = S$. Wegen $\chi_G^1(S) = \emptyset$ ist $\Phi_G(\chi_G^1(S)) = \emptyset$ und folglich $\phi_G^1(S) = S$. Somit gilt $\phi_G^{0+1}(S) = \phi_G^0(S)$. Da die kleinste Zahl ist mit $\phi_G^{r+1}(S) = \phi_G^r(S)$ folgt daraus $r \leq 0$, d.h. $r = 0$.
Fall 2: Sei $s \geq 1$. Gemäss Definition ist $\phi_G^{s+1}(S) = \phi_G^s(\phi_G^s(S)) = \phi_G^s(S) \cup \Phi_G(\chi_G^s(\phi_G^s(S)))$, d.h. $\phi_G^{s+1}(S) = \phi_G^s(S) \cup \Phi_G(\chi_G^{s+1}(S))$. Analog ist $\phi_G^s(S) = \phi_G^{s-1}(S) \cup \Phi_G(\chi_G^s(S))$. Somit ist $\phi_G^{s+1}(S) = \phi_G^{s-1}(S) \cup \Phi_G(\chi_G^s(S)) \cup \Phi_G(\chi_G^{s+1}(S))$. Da $\chi_G^{s+1}(S) = \chi_G^s(S)$ gilt somit $\phi_G^{s+1}(S) = \phi_G^{s-1}(S) \cup \Phi_G(\chi_G^s(S))$, d.h. $\phi_G^{s+1}(S) = \phi_G^s(S)$. Da die kleinste Zahl ist mit $\phi_G^{r+1}(S) = \phi_G^r(S)$, folgt daraus $r \leq s$.

Zeige zweitens, dass $s \leq r+1$. Gemäss Definition ist $\chi_G^{r+2}(S) = \chi_G(\phi_G^{r+1}(S))$ und $\chi_G^{r+1}(S) = \chi_G(\phi_G^r(S))$. Da $\phi_G^{r+1}(S) = \phi_G^r(S)$ gilt somit $\chi_G^{(r+1)+1}(S) = \chi_G^{r+1}(S)$. Da die kleinste Zahl mit $\chi_G^{s+1}(S) = \chi_G^s(S)$ ist, folgt daraus $s \leq r+1$. ♦

Gegenseitige Abhängigkeit von χ_G^r und ϕ_G^r

Lemma 5.29. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Für alle $r \geq 0$ gilt:

- (i) $\chi_G^{r+1}(S) = \chi_G(\phi_G^r(S))$
- (ii) $\phi_G^r(S) = S \cup \Phi_G(\chi_G^r(S))$

Daraus folgt insbesondere, dass $\chi_G^*(S) = \chi_G(\phi_G^*(S))$ und $\phi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$.

Beweis:

Beweis von (i): Gemäss Definition ist $\chi_G^r(S) = \chi_G(\phi_G^{r-1}(S))$ für $r \geq 1$. Daraus folgt $\chi_G^{r+1}(S) = \chi_G(\phi_G^r(S))$ für alle $r \geq 0$.

Beweis von (ii): Der Beweis erfolgt mittels Induktion über r . (1) Induktionsbehauptung: Für alle $r \geq 0$ gilt $\mathcal{P}(r)$: $\phi_G^r(S) = S \cup \Phi_G(\chi_G^r(S))$. (2) Induktionsverankerung: Zeige, dass $\mathcal{P}(0)$ gilt. Gemäss Definition ist $\phi_G^0(S) = S$. Da $\chi_G^0(S) = \emptyset$ gilt $\Phi_G(\chi_G^0(S)) = \emptyset$. Folglich ist $\phi_G^0(S) = S \cup \Phi_G(\chi_G^0(S))$. (3) Induktionsschritt: Zeige, dass für alle $r \geq 0$ gilt: $\mathcal{P}(r) \Rightarrow \mathcal{P}(r+1)$. Sei $r \geq 1$ und gelte $\mathcal{P}(r)$. Zeige, dass $\mathcal{P}(r+1)$ gilt. Gemäss Definition ist $\phi_G^{r+1}(S) = \phi_G(\phi_G^r(S)) = \phi_G^r(S) \cup \Phi_G(\chi_G^r(\phi_G^r(S)))$. Da $\chi_G(\phi_G^r(S)) = \chi_G^{r+1}(S)$ gilt somit $\phi_G^{r+1}(S) = \phi_G^r(S) \cup \Phi_G(\chi_G^{r+1}(S))$. Gemäss $\mathcal{P}(r)$ ist $\phi_G^r(S) = S \cup \Phi_G(\chi_G^r(S))$ und folglich $\phi_G^{r+1}(S) = S \cup \Phi_G(\chi_G^r(S)) \cup \Phi_G(\chi_G^{r+1}(S))$. Da $\chi_G^r(S) \subseteq \chi_G^{r+1}(S)$ gilt $\Phi_G(\chi_G^r(S)) \subseteq \Phi_G(\chi_G^{r+1}(S))$. Somit ist $\phi_G^{r+1}(S) = S \cup \Phi_G(\chi_G^{r+1}(S))$.

Zeige noch, dass $\chi_G^*(S) = \chi_G(\phi_G^*(S))$ und $\phi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$. Sei r die kleinste Zahl mit $\phi_G^{r+1}(S) = \phi_G^r(S)$ und $s \geq 0$ die kleinste Zahl mit $\chi_G^{s+1}(S) = \chi_G^s(S)$. Gemäss Lemma 5.27. ist $\phi_G^{r+n}(S) = \phi_G^r(S)$ und $\chi_G^{s+n}(S) = \chi_G^s(S)$ für alle $n \geq 1$. Gemäss Lemma 5.28. ist $\phi_G^*(S) = \phi_G^r(S)$ und $\chi_G^*(S) = \chi_G^s(S)$. Zeige erstens, dass $\chi_G^*(S) = \chi_G(\phi_G^*(S))$. Gemäss vorangehendem Teil des Beweises ist $\chi_G^{r+n+1}(S) = \chi_G(\phi_G^{r+n}(S))$ für $n \geq 0$. Da $\phi_G^{r+n}(S) = \phi_G^r(S)$ für alle $n \geq 1$ gilt $\chi_G^{r+n+1}(S) = \chi_G(\phi_G^r(S))$ für alle $n \geq 1$. Somit ist $\chi_G^\infty(S) = \chi_G(\phi_G^r(S))$ d.h. $\chi_G^*(S) = \chi_G(\phi_G^*(S))$. Zeige zweitens, dass $\phi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$. Gemäss vorangehendem Teil des Beweises ist $\phi_G^{s+n}(S) = S \cup \Phi_G(\chi_G^{s+n}(S))$ für $n \geq 0$. Da $\chi_G^{s+n}(S) = \chi_G^s(S)$ für alle $n \geq 1$ gilt $\phi_G^{s+n}(S) = S \cup \Phi_G(\chi_G^s(S))$ für alle $n \geq 1$. Somit ist $\phi_G^\infty(S) = S \cup \Phi_G(\chi_G^s(S))$. $\phi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$. ♦

Konjunktiver Abschluss

Lemma 5.30. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt:

- (i) $\varphi_G(\varphi_G^*(S)) = \varphi_G^*(S)$
- (ii) $\varphi_G(S) = S \Leftrightarrow \varphi_G^*(S) = S$

Beweis:

Beweis von (i): Sei $r \geq 0$ die kleinste Zahl mit $\varphi_G^{r+1}(S) = \varphi_G^r(S)$. Gemäss Lemma 5.28. ist $\varphi_G^*(S) = \varphi_G^r(S)$. Somit ist $\varphi_G(\varphi_G^*(S)) = \varphi_G(\varphi_G^r(S)) = \varphi_G^{r+1}(S) = \varphi_G^r(S)$. Da $\varphi_G^{r+1}(S) = \varphi_G^r(S)$ folgt daraus $\varphi_G(\varphi_G^*(S)) = \varphi_G^*(S)$.

Beweis von (ii): Sei einerseits $\varphi_G(S) = S$. Zeige, dass $\varphi_G^*(S) = S$. Da $\varphi_G(S) = S$ und $\varphi_G^0(S) = S$, gilt $\varphi_G^1(S) = \varphi_G^0(S)$. Gemäss Lemma 5.27. folgt daraus $\varphi_G^{1+n}(S) = \varphi_G^0(S)$ für alle $n \geq 1$. Somit ist $\varphi_G^\infty(S) = \varphi_G^0(S)$ d.h. $\varphi_G^*(S) = S$. Sei andererseits $\varphi_G^*(S) = S$. Zeige, dass $\varphi_G(S) = S$. Gemäss Lemma 5.25. ist $\varphi_G(S) \subseteq \varphi_G^*(S)$ und folglich $\varphi_G(S) \subseteq S$. Da auch $S \subseteq \varphi_G(S)$, gilt $\varphi_G(S) = S$. ♦

Lemma 5.31. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$.

- (i) Falls $\varphi_G(S) = S$, dann ist $G[S]$ konjunktiv abgeschlossen.
- (ii) Falls $G[S]$ konjunktiv abgeschlossen und positiv azyklisch ist, dann ist $\varphi_G(S) = S$.

Daraus folgt insbesondere, dass $G[\varphi_G^*(S)]$ konjunktiv abgeschlossen ist.

Beweis:

Beweis von (i): Sei $\varphi_G(S) = S$. Bezeichne $H = G[S]$. Gemäss Definition ist H konjunktiv abgeschlossen, falls $\varphi(H) = \varphi_H(\emptyset) = \emptyset$. Zeige, dass $\varphi_H(\emptyset) = \emptyset$. Nehme an, dass $\varphi_H(\emptyset) \neq \emptyset$. Sei $e \in \varphi_H(\emptyset)$. Gemäss Definition ist $\varphi_H(\emptyset) = \emptyset \cup \Phi_H(\chi_H(\emptyset))$. Da $e \in \Phi_H(\chi_H(\emptyset))$ existiert ein $D \in \mathcal{E}_H$ mit $e \in D$, so dass $e' \in \chi_H(\emptyset)$ für alle $e' \in D - e$. Sei $e' \in D - e$. Da $e' \in \chi_H(\emptyset)$ ist $H(\emptyset \cup e')$ positiv zyklisch. Es gilt $H(\emptyset \cup e') = G[S](e') = G(S \cup e')$. Da $G(S \cup e')$ positiv zyklisch ist, gilt $e' \in \chi_G(S)$. Somit ist gezeigt, dass ein $D \in \mathcal{E}_H$ existiert mit $e \in D$ so dass $e' \in \chi_G(S)$ für alle $e' \in D - e$. Daraus folgt, dass $e \in \varphi_G(S)$ ist. Da $\varphi_G(S) = S$, gilt somit $e \in S$. Da $e \in D$ und $e \in S$ gilt $D \cap S \neq \emptyset$ und folglich $D \in \mathcal{E}(S)$. Gemäss Definition ist aber $\mathcal{E}_H = \mathcal{E} - \mathcal{E}(S)$. Da $D \in \mathcal{E}_H$ gilt somit $D \notin \mathcal{E}(S)$. Ein Widerspruch.

Beweis von (ii): Sei $G[S]$ konjunktiv abgeschlossen und positiv azyklisch. Zeige, dass $S = \varphi_G(S)$. Da $S \subseteq \varphi_G(S)$ muss nur gezeigt werden, dass $\varphi_G(S) \subseteq S$. Nehme an, dass ein $e \in \varphi_G(S)$ existiert mit $e \notin S$. Gemäss Definition ist $\varphi_G(S) = S \cup \Phi(\chi_G(S))$. Da $e \notin S$ existiert

somit ein $D \in \mathcal{E}$ mit $e \in D$ so dass $e' \in \chi_G(S)$ für alle $e' \in D - e$. Da $G[S]$ disjunktiv abgeschlossen ist, gilt $\varphi(G[S]) = \varphi_{G[S]}(\emptyset) = \emptyset$. Gemäss Definition ist $\varphi_{G[S]}(\emptyset) = \emptyset \cup \Phi_{G[S]}(\chi_{G[S]}(\emptyset))$. Daraus folgt $\Phi_{G[S]}(\chi_{G[S]}(\emptyset)) = \emptyset$. Somit gilt $e \notin \Phi_{G[S]}(\chi_{G[S]}(\emptyset))$. D.h. es existiert kein $D' \in \mathcal{E}_{G[S]}$ mit $e \in D'$ so dass $e' \in \chi_{G[S]}(\emptyset)$ für alle $e' \in D' - e$. Gemäss Definition ist $\chi_{G[S]}(\emptyset) = \{f \in \mathcal{E}_{G[S]}^\circ : G[S](\emptyset \cup f) \text{ ist positiv zyklisch}\}$. Da $G[S](\emptyset \cup f) = G(S \cup f)$ gilt somit $\chi_{G[S]}(\emptyset) = \chi_G(S) \cap \mathcal{E}_{G[S]}^\circ$. Folglich existiert kein $D' \in \mathcal{E}_{G[S]}$ mit $e \in D'$, so dass $e' \in \chi_G(S) \cap \mathcal{E}_{G[S]}^\circ$ für alle $e' \in D' - e$. Da $D' \subseteq \mathcal{E}_{G[S]}^\circ$ für alle $D' \in \mathcal{E}_{G[S]}$, gilt für alle $D' \in \mathcal{E}_{G[S]}$ und alle $e' \in D'$ dass die Bedingung $e' \in \chi_G(S) \cap \mathcal{E}_{G[S]}^\circ$ äquivalent ist zur Bedingung $e' \in \chi_G(S)$. Daraus folgt unter Berücksichtigung von $\mathcal{E}_{G[S]} = \mathcal{E} - \mathcal{E}(S)$, dass kein $D' \in \mathcal{E} - \mathcal{E}(S)$ existiert mit $e \in D'$ so dass $e' \in \chi_G(S)$ für alle $e' \in D' - e$. Wie aber schon gezeigt wurde, existiert ein $D \in \mathcal{E}$ mit $e \in D$ so dass $e' \in \chi_G(S)$ für alle $e' \in D - e$. Folglich muss $D \in \mathcal{E}(S)$ sein. Also existiert ein $e' \in D$ mit $e' \in S$. Da $e \notin S$ gilt $e' \neq e$. Da $e' \in D - e$ gilt $e' \in \chi_G(S)$. Folglich ist $G(S \cup e)$ positiv zyklisch. Da $e' \in S$ ist somit $G(S)$ positiv zyklisch. Ein Widerspruch zur Annahme, dass $G[S]$ positiv azyklisch ist. ♦

Lemma 5.32. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$ und $R \subseteq S$. Dann gilt:

$$\varphi_G^*(\varphi_G^*(R) \cup S) = \varphi_G^*(S)$$

Beweis:

Da $R \subseteq S$, gilt aufgrund der Monotonie von φ_G^* dass $\varphi_G^*(R) \subseteq \varphi_G^*(S)$. Da auch $S \subseteq \varphi_G^*(S)$ gilt somit $\varphi_G^*(R) \cup S \subseteq \varphi_G^*(S)$. Aus der Monotonie von φ_G^* folgt $\varphi_G^*(\varphi_G^*(R) \cup S) \subseteq \varphi_G^*(\varphi_G^*(S)) = \varphi_G^*(S)$. Da umgekehrt $S \subseteq \varphi_G^*(R) \cup S$ folgt aus der Monotonie von φ_G^* , dass $\varphi_G^*(S) \subseteq \varphi_G^*(\varphi_G^*(R) \cup S)$. Somit gilt $\varphi_G^*(\varphi_G^*(R) \cup S) = \varphi_G^*(S)$ ♦

Positiv zyklische Graphen

Lemma 5.33. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. Falls $\varphi_G^*(S)$ positiv zyklisch ist in G , dann gilt $\chi_G^*(S) = E$ und $\varphi_G^*(S) = E$.

Beweis:

Sei $\varphi_G^*(S)$ positiv zyklisch ist in G . Zeige erstens, dass $\chi_G^*(S) = E$. Da $\chi_G^*(S) \subseteq E$ muss nur gezeigt werden, dass $E \subseteq \chi_G^*(S)$. Sei $e \in E$. Nehme an, dass $e \notin \chi_G^*(S)$. Gemäss Lemma 5.29. ist $\chi_G^*(S) = \chi_G(\varphi_G^*(S))$. Da $e \notin \chi_G^*(S)$ ist $G(\varphi_G^*(S) \cup e)$ positiv azyklisch. Da aber $G(\varphi_G^*(S))$ positiv zyklisch ist, muss auch $G(\varphi_G^*(S) \cup e)$ positiv zyklisch sein. Ein Widerspruch. Zeige zweitens, dass $\varphi_G^*(S) = E$. Da $\varphi_G^*(S) \subseteq E$ muss nur gezeigt werden, dass $E \subseteq \varphi_G^*(S)$. Sei $e \in E$. Nehme an, dass $e \notin \varphi_G^*(S)$. Gemäss Lemma 5.29. ist $\varphi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$. Gemäss Beweis von (i) ist

$\chi_G^*(S) = E$. Da $e \in E$, existiert ein $D \in \mathcal{E}$ mit $e \in D$. Da $\chi_G^*(S) = E$, gilt $e' \in \chi_G^*(S)$ für alle $e' \in D - e$. Folglich ist $e \in \Phi_G(\chi_G^*(S))$ d.h. $e \in \Phi_G^*(S)$. Ein Widerspruch. ♦

Lemma 5.34. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. Es gilt:

- (i) Falls $\chi_G^*(S) = E$, dann ist $\Phi_G^*(S)$ positiv zyklisch in G , oder es gilt $S = \emptyset$.
- (ii) Falls $\Phi_G^*(S) = E$, dann ist $\Phi_G^*(S)$ positiv zyklisch in G , oder es gilt $S = E$.

Beweis:

Beweis von (i): Sei $\chi_G^*(S) = E$. Sei $S \neq \emptyset$. Nehme an, dass $\Phi_G^*(S)$ positiv azyklisch ist in G . Da $S \subseteq \Phi_G^*(S)$, gilt $\Phi_G^*(S) \neq \emptyset$. Sei $e \in \Phi_G^*(S)$. Da $\chi_G^*(S) = E$, gilt $e \in \chi_G^*(S)$. Gemäss Lemma 5.29. ist $\chi_G^*(S) = \chi_G(\Phi_G^*(S))$. Somit ist $G(\Phi_G^*(S) \cup e) = G(\Phi_G^*(S))$ positiv zyklisch. Ein Widerspruch.

Beweis von (ii): Sei $\Phi_G^*(S) = E$. Sei $S \neq E$. Nehme an, dass $\Phi_G^*(S)$ positiv azyklisch ist in G . Gemäss Lemma 5.29. ist $\Phi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$. Da $\Phi_G^*(S) = E$ und $S \neq E$, existiert ein $e \in \Phi_G(\chi_G^*(S))$. Folglich existiert ein $D \in \mathcal{E}$ mit $e \in D$ so dass $e' \in \chi_G^*(S)$ für alle $e' \in D - e$. Sei $e' \in D - e$. Gemäss Lemma 5.29. ist $\chi_G^*(S) = \chi_G(\Phi_G^*(S))$. Da $e' \in \chi_G^*(S)$ ist $G(\Phi_G^*(S) \cup e')$ positiv zyklisch. Da $\Phi_G^*(S) = E$ ist $e' \in \Phi_G^*(S)$. Folglich ist $G(\Phi_G^*(S))$ positiv zyklisch. Ein Widerspruch. ♦

5.4.4. Global verbotene und erzwungene Bogen

In diesem Abschnitt werden die Mengen $\chi_G(S)$ und $\Phi_G(S)$ definiert, welche die bei Wahl von S global verbotenen bzw. erzwungenen Bogen in einem disjunktiven Graphen G darstellen. Anschliessend wird gezeigt, dass die $\chi_G(S)$ bzw. $\Phi_G(S)$ alle Bogen umfasst, welche in keiner bzw. jeder Selektion S' vorkommen, welche S enthält. Danach wird der Spezialfall untersucht, wenn G keine zulässigen Selektionen besitzt. Schliesslich wird die schon mehrfach erwähnte Tatsache bewiesen, dass die verbotenen bzw. erzwungenen Bogen $\chi_G^*(S)$ bzw. $\Phi_G^*(S)$ eine Teilmenge der global verbotenen bzw. erzwungenen Bogen $\chi_G(S)$ bzw. $\Phi_G(S)$ darstellen. Weiter wird gezeigt, dass im allgemeinen mit den verbotenen bzw. erzwungenen Bogen nicht alle global verbotenen bzw. erzwungenen Bogen erfasst werden.

Definition 5.35. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$.

- (i) Die Menge der global verbotenen disjunktiven Bogen in G bei Wahl von S ist definiert durch:

$$\chi_G(S) = \{e \in E : G(S') \text{ ist positiv zyklisch für alle vollständigen Selektionen } S' \text{ in } G \text{ mit } S \subseteq S' \text{ und } e \in S'\}$$

- (ii) Die Menge der global erzwungenen disjunktiven Bogen in G bei Wahl von S ist definiert durch:

$$\Phi_G(S) = \{e \in E : G(S') \text{ ist positiv zyklisch f\"ur alle vollst\"andigen Selektionen } S' \\ \text{in } G \text{ mit } S \subseteq S' \text{ und } e \notin S'\}$$

F\"ur den Spezialfall $S = \emptyset$ gelten die Bezeichnungen $\chi_G(\emptyset) = \chi(G)$ und $\Phi_G(\emptyset) = \Phi(G)$.

Lemma 5.36. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt:

- (i) $\chi_G(S) = \{e \in E : e \notin S' \text{ f\"ur alle zul\"assigen Selektionen } S' \text{ in } G \text{ mit } S \subseteq S'\}$
- (ii) $\Phi_G(S) = \{e \in E : e \in S' \text{ f\"ur alle zul\"assigen Selektionen } S' \text{ in } G \text{ mit } S \subseteq S'\}$

Daraus folgt insbesondere, dass $\chi_G(S) \cap S' = \emptyset$ und $\Phi_G(S) \subseteq S'$ f\"ur alle zul\"assigen Selektionen S' in G mit $S \subseteq S'$.

Beweis:

Beweis von (i): Bezeichne $W = \{e \in E : e \notin S' \text{ f\"ur alle zul\"assigen Selektionen } S' \text{ in } G \text{ mit } S \subseteq S'\}$. Zeige einerseits, dass $\chi_G(S) \subseteq W$. Sei $e \in \chi_G(S)$. Nehme an, dass $e \notin W$. Dann existiert eine zul\"assige Selektion S' in G mit $S \subseteq S'$ und $e \in S'$. Da S' vollst\"andig ist in G , gilt wegen $e \in \chi_G(S)$, dass $G(S')$ positiv zyklisch ist. Also ist S' nicht zul\"assig in G . Ein Widerspruch. Zeige andererseits, dass $W \subseteq \chi_G(S)$. Sei $e \in W$. Nehme an, dass $e \notin \chi_G(S)$. Dann existiert eine vollst\"andige Selektion S' in G mit $S \subseteq S'$ und $e \in S'$, so dass $G(S')$ positiv azyklisch ist. S' ist eine zul\"assige Selektion in G . Da aber $e \in W$, gilt $e \notin S'$. Ein Widerspruch.

Beweis von (ii): Bezeichne

$W = \{e \in E : e \in S' \text{ f\"ur alle zul\"assigen Selektionen } S' \text{ in } G \text{ mit } S \subseteq S'\}$. Zeige einerseits, dass $\Phi_G(S) \subseteq W$. Sei $e \in \Phi_G(S)$. Zeige, dass $e \in W$. Nehme an, dass $e \notin W$. Dann existiert eine zul\"assige Selektion S' in G mit $S \subseteq S'$ und $e \notin S'$. Da S' vollst\"andig ist in G , gilt wegen $e \in \Phi_G(S)$, dass $G(S')$ positiv zyklisch ist. Also ist S' nicht zul\"assig in G . Ein Widerspruch. Zeige andererseits, dass $W \subseteq \Phi_G(S)$. Sei $e \in W$. Nehme an, dass $e \notin \Phi_G(S)$. Dann existiert eine vollst\"andige Selektion S' in G mit $S \subseteq S'$ und $e \notin S'$, so dass $G(S')$ positiv azyklisch ist. S' ist eine zul\"assige Selektion in G . Da aber $e \in W$, gilt $e \in S'$. Ein Widerspruch. ♦

Lemma 5.37. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. Falls G keine zul\"assige Selektion besitzt, welche S enth\"alt, dann gilt $\chi_G(S) = E$ und $\Phi_G(S) = E$.

Beweis:

Nehme an, dass G keine zul\"assige Selektion besitzt, welche S enth\"alt. Zeige erstens, dass $\chi_G(S) = E$. Da $\chi_G(S) \subseteq E$, muss noch gezeigt werden, dass $E \subseteq \chi_G(S)$. Sei $e \in E$. Nehme an, dass $e \notin \chi_G(S)$. Dann existiert eine vollst\"andige Selektion S' in G mit $S \subseteq S'$ und $e \in S'$, so

dass $G(S')$ positiv azyklisch ist. S' ist eine zulässige Selektion in G mit $S \subseteq S'$. Ein Widerspruch. Zeige zweitens, dass $\Phi_G(S) = E$. Da $\Phi_G(S) \subseteq E$ muss noch gezeigt werden, dass $E \subseteq \Phi_G(S)$. Sei $e \in E$. Nehme an, dass $e \notin \Phi_G(S)$. Dann existiert eine vollständige Selektion S' in G mit $S \subseteq S'$ und $e \notin S'$, so dass $G(S')$ positiv azyklisch ist. S' ist eine zulässige Selektion in G mit $S \subseteq S'$. Ein Widerspruch. ♦

Lemma 5.38. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$.

- (i) Falls $\chi_G(S) = E$, dann besitzt G keine zulässige Selektion, welche S enthält, oder es gilt $S = \emptyset$.
- (ii) Falls $\Phi_G(S) = E$, dann besitzt G keine zulässige Selektion, welche S enthält, oder es gilt $S = E$.

Beweis:

Beweis von (i): Sei $\chi_G(S) = E$. Sei $S \neq \emptyset$. Nehme an, dass eine zulässige Selektion S' in G existiert mit $S \subseteq S'$. Gemäss Lemma 5.36. ist $S \subseteq \Phi_G(S)$. Da $S \neq \emptyset$ ist $\Phi_G(S) \neq \emptyset$. Somit existiert ein $e \in \Phi_G(S)$. Gemäss Lemma 5.36. ist $e \in S'$. Da aber $e \in \chi_G(S)$ gilt gemäss Lemma 5.36., dass $e \notin S'$. Ein Widerspruch.

Beweis von (ii): Sei $\Phi_G(S) = E$. Sei $S \neq E$. Nehme an, dass eine zulässige Selektion S' in G existiert mit $S \subseteq S'$. Gemäss Lemma 5.36. ist $\Phi_G(S) \subseteq S'$ und folglich $S' = E$. Da $S \neq E$, existiert ein $e \in S' - S$. Sei $R = S' - e$. Da S' positiv azyklisch ist in G und $R \subseteq S'$ ist R positiv azyklisch in G . Zeige, dass R vollständig ist in G . Nehme an, dass ein $D \in \mathcal{E}$ existiert mit $D \cap R = \emptyset$. Es gilt $D \cap R = D \cap (S' - e) = (D \cap S') - e$. Da $S' = E$, ist $D \subseteq S'$. Folglich gilt $D \cap R = D - e$. Da $|D| \geq 2$, ist $D \cap R \neq \emptyset$. Ein Widerspruch. Somit ist R vollständig in G . Da $S \subseteq S'$ und $e \notin S$, ist $S \subseteq R$. Also ist R eine zulässige Selektion in G mit $S \subseteq R$. Da $e \in \Phi_G(S)$ gilt gemäss Lemma 5.36., dass $e \in R$. Ein Widerspruch. ♦

Lemma 5.39. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt:

- (i) $\chi_G^*(S) \subseteq \chi_G(S)$
- (ii) $\Phi_G^*(S) \subseteq \Phi_G(S)$

Beweis:

Der Beweis erfolgt mittels Induktion über r . Die Induktion wird simultan für (i) und (ii) durchgeführt. (1) Induktionsbehauptung: Für alle $r \geq 0$ gilt $\mathcal{P}(r)$ (i) $\chi_G^r(S) \subseteq \chi_G(S)$ und (ii) $\Phi_G^r(S) \subseteq \Phi_G(S)$. (2) Induktionsverankerung: Zeige, dass $\mathcal{P}(0)$ gilt. Zu (i): Gemäss Definition ist $\chi_G^0(S) = \emptyset \subseteq \chi_G(S)$. Zu (ii): Gemäss Definition ist $\Phi_G^0(S) = S$. Sei $e \in S$. Nehme an, dass $e \notin \Phi_G(S)$. Dann existiert eine vollständige Selektion S' in G mit $S \subseteq S'$ und $e \notin S'$, so dass $G(S')$ positiv azyklisch ist. Da aber $e \in S$ und $S \subseteq S'$, gilt $e \in S'$. Ein Widerspruch. (3) Indukti-

onsschritt: Zeige, dass für alle $r \geq 0$ gilt: $\mathcal{P}(r) \Rightarrow \mathcal{P}(r+1)$. Sei $r \geq 0$ und gelte $\mathcal{P}(r)$. Zeige, dass $\mathcal{P}(r+1)$ gilt.

Zu (i): Sei $e \in \chi_G^{r+1}(S)$. Zeige, dass $e \in \chi_G(S)$. Nehme an, dass $e \notin \chi_G(S)$. Dann existiert gemäss Definition eine vollständige Selektion S' in G mit $S \subseteq S'$ und $e \in S'$, so dass $G(S')$ positiv azyklisch ist. S' ist zulässig in G . Gemäss Definition ist $\chi_G^{r+1}(S) = \chi_G(\varphi_G^r(S))$. Aus $e \in \chi_G^{r+1}(S)$ folgt somit, dass $G(\varphi_G^r(S) \cup e)$ positiv zyklisch ist. Gemäss $\mathcal{P}(r)$ ist $\varphi_G^r(S) \subseteq \varphi_G(S)$. Da S' eine zulässige Selektion in G ist mit $S \subseteq S'$, gilt gemäss Lemma 5.36., dass $\varphi_G(S) \subseteq S'$ und folglich $\varphi_G^r(S) \subseteq S'$. Somit ist $\varphi_G^r(S) \cup e \subseteq S'$. Da $G(S')$ positiv azyklisch ist, muss auch $G(\varphi_G^r(S) \cup e)$ positiv azyklisch sein. Ein Widerspruch.

Zu (ii): Sei $e \in \varphi_G^{r+1}(S)$. Zeige, dass $e \in \varphi_G(S)$. Nehme an, dass $e \notin \varphi_G(S)$. Dann existiert gemäss Definition eine vollständige Selektion S' in G mit $S \subseteq S'$ und $e \notin S'$, so dass $G(S')$ positiv azyklisch ist. Gemäss Lemma 5.29. ist $\varphi_G^{r+1}(S) = S \cup \Phi_G(\chi_G^{r+1}(S))$. Fall 1: Sei $e \in S$. Wegen $S \subseteq S'$ gilt $e \in S'$. Ein Widerspruch. Fall 2: Sei $e \notin S$. Dann ist $e \in \Phi_G(\chi_G^{r+1}(S))$. Wie schon gezeigt wurde, gilt $\chi_G^{r+1}(S) \subseteq \chi_G(S)$. Daraus folgt $e \in \Phi_G(\chi_G(S))$, d.h. es existiert ein $D \in \mathcal{E}$ mit $e \in D$, so dass $e' \in \chi_G(S)$ für alle $e' \in D - e$. Da S' eine zulässige Selektion in G ist mit $S \subseteq S'$, gilt gemäss Lemma 5.36., dass $S' \cap \chi_G(S) = \emptyset$. Folglich ist $e' \notin S'$ für alle $e' \in D - e$. Da S' vollständig ist in G , muss $D \cap S' \neq \emptyset$ sein. Folglich ist $e \in S'$. Ein Widerspruch. ♦

Satz 5.40. *Es existieren disjunktive Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$ mit $S \subseteq E$, so dass eine oder beide der folgenden Aussagen zutrifft:*

- (i) $\chi_G^*(S) \neq \chi_G(S)$
- (ii) $\varphi_G^*(S) \neq \varphi_G(S)$

Beweis:

(1) Zeige ein Beispiel für G , bei welchem beide Aussagen (i) und (ii) zutreffen. Betrachte Abbildung 5.9. Sei $A = \{a\}$, $\mathcal{E} = \{\{e, e'\}, \{f, f'\}\}$ und $c = 1$. Sei $S = \emptyset$. Es gilt $\chi_G^*(S) = \emptyset$ und $\varphi_G^*(S) = \emptyset$. Die Menge der zulässigen Selektionen in G ist gegeben durch $\{\{e', f\}, \{e', f'\}, \{e', f, f'\}\}$. Somit ist $\chi_G(S) = \{e\}$ und $\varphi_G(S) = \{e'\}$. Folglich trifft (i) und (ii) zu.

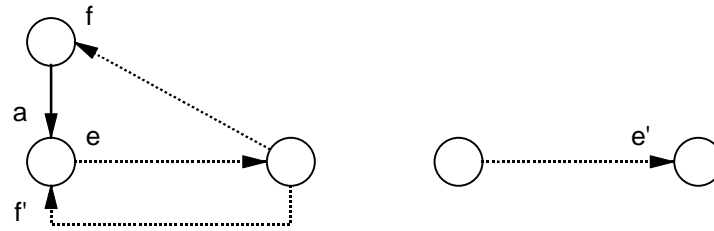


Abbildung 5.9. Disjunktiver Graph G , für welchen (i) und (ii) zutrifft

(2) Zeige ein Beispiel für G , bei welchem nur Aussage (i) zutrifft. Betrachte Abbildung 5.10. Sei $A = \{a, b\}$, $\mathcal{E} = \{\{e, e'\}, \{e', e''\}, \{f, f'\}\}$ und $c > 0$. Sei $S = \emptyset$. Es gilt $\chi_G^*(S) = \{a, b\}$ und $\phi_G^*(S) = \{e'\}$. Die Menge der zulässigen Selektionen in G ist gegeben durch $\{\{e', f\}, \{e', f'\}, \{e', f, f'\}\}$. Somit ist $\chi_G(S) = \{e, e''\}$ und $\phi_G(S) = \{e'\}$. Folglich trifft nur (i) zu.

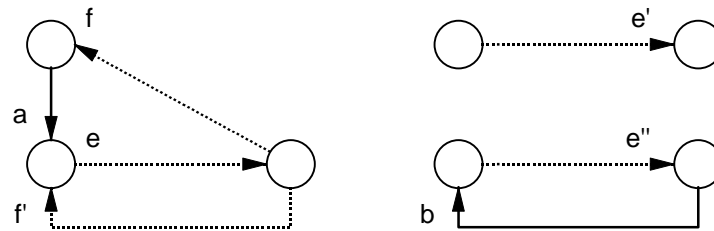


Abbildung 5.10. Disjunktiver Graph G , für welchen nur (i) zutrifft

Zeige ein Beispiel für G , bei welchem nur Aussage (ii) zutrifft. Betrachte Abbildung 5.11. Sei $A = \emptyset$, $\mathcal{E} = \{\{e, e'\}, \{e, e''\}\}$ und $c > 0$. Sei $S = \emptyset$. Es gilt $\chi_G^*(S) = \emptyset$ und $\phi_G^*(S) = \emptyset$. Die Menge der zulässigen Selektionen ist gegeben durch $\{\{e\}, \{e, e'\}, \{e, e''\}\}$. Somit ist $\chi_G(S) = \emptyset$ und $\phi_G(S) = \{e\}$. Folglich trifft nur (ii) zu. ♦

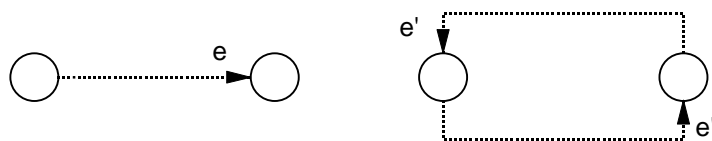


Abbildung 5.11. Disjunktiver Graph G , für welchen nur (ii) zutrifft

5.4.5. Allgemeines Prinzip der konjunktiven Erweiterung

Es wurde schon früher darauf hingewiesen, dass die zur Bestimmung der verbotenen bzw. erzwungenen Bogen benützten Konzepte in einem gewissen Sinne eine lokale Sichtweise darstellen. In diesem Abschnitt wird die Bedeutung dieser lokalen Sichtweise näher erläutert. Davon ausgehend wird eine Verallgemeinerung des Konzepts der verbotenen und erzwungenen Bogen erarbeitet.

Definition 5.41. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$ und $\mathcal{F} \subseteq \mathcal{E}$.

- (i) Die Menge der durch S direkt verbotenen disjunktiven Bogen bezüglich \mathcal{F} in G ist gegeben durch:

$$\chi_{G, \mathcal{F}}(S) = \{e \in E_{G_{\mathcal{F}}} : G(S \cup R) \text{ ist positiv zyklisch} \\ \text{für alle vollständigen Selektionen } R \text{ in } G_{\mathcal{F}} \text{ mit } e \in R\}$$

- (ii) Die Menge der durch S direkt erzwungenen disjunktiven Bogen bezüglich \mathcal{F} in G ist gegeben durch:

$$\varphi_{G, \mathcal{F}}(S) = S \cup \{e \in E_{G_{\mathcal{F}}} : G(S \cup R) \text{ ist positiv zyklisch} \\ \text{für alle vollständigen Selektionen } R \text{ in } G_{\mathcal{F}} \text{ mit } e \notin R\}$$

In diesen Definitionen wird das in Abschnitt 5.4.2. eingeführte Konzept der direkt verbotenen bzw. erzwungenen Bogen, bezeichnet als $\chi_G(S)$ und $\varphi_G(S)$ verallgemeinert für beliebige Teilfamilien $\mathcal{F} \subseteq \mathcal{E}$ von disjunktiven Mengen. Das folgende Lemma zeigt, dass die so definierten Bogenmengen für den Spezialfall $\mathcal{F} = \mathcal{E}$ den global verbotenen bzw. erzwungenen Bogen entsprechen.

Lemma 5.42. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. Es gilt:

- (i) $\chi_{G, \mathcal{E}}(S) = \chi_G(S)$
(ii) $\varphi_{G, \mathcal{E}}(S) = \varphi_G(S)$

Beweis:

Beweis von (i): Zeige einerseits, dass $\chi_{G, \mathcal{E}}(S) \subseteq \chi_G(S)$ Sei $e \in \chi_{G, \mathcal{E}}(S)$ Nehme an, dass $e \notin \chi_G(S)$. Dann existiert eine vollständige Selektion S' in G mit $S \subseteq S'$ und $e \in S'$ so dass $G(S')$ positiv azyklisch ist. Sei $R = S'$. R ist eine vollständige Selektion in $G_{\mathcal{E}} = G$ mit $e \in R$ Da $e \in \chi_{G, \mathcal{E}}(S)$, ist $G(S \cup R)$ positiv zyklisch. Da $S \cup R = S'$ ist auch $G(S')$ positiv zyklisch. Ein Widerspruch. Zeige andererseits, dass $\chi_G(S) \subseteq \chi_{G, \mathcal{E}}(S)$ Sei $e \in \chi_G(S)$ Nehme an, dass $e \notin \chi_{G, \mathcal{E}}(S)$. Dann existiert eine vollständige Selektion R in $G_{\mathcal{E}} = G$ mit $e \in R$ so dass $G(S \cup R)$ positiv azyklisch ist. Sei $S' = S \cup R$ S' ist eine vollständige Selektion in G mit $S \subseteq S'$

und $e \in S'$. Da $e \in \chi_G(S)$ ist $G(S')$ positiv zyklisch. Da $S' = S \cup R$ ist auch $G(S \cup R)$ positiv zyklisch. Ein Widerspruch.

Beweis von (ii): Zeige einerseits, dass $\Phi_{G, \mathcal{E}}(S) \subseteq \Phi_G(S)$. Sei $e \in \Phi_{G, \mathcal{E}}(S)$. Nehme an, dass $e \notin \Phi_G(S)$. Dann existiert eine vollständige Selektion S' in G mit $S \subseteq S'$ und $e \notin S'$ so dass $G(S')$ positiv azyklisch ist. Sei $R = S'$. R ist eine vollständige Selektion in $G_{\mathcal{E}} = G$ mit $e \notin R$. Da $e \in \Phi_{G, \mathcal{E}}(S)$, ist $G(S \cup R)$ positiv zyklisch. Da $S \cup R = S'$ ist auch $G(S')$ positiv zyklisch. Ein Widerspruch. Zeige andererseits, dass $\Phi_G(S) \subseteq \Phi_{G, \mathcal{E}}(S)$. Sei $e \in \Phi_G(S)$. Nehme an, dass $e \notin \Phi_{G, \mathcal{E}}(S)$. Dann ist $e \notin S$, und es existiert eine vollständige Selektion R in $G_{\mathcal{E}} = G$ mit $e \notin R$ so dass $G(S \cup R)$ positiv azyklisch ist. Sei $S' = S \cup R$. S' ist eine vollständige Selektion in G mit $S \subseteq S'$ und $e \notin S'$. Da $e \in \Phi_G(S)$ ist $G(S')$ positiv zyklisch. Da $S' = S \cup R$ ist auch $G(S \cup R)$ positiv zyklisch. Ein Widerspruch. ♦

Als nächstes wird gezeigt, dass die Mengenfunktionen $\chi_{G, \mathcal{F}}$ und $\Phi_{G, \mathcal{F}}$ monoton sind bezüglich \mathcal{F} . Dies bedeutet, dass die bezüglich \mathcal{F} direkt verbotenen bzw. erzwungenen Bogen in den bezüglich \mathcal{F}' direkt verbotenen bzw. erzwungenen Bogen enthalten sind, falls \mathcal{F} in \mathcal{F}' enthalten ist. Daraus folgt insbesondere, dass die bezüglich einer beliebigen Teilfamilie \mathcal{F} verbotenen bzw. erzwungenen Bogen auch global verboten bzw. erzwungen sind.

Lemma 5.43. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$ und $\mathcal{F}, \mathcal{F}' \subseteq \mathcal{E}$ mit $\mathcal{F} \subseteq \mathcal{F}'$. Es gilt:

$$(i) \quad \chi_{G, \mathcal{F}}(S) \subseteq \chi_{G, \mathcal{F}'}(S)$$

$$(ii) \quad \Phi_{G, \mathcal{F}}(S) \subseteq \Phi_{G, \mathcal{F}'}(S)$$

Daraus folgt insbesondere, dass $\chi_{G, \mathcal{F}}(S) \subseteq \chi_G(S)$ und $\Phi_{G, \mathcal{F}}(S) \subseteq \Phi_G(S)$.

Beweis:

Beweis von (i): Sei $e \in \chi_{G, \mathcal{F}}(S)$. Nehme an, dass $e \notin \chi_{G, \mathcal{F}'}(S)$. Da $e \in \chi_{G, \mathcal{F}}(S)$ ist $e \in E_{G_{\mathcal{F}}}$. Da $\mathcal{F} \subseteq \mathcal{F}'$, ist $E_{G_{\mathcal{F}}} \subseteq E_{G_{\mathcal{F}'}}$ und folglich $e \in E_{G_{\mathcal{F}'}}$. Da $e \notin \chi_{G, \mathcal{F}'}(S)$ existiert somit eine vollständige Selektion R' in $G_{\mathcal{F}'}$ mit $e \in R'$ so dass $G(S \cup R')$ positiv azyklisch ist. Sei $R = R' \cap E_{G_{\mathcal{F}}}$. Da $R \subseteq E_{G_{\mathcal{F}'}}$ ist R eine Selektion in $G_{\mathcal{F}}$. Da R' vollständig ist in $G_{\mathcal{F}'}$ ist R vollständig in $G_{\mathcal{F}}$. Da $e \in R'$ und $e \in E_{G_{\mathcal{F}'}}$ ist $e \in R$. Folglich ist R eine vollständige Selektion in $G_{\mathcal{F}}$ mit $e \in R$. Da $e \in \chi_{G, \mathcal{F}}(S)$ ist folglich $G(S \cup R)$ positiv zyklisch. Da aber $R \subseteq R'$ ist auch $G(S \cup R')$ positiv zyklisch. Ein Widerspruch.

Beweis von (ii): Sei $e \in \Phi_{G, \mathcal{F}}(S)$. Nehme an, dass $e \notin \Phi_{G, \mathcal{F}'}(S)$. Da $e \in \Phi_{G, \mathcal{F}}(S)$ ist $e \in E_{G_{\mathcal{F}}}$. Da $\mathcal{F} \subseteq \mathcal{F}'$, ist $E_{G_{\mathcal{F}}} \subseteq E_{G_{\mathcal{F}'}}$ und folglich $e \in E_{G_{\mathcal{F}'}}$. Da $e \notin \Phi_{G, \mathcal{F}'}(S)$ existiert somit eine vollständige Selektion R' in $G_{\mathcal{F}'}$ mit $e \notin R'$ so dass $G(S \cup R')$ positiv azyklisch ist. Sei

$R = R' \cap E_{G_{\mathcal{F}}}$. Da $R \subseteq E_{G_{\mathcal{F}}}$ ist R eine Selektion in $G_{\mathcal{F}}$. Da R' vollständig ist in $G_{\mathcal{F}}$ ist R vollständig in $G_{\mathcal{F}}$. Da $e \notin R'$, gilt $e \notin R$. Folglich ist R eine vollständige Selektion in $G_{\mathcal{F}}$ mit $e \notin R$. Da $e \in \Phi_{G, \mathcal{F}}(S)$ und $e \notin R$ gilt somit, dass $G(S \cup R)$ positiv zyklisch ist. Da aber $R \subseteq R'$ ist auch $G(S \cup R')$ positiv zyklisch. Ein Widerspruch.

Zeige noch, dass $\chi_{G, \mathcal{F}}(S) \subseteq \chi_G(S)$ und $\Phi_{G, \mathcal{F}}(S) \subseteq \Phi_G(S)$. Da $\chi_{G, \mathcal{F}}(S) \subseteq \chi_{G, \mathcal{E}}(S)$ und $\Phi_{G, \mathcal{F}}(S) \subseteq \Phi_{G, \mathcal{E}}(S)$, folgt gemäss Lemma 5.42. die Behauptung. ♦

Im folgenden Lemma erklärt, inwiefern das Prinzip der direkt verbotenen bzw. erzwungenen Bogen eine lokale Sichtweise darstellt. Es wird gezeigt, dass die Menge der direkt verbotenen bzw. erzwungenen Bogen erhalten wird, indem nur Teilfamilien $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| = 1$ betrachtet werden, bezüglich welcher die direkt verbotenen bzw. erzwungenen Bogen bestimmt werden.

Lemma 5.44. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Es gilt:

- (i) $\chi_G(S) = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1} \chi_{G, \mathcal{F}}(S)$
- (ii) $\Phi_G(S) = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1} \Phi_{G, \mathcal{F}}(S)$

Beweis:

Beweis von (i): Bezeichne $W = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1} \chi_{G, \mathcal{F}}(S)$

Zeige einerseits, dass $\chi_G(S) \subseteq W$. Sei $e \in \chi_G(S)$. Nehme an, dass $e \notin W$. Dann existiert kein $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| = 1$, so dass $e \in \chi_{G, \mathcal{F}}(S)$. Sei $D \in \mathcal{E}$ mit $e \in D$. Sei $\mathcal{F} = \{D\}$. Da $|\mathcal{F}| = 1$, gilt $e \in \chi_{G, \mathcal{F}}(S)$. Da $e \in D$ und $E_{G_{\mathcal{F}}} = \mathcal{F}^\circ = D$ ist $e \in E_{G_{\mathcal{F}}}$. Somit existiert eine vollständige Selektion R in $G_{\mathcal{F}}$ mit $e \in R$, so dass $G(S \cup R)$ positiv azyklisch ist. Da $e \in R$, muss auch $G(S \cup e)$ positiv azyklisch sein. Da aber $e \in \chi_G(S)$ ist $G(S \cup e)$ positiv zyklisch. Ein Widerspruch.

Zeige andererseits, dass $W \subseteq \chi_G(S)$. Sei $e \in W$. Nehme an, dass $e \notin \chi_G(S)$. Dann ist $G(S \cup e)$ positiv azyklisch. Da $e \in W$, existiert ein $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| = 1$, so dass $e \in \chi_{G, \mathcal{F}}(S)$. Sei $R = \{e\}$. Da $e \in \chi_{G, \mathcal{F}}(S)$ ist $e \in E_{G_{\mathcal{F}}}$ d.h. R ist eine Selektion in $G_{\mathcal{F}}$. Da $|\mathcal{F}| = 1$ gilt $\mathcal{F} = \{D\}$ für ein $D \in \mathcal{E}$. Da $E_{G_{\mathcal{F}}} = \mathcal{F}^\circ = D$ ist $e \in D$. Da $\mathcal{E}_{G_{\mathcal{F}}} = \mathcal{F} = \{D\}$ ist $e \in E_{G_{\mathcal{F}}}$. Dann existiert eine vollständige Selektion in $G_{\mathcal{F}}$ mit $e \in R$. Folglich gilt wegen $e \in \chi_{G, \mathcal{F}}(S)$, dass $G(S \cup R) = G(S \cup e)$ positiv zyklisch ist. Da aber $e \notin \chi_G(S)$ ist $G(S \cup e)$ positiv azyklisch. Ein Widerspruch.

Beweis von (ii): Bezeichne $W = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1} \Phi_{G, \mathcal{F}}(S)$

Zeige einerseits, dass $\Phi_G(S) \subseteq W$. Sei $e \in \Phi_G(S)$. Nehme an, dass $e \notin W$. Da $e \in \Phi_G(S)$ und $\Phi_G(S) = S \cup \Phi_G(\chi_G(S))$, gilt $e \in S$ oder $e \in \Phi_G(\chi_G(S))$. Fall 1: Sei $e \in S$. Da $E \neq \emptyset$ und folglich $\mathcal{E} \neq \emptyset$, existiert ein $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| = 1$. Da $\Phi_{G, \mathcal{F}}(S) \subseteq W$ und $S \subseteq \Phi_{G, \mathcal{F}}(S)$ gilt $e \in W$. Ein Widerspruch. Fall 2: Sei $e \in \Phi_G(\chi_G(S))$. Dann existiert ein $D \in \mathcal{E}$ mit $e \in D$. So dass $e' \in \chi_G(S)$

für alle $e' \in D - e$. Somit ist $G(S \cup e')$ positiv zyklisch für alle $e' \in D - e$. Da $e \notin W$, existiert kein $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| = 1$ so dass $e \in \Phi_{G, \mathcal{F}}(S)$. Folglich gilt für alle $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| \geq 2$ dass $e \notin \Phi_{G, \mathcal{F}}(S)$. Sei $\mathcal{F} = \{D\}$. Da $E_{G_{\mathcal{F}}} = \mathcal{F}^\circ = D$ und $e \in D$ gilt $e \in E_{G_{\mathcal{F}}}$. Da $e \notin \Phi_{G, \mathcal{F}}(S)$ und $e \in E_{G_{\mathcal{F}}}$, existiert eine vollständige Selektion R in $G_{\mathcal{F}}$ mit $e \notin R$ so dass $G(S \cup R)$ positiv azyklisch ist. Da $\mathcal{E}_{G_{\mathcal{F}}} = \mathcal{F} = \{D\}$ und R vollständig ist in $G_{\mathcal{F}}$ gilt $R \cap D \neq \emptyset$. Da $e \notin R$ existiert somit ein $e' \in D - e$ mit $e' \in R$. Da aber $G(S \cup e')$ positiv zyklisch ist, muss auch $G(S \cup R)$ positiv zyklisch sein. Ein Widerspruch.

Zeige andererseits, dass $W \subseteq \Phi_G(S)$. Sei $e \in W$. Nehme an, dass $e \notin \Phi_G(S)$. Da $\Phi_G(S) = S \cup \Phi_G(\chi_G(S))$, gilt $e \notin S$ und $e \notin \Phi_G(\chi_G(S))$. Da $e \notin \Phi_G(\chi_G(S))$ existiert für alle $D \in \mathcal{E}$ mit $e \in D$ ein $e' \in D - e$ so dass $e' \notin \chi_G(S)$. Somit existiert für alle $D \in \mathcal{E}$ mit $e \in D$ ein $e' \in D - e$, so dass $G(S \cup e')$ positiv azyklisch ist. Da $e \in W$ existiert ein $\mathcal{F} \subseteq \mathcal{E}$ mit $|\mathcal{F}| = 1$, so dass $e \in \Phi_{G, \mathcal{F}}(S)$. Somit gilt für alle vollständigen Selektionen R in $G_{\mathcal{F}}$ mit $e \notin R$, dass $G(S \cup R)$ positiv zyklisch ist. Da $|\mathcal{F}| = 1$, gilt $\mathcal{F} = \{D\}$ für ein $D \in \mathcal{E}$. Da $e \in \Phi_{G, \mathcal{F}}(S)$ ist $e \in E_{G_{\mathcal{F}}}$. Da $E_{G_{\mathcal{F}}} = \mathcal{F}^\circ = D$ gilt somit $e \in D$. Sei $e' \in D - e$ und $R = \{e'\}$. Da $e' \in E_{G_{\mathcal{F}}}$ ist R eine Selektion in $G_{\mathcal{F}}$. Da $\mathcal{E}_{G_{\mathcal{F}}} = \mathcal{F} = \{D\}$ ist R eine vollständige Selektion in $G_{\mathcal{F}}$ mit $e \notin R$. Folglich gilt wegen $e \in \Phi_{G, \mathcal{F}}(S)$ dass $G(S \cup R) = G(S \cup e')$ positiv zyklisch ist. Somit ist gezeigt, dass $G(S \cup e')$ positiv zyklisch ist für alle $e' \in D - e$. Da aber $e \notin \Phi_G(\chi_G(S))$ existiert ein $e' \in D - e$ so dass $G(S \cup e')$ positiv azyklisch ist. Ein Widerspruch. ♦

Aus den vorangehenden Ausführungen lässt sich ein verallgemeinertes Konzept zur Bestimmung von verbotenen und erzwungenen Bogen ableiten. Gemäss Lemma 5.43. gilt $\chi_{G, \mathcal{F}}(S) \subseteq \chi_G(S)$ und $\Phi_{G, \mathcal{F}}(S) \subseteq \Phi_G(S)$ für beliebige Teilfamilien $\mathcal{F} \subseteq \mathcal{E}$. Folglich gilt auch für beliebige Mengen $F \subseteq \text{Pot}(\mathcal{E})$ von Teilfamilien, dass $\bigcup_{\mathcal{F} \subseteq F} \chi_{G, \mathcal{F}}(S) \subseteq \chi_G(S)$ bzw. $\bigcup_{\mathcal{F} \subseteq F} \Phi_{G, \mathcal{F}}(S) \subseteq \Phi_G(S)$. Somit können als Grundlage zur Bestimmung von verbotenen bzw. erzwungenen Bogen beliebige Mengenfunktionen $\chi_G^F(S) = \bigcup_{\mathcal{F} \subseteq F} \chi_{G, \mathcal{F}}(S)$ bzw. $\Phi_G^F(S) = \bigcup_{\mathcal{F} \subseteq F} \Phi_{G, \mathcal{F}}(S)$ gewählt werden. Durch Iteration dieser Funktionen analog zu $\chi_G^r(S)$ und $\Phi_G^r(S)$ resultiert daraus ein verallgemeinertes Konzept zur Bestimmung von verbotenen bzw. erzwungenen Bogen. $\chi_G^r(S)$ und $\Phi_G^r(S)$ stellen einen Spezialfall dieses Konzepts dar, wo nur Familien von disjunktiven Mengen betrachtet werden, welche genau eine disjunktive Menge umfassen. Eine mögliche Verallgemeinerung wäre beispielsweise, alle Familien zu betrachten, welche zwei oder allgemeiner $k \geq 1$ disjunktive Mengen umfassen.

Es ist offensichtlich, dass der Rechenaufwand zunimmt, je grösser die betrachteten Familien $\mathcal{F} \subseteq \mathcal{E}$ sind und je umfangreicher $F \subseteq \text{Pot}(\mathcal{E})$ gewählt wird. Während der Aufwand zur Bestimmung von $\chi_G^*(S)$ und $\Phi_G^*(S)$ polynomial ist (vgl. Abschnitt 5.4.8.), wird sich zeigen, dass der Aufwand zur Bestimmung von $\chi_{G, \mathcal{E}}(S) = \chi_G(S)$ und $\Phi_{G, \mathcal{E}}(S) = \Phi_G(S)$ im Gegensatz dazu NP-hart ist (vgl. Abschnitt 5.5.2.). Zum Abschluss sollen die erarbeiteten Konzepte anhand eines Beispiels illustriert werden.

Beispiel:

Betrachte den nachfolgenden disjunktiven Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$ wobei $A = \emptyset$, $\mathcal{E} = \{\{e, f\}, \{e, g\}, \{h, i\}\}$ und $c > 0$. Sei $S = \{i\}$. Für $\mathcal{F} \subseteq \mathcal{E}$ bezeichne $\mathcal{R}_{\mathcal{F}}^-$ die Menge der vollständigen Selektionen R in $G_{\mathcal{F}}$ für welche $G(S \cup R)$ positiv azyklisch ist, und $\mathcal{R}_{\mathcal{F}}^+$ die Menge der vollständigen Selektionen R in $G_{\mathcal{F}}$ für welche $G(S \cup R)$ positiv zyklisch ist.

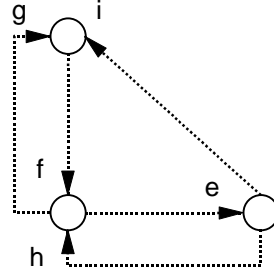


Abbildung 5.12. Verallgemeinertes Prinzip der konjunktiven Erweiterung.

Im folgenden werden für alle Teilfamilien $\mathcal{F} \subseteq \mathcal{E}$, $\mathcal{F} \neq \emptyset$ die entsprechenden durch \mathcal{S} direkt verbotenen bzw. erzwungenen Bogen $\chi_{G, \mathcal{F}}(S)$ und $\Phi_{G, \mathcal{F}}(S)$ berechnet:

- $\mathcal{F} = \{\{e, f\}\}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{e\}, \{f\}\}, \mathcal{R}_{\mathcal{F}}^+ = \{\{e, f\}\}, \chi_{G, \mathcal{F}}(S) = \emptyset, \Phi_{G, \mathcal{F}}(S) = \{i\}$$

- $\mathcal{F} = \{\{e, g\}\}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{e\}, \{g\}, \{e, g\}\}, \mathcal{R}_{\mathcal{F}}^+ = \emptyset, \chi_{G, \mathcal{F}}(S) = \emptyset, \Phi_{G, \mathcal{F}}(S) = \{i\}$$

- $\mathcal{F} = \{\{h, i\}\}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{h\}, \{i\}, \{h, i\}\}, \mathcal{R}_{\mathcal{F}}^+ = \emptyset, \chi_{G, \mathcal{F}}(S) = \emptyset, \Phi_{G, \mathcal{F}}(S) = \{i\}$$

- $\mathcal{F} = \{\{e, f\}, \{e, g\}\}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{e\}, \{e, g\}\}, \mathcal{R}_{\mathcal{F}}^+ = \{\{e, f\}, \{f, g\}, \{e, f, g\}\}, \chi_{G, \mathcal{F}}(S) = \{f\}, \Phi_{G, \mathcal{F}}(S) = \{e, i\}$$

- $\mathcal{F} = \{\{e, f\}, \{h, i\}\}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{e, i\}, \{f, h\}, \{f, i\}, \{f, h, i\}\},$$

$$\mathcal{R}_{\mathcal{F}}^+ = \{\{e, h\}, \{e, f, h\}, \{e, f, i\}, \{e, h, i\}, \{e, f, h, i\}\},$$

$$\chi_{G, \mathcal{F}}(S) = \emptyset, \Phi_{G, \mathcal{F}}(S) = \{i\}$$

- $\mathcal{F} = \{\{e, g\}, \{h, i\}\}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{e, i\}, \{g, h\}, \{g, i\}, \{e, g, i\}, \{g, h, i\}\},$$

$$\mathcal{R}_{\mathcal{F}}^+ = \{\{e, h\}, \{e, g, h\}, \{e, h, i\}, \{e, g, h, i\}\},$$

$$\chi_{G, \mathcal{F}}(S) = \emptyset, \quad \Phi_{G, \mathcal{F}}(S) = \{i\}$$

- $\mathcal{F} = \{\{e, f\}, \{e, g\}, \{h, i\}\} = \mathcal{E}$:

$$\mathcal{R}_{\mathcal{F}}^- = \{\{e, i\}, \{e, g, i\}\},$$

$$\mathcal{R}_{\mathcal{F}}^+ = \{\{e, h\}, \{e, f, h\}, \{e, f, i\}, \{e, g, h\}, \{f, g, h\}, \{f, g, i\},$$

$$\{e, f, g, h\}, \{e, f, g, i\}, \{f, g, h, i\}, \{e, f, g, h, i\}\}$$

$$\chi_{G, \mathcal{F}}(S) = \{f, h\}, \quad \Phi_{G, \mathcal{F}}(S) = \{e, i\}$$

Betrachte nun beispielsweise die beiden Mengen von Teilfamilien $F_1 = \{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1\}$ und $F_2 = \{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=2\}$. Es gilt:

$$\chi_{G, F_1}^1(S) = \chi_G(S) = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1} \chi_{G, \mathcal{F}}(S) = \emptyset, \quad \Phi_{G, F_1}^1(S) = \Phi_G(S) = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=1} \Phi_{G, \mathcal{F}}(S) = \{i\}$$

$$\chi_{G, F_2}^2(S) = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=2} \chi_{G, \mathcal{F}}(S) = \{f\}, \quad \Phi_{G, F_2}^2(S) = \bigcup_{\mathcal{F} \subseteq \mathcal{E}, |\mathcal{F}|=2} \Phi_{G, \mathcal{F}}(S) = \{e, i\}$$

Die Menge der zulässigen Selektionen S' in G mit $S \subseteq S'$ ist gegeben durch $\{\{e, i\}, \{e, g, i\}\}$. Daraus lässt sich erkennen, dass die Menge der global verbotenen bzw. erzwungenen Bogen gegeben ist durch:

$$\chi_G(S) = \chi_{G, \mathcal{E}}(S) = \{f, h\}, \quad \Phi_G(S) = \Phi_{G, \mathcal{E}}(S) = \{e, i\}$$

Somit werden weder durch F_1 noch durch F_2 alle global verbotenen bzw. erzwungenen Bogen erfasst. ♦

5.4.6. Relationen

Im Zusammenhang mit dem Konzept der verbotenen bzw. erzwungenen Bogen in einem disjunktiven Graphen G drängen sich Fragen der folgenden Form auf: Wenn ein Bogen e einen Bogen f erzwingt, gilt dann umgekehrt auch, dass f den Bogen e erzwingt? Oder: Wenn ein Bogen e einen Bogen f erzwingt, und f einen Bogen g , erzwingt dann e auch g ?

Allgemeiner ausgedrückt stellt sich die Frage, welche Eigenschaften die Relationen " e verbietet f " bzw. " e erzwingt f " aufweisen. Unter den Relationen " e verbietet f " bzw. " e erzwingt f " werden hierbei die Relationen " $f \in \chi_G^*(e)$ " bzw. " $f \in \Phi_G^*(e)$ " verstanden. Diese beiden Relationen sollen im folgenden untersucht werden in Hinblick auf drei grundlegende Eigenschaften: (1) Reflexivität, (2) Symmetrie und (3) Transitivität. Die Bedeutung dieser Eigenschaften für eine allgemeine Relation $R \subseteq E^2$ ist folgende: (1) R ist reflexiv, falls $(e, e) \in R$ für alle $e \in E$ (2) R ist symmetrisch,

falls aus $(e, f) \in R$ folgt $(f, e) \in R$ für alle $e, f \in R$ (3) Ist transitiv, falls aus $(e, f) \in R$ und $(f, g) \in R$ folgt $(e, g) \in R$ für alle $e, f, g \in R$

Die Relation " $f \in \phi_G^*(e)$ " ist offensichtlich reflexiv. Auf den ersten Blick würde man vielleicht vermuten, dass sie auch die Eigenschaften der Symmetrie und Transitivität aufweist. In diesem Fall würde sie eine Äquivalenzrelation darstellen, d.h. die Menge der disjunktiven Bogen liesse sich partitionieren in Äquivalenzklassen, deren Elemente sich gegenseitig erzwingen. Dies ist jedoch nicht der Fall, wie im folgenden gezeigt wird. Als Verallgemeinerung werden in den nachfolgenden Ausführungen diese Relationen nicht nur bezüglich χ_G^* und ϕ_G^* betrachtet, sondern bezüglich aller Funktionen χ_G^r und ϕ_G^r mit $r \geq 1$.

Das Ziel dieses Abschnittes ist einerseits, zu zeigen, dass " $f \in \phi_G^*(e)$ " keine Äquivalenzrelation darstellt, und andererseits, durch die angeführten Gegenbeispiele ein Vorstellung davon zu vermitteln, wie das Verbieten und Erzwingen von Bogen in disjunktiven Graphen zu Stande kommt.

Definition 5.45. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Es werden die folgenden Relationen auf E definiert:

- (i) " e verbietet r -stufig f in G ": $\mathcal{R}(\chi_G^r) = \{(e, f) \in E^2 : f \in \chi_G^r(e)\}$
- (ii) " e erzwingt r -stufig f in G ": $\mathcal{R}(\phi_G^r) = \{(e, f) \in E^2 : f \in \phi_G^r(e)\}$

Satz 5.46. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Die folgende Tabelle zeigt für die Relationen $\mathcal{R}(\chi_G^r)$ und $\mathcal{R}(\phi_G^r)$, welche Eigenschaften für welche Werte von $r \geq 1$ vorhanden sind.

Relation:	reflexiv	symmetrisch	transitiv
$\mathcal{R}(\chi_G^r)$	-	$r = 1$	-
$\mathcal{R}(\phi_G^r)$	$r \geq 1$	-	$r = \infty$

Abbildung 5.13. Eigenschaften der Relationen

Beweis:

Beweis für $\mathcal{R}(\chi_G^r)$:

(1) Nicht reflexiv für alle $r \geq 1$: Zeige: Für alle $r \geq 1$ existiert ein disjunktiver Graph G und ein $e \in E$, so dass $e \notin \chi_G^r(e)$. Sei $r \geq 1$ und G ein beliebiger disjunktiver Graph, für welchen eine zulässige Selektion $S \subseteq E$ existiert. Sei $e \in S$. Nehme an, dass $e \in \chi_G^r(e)$. Da $e \in S$, ist aufgrund der Monotonie $\chi_G^r(e) \subseteq \chi_G^r(S)$. Gemäss Lemma 5.39. ist $\chi_G^r(S) \subseteq \chi_G^r(S)$. Da S zulässig ist in G

und $e \in S$, gilt gemäss Lemma 5.36., dass $\chi_G(e) \cap S = \emptyset$. Da aber $e \in \chi_G^r(e)$ und $e \in S$ ist $e \in \chi_G(e) \cap S$. Ein Widerspruch.

(2) Symmetrisch für $r = 1$: Zeige erstens: Für alle disjunktiven Graphen G und alle $e, f \in E$ gilt, dass aus $f \in \chi_G(e)$ folgt $e \in \chi_G(f)$. Sei G ein disjunktiver Graph und $e, f \in E$ mit $f \in \chi_G(e)$. Da $f \in \chi_G(e)$, ist $G(e \cup f)$ positiv zyklisch. Da $G(e \cup f) = G(f \cup e)$ gilt $e \in \chi_G(f)$. Zeige zweitens: Für alle $r \geq 2$ existiert ein disjunktiver Graph G und $e, f \in E$ so dass aus $f \in \chi_G^r(e)$ nicht folgt $e \in \chi_G^r(f)$. Betrachte den disjunktiven Graphen G in Abbildung 5.14., wobei $\mathcal{E} = \{\{e, e'\}, \{f, f'\}, \{g, g'\}, \{h, h'\}\}$ und $c > 0$ ist. Es ist $\chi_G^2(e) = \chi_G^*(e) = \{e', f', g', h'\}$ und $\chi_G^2(h') = \chi_G^*(h') = \{h\}$. Somit gilt für alle $r \geq 2$, dass $h' \in \chi_G^r(e)$ aber nicht $e \in \chi_G^r(h')$.

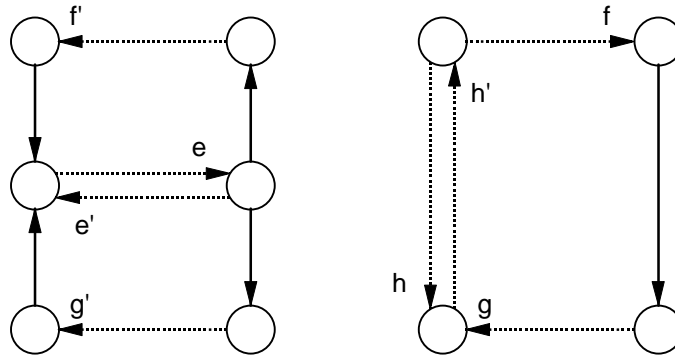


Abbildung 5.14. G mit nicht-symmetrischer Relation $\mathcal{R}(\chi_G^r)$ für $r \geq 2$

(3) Nicht transitiv für alle $r \geq 1$: Zeige: Für alle $r \geq 1$ existiert ein disjunktiver Graph G und $e, f, g \in E$, so dass aus $f \in \chi_G^r(e)$ und $g \in \chi_G^r(f)$ nicht folgt $g \in \chi_G^r(e)$. Betrachte den disjunktiven Graphen G (ein Lager-Graph) in Abbildung 5.15., wobei $\mathcal{E} = \{\{e, e'\}, \{f, f'\}, \{g, g'\}\}$ und

$c > 0$. Es ist $\chi_G^1(e) = \{e', f'\}$ und $\chi_G^2(e) = \chi_G^*(e) = \{e', f', g\}$ sowie $\chi_G^1(f') = \chi_G^*(f') = \{e, f, g\}$.
Somit gilt für alle $r \geq 1$, dass $f' \in \chi_G^r(e)$ und $g \in \chi_G^r(f')$, aber nicht $g \in \chi_G^r(e)$.

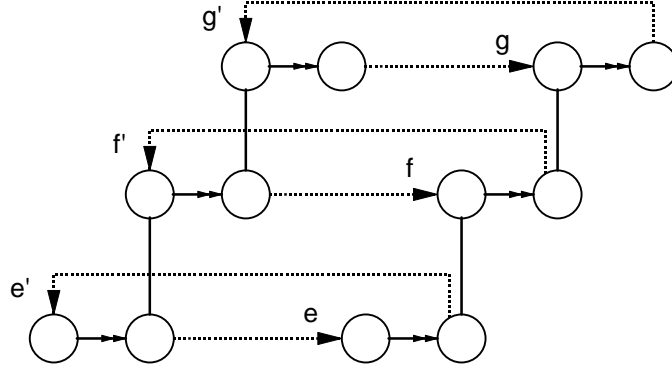


Abbildung 5.15. G mit nicht-transitiver Relation $\mathcal{R}(\chi_G^r)$ für $r \geq 1$

Beweis für $\mathcal{R}(\phi_G^r)$:

- (1) reflexiv für alle $r \geq 1$: Zeige: Für alle $r \geq 1$ und alle disjunktiven Graphen G und alle $e \in E$ gilt, dass $e \in \phi_G^r(e)$. Sei $r \geq 1$. Sei G ein disjunktiver Graph und $e \in E$. Gemäss Lemma 5.25. ist $\{e\} = \phi_G^0(e) \subseteq \phi_G^r(e)$.
- (2) nicht symmetrisch für alle $r \geq 1$: Zeige: Für alle $r \geq 1$ existiert ein disjunktiver Graph G und $e, f \in E$, so dass aus $f \in \phi_G^r(e)$ nicht folgt $e \in \phi_G^r(f)$. Betrachte den disjunktiven Graphen G (ein Job-Shop-Graph) in Abbildung 5.16., wobei $\mathcal{E} = \{\{e, e'\}, \{f, f'\}\}$ und $c > 0$ ist. Es ist $\phi_G^1(e) = \phi_G^*(e) = \{f\}$ und $\phi_G^1(f) = \phi_G^*(f) = \emptyset$. Somit gilt für alle $r \geq 1$, dass $f \in \phi_G^r(e)$, aber nicht $e \in \phi_G^r(f)$.

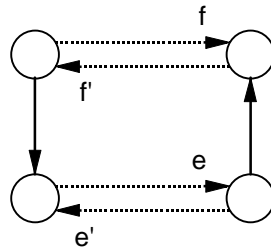


Abbildung 5.16. G mit nicht-symmetrischer Relation $\mathcal{R}(\phi_G^r)$ für $r \geq 1$

- (3) transitiv für $r = \infty$: Zeige: Für alle disjunktiven Graphen G und alle $e, f, g \in E$ gilt, dass aus $f \in \phi_G^*(e)$ und $g \in \phi_G^*(f)$ folgt $g \in \phi_G^*(e)$. Sei G ein disjunktiver Graph und $e, f, g \in E$. Sei

$f \in \varphi_G^*(e)$ und $g \in \varphi_G^*(f)$. Da $\{f\} \subseteq \varphi_G^*(e)$ gilt aufgrund der Monotonie von φ_G^* dass $\varphi_G^*(f) \subseteq \varphi_G^*(\varphi_G^*(e)) = \varphi_G^*(e)$. Somit folgt aus $g \in \varphi_G^*(f)$ dass $g \in \varphi_G^*(e)$.

Zeige weiter: Für alle $1 \leq r < \infty$ existiert ein disjunktiver Graph G und $e, f, g \in E$ so dass aus $f \in \varphi_G^r(e)$ und $g \in \varphi_G^r(f)$ nicht folgt $g \in \varphi_G^r(e)$. Sei $r = 1$. Betrachte den Graphen G aus Abbildung 5.15. Es gilt $f \in \varphi_G^1(e)$ und $g \in \varphi_G^1(f)$ aber nicht $g \in \varphi_G^1(e)$. Beachte, dass jedoch gilt $g \in \varphi_G^2(e)$. Analog kann durch entsprechende Erweiterung dieses Graphen G gezeigt werden, dass die Behauptung für alle $1 \leq r < \infty$ stimmt. ♦

5.4.7. Fixpunkttheorie

In diesem Abschnitt wird die Mengenfunktion $\varphi_G : Pot(E) \rightarrow Pot(E)$ unter dem Gesichtspunkt der Fixpunkttheorie betrachtet. $\varphi_G(S)$ stellt hierbei gemäss Definition die Menge der direkt erzwungenen Bogen im disjunktiven Graphen G dar. Für eine Zusammenfassung der wesentlichen Punkte der Fixpunkttheorie von Mengenfunktionen siehe beispielsweise [Engeler, E., Läuchli, P. (1988)].

Zunächst werden die Begriffe der abgeschlossenen Menge und des Fixpunktes eingeführt. Anschliessend wird eine Charakterisierung der abgeschlossenen Mengen und der Fixpunkte von φ_G gegeben. Insbesondere wird gezeigt, dass alle abgeschlossenen Mengen auch Fixpunkte sind (und umgekehrt). Eine grundlegende Aussage der Fixpunkttheorie besagt, dass jede monotone Mengenfunktion einen kleinsten Fixpunkt besitzt. Da φ_G monoton ist (vgl. Abschnitt 5.4.3.), hat φ_G einen kleinsten Fixpunkt. Es wird gezeigt, dass dieser gegeben ist durch $\varphi^*(G) = \varphi_G^*(\emptyset)$ d.h. durch das konjunktive Erzeugnis in G . Schliesslich wird noch die Frage untersucht, welches die kleinste Menge S ist, für welche der disjunktive Subgraph $G[S]$ konjunktiv abgeschlossen ist. Es wird gezeigt, dass $S \subseteq \varphi^*(G)$ und $S = \varphi^*(G)$ falls $\varphi^*(G)$ positiv azyklisch ist in G .

Definition 5.47. Sei $\varphi : Pot(E) \rightarrow Pot(E)$ eine Mengenfunktion auf der Grundmenge E .

- (i) $S \subseteq E$ heisst eine abgeschlossene Menge von φ , falls $\varphi(S) \subseteq S$ ist.
- (ii) $S \subseteq E$ heisst ein Fixpunkt von φ , falls $\varphi(S) = S$ ist.

Lemma 5.48. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Betrachte die Mengenfunktion $\varphi_G : Pot(E) \rightarrow Pot(E)$. Sei $S \subseteq E$. Es gilt:

- (i) S ist genau dann eine abgeschlossene Menge von φ_G , wenn $\varphi_G^*(S) = S$ ist.
- (ii) S ist genau dann ein Fixpunkt von φ_G , wenn $\varphi_G^*(S) = S$ ist.

Beweis:

Beweis von (i): Zeige einerseits: Falls S eine abgeschlossene Menge von φ_G ist, dann gilt $\varphi_G^*(S) = S$. Sei S eine abgeschlossene Menge von φ_G . Dann gilt $\varphi_G(S) \subseteq S$. Da auch $S \subseteq \varphi_G(S)$ ist, $\varphi_G(S) = S$. Gemäss Lemma 5.30. folgt daraus $\varphi_G^*(S) = S$. Zeige andererseits: Falls $\varphi_G^*(S) = S$, dann ist S eine abgeschlossene Menge von φ_G . Sei $\varphi_G^*(S) = S$. Gemäss Lemma 5.25. ist $\varphi_G(S) \subseteq \varphi_G^*(S)$. Daraus folgt $\varphi_G(S) \subseteq S$.

Beweis von (ii): Zeige einerseits: Falls S ein Fixpunkt von φ_G ist, dann gilt $\varphi_G^*(S) = S$. Sei S ein Fixpunkt von φ_G . Dann gilt $\varphi_G(S) = S$. Gemäss Lemma 5.30. folgt daraus $\varphi_G^*(S) = S$. Zeige andererseits: Falls $\varphi_G^*(S) = S$, dann ist S ein Fixpunkt von φ_G . Sei $\varphi_G^*(S) = S$. Gemäss Lemma 5.30. folgt daraus $\varphi_G(S) = S$. ♦

Lemma 5.49. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Betrachte die Mengenfunktion $\varphi_G : Pot(E) \rightarrow Pot(E)$. Es gilt:

- (i) $\varphi^*(G) = \varphi_G^*(\emptyset)$ ist die kleinste abgeschlossene Menge von φ_G .
- (ii) $\varphi^*(G) = \varphi_G^*(\emptyset)$ ist der kleinste Fixpunkt von φ_G .

Beweis:

Beweis von (i): Gemäss Lemma 5.30. ist $\varphi_G(\varphi^*(G)) = \varphi^*(G)$ d.h. $\varphi^*(G)$ ist eine abgeschlossene Menge von φ_G . Nehme an, dass eine kleinere abgeschlossene Menge $S \subset \varphi^*(G)$ existiert. Da S abgeschlossen ist, gilt gemäss Lemma 5.48., dass $\varphi_G^*(S) = S$. Da $\emptyset \subseteq S$ gilt aufgrund der Monotonie von φ_G^* , dass $\varphi_G^*(\emptyset) \subseteq \varphi_G^*(S)$. Folglich ist $\varphi^*(G) \subseteq S$. Ein Widerspruch.

Beweis von (ii): Gemäss Lemma 5.30. ist $\varphi_G(\varphi^*(G)) = \varphi^*(G)$ d.h. $\varphi^*(G)$ ist ein Fixpunkt von φ_G . Da jeder Fixpunkt eine abgeschlossene Menge ist und $\varphi^*(G)$ gemäss Beweis von (i) die kleinste abgeschlossene Menge ist, ist $\varphi^*(G)$ der kleinste Fixpunkt von φ_G . ♦

Das vorangehende Lemma besagt also, dass $S = \varphi^*(G)$ die kleinste Selektion in G ist, welche keine weiteren Bogen erzwingt in G , d.h. für welche gilt $\varphi_G^*(S) = S$.

Satz 5.50. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$ die kleinste Menge, für welche $G[S]$ konjunktiv abgeschlossen ist. Es gilt:

- (i) $S \subseteq \varphi^*(G)$
- (ii) $S = \varphi^*(G)$, falls $\varphi^*(G)$ positiv azyklisch ist in G

Beweis:

Beweis von (i): Gemäss Lemma 5.31. ist $G[\varphi^*(G)] = G[\varphi_G^*(\emptyset)]$ konjunktiv abgeschlossen. Da S die kleinste Menge ist, für welche $G[S]$ konjunktiv abgeschlossen ist, gilt $S \subseteq \varphi^*(G)$

Beweis von (ii): Sei $\varphi^*(G)$ positiv azyklisch in G . Gemäss Lemma 5.31. ist $G[\varphi^*(G)] = G[\varphi_G^*(\emptyset)]$ konjunktiv abgeschlossen. Nehme an, dass eine kleinere Menge $S' \subset \varphi_G^*(\emptyset)$ existiert, so dass $G[S']$ konjunktiv abgeschlossen ist. Da $\varphi_G^*(\emptyset)$ positiv azyklisch ist in G , ist auch S' positiv azyklisch in G . Somit gilt gemäss Lemma 5.31., dass $S' = \varphi_G(S')$. Gemäss Lemma 5.30. folgt daraus $S' = \varphi_G^*(S')$. Aufgrund der Monotonie von φ_G^* gilt $\varphi_G^*(\emptyset) \subseteq \varphi_G^*(S')$. Also ist $\varphi_G^*(\emptyset) \subseteq S'$. Ein Widerspruch. ♦

5.4.8. Komplexität

Die Berechnung der verbotenen und erzwungenen Bogen $\chi_G^*(S)$ bzw. $\varphi_G^*(S)$ in einem disjunktiven Graphen G ist mit polynomialem Aufwand möglich, im Gegensatz zu den global verbotenen bzw. erzwungenen Bogen $\chi_G(S)$ bzw. $\varphi_G(S)$. Der nachfolgende Satz zeigt, dass der Aufwand für die Berechnung von $\chi_G^*(S)$ und $\varphi_G^*(S)$ kubisch ist in der Anzahl konjunktiver Bogen und der Betragssumme der disjunktiven Mengen. Falls jeder disjunktive Bogen in genau einer disjunktiven Menge vorkommt, ergibt sich daraus ein kubischer Aufwand in der Anzahl (konjunktiver und disjunktiver) Bogen von G .

Satz 5.51. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph. Sei $S \subseteq E$. Sei $m = |A| + \sum_{D \in \mathcal{E}} |D|$.

- (i) Der Aufwand für die Berechnung von $\chi_G^*(S)$ ist von der Grössenordnung $O(m^3)$.
- (ii) Der Aufwand für die Berechnung von $\varphi_G^*(S)$ ist von der Grössenordnung $O(m^3)$.

Beweis:

Beweis von (i): (1) Sei $r \geq 1$. Zeige, dass der Aufwand für die Berechnung von $\varphi_G^r(S)$ bei gegebenem $\chi_G^r(S)$ von der Grössenordnung $O(\sum_{D \in \mathcal{E}} |D|)$ ist. Gemäss Lemma 5.29. ist $\varphi_G^r(S) = S \cup \Phi_G(\chi_G^r(S))$. Somit entspricht der Aufwand zur Berechnung von $\varphi_G^r(S)$ dem Aufwand zur Berechnung von $\Phi_G(\chi_G^r(S))$. Gemäss Definition ist $\Phi_G(\chi_G^r(S))$ die Menge aller Bogen $e \in E$, für welche ein $D \in \mathcal{E}$ existiert mit $e \in D$ so dass $e' \in \chi_G^r(S)$ für alle $e' \in D \setminus e$.

$\Phi_G(\chi_G^r(S))$ kann bei gegebenen $\chi_G^r(S)$ folgendermassen berechnet werden: "Setze $Q = \emptyset$ und führe für alle $D \in \mathcal{E}$ die folgenden Schritte durch: Bestimme die Menge $D' = D - \chi_G^r(S)$. Falls $|D'| = 0$, dann setze $Q = Q \cup D$. Falls $|D'| = 1$ dann setze $Q = Q \cup D$. Falls $|D'| > 1$ dann tue nichts." Als Resultat ergibt sich $\Phi_G(\chi_G^r(S)) = Q$. Die Menge $D' = D - \chi_G^r(S)$ kann berechnet werden, indem für alle $e \in D$ bestimmt wird, ob $e \in \chi_G^r(S)$. Falls $e \notin \chi_G^r(S)$ wird e zu D' hinzugefügt. Somit kann D' mit Aufwand $O(|D|)$ berechnet werden, falls die Entscheidung

$e \in \chi_G^r(S)$ mit Aufwand $O(1)$ bestimmt werden kann, was bei entsprechender Implementation möglich ist. Folglich ist der gesamte Aufwand zur Berechnung von $\Phi_G(\chi_G^r(S))$ von der Grössenordnung $O(\sum_{D \in \mathcal{E}} |D|)$

(2) Sei $r \geq 1$. Zeige, dass der Aufwand für die Berechnung von $\chi_G^r(S)$ bei gegebenem $\phi_G^{r-1}(S)$ von der Grössenordnung $O(|A \cup E|^2)$ ist. Gemäss Definition ist $\chi_G^r(S) = \chi_G(\phi_G^{r-1}(S))$ d.h. $\chi_G^r(S)$ entspricht der Menge aller Bogen $e \in E$, für welche $G(\phi_G^{r-1}(S) \cup e)$ positiv zyklisch ist. Somit kann $\chi_G^r(S)$ berechnet werden, indem für jeden Bogen $e \in E$ geprüft wird, ob $G(\phi_G^{r-1}(S) \cup e)$ positiv zyklisch ist. Der Test, ob ein gerichteter Graph positiv zyklisch ist, kann bekanntlich mittels Topologischer Sortierung mit linearem Aufwand bezüglich der Anzahl Bogen gemacht werden. Da $G(\phi_G^{r-1}(S) \cup e)$ maximal $|A \cup E|$ Bogen hat, ist der Aufwand für den Test, ob $G(\phi_G^{r-1}(S) \cup e)$ positiv zyklisch ist, von der Grössenordnung $O(|A \cup E|)$. Da dieser Test für jeden disjunktiven Bogen $e \in E$ durchgeführt werden muss, ergibt sich für die Berechnung von $\chi_G^r(S)$ ein Gesamtaufwand $|E| \cdot O(|A \cup E|)$ d.h. $O(|A \cup E|^2)$.

(3) Zeige, dass der Aufwand für die Berechnung von $\chi_G^*(S)$ von der Grössenordnung $O(m^3)$ ist. Gemäss Lemma 5.28. existiert ein $s \leq |E|$, so dass $\chi_G^*(S) = \chi_G^s(S)$. Gemäss Definition kann $\chi_G^s(S)$ berechnet werden, indem der Reihe nach für $r = 1, 2, \dots, s$ zuerst $\phi_G^{r-1}(S) = S \cup \Phi_G(\chi_G^{r-1}(S))$ und anschliessend $\chi_G^r(S) = \chi_G(\phi_G^{r-1}(S))$ berechnet wird. Die Berechnung von $\phi_G^{r-1}(S)$ bei gegebenem $\chi_G^{r-1}(S)$ hat wie gezeigt den Aufwand $O(\sum_{D \in \mathcal{E}} |D|)$ bzw. $O(m)$, da $\sum_{D \in \mathcal{E}} |D| \leq m$ ist. Die Berechnung von $\chi_G^r(S)$ bei gegebenem $\phi_G^{r-1}(S)$ hat wie gezeigt den Aufwand $O(|A \cup E|^2)$ bzw. $O(m^2)$, da $|A \cup E| \leq m$ ist. Somit hat jeder Rechendurchgang für $r = 1, 2, \dots, s$ den Gesamtaufwand von $O(m) + O(m^2)$ d.h. $O(m^2)$. Da $s \leq |E|$ müssen insgesamt höchstens $|E|$ Rechendurchgänge gemacht werden. Somit ergibt sich ein Totalaufwand von $|E| \cdot O(m^2)$ bzw. $O(m^3)$, da $|E| \leq m$ ist.

Beweis von (ii): Gemäss Lemma 5.29. ist $\phi_G^*(S) = S \cup \Phi_G(\chi_G^*(S))$. Gemäss Beweis von (i) ist der Aufwand für die Berechnung von $\Phi_G(\chi_G^*(S))$ bei gegebenem $\chi_G^*(S)$ von der Grössenordnung $O(\sum_{D \in \mathcal{E}} |D|)$ und für die Berechnung von $\chi_G^*(S)$ von der Grössenordnung $O(m^3)$. Somit ergibt sich ein Gesamtaufwand von $O(\sum_{D \in \mathcal{E}} |D|) + O(m^3)$ bzw. $O(m^3)$, da $\sum_{D \in \mathcal{E}} |D| \leq m$ ist. ♦

5.5. Zulässigkeitsproblem

5.5.1. Einführung

Das Zulässigkeitsproblem befasst sich mit der Frage, ob in einem disjunktiven Graphen G eine zulässige Selektion existiert. Für gewisse Klassen von disjunktiven Graphen ist diese Fragestel-

lung trivial. Betrachte beispielsweise die Klasse der Job-Shop-Graphen oder Lager-Graphen. In einem Job-Shop- oder Lager-Graphen existiert immer eine zulässige Selektion. Eine solche kann konstruiert werden, indem die Jobs in beliebiger Reihenfolge einer nach dem andern abgearbeitet werden (vgl. Abschnitt 6.6.).

Schwieriger wird das Problem, wenn eine Teilselektion Q von disjunktiven Bogen vorgegeben ist. Das Zulässigkeitsproblem lautet dann, ob der disjunktive Subgraph $G[Q]$ eine zulässige Selektion besitzt. Eine notwendige Bedingung für die Existenz einer zulässigen Selektion in $G[Q]$ ist, dass Q positiv azyklisch ist in G . Es wird sich herausstellen, dass diese Bedingung für klassische disjunktive Graphen (und somit für Job-Shop-Graphen) auch hinreichend ist. Dies gilt jedoch nicht für verallgemeinerte disjunktive Graphen. Als ein zentrales Resultat dieses Abschnittes wird gezeigt, dass das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen NP-hart ist. Weiter wird gezeigt, dass das Zulässigkeitsproblem für Lager-Graphen mit vorgegebener Selektion Q auch NP-hart ist. Die daraus resultierenden (weitreichenden) Konsequenzen in Hinblick auf die Lösung des Lager-Problems, d.h. des Steuerungsproblems für automatisierte Kompaktlager, werden in Abschnitt 6.1. diskutiert.

Aus diesen Ausführungen wird ersichtlich, dass bei der Diskussion des Zulässigkeitsproblems für eine bestimmte Graphenklasse unterschieden werden muss, ob eine Teilauswahl Q von disjunktiven Bogen vorgegeben ist oder nicht. Eine vorgegebene Teilauswahl Q wird im weiteren als Präselektion bezeichnet. Nachfolgend wird eine formale Definition des Zulässigkeitsproblems gegeben, wobei unterschieden wird zwischen dem Zulässigkeitsproblem ohne Präselektion (auch kurzerhand Zulässigkeitsproblem genannt) und dem Zulässigkeitsproblem mit Präselektion.

Definition 5.52. Sei \mathcal{G} eine Klasse von disjunktiven Graphen. Bezeichne \mathcal{G}^* die Klasse aller disjunktiven Subgraphen von \mathcal{G} , d.h. $\mathcal{G}^* = \{G[Q] : G \in \mathcal{G}, Q \text{ ist eine Selektion von } G\}$.

- (i) Das Zulässigkeitsproblem (ohne Präselektion) für \mathcal{G} entspricht dem Entscheidungsproblem, ob ein disjunktiver Graph $G \in \mathcal{G}$ eine zulässige Selektion besitzt.
- (ii) Das Zulässigkeitsproblem mit Präselektion für \mathcal{G} entspricht dem Entscheidungsproblem, ob ein disjunktiver Graph $G \in \mathcal{G}^*$ eine zulässige Selektion besitzt.

Beachte, dass $\mathcal{G} \subseteq \mathcal{G}^*$ ist. Das Zulässigkeitsproblem ohne Präselektion entspricht offensichtlich dem Spezialfall des Zulässigkeitsproblems mit Präselektion, wenn nur Präselektionen $Q = \emptyset$ betrachtet werden. Folglich bildet für eine gegebene Graphenklasse \mathcal{G} das Zulässigkeitsproblem ohne Präselektion ein Teilproblem des Zulässigkeitsproblems mit Präselektion und ist somit komplexitätstheoretisch höchstens so schwierig wie letzteres. Falls für eine Graphenklasse $\mathcal{G} = \mathcal{G}^*$ gilt, sind die beiden Probleme äquivalent. Dies trifft beispielsweise zu für die klassischen und die verallgemeinerten disjunktiven Graphen.

In den folgenden Abschnitten wird das Zulässigkeitsproblem für die verschiedenen in Abschnitt 5.3. definierten Klassen von disjunktiven Graphen untersucht: Verallgemeinerte disjunktive Graphen, klassische disjunktive Graphen, Job-Shop-Graphen, Lager-Graphen und 1-Job-Lager-Graphen. Es werden verschiedene komplexitätstheoretische und strukturelle Resultate erarbeitet, von welchen einige bei der Lösung des Lager-Problems in Kapitel 6 von Bedeutung sind. Weiter wird der Zusammenhang zwischen dem Zulässigkeitsproblem und dem Konzept der verbotenen und erzwungenen disjunktiven Bogen diskutiert.

5.5.2. Verallgemeinerte disjunktive Graphen

Bedingungen für Zulässigkeit

Der folgende Satz zeigt notwendige und hinreichende Bedingungen für die Existenz einer zulässigen Selektion in einem disjunktiven Graphen G mit Präselektion Q . Er basiert im wesentlichen auf der Definition von $\chi_G(Q)$ und $\phi_G(Q)$ sowie auf der Tatsache, dass $\chi_G(Q) = \phi_G(Q) \neq E$ falls $G[Q]$ keine zulässige Selektion hat.

Satz 5.53. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $Q \subseteq E$ eine Präselektion in G . Jede der beiden folgenden Bedingungen ist notwendig und hinreichend dafür, dass $G[Q]$ eine zulässige Selektion besitzt:

- (i) Es existiert kein $D \in \mathcal{E}$ mit $D \subseteq \chi_G(Q)$.
- (ii) $\phi_G(Q)$ ist positiv azyklisch in G .

Beweis:

Beweis von (i): Zeige einerseits, dass die Bedingung notwendig ist. Sei S eine zulässige Selektion in $G[Q]$. Nehme an, dass ein $D \in \mathcal{E}$ existiert mit $D \subseteq \chi_G(Q)$. Gemäss Lemma 5.11. ist $S' = Q \cup S$ eine zulässige Selektion in G . Da S' vollständig ist in G gilt $D \cap S' \neq \emptyset$. Sei $e \in D \cap S'$. S' ist eine vollständige Selektion in G mit $Q \subseteq S'$ und $e \in S'$ so dass $G(S')$ positiv azyklisch ist. Folglich gilt $e \notin \chi_G(Q)$. Da aber $e \in D$ und $D \subseteq \chi_G(Q)$ gilt $e \in \chi_G(Q)$ in Widerspruch. Zeige andererseits, dass die Bedingung hinreichend ist. Es gelte, dass kein $D \in \mathcal{E}$ existiert mit $D \subseteq \chi_G(Q)$. Zeige, dass $G[Q]$ eine zulässige Selektion besitzt. Aufgrund der Voraussetzung gilt $\chi_G(Q) \neq E$. Gemäss Lemma 5.37. folgt daraus, dass $G[Q]$ eine zulässige Selektion besitzt.

Beweis von (ii): Zeige einerseits, dass die Bedingung notwendig ist. Sei S eine zulässige Selektion in $G[Q]$. Nehme an, dass $\phi_G(Q)$ positiv zyklisch ist in G . Gemäss Lemma 5.11. ist $S' = Q \cup S$ eine zulässige Selektion in G . Da $Q \subseteq S'$ gilt gemäss Lemma 5.36., dass $\phi_G(Q) \subseteq S'$. Da S' positiv azyklisch ist in G , muss auch $\phi_G(Q)$ positiv azyklisch sein in G . Ein

Widerspruch. Zeige andererseits, dass die Bedingung hinreichend ist. Sei $\varphi_G(Q)$ positiv azyklisch in G . Zeige, dass $G[Q]$ eine zulässige Selektion besitzt. Fall 1: Sei $\varphi_G(Q) = E$. Sei $S' = \varphi_G(Q)$. S' ist vollständig in G . Da $\varphi_G(Q)$ positiv azyklisch ist in G und $Q \subseteq \varphi_G(Q)$ ist S' eine zulässige Selektion in G mit $Q \subseteq S'$. Sei $S = S' \cap E_{G[Q]}$. Gemäss Lemma 5.11. ist S eine zulässige Selektion in $G[Q]$. Fall 2: Sei $\varphi_G(Q) \neq E$. Gemäss Lemma 5.37. folgt daraus, dass $G[Q]$ eine zulässige Lösung besitzt. ♦

Durch den vorangehenden Satz wird ein Zusammenhang hergestellt zwischen dem Zulässigkeitsproblem und dem Konzept der global verbotenen bzw. erzwungenen Bogen. Falls die Menge der global verbotenen bzw. erzwungenen Bogen bekannt ist, kann mit polynomialem Aufwand festgestellt werden, ob eine zulässige Selektion existiert. Somit ist die Bestimmung der global verbotenen bzw. erzwungenen Bogen komplexitätstheoretisch mindestens so schwierig wie das Zulässigkeitsproblem. Da das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen NP-hart ist (vgl. Korollar 5.57.), ist somit auch die Bestimmung der global verbotenen bzw. erzwungenen Bogen NP-hart (vgl. Korollar 5.58.).

Im folgenden Satz werden zwei notwendige Bedingungen gezeigt für die Existenz einer zulässigen Selektion in einem disjunktiven Graphen mit Präselektion Q : (1) Es dürfen nicht alle Bogen einer disjunktiven Menge verboten sein durch Q . (2) Das konjunktive Erzeugnis von Q muss positiv azyklisch sein. Anhand eines Gegenbeispiels wird gezeigt, dass diese beiden Bedingungen im allgemeinen nicht hinreichend sind.

Satz 5.54. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $Q \subseteq E$ eine Präselektion in G . Die beiden folgenden Bedingungen sind notwendig, aber nicht hinreichend dafür, dass $G[Q]$ eine zulässige Selektion besitzt:

- (i) Es existiert kein $D \in \mathcal{E}$ mit $D \subseteq \chi_G^*(Q)$.
- (ii) $\varphi_G^*(Q)$ ist positiv azyklisch in G .

Beweis:

Beweis von (i): Zeige, dass die Bedingung notwendig ist. Sei S eine zulässige Selektion in $G[Q]$. Gemäss Satz 5.53. existiert kein $D \in \mathcal{E}$ mit $D \subseteq \chi_G(Q)$. Gemäss Lemma 5.39. gilt $\chi_G^*(S) \subseteq \chi_G(S)$. Daraus folgt die Behauptung.

Beweis von (ii): Zeige, dass die Bedingung notwendig ist. Sei S eine zulässige Selektion in $G[Q]$. Gemäss Satz 5.53. ist $\varphi_G(Q)$ positiv azyklisch in G . Gemäss Lemma 5.39. gilt $\varphi_G^*(Q) \subseteq \varphi_G(Q)$. Folglich ist auch $\varphi_G^*(Q)$ positiv azyklisch in G .

Zeige noch, dass die beiden Bedingungen nicht hinreichend sind. Betrachte das folgende Gegenbeispiel. Gegeben ist der untenstehende disjunktive Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ wobei $\mathcal{E} = \{\{e, e'\}, \{f, f'\}\}$ und $c > 0$. Sei $Q = \emptyset$. Da $\chi_G^*(Q) = \emptyset$ ist Bedingung (i) erfüllt. Da $\varphi_G^*(Q) = \emptyset$ und G positiv azyklisch, ist Bedingung (ii) erfüllt. Aber G besitzt keine zulässige Selektion. Folglich sind die beiden Bedingungen nicht hinreichend. ♦

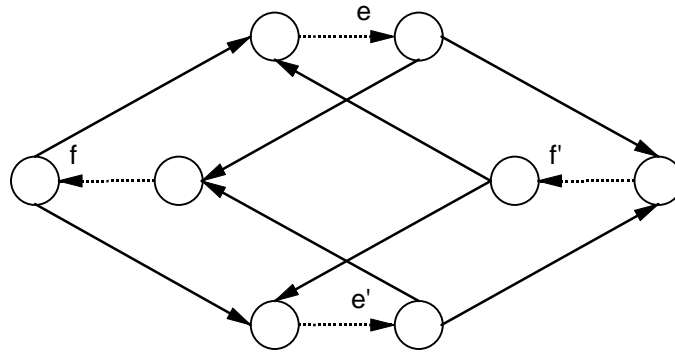


Abbildung 5.17. G hat keine zulässige Selektion

Komplexität

Im folgenden wird gezeigt, dass das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen NP-hart ist. Für den Beweis wird eine spezielle Klasse von disjunktiven Graphen betrachtet, welche als *th*-Graphen bezeichnet wird. *th*-Graphen sind ähnlich aufgebaut wie Lager-Graphen und bilden eine Verallgemeinerung der letzteren. Sie unterscheiden sich im wesentlichen dadurch, dass auf die Job-Struktur verzichtet wird. Untenstehend wird eine genaue Definition der *th*-Graphen gegeben. Die Beweisführung für die Komplexität des Zulässigkeitsproblems basiert darauf, dass das SAT-Problem polynomial zurückgeführt wird auf das Zulässigkeitsproblem in *th*-Graphen.

Das SAT-Problem bildet ein Standardproblem in der Komplexitätstheorie, für welches bewiesen ist, dass es NP-vollständig ist (siehe beispielsweise [Garey, M. R., Johnson D. S. (1979)], [Davis, M. D., Weyuker, E. J. (1983)] oder [Papadimitriou, Ch. H. (1994)]). Es entspricht dem Entscheidungsproblem, ob ein aussagenlogischer Ausdruck in konjunktiver Normalform eine Wertebelegung besitzt, so dass der Ausdruck wahr ist. Das SAT-Problem lässt sich folgendermassen formulieren: Gegeben ist eine Menge X von aussagenlogischen Variablen, welche die Werte *true* und *false* annehmen können. Eine einzelne Variable $x \in X$ oder deren Negation \bar{x} werden als Literale bezeichnet. Die Menge der Literale ist gegeben durch $X' = \{x, \bar{x} : x \in X\}$. x wird als das positive und \bar{x} als das negative Literal zur Variablen x bezeichnet. Eine Klausel C in X ist eine Disjunktion von Literalen aus X' und wird dargestellt als Teilmenge $C \subseteq X'$. Eine Instanz des SAT-Problems kann dargestellt werden als ein Paar (X, Q) , wobei X die Grundmenge der logischen

Variablen und $Q \subseteq \text{Pot}(X')$ eine Menge von Klauseln in X ist. Das SAT-Problem entspricht dem Entscheidungsproblem, ob für eine Instanz (X, Q) eine Wertebelegung für die Variablen X existiert, so dass jede Klausel den Wert *true* ergibt. Eine Klausel C ergibt den Wert *true* wenn ein positives Literal x in C existiert mit $x = \text{true}$ oder ein negatives Literal mit $x = \text{false}$.

Definition 5.55. Ein *th-Graph* ist ein spezieller disjunktiver Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ mit folgenden Eigenschaften:

- (i) Die Knotenmenge ist gegeben durch $V = \{t_i, h_i : i \in I\} \cup \{\sigma, \tau\}$. Sie besteht aus einer Menge von paarweise zusammengehörenden Knoten t_i und h_i , wobei I eine beliebige Grundmenge (von Operationen) darstellt. t_i wird als *t-Knoten* und h_i als *h-Knoten* von $i \in I$ bezeichnet werden. t_i und h_i bilden das zu $i \in I$ gehörende *th-Knotenpaar*. σ ist ein künstlicher Start- und τ ein künstlicher Endknoten.
- (ii) Die Menge der konjunktiven Bogen ist gegeben durch $A = A_1 \cup A_2 \cup A_\sigma \cup A_\tau$, wobei diese Bogenteilmengen paarweise disjunkt sind. Die Bogen in A_1 verbinden für jedes *th-Knotenpaar* den *t-Knoten* mit dem *h-Knoten*, d.h. $A_1 = \{(t_i, h_i) : i \in I\}$. Die Menge $A_2 \subseteq \{(h_i, t_j) : i, j \in I, i \neq j\}$, $A_2 \cap E = \emptyset$, stellt eine beliebige Menge von zusätzlichen Bogen dar, wobei die Bogen jeweils einen *h-Knoten* mit einem *t-Knoten* verbinden. Die Mengen $A_\sigma = \{(\sigma, t_i) : i \in I\}$ bzw. $A_\tau = \{(h_i, \tau) : i \in I\}$ bilden die zum Startknoten σ bzw. Endknoten τ gehörenden Start- bzw. Endbogen.
- (iii) Die beiden Bogen einer disjunktiven Menge verbinden die *h-Knoten* mit den *t-Knoten* von zwei *th-Knotenpaaren*, d.h. $\mathcal{E} \subseteq \{(h_i, t_j), (h_j, t_i) : i, j \in I, i \neq j\}$.
- (iv) Die Menge der disjunktiven Bogen ist gegeben durch $E = \mathcal{E}^\circ$.
- (v) Die Bogengewichte $c \geq 0$ sind nicht negativ und für alle k -Zyklen Z in $G(E)$ mit $k \geq 3$ wird ein positives Gewicht $c(Z) > 0$ vorausgesetzt.

Satz 5.56. Das Zulässigkeitsproblem für *th-Graphen* ist NP-hart.

Beweis:

Zur Beweisführung wird das SAT-Problem polynomial zurückgeführt auf das Zulässigkeitsproblem (ohne Präselektion) in *th-Graphen*. Beachte noch: Da das Zulässigkeitsproblem ein Spezialfall des Zulässigkeitsproblems mit Präselektion ist, folgt aus der Behauptung auch, dass das Zulässigkeitsproblem mit Präselektion für *th-Graphen* NP-hart ist.

- (1) Das SAT-Problem wird hier in folgender Form geschrieben: X bezeichnet die Grundmenge der logischen Variablen. Zu jeder Variablen $x \in X$ werden zwei komplementäre 0-1-Variablen x^0 und x^1 definiert, wobei x^0 das negative und x^1 das positive Literal von $x \in X$ darstellt. $x^0 = 1$ bedeutet, dass $x = \text{false}$ ist und $x^1 = 1$ dass $x = \text{true}$ ist. Die Menge der Literale ist gegeben durch

$X' = \{x^0, x^1 : x \in X\}$. Eine Klausel in X ist eine Teilmenge $C \subseteq X'$. Eine Instanz des SAT-Problems entspricht einem Paar (X, Q) , wobei $Q \subseteq \text{Pot}(X')$ eine Menge von Klauseln in X darstellt. Das SAT-Problem entspricht dem Entscheidungsproblem, ob für eine Instanz (X, Q) das folgende ganzzahlig-lineare System $P(X, Q)$ eine zulässige Lösung besitzt:

$$\sum_{x^s \in C} x^s \geq 1, \quad C \in Q \quad (5.12)$$

$$x^0 = 1 - x^1, \quad x \in X \quad (5.13)$$

$$x^0, x^1 \in \{0, 1\}, \quad x \in X \quad (5.14)$$

In dieser Darstellung des SAT-Problems wird aufgrund der Mengendarstellung für Klauseln angenommen, dass ein Literal nicht mehrmals in einer Klausel auftritt. Dies ist keine Einschränkung der Allgemeinheit, da bei mehrmaligem Vorhandensein eines Literals in einer Klausel alle betreffenden Literale bis auf eines weggelassen werden können. Als weitere Annahme soll gelten, dass in einer Klausel nicht gleichzeitig ein positives und ein negatives Literal x^0 und x^1 von $x \in X$ vorkommt. Auch dies ist keine Einschränkung der Allgemeinheit, da eine solche Klausel einer Tautologie entspricht und weggelassen werden kann. Somit enthält eine Klausel höchstens ein Literal zu einer Variablen $x \in X$. Dies bedeutet, dass nur SAT-Instanzen betrachtet werden, welche sich in der sogenannten kanonischen konjunktiven Normalform befinden.

Weiter wird angenommen, dass eine Klausel mindestens zwei Literale umfasst, d.h. $|C| \geq 2$ für alle $C \in Q$. Dies bedeutet ebenfalls keine Einschränkung der Allgemeinheit, denn es ist einfach zu zeigen, dass eine Klausel $C = \{x^s\}$ mit einem einzelnen Literal eliminiert werden kann, indem alle Klauseln $D \in Q$ mit $x^s \in D$ aus Q entfernt werden und alle Klauseln $D \in Q$ mit $x^{1-s} \in D$ ersetzt werden durch $D' = D - \{x^{1-s}\}$.

Bezeichne schliesslich $Q(x) = \{C \in Q : x^0 \in C \text{ oder } x^1 \in C\}$ die Menge aller Klauseln in Q , welche (genau) ein Literal von $x \in X$ enthalten. Es wird angenommen, dass in jeder Instanz die Grundmenge X der Variablen minimal ist, d.h. zu jeder Variablen $x \in X$ existiert mindestens eine Klausel mit einem Literal von x , d.h. $Q(x) \neq \emptyset$ für alle $x \in X$.

(2) Im folgenden wird gezeigt, wie jeder Instanz (X, Q) des SAT-Problems ein th -Graph G zugeordnet werden kann, so dass $P(X, Q)$ genau dann eine Lösung hat, wenn G eine zulässige Selektion besitzt. Die Konstruktion von G ist polynomial.

(3) Die Konstruktion des th -Graphen G geschieht wie folgt: Die Menge V der Knoten ist gegeben durch

$$V = \{t_C(x^0), h_C(x^0), t_C(x^1), h_C(x^1) : C \in Q, x^s \in C\} \cup \{\sigma, \tau\}$$

d.h. zu jeder Klausel $C \in Q$ und jedem Literal $x^s \in C$ dieser Klausel werden zwei th -Knotenpaare definiert, wobei das eine x^0 und das andere x^1 zugeordnet wird. Zusätzlich wird ein Startknoten σ und ein Endknoten τ eingeführt.

Die beiden th -Knotenpaare werden je mit einem konjunktiven Bogen vom t -Knoten zum h -Knoten verbunden:

$$A_1 = \{(t_C(x^0), h_C(x^0)), (t_C(x^1), h_C(x^1)) : C \in Q, x^s \in C\}$$

Die beiden th -Knotenpaare werden weiter mit zwei disjunktiven Bogen verbunden, wobei jeweils der h -Knoten des einen Paares mit dem t -Knoten des andern Paares verbunden wird. Diese beiden Bogen bilden jeweils eine disjunktive Menge. Die so definierten disjunktiven Bogen und disjunktiven Mengen sind die einzigen in G :

$$E = \{(h_C(x^0), t_C(x^1)), (h_C(x^1), t_C(x^0)) : C \in Q, x^s \in C\}$$

$$\mathcal{E} = \{\{(h_C(x^0), t_C(x^1)), (h_C(x^1), t_C(x^0))\} : C \in Q, x^s \in C\}$$

Die beiden disjunktiven Bogen einer disjunktiven Menge werden nach folgender Konvention den entsprechenden Variablen zugeordnet und es werden die folgenden Bezeichnungen verwendet:

$$e_C(x^0) = (h_C(x^0), t_C(x^1)), \quad C \in Q, x^s \in C$$

$$e_C(x^1) = (h_C(x^1), t_C(x^0)), \quad C \in Q, x^s \in C$$

Die untenstehende Abbildung zeigt die Knoten sowie die konjunktiven und disjunktiven Bogen, welche jedem einzelnen Literal $x^s \in C$ einer Klausel $C \in Q$ zugeordnet werden:

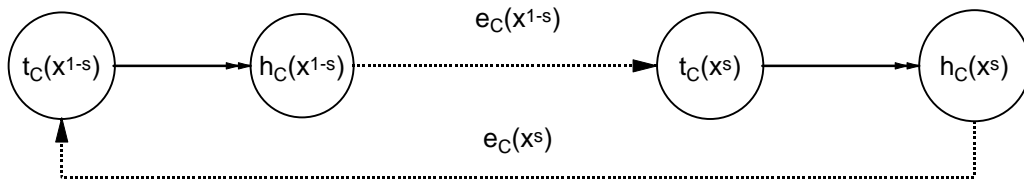


Abbildung 5.18. Knoten und Bogen zu einem Literal x^s

Neben den bisher erwähnten Bogen umfasst der Graph G weitere konjunktive Bogen. Für alle $x \in X$ und alle Klauseln $C, D \in Q(x)$ $C \neq D$ werden die folgenden vier konjunktiven Bogen definiert:

$$A_2 = \{(t_C(x^1), h_D(x^1)), (t_D(x^0), h_C(x^0)), (t_C(x^0), h_D(x^0)), (t_D(x^1), h_C(x^1)) : C, D \in Q(x), C \neq D, x \in X\}$$

Für jede Klausel $C \in Q$, $C = \{x_i^{s(i)} : i = 1, \dots, n\}$ $n \geq 2$ werden weiter die folgenden konjunktiven Bogen eingeführt:

$$A_3 = \{(t_C(x_i^{s(i)}), h_C(x_{i+1}^{1-s(i+1)})) : C \in Q, C = \{x_i^{s(i)} : i = 1, \dots, n\}\}$$

wobei $x_{n+1}^{1-s(n+1)} = x_1^{1-s(1)}$ ist.

Schliesslich werden noch die Startbogen $A_\sigma = \{(\sigma, t_C(x^0)), (\sigma, t_C(x^1)) : C \in Q, x^s \in C\}$ und die Endbogen $A_\tau = \{(h_C(x^0), \tau), (h_C(x^1), \tau) : C \in Q, x^s \in C\}$ definiert.

Die Gesamtmenge der konjunktiven Bogen ist gegeben durch $A = A_1 \cup A_2 \cup A_3 \cup A_\sigma \cup A_\tau$. Der disjunktive Graph G , welcher einer SAT-Probleminstanz (X, Q) zugeordnet wird, ist definiert als

$$G = \langle V, A, E, \mathcal{E}, c \rangle$$

wobei $c > 0$ einen beliebigen positiven Gewichtsvektor darstellt.

Die untenstehende Abbildung zeigt als Beispiel den disjunktiven Graphen G für eine Probleminstanz (X, Q) mit Variablen $X = \{x, y, z, w\}$ und Klauseln $Q = \{\{x^1, y^1, z^0\}, \{x^1, y^0, w^1\}\}$. Die disjunktiven Bogen sind mit den zugeordneten Variablen angeschrieben. Die Start- und Endknoten sowie die entsprechenden Bogen sind zur Vereinfachung nicht eingezeichnet.

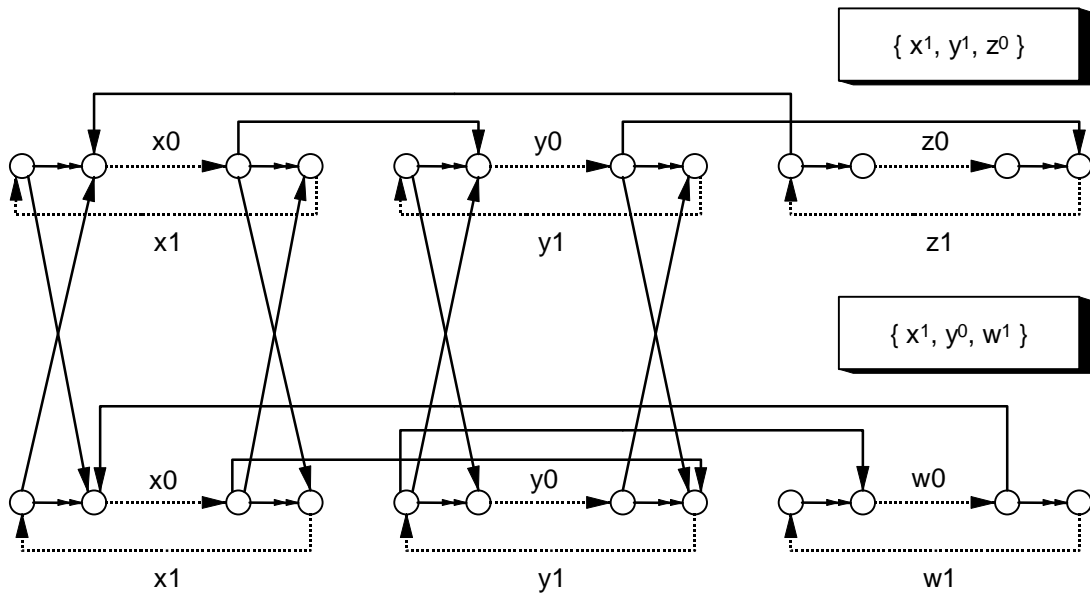


Abbildung 5.19. Disjunktiver th -Graph für Beispiel mit 2 Klauseln

- (4) G ist offensichtlich ein th -Graph und die Konstruktion von G ist polynomial.
- (5) Als erstes wird gezeigt: Wenn in G eine zulässige Selektion existiert, dann existiert in $P(X, Q)$ eine zulässige Lösung. Sei S eine zulässige Selektion in G .
- (i) Für alle $x \in X$ gilt: Für jede Klausel $C \in Q(x)$ ist genau einer der beiden Bogen $e_C(x^0)$ und $e_C(x^1)$ in der Selektion S . Beweis: Die beiden Bogen bilden eine disjunktive Menge in G . Da S vollständig ist, muss mindestens einer der beiden Bogen in S sein. Wenn beide Bogen in S wären, dann hätte $G(S)$ den 4-Zyklus $Z = \{e_C(x^0), a_1, e_C(x^1), a_2\}$, wobei $a_1, a_2 \in A_1$

$a_1 = (t_C(x^1), h_C(x^1))$ und $a_2 = (t_C(x^0), h_C(x^0))$ Ein Widerspruch zur Annahme, dass S positiv azyklisch ist.

(ii) Für alle $x \in X$ gilt: Entweder ist $e_C(x^0) \in S$ für alle $C \in Q(x)$ oder $e_C(x^1) \in S$ für alle $C \in Q(x)$. Beweis: Gemäss (i) ist für jedes $C \in Q(x)$ genau einer der beiden Bogen $e_C(x^0)$ und $e_C(x^1)$ in S . Nehme an, dass $C, D \in Q(x)$ existieren mit $e_C(x^0) \in S$ und $e_D(x^1) \in S$. Gemäss (i) ist $C \neq D$. Dann existiert ein 4-Zyklus $Z = \{e_C(x^0), a_1, e_D(x^1), a_2\}$ in $G(S)$, wobei $a_1, a_2 \in A_2$ $a_1 = (t_C(x^1), h_D(x^1))$ und $a_2 = (t_D(x^0), h_C(x^0))$ Ein Widerspruch zur Annahme, dass S positiv azyklisch ist.

(iii) Für alle Klauseln $C \in Q$ $C = \{x_i^{s(i)} : i = 1, \dots, n\}$ $n \geq 2$ gilt: Für mindestens ein $i \in \{1, \dots, n\}$ ist $e_C(x_i^{s(i)}) \in S$ Beweis: Gemäss (i) ist für alle $i \in \{1, \dots, n\}$ genau einer der beiden Bogen $e_C(x_i^{s(i)})$ und $e_C(x_i^{1-s(i)})$ in S . Nehme an, dass für alle $i \in \{1, \dots, n\}$ der Bogen $e_C(x_i^{1-s(i)})$ in S ist. Dann existiert der 2-Zyklus $Z = (e_C(x_1^{1-s(1)}), a_1, e_C(x_2^{1-s(2)}), a_2, \dots, e_C(x_n^{1-s(n)}), a_n)$ in $G(S)$, wobei $a_i \in A_3$ und $a_i = (t_C(x_i^{s(i)}), h_C(x_{i+1}^{1-s(i+1)}))$ für alle $i \in \{1, \dots, n\}$ und $x_{n+1}^{1-s(n+1)} = x_1^{1-s(1)}$ ist. Ein Widerspruch zur Annahme, dass S positiv azyklisch ist.

Definiere nun für alle $x \in X$ die folgenden Werte für die Literale x^0 und x^1 .

$$x^0 = 1, x^1 = 0, \quad \text{falls } e_C(x^0) \in S \text{ für alle } C \in Q(x)$$

$$x^0 = 0, x^1 = 1, \quad \text{falls } e_C(x^1) \in S \text{ für alle } C \in Q(x)$$

Aufgrund von (ii) und der Annahme $Q(x) \neq \emptyset$, $x \in X$ folgt, dass für jedes $x \in X$ genau einer dieser beiden Fälle zutrifft. Somit sind alle Literale wohldefiniert. Offensichtlich bilden diese Werte eine zulässige Lösung des Systems $P(X, Q)$. Die Restriktionen (5.13) und (5.14) sind trivialerweise erfüllt. Die Restriktionen (5.12) sind erfüllt, da aufgrund von (iii) folgt, dass für alle Klauseln $C \in Q$, $C = \{x_i^{s(i)} : i = 1, \dots, n\}$ $n \geq 2$ mindestens eines der Literale $x_i^{s(i)}$ $i \in \{1, \dots, n\}$ den Wert 1 hat.

(6) Als zweites wird gezeigt: Wenn in $P(X, Q)$ eine zulässige Lösung existiert, dann existiert in G eine zulässige Selektion. Sei $\{x^0, x^1 : x \in X\}$ eine zulässige Lösung von $P(X, Q)$. Sei $S = \{e_C(x^s) : C \in Q, x^s \in C \text{ mit } x^s = 1\}$ die durch diese Lösung definierte Selektion in G .

Zeige zunächst, dass S vollständig ist. Gemäss (5.13) ist entweder $x^0 = 1$ oder $x^1 = 1$ für alle $x \in X$. Somit ist für jede Klausel $C \in Q$ und jedes Literal $x^s \in C$ mindestens einer der beiden disjunktiven Bogen $e_C(x^0)$ und $e_C(x^1)$ in S .

Zeige nun, dass S positiv azyklisch ist. Beachte zunächst, dass ein Zyklus in $G(S)$ keinen Bogen $a \in A_\sigma \cup A_\tau$ enthalten kann, da in σ keine Bogen enden und in τ keine Bogen starten. Bezeichne $A' = A_1 \cup A_2 \cup A_3$. Beachte weiter, dass jeder Zyklus in $G(S)$ mindestens einen konjunktiven Bogen $a \in A'$ und mindestens einen disjunktiven Bogen $e \in E$ enthält. Dies folgt direkt aus der Tatsache, dass alle konjunktiven Bogen $a \in A'$ in einem t -Knoten starten und in

einem h -Knoten enden und alle disjunktiven Bogen $e \in E$ in einem h -Knoten starten und in einem t -Knoten enden.

(i) Nehme an, dass $G(S)$ einen Zyklus Z hat, welcher einen konjunktiven Bogen $a \in A_1$ enthält. Sei $a = (t_C(x^s), h_C(x^s))$ wobei $C \in Q(x)$ und $x \in X$. Der einzige Bogen in $A' \cup E$ der im Startknoten $t_C(x^s)$ von a endet, ist $e_C(x^{1-s})$ und der einzige Bogen in $A' \cup E$ der im Endknoten $h_C(x^s)$ von a startet, ist $e_C(x^s)$. Folglich müssen diese beiden disjunktiven Bogen in Z und somit auch in S sein. Also muss $x^s = 1$ und $x^{1-s} = 1$ sein. Ein Widerspruch zur Annahme, dass $\{x^0, x^1 : x \in X\}$ eine zulässige Lösung von $P(Q, X)$ ist.

(ii) Nehme an, dass $G(S)$ einen Zyklus Z hat, welcher einen konjunktiven Bogen $a \in A_2$ enthält. Sei $a = (t_C(x^s), h_D(x^s))$ wobei $C, D \in Q(x)$ $C \neq D$ und $x \in X$. Der einzige Bogen in $A' \cup E$ der im Startknoten $t_C(x^s)$ von a endet, ist $e_C(x^{1-s})$ und der einzige Bogen in $A' \cup E$ der im Endknoten $h_D(x^s)$ von a startet, ist $e_D(x^s)$. Folglich müssen diese beiden disjunktiven Bogen in Z und somit auch in S sein. Also muss $x^s = 1$ und $x^{1-s} = 1$ sein. Ein Widerspruch zur Annahme, dass $\{x^0, x^1 : x \in X\}$ eine zulässige Lösung von $P(Q, X)$ ist.

(iii) Aus (i) und (ii) folgt, dass ein Zyklus in $G(S)$ nur Bogen aus $A_3 \cup E$ enthalten kann. Nehme an, dass $G(S)$ einen Zyklus Z hat, welcher einen konjunktiven Bogen $a \in A_3$ enthält. Sei $C \in Q$ mit $C = \{x_i^{s(i)} : i = 1, \dots, n\}$ $n \geq 2$. Sei $a = (t_C(x_i^{s(i)}), h_C(x_{i+1}^{1-s(i+1)}))$ für ein $i \in \{1, \dots, n\}$ wobei $x_{n+k}^{1-s(n+k)} = x_k^{1-s(k)}$ für $1 \leq k \leq n$ ist. Der einzige Bogen in $A_3 \cup E$ der im Endknoten $h_C(x_{i+1}^{1-s(i+1)})$ von a startet, ist der disjunktive Bogen $e_C(x_{i+1}^{1-s(i+1)}) = (h_C(x_{i+1}^{1-s(i+1)}), t_C(x_{i+1}^{s(i+1)}))$. Somit muss auch dieser disjunktive Bogen in Z sein. Weiter ist der einzige Bogen, welcher im Endknoten $t_C(x_{i+1}^{s(i+1)})$ dieses disjunktiven Bogens startet und gleichzeitig in $A_3 \cup E$ ist, gegeben durch $a' \in A_3$ $a' = (t_C(x_{i+1}^{s(i+1)}), h_C(x_{i+2}^{1-s(i+2)}))$. Dieser konjunktive Bogen muss ebenfalls in Z sein. Daraus lässt sich einfach durch Induktion ableiten, dass alle disjunktiven Bogen $e_C(x_i^{1-s(i)})$ $i = 1, \dots, n$, in Z und somit auch in S sein müssen. Folglich ist $x_i^{1-s(i)} = 1$ und $x_i^{s(i)} = 0$ für alle $i \in \{1, \dots, n\}$. Ein Widerspruch zur Annahme, dass $\{x^0, x^1 : x \in X\}$ eine zulässige Lösung von $P(Q, X)$ ist.

Zusammenfassend wurde gezeigt, dass ein Zyklus in $G(S)$ keinen konjunktiven Bogen $a \in A_1 \cup A_2 \cup A_3$ enthält. Da ein Zyklus aber mindestens einen konjunktiven Bogen enthalten muss, existiert folglich kein Zyklus in $G(S)$. Also ist S eine zulässige Selektion. ♦

Korollar 5.57. Das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen ist NP-hart.

Beweis:

th -Graphen sind spezielle disjunktive Graphen. Daraus folgt aufgrund von Satz 5.56. die Behauptung. ♦

Konjunktive Erweiterung

Aufgrund von Satz 5.53. und der Tatsache, dass das Zulässigkeitsproblem NP-hart ist, folgt unmittelbar, dass die Bestimmung der global verbotenen bzw. erzwungenen Bogen in verallgemeinerten disjunktiven Graphen ebenfalls NP-hart ist.

Korollar 5.58. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph und $S \subseteq E$. Die Bestimmung von $\chi_G(S)$ und $\varnothing_G(S)$ ist NP-hart.

Beweis:

Es wird gezeigt, dass die beiden Entscheidungsprobleme, ob $e \in \chi_G(S)$ und $e \in \varnothing_G(S)$ ist, NP-hart sind. Dazu wird gezeigt, dass sich das Zulässigkeitsproblem in disjunktiven Graphen polynomial auf diese Entscheidungsprobleme zurückführen lässt. Gemäss Korollar 5.57. ist das Zulässigkeitsproblem für disjunktive Graphen NP-hart. Zeige erstens, dass sich das Zulässigkeitsproblem für $G[S]$ polynomial zurückführen lässt auf das Entscheidungsproblem, ob $e \in \chi_G(S)$. Gemäss Satz 5.53. hat $G[S]$ genau dann eine zulässige Selektion, wenn kein $D \in \mathcal{E}$ existiert mit $D \subseteq \chi_G(S)$. $\chi_G(S)$ lässt sich bestimmen, indem für alle $e \in E$ geprüft wird, ob $e \in \chi_G(S)$. Zeige zweitens, dass sich das Zulässigkeitsproblem für $G[S]$ polynomial zurückführen lässt auf das Entscheidungsproblem, ob $e \in \varnothing_G(S)$. Gemäss Satz 5.53. hat $G[S]$ genau dann eine zulässige Selektion, wenn $\varnothing_G(S)$ positiv azyklisch ist in G . $\varnothing_G(S)$ lässt sich bestimmen, indem für alle $e \in E$ geprüft wird, ob $e \in \varnothing_G(S)$. ♦

5.5.3. Klassische disjunktive Graphen

Bedingungen für Zulässigkeit

Im nachfolgenden Satz wird gezeigt, dass für klassische disjunktive Graphen G mit positiven Gewichten die Bedingung, dass G positiv azyklisch ist, notwendig und hinreichend ist für die Existenz einer zulässigen Selektion.

Satz 5.59. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein klassischer disjunktiver Graph mit $c > 0$ und $Q \subseteq E$ eine Präselektion in G . $G[Q]$ hat genau dann eine zulässige Selektion, wenn Q positiv azyklisch ist in G .

Beweis:

Die Bedingung, dass Q positiv azyklisch ist in G , ist äquivalent zur Bedingung, dass $G[Q]$ positiv azyklisch ist. Bezeichne $H = G[Q]$. Zu zeigen ist: H hat genau dann eine zulässige Selektion, wenn H positiv azyklisch ist. Beachte zunächst: Wenn H eine zulässige Selektion hat, muss H

positiv azyklisch sein. Somit muss nur gezeigt werden: Wenn H positiv azyklisch ist, dann hat H eine zulässige Selektion.

Der Beweis geschieht durch Induktion über die Anzahl disjunktiver Mengen von H . (1) Induktionsaussage $\mathcal{P}(n)$: Für alle H mit $|\mathcal{E}_H| = n$ gilt: Wenn H positiv azyklisch ist, dann hat H eine zulässige Selektion. (2) Verankerung: Zeige, dass $\mathcal{P}(0)$ gilt. Sei H mit $|\mathcal{E}_H| = 0$. Dann ist $S = \emptyset$ eine zulässige Selektion in H , denn S ist vollständig, da $|\mathcal{E}_H| = 0$ ist, und $H(S) = H(\emptyset)$ ist gemäss Voraussetzung positiv azyklisch. (3) Induktionsschritt: Zeige, dass für alle $n \geq 0$ gilt $\mathcal{P}(n) \Rightarrow \mathcal{P}(n+1)$. Sei $n \geq 0$ und gelte $\mathcal{P}(n)$. Zeige, dass $\mathcal{P}(n+1)$ gilt.

Sei H mit $|\mathcal{E}_H| = n+1$. Sei H positiv azyklisch. Zeige: H hat eine zulässige Selektion. Da $n \geq 0$ und somit $|\mathcal{E}| = n+1 \geq 1$ existiert ein $D \in \mathcal{E}$. Sei $D = \{e_1, e_2\}$. Betrachte die beiden disjunktiven Subgraphen $H_1 = H[e_1] = \langle V, A_1, E_1, \mathcal{E}_1, c \rangle$ und $H_2 = H[e_2] = \langle V, A_2, E_2, \mathcal{E}_2, c \rangle$. Sei $i \in \{1, 2\}$. Gemäss Definition ist $\mathcal{E}_i = \mathcal{E}_H - \mathcal{E}_H(e_i)$. Da H ein klassischer disjunktiver Graph ist, gilt $\mathcal{E}_H(e_i) = \{D\}$ und folglich $|\mathcal{E}_i| = |\mathcal{E}_H| - 1 = n$. Somit ist H_i ein klassischer disjunktiver Graph mit n disjunktiven Mengen, und die Induktionsannahme $\mathcal{P}(n)$ trifft zu für H_i .

Betrachte nun den Fall, wenn H_i positiv azyklisch ist. Dann hat H_i aufgrund von $\mathcal{P}(n)$ eine zulässige Selektion S . Sei $S' = S \cup e_i$. Da S zulässig ist in H_i , ist S' eine zulässige Selektion in H .

Zeige schliesslich, dass mindestens einer der beiden Graphen H_1 und H_2 positiv azyklisch ist. Nehme an, dass H_1 und H_2 beide positiv zyklisch sind. Für beide $i \in \{1, 2\}$ gilt: Sei Z_i ein positiver Zyklus in H_i . Da $A_i = A_H \cup e_i$ und H_i positiv azyklisch ist, muss $e_i \in Z_i$ sein. Somit hat Z_i die Form $Z_i = (e_i, W_i)$ wobei W_i ein Weg von $\text{head}(e_i)$ nach $\text{tail}(e_i)$ ist in $H(\emptyset)$. Da H ein klassischer disjunktiver Graph ist, gilt $\text{tail}(e_1) = \text{head}(e_2)$ und $\text{head}(e_1) = \text{tail}(e_2)$. Folglich bildet $Z' = (W_1, W_2)$ einen geschlossenen Bogenzug von $\text{head}(e_1)$ über $\text{tail}(e_1) = \text{head}(e_2)$ nach $\text{tail}(e_2) = \text{head}(e_1)$ in $H(\emptyset)$. Da $c > 0$ ist $c(Z') > 0$. Somit ist $H(\emptyset)$ positiv zyklisch. Ein Widerspruch zur Annahme, dass H positiv azyklisch ist. ♦

Komplexität

Aus dem vorangehenden Satz folgt unmittelbar, dass das Zulässigkeitsproblem für klassische disjunktive Graphen mit positiven Bogengewichten polynomial lösbar ist. Falls jedoch negative Bogengewichte zugelassen werden, ist das Zulässigkeitsproblem NP-hart, wie in Satz 5.61. gezeigt wird.

Korollar 5.60. *Das Zulässigkeitsproblem (ohne und mit Präselektion) für klassische disjunktive Graphen mit positiven Bogengewichten ist polynomial lösbar.*

Beweis:

Sei G ein klassischer disjunktiver Graph und Q eine Präselektion in G . Gemäss Satz 5.59. hat $G[Q]$ genau dann eine zulässige Selektion, wenn $G[Q]$ positiv azyklisch ist. Das Entscheidungsproblem, ob ein Graph positiv azyklisch ist, kann mit polynomialem Aufwand gelöst werden. ♦

Satz 5.61. *Das Zulässigkeitsproblem für klassische disjunktive Graphen mit beliebigen (auch negativen) Bogengewichten ist NP-hart.*

Beweis:

Betrachte die Entscheidungsversion des Job-Shop-Problems: Gegeben ist ein Job-Shop-Problem $\Pi = (M, I, \mathcal{J}, \mu, d)$ und eine Zahl $b > 0$. Zu beantworten ist die Frage, ob der Optimalwert $\omega(\Pi)$ kleiner gleich b ist. Dieses Entscheidungsproblem ist bekanntlich NP-hart (siehe beispielsweise [Garey, M. R., Johnson D. S. (1979)]). Zeige, dass sich das Job-Shop-Entscheidungsproblem polynomial zurückführen lässt auf das Zulässigkeitsproblem in einem klassischen disjunktiven Graphen. Betrachte den durch Π definierten Job-Shop-Graphen $G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$. Es gilt $\omega(\Pi) \leq b$ genau dann, wenn eine zulässige Selektion S existiert in G^Π , so dass $\lambda(G^\Pi(S)) \leq b$ ist. Hierbei bezeichnet λ die Länge eines längsten $\sigma\tau$ -Weges im entsprechenden Graphen. Füge in G^Π einen konjunktiven Bogen von τ nach σ ein mit dem (negativen) Gewicht $c_{\tau\sigma} = -b$ und bezeichne den resultierenden disjunktiven Graphen als $G' = \langle V, A', E, \mathcal{E}, c' \rangle$ wobei $A' = A \cup (\tau, \sigma)$. Zeige nun: G' besitzt genau dann eine zulässige Selektion, wenn G^Π eine zulässige Selektion besitzt mit $\lambda(G^\Pi(S)) \leq b$. Beachte zunächst, dass $\lambda(G^\Pi(S)) = \lambda(G'(S))$ für alle Selektionen $S \subseteq E$. Dies trifft zu, da alle Bogengewichte in G^Π nicht-negativ sind und durch das Hinzufügen eines Bogen negativen Gewichts die Länge eines längsten Weges in G^Π nicht verändert wird.

Zeige einerseits: Falls G' eine zulässige Selektion hat, dann hat G^Π eine zulässige Selektion mit $\lambda(G^\Pi(S)) \leq b$. Sei S eine zulässige Selektion in G' . Dann ist $G'(S)$ positiv azyklisch. Folglich ist $\lambda(G'(S)) \leq b$, denn andernfalls würde ein längster $\sigma\tau$ -Weg zusammen mit dem Bogen (τ, σ) einen positiven Zyklus bilden. Da S zulässig ist in G' , ist S auch zulässig in G^Π . Aus $\lambda(G^\Pi(S)) = \lambda(G'(S))$ folgt somit die Behauptung.

Zeige andererseits: Falls G^Π eine zulässige Selektion hat mit $\lambda(G^\Pi(S)) \leq b$ dann hat G' eine zulässige Selektion. Sei S eine zulässige Selektion in G^Π mit $\lambda(G^\Pi(S)) \leq b$. Nehme an, dass $G'(S)$ einen positiven Zyklus Z hat. Da Z in G^Π nicht vorhanden ist, muss $(\tau, \sigma) \in Z$ sein. Da das Gewicht von Z positiv ist, existiert ein Weg von σ nach τ in G^Π mit einem Gewicht grösser als b . Ein Widerspruch. Somit ist S positiv azyklisch in G' . Da S vollständig ist in G^Π , ist S auch vollständig in G' . Folglich ist S zulässig in G' .

Zusammenfassend ist somit gezeigt, dass $\omega(\Pi) \leq b$ genau dann, wenn G' eine zulässige Selektion besitzt. ♦

Konjunktive Erweiterung

Für klassische disjunktive Graphen mit positiven Bogengewichten weisen die Mengen der global verbotenen bzw. erzwungenen Bogen eine spezielle Struktur auf. Im nachfolgenden Satz wird gezeigt, dass die global verbotenen bzw. erzwungenen Bogen identisch sind mit den direkt verbotenen bzw. erzwungenen Bogen. Weiter wird gezeigt, dass das konjunktive Erzeugnis einer Selektion S genau dann positiv azyklisch ist, wenn S positiv azyklisch ist.

Satz 5.62. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein klassischer disjunktiver Graph mit $c > 0$. Sei $S \subseteq E$. Es gilt:

$$(i) \quad \chi_G(S) = \chi_G^*(S) = \chi_G(S)$$

$$(ii) \quad \phi_G(S) = \phi_G^*(S) = \phi_G(S)$$

Beweis:

Beweis von (i): Da $\chi_G(S) \subseteq \chi_G^*(S)$ und $\chi_G^*(S) \subseteq \chi_G(S)$ genügt es, zu zeigen, dass $\chi_G(S) \subseteq \chi_G(S)$. Sei $e \in \chi_G(S)$. Zeige, dass $e \in \chi_G(S)$. Nehme an, dass $e \notin \chi_G(S)$. Dann gilt gemäss Definition, dass $G(S \cup e)$ positiv azyklisch ist. Gemäss Satz 5.59. besitzt $G[S \cup e]$ folglich eine zulässige Selektion R . Sei $R' = S \cup e \cup R$. Da R zulässig ist in $G[S \cup e]$ gilt gemäss Lemma 5.11., dass R' eine zulässige Selektion in G ist. Da $S \subseteq R'$ und $e \in \chi_G(S)$ gilt gemäss Lemma 5.36., dass $e \notin R'$. Ein Widerspruch.

Beweis von (ii): Da $\phi_G(S) \subseteq \phi_G^*(S)$ und $\phi_G^*(S) \subseteq \phi_G(S)$ genügt es, zu zeigen, dass $\phi_G(S) \subseteq \phi_G(S)$. Sei $e \in \phi_G(S)$. Zeige, dass $e \in \phi_G(S)$. Nehme an, dass $e \notin \phi_G(S)$. Gemäss Definition ist $\phi_G(S) = S \cup \Phi_G(\chi_G(S))$. Da $e \notin \phi_G(S)$ gilt $e \notin \Phi_G(\chi_G(S))$. Somit existiert kein $D \in \mathcal{E}$ mit $e \in D$, so dass $e' \in \chi_G(S)$ für alle $e' \in D - e$. Da G ein klassischer disjunktiver Graph ist, existiert genau eine disjunktive Menge D , welche e enthält. Sei $D = \{e, e'\}$. Es gilt also $e' \notin \chi_G(S)$. Da $\chi_G(S) = \chi_G(S)$ ist $e' \notin \chi_G(S)$. Gemäss Lemma 5.36. existiert somit eine zulässige Selektion S' in G mit $S \subseteq S'$ und $e' \in S'$. Da G ein klassischer disjunktiver Graph mit $c > 0$ ist, bilden e und e' einen positiven 2-Zyklus. Folglich kann e nicht in S' sein. Da aber $S \subseteq S'$ und $e \in \phi_G(S)$ gilt gemäss Lemma 5.36., dass $e \in S'$. Ein Widerspruch. ♦

Satz 5.63. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein klassischer disjunktiver Graph mit $c > 0$. Sei $S \subseteq E$. $\phi_G^*(S)$ ist genau dann positiv azyklisch in G , wenn S positiv azyklisch ist in G .

Beweis:

Zeige einerseits: Falls $\phi_G^*(S)$ positiv azyklisch ist in G , dann ist S positiv azyklisch in G . Dies trifft zu, da $S \subseteq \phi_G^*(S)$. Zeige andererseits: Falls S positiv azyklisch ist in G , dann ist $\phi_G^*(S)$ positiv azyklisch in G . Sei G positiv azyklisch. Nehme an, dass $\phi_G^*(S)$ positiv zyklisch ist. Gemäss Lemma 5.39. ist $\phi_G^*(S) \subseteq S'$ für alle zulässigen Selektionen S' in G mit $S \subseteq S'$. Folglich

existiert keine zulässige Selektion S' in G mit $S \subseteq S'$. Da $G[S]$ positiv azyklisch ist, existiert gemäss Satz 5.59. eine zulässige Selektion R' in $G[S]$. Gemäss Lemma 5.11. ist $R' = S \cup R$ eine zulässige Selektion in G mit $S \subseteq R'$. Ein Widerspruch. ♦

5.5.4. Job-Shop-Graphen

Job-Shop-Graphen sind spezielle klassische disjunktive Graphen mit positiven Bogengewichten. Somit gelten alle im vorangehenden Abschnitt gezeigten Resultate auch für Job-Shop-Graphen.

5.5.5. Lager-Graphen

Bedingungen für Zulässigkeit

Im folgenden Satz wird eine notwendige, aber im allgemeinen nicht hinreichende Bedingung gezeigt, damit ein Lager-Graph eine zulässige Selektion hat. Der Beweis liefert ein Verfahren, mit welchem eine zulässige Selektion konstruiert werden kann, falls die angegebene Bedingung erfüllt ist.

Satz 5.64. Sei $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^s)$ ein Lager-Problem und $G = G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ der zugehörige Lager-Graph. Sei $Q \subseteq E$ eine Präselektion in G . Die folgende Bedingung ist hinreichend, aber nicht notwendig dafür, dass $G[Q]$ eine zulässige Selektion besitzt:

Es existiert eine Numerierung der Jobs von Π , d.h. eine Funktion $n : \mathcal{J} \rightarrow \{1, 2, \dots, |\mathcal{J}|\}$ mit $n(J) \neq n(J')$ für alle $J, J' \in \mathcal{J}$ mit $J \neq J'$, so dass für alle Bogen $(v, w) \in Q$ gilt, dass $n(J_v) < n(J_w)$.

Beweis:

Zur Erinnerung sei bemerkt, dass gemäss Definition 5.16. J_v denjenigen Job bezeichnet, welcher die zum Knoten v gehörende Operation enthält. Sei nun n eine Numerierung der Jobs von Π , so dass $n(J_v) < n(J_w)$ für alle $(v, w) \in Q$. Bezeichne J_i den Job, welcher die Operation $i \in I$ enthält. Bezeichne $S_i = \{(h_i, t_j) : j \in I_{\mu(i)}, n(J_i) < n(J_j)\} \cap E_{G[Q]}$ für $i \in I$. Sei $S = \bigcup_{i \in I} S_i$ eine vollständige Selektion in $G[Q]$. Zeige, dass S positiv azyklisch ist in $G[Q]$. Nehme an, dass $G[Q](S) = G(Q \cup S)$ einen positiven Zyklus $Z = \{v_1, v_2, \dots, v_k\}$ mit $k \geq 2$ hat. Da $v_1 = v_k$ muss $n(J_{v_1}) = n(J_{v_k})$ sein. Da aber für alle Bogen $(v, w) \in Q \cup S$ gilt $n(J_v) < n(J_w)$ ist $n(J_{v_r}) < n(J_{v_{r+1}})$ für alle $r = 1, \dots, k-1$. Folglich gilt $n(J_{v_1}) < n(J_{v_k})$. Ein Widerspruch. Somit bildet S eine zulässige Selektion in $G[Q]$.

Zeige noch, dass die Bedingung nicht notwendig ist. Betrachte als Gegenbeispiel den folgenden Lager-Graphen mit Präselektion $Q = \{a, b\}$

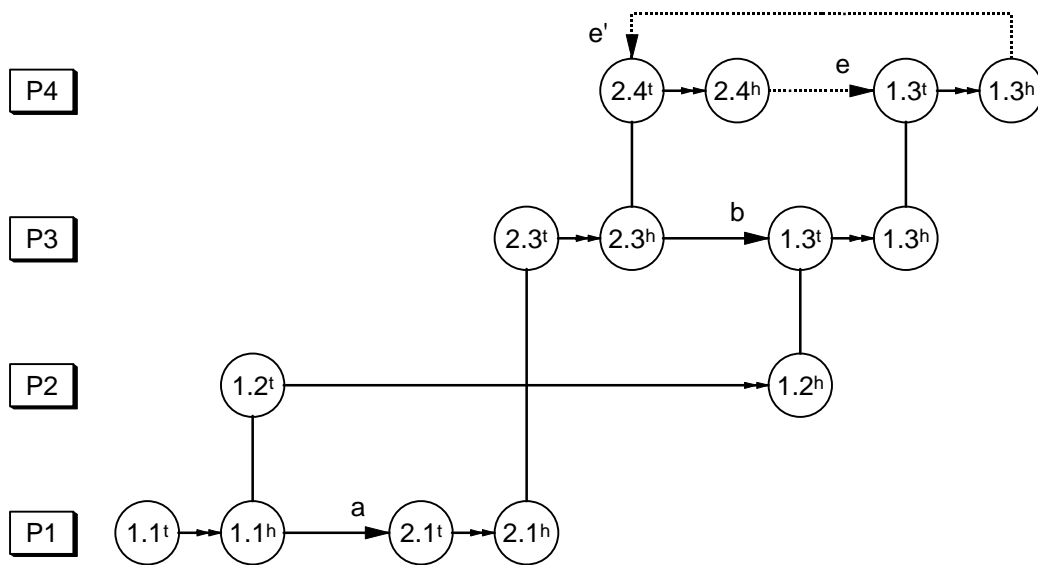


Abbildung 5.20. Bedingung ist nicht notwendig für Existenz einer zulässigen Selektion

Da a von Job 1 nach Job 2 führt und b von Job 2 nach Job 1, existiert keine Jobnumerierung n mit den oben genannten Eigenschaften. Aber offensichtlich ist $S = \{e\}$ eine zulässige Selektion in $G[Q]$. ♦

Im vorangehenden Beweis wurde gezeigt, wie in einem Lager-Graphen $G[Q]$ mit Präselektion Q eine zulässige Selektion $S = \bigcup_{i \in I} S_i$ konstruiert werden kann, falls für die Jobs von G eine Numerierung mit den erwähnten Eigenschaften existiert. Gemäss Konstruktion stellt S einen Ablaufplan dar, bei welchem die Jobs von G der Reihe nach in der durch n vorgegebenen Reihenfolge abgearbeitet werden. Dies bedeutet, dass die Reihenfolge der Operationen auf jedem Prozessor der durch n vorgegebenen Jobreihenfolge entspricht, d.h. für alle Prozessoren $m \in M$ und alle Operationen $i, j \in I_m$ mit $i \in J$ und $j \in J'$ gilt, dass i vor j ausgeführt wird, falls $n(J) < n(J')$.

Für Lager-Graphen $G = G[\emptyset]$ ohne Präselektion, d.h. für den Spezialfall $Q = \emptyset$ existiert trivialerweise immer eine Numerierung mit den genannten Eigenschaften. Daraus ergeben sich in diesem Fall die folgenden einfachen Erkenntnisse: (1) Es existiert immer eine zulässige Selektion in G . (2) Eine zulässige Selektion kann konstruiert werden, indem die Jobs von G in beliebiger Reihenfolge durchnummeriert und entsprechend abgearbeitet werden.

Eine wichtige Bedeutung hat Satz 5.64. bei der Konstruktion einer zulässigen Startlösung für das Lager-Problem. In Abschnitt 4.4. wurde erläutert, wie die räumlichen Nebenbedingungen in

den Korridoren zu einer Präselektion Q von disjunktiven Bogen führen. In Abschnitt 6.6. wird gezeigt, dass für ein Lagerproblem, bei welchem keine Umlagerungen vorgenommen werden, genau dann eine Jobnumerierung mit den erwähnten Eigenschaften existiert, wenn Q positiv azyklisch ist in G . Dies trifft jedoch nicht zu, falls Umlagerungen ausgeführt werden.

Ein Lagerproblem ohne Umlagerungen hat also genau dann eine zulässige Lösung, wenn Q positiv azyklisch ist in G , d.h. wenn die räumlichen Nebenbedingungen konsistent sind. Somit ist das Problem der zulässigen Startlösung in Lager-Graphen ohne Umlagerungen polynomial lösbar, obwohl für allgemeine Präselektionen Q das Zulässigkeitsproblem NP-hart ist, wie im folgenden Satz gezeigt wird.

Komplexität

In Satz 5.56. wurde gezeigt, dass das Zulässigkeitsproblem für th -Graphen NP-hart ist. In der Beweisführung wurde das SAT-Problem polynomial zurückgeführt auf das Zulässigkeitsproblem in th -Graphen. Es wurde gezeigt, wie zu jeder Instanz (X, Q) des SAT-Problems ein th -Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ konstruiert werden kann, welcher genau dann eine zulässige Selektion besitzt, wenn das SAT-Problem eine zulässige Lösung hat.

Die th -Graphen, welche von den SAT-Instanzen abgeleitet werden, sind keine Lager-Graphen, sondern stellen eine Verallgemeinerung derselben dar. Somit wäre es prinzipiell möglich, dass das Zulässigkeitsproblem mit Präselektion für Lager-Graphen polynomial lösbar ist, obwohl das Zulässigkeitsproblem für th -Graphen NP-hart ist. Der nächste Satz zeigt, dass dies nicht der Fall ist und das Zulässigkeitsproblem mit Präselektion für Lager-Graphen NP-hart ist.

Satz 5.65.

- (i) *Das Zulässigkeitsproblem ohne Präselektion für Lager-Graphen ist polynomial lösbar.*
- (ii) *Das Zulässigkeitsproblem mit Präselektion für Lager-Graphen ist NP-hart. Dies gilt schon für Probleme mit höchstens zwei Operationen auf jedem Prozessor.*

Beweis:

Beweis von (i): Die Jobs können in beliebiger Reihenfolge abgearbeitet werden. Siehe dazu die Bemerkungen zum Beweis von Satz 5.64.

Beweis von (ii): Sei (X, Q) eine Instanz des SAT-Problems und $G = \langle V, A, E, \mathcal{E}, c \rangle$ der th -Graph, welcher gemäss Beweis von Satz 5.56. zu dieser Instanz konstruiert wird. Im folgenden wird gezeigt, wie sich zum th -Graphen G (mit polynomialen Aufwand) ein Lager-Graph $G^\Pi[R] = \langle V', A', E', \mathcal{E}', c' \rangle$ mit Präselektion R konstruieren lässt, welcher genau dann eine zulässige Selektion besitzt, wenn G eine zulässige Selektion hat. Dies bedeutet, dass sich das SAT-

Problem polynomial zurückführen lässt auf das Zulässigkeitsproblem mit Präselektion in Lager-Graphen. Daraus folgt, dass dieses Zulässigkeitsproblem NP-hart ist.

(1) Zeige, dass der th -Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$ einen Lager-Graphen darstellt, wenn die konjunktiven Bogen $A_2 \cup A_3$ aus G entfernt werden. Gemäss Beweis von Satz 5.56. ist G wie folgt definiert. Die Knotenmenge ist gegeben durch:

$$V = \{t_C(x^0), h_C(x^0), t_C(x^1), h_C(x^1) : C \in \mathcal{Q}, x^s \in C\} \cup \{\sigma, \tau\}$$

Für die konjunktiven Bogen gilt $A = A_1 \cup A_2 \cup A_3 \cup A_\sigma \cup A_\tau$ wobei:

$$A_1 = \{(t_C(x^0), h_C(x^0)), (t_C(x^1), h_C(x^1)) : C \in \mathcal{Q}, x^s \in C\}$$

$$A_2 = \{(t_C(x^1), h_D(x^1)), (t_D(x^0), h_C(x^0)), (t_C(x^0), h_D(x^0)), \\ (t_D(x^1), h_C(x^1)) : C, D \in \mathcal{Q}(x), C \neq D, x \in X\}$$

$$A_3 = \{(t_C(x_i^{s(i)}), h_C(x_{i+1}^{1-s(i+1)})) : C \in \mathcal{Q}, C = \{x_i^{s(i)} : i = 1, \dots, n\}\}$$

$$A_\sigma = \{(\sigma, t_C(x^0)), (\sigma, t_C(x^1)) : C \in \mathcal{Q}, x^s \in C\}$$

$$A_\tau = \{(h_C(x^0), \tau), (h_C(x^1), \tau) : C \in \mathcal{Q}, x^s \in C\}$$

Die disjunktiven Bogen und Mengen sind gegeben durch:

$$E = \{(h_C(x^0), t_C(x^1)), (h_C(x^1), t_C(x^0)) : C \in \mathcal{Q}, x^s \in C\}$$

$$\mathcal{E} = \{\{(h_C(x^0), t_C(x^1)), (h_C(x^1), t_C(x^0))\} : C \in \mathcal{Q}, x^s \in C\}$$

Betrachte den Teilgraphen $H = \langle V, A_1 \cup A_\sigma \cup A_\tau, E, \mathcal{E}, c \rangle$ welcher resultiert, wenn die konjunktiven Bogen A_2 und A_3 ignoriert werden. Dieser Teilgraph kann als Lagergraph $G^{\Pi'}$ aufgefasst werden. Das entsprechende Lager-Problem $\Pi' = (M', I', \mathcal{J}', \mu', d, d^t, d^h, d^s)$ wird wie folgt definiert: Zu jedem Literal x^s in einer Klausel $C \in \mathcal{Q}$ wird eine Maschine $m(x_C^s)$ definiert, d.h. $M' = \{m(x_C^s) : C \in \mathcal{Q}, x^s \in C\}$. Zu jedem Literal x^s in einer Klausel $C \in \mathcal{Q}$ werden weiter zwei Operationen x_C^0 und x_C^1 definiert, d.h. $I' = \{x_C^0, x_C^1 : C \in \mathcal{Q}, x^s \in C\}$. Die Jobs von H bestehen alle aus einer einzigen Operation und sind gegeben durch $\mathcal{J}' = \{\{x_C^0\}, \{x_C^1\} : C \in \mathcal{Q}, x^s \in C\}$. Für die Zuordnung der Prozessoren zu den Operationen gilt, dass die beiden Operationen x_C^0, x_C^1 welche zu jedem Literal x^s einer Klausel C gehören, auf demselben Prozessor $m(x_C^s)$ ausgeführt werden, d.h. $\mu'(x_C^0) = \mu'(x_C^1) = m(x_C^s)$ für alle $C \in \mathcal{Q}, x^s \in C$. Für die Gewichte d, d^t, d^h, d^s können beliebige positive Zahlen gewählt werden, da die Gewichtung in G bei der Beweisführung keine Rolle spielt (ausser der Bedingung der Positivität).

Offensichtlich entspricht der auf diese Weise konstruierte Lager-Graph $G^{\Pi'}$ (abgesehen von den Gewichten) offensichtlich dem Teilgraphen H , d.h. es gilt $H = G^{\Pi'}$. Nicht berücksichtigt im Subgraphen H sind die konjunktiven Bogen $A_2 \cup A_3$. Für alle Bogen in $A_2 \cup A_3$ gilt, dass sie von einem t -Knoten zu einem h -Knoten führen. In einem Lager-Graphen gibt es zwei Typen von Bogen, welche von einem t -Knoten zu einem h -Knoten führen: (1) th -Bogen und (2) Synchronisationsbogen. Die Bogen in $A_2 \cup A_3$ können nicht als th -Bogen interpretiert werden, da sie die

Knoten von zwei verschiedenen Operationen verbinden. Sie können auch nicht als Synchronisationsbogen aufgefasst werden, da diese immer paarweise in entgegengesetzter Richtung auftreten. Somit lässt sich der gesamte th -Graph G nicht als Lager-Graph interpretieren. Im folgenden wird gezeigt, wie sich der th -Graph G durch Hinzufügen von Knoten und konjunktiven Bogen so erweitern lässt, dass er einen Lager-Graphen darstellt.

(2) Konstruiere aus dem th -Graphen G ein Lager-Problem $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^s)$. Die Operationen I von Π umfassen erstens die Operationen von Π' , welche im folgenden als $I^Q = I'$ bezeichnet werde. Weiter werden für jede Operation $i \in I^Q$ zwei zusätzliche Operationen a^i und b^i definiert, welche zusammen mit i einen Job bilden. Darüber hinaus werden zu jeder Operation $i \in I^Q$ zwei zusätzliche Jobs $A^i = \{A_0^i, A_1^i, \dots, A_{k(i)}^i\}$ und $B^i = \{B_1^i, B_2^i, \dots, B_{l(i)}^i, B_{l(i)+1}^i\}$ definiert.

Die Anzahl Operationen von Job A^i bzw. B^i ist gegeben durch $k(i) + 1$ bzw. $l(i) + 1$, wobei $k(i)$ bzw. $l(i)$ die Anzahl der aus dem Knoten t_i hinausführenden bzw. in den Knoten h_i hineinführenden Bogen in $A_2 \cup A_3$ entspricht. Bezeichne A_i^+ bzw. A_i^- die hinein- bzw. hinausführenden Bogen in G , d.h.

$$A_i^+ = \delta^+(t_i) \cap (A_2 \cup A_3) \quad \text{und} \quad A_i^- = \delta^-(h_i) \cap (A_2 \cup A_3), \quad i \in I^Q$$

Es gilt $k(i) = |A_i^+|$ und $l(i) = |A_i^-|$. Seien die Bogen in A_i^+ bzw. A_i^- beliebiger Reihenfolge durchnummeriert, d.h.

$$A_i^+ = \{e_1^i, e_2^i, \dots, e_{k(i)}^i\} \quad \text{und} \quad A_i^- = \{f_1^i, f_2^i, \dots, f_{l(i)}^i\}, \quad i \in I^Q$$

Da jeder Bogen $e \in A_2 \cup A_3$ in einem \mathcal{K} -Knoten startet und in einem \mathcal{K} -Knoten endet, gilt $A_2 \cup A_3 = \bigcup_{i \in I^Q} A_i^+ = \bigcup_{i \in I^Q} A_i^-$.

Weiter gilt, dass für jeden Bogen $e \in A_2 \cup A_3$ eindeutige Operationen $i, j \in I^Q$ und eindeutige Indizes $1 \leq r \leq k(i)$ bzw. $1 \leq s \leq l(j)$ existieren, so dass $e = e_r^i = f_s^j$ ist.

Die Gesamtmenge der Operationen von Π ist gegeben durch $I = I^Q \cup I^{a,b} \cup I^{A,B}$, wobei:

$$I^Q = I' = \{x_C^0, x_C^1 : C \in \mathcal{Q}, x^s \in C\}$$

$$I^{a,b} = \{a^i, b^i : i \in I^Q\}$$

$$I^{A,B} = \{A_0^i, A_1^i, \dots, A_{k(i)}^i, B_1^i, B_2^i, \dots, B_{l(i)+1}^i : i \in I^Q\}$$

Die Menge der Jobs ist gegeben durch $\mathcal{J} = \mathcal{J}^Q \cup \mathcal{J}^{A,B}$, wobei:

$$\mathcal{J}^Q = \{\{a^i, i, b^i\} : i \in I^Q\}$$

$$\mathcal{J}^{A,B} = \{\{A_0^i, A_1^i, \dots, A_{k(i)}^i\}, \{B_1^i, B_2^i, \dots, B_{l(i)+1}^i\} : i \in I^Q\}$$

Die Operationen dieser Jobs sind der Reihe nach gemäss ihrer Ausführungsreihenfolge aufgeführt.

Die Menge der Prozessoren umfasst erstens die Prozessoren von Π' , welche im folgenden als $M^Q = M'$ bezeichnet werden. Dazu kommt für jede Operation a^i bzw. b^i , $i \in I^Q$, ein weiterer Prozessor $m(a^i)$ bzw. $m(b^i)$. Schliesslich wird für jeden Bogen $e \in A_2 \cup A_3$ ein weiterer Prozessor $m(e)$ eingeführt. Die Gesamtmenge der Prozessoren ist somit gegeben als $M = M^Q \cup M^{a,b} \cup M^{A_2 \cup A_3}$, wobei:

$$M^Q = M' = \{m(x_C^s) : C \in Q, x^s \in C\}$$

$$M^{a,b} = \{m(a^i), m(b^i) : i \in I^Q\}$$

$$M^{A_2 \cup A_3} = \{m(e) : e \in A_2 \cup A_3\}$$

Die Zuordnung der Prozessoren zu den Operationen ist wie folgt:

$$\mu(x_C^0) = \mu(x_C^1) = m(x_C^s), \quad C \in Q, x^s \in C$$

$$\mu(a^i) = \mu(A_0^i) = m(a^i) \quad \text{und} \quad \mu(b^i) = \mu(B_{l(i)+1}^i) = m(b^i), \quad i \in I^Q$$

$$\mu(A_r^i) = \mu(B_s^j) = m(e), \quad e \in A_2 \cup A_3, e = e_r^i = f_s^j$$

(3) Zu dem beschriebenen Lager-Problem Π wird der zugehörige Lager-Graph G^Π konstruiert. In G^Π wird auf folgende Weise eine Präselektion R von disjunktiven Bogen ausgewählt: Für alle $i \in I^Q$ wird die Operation a^i vor A_0^i und die Operation $B_{l(i)+1}^i$ vor b^i ausgeführt. Weiter gilt für jeden Bogen $e \in A_2 \cup A_3$ mit $e = e_r^i = f_s^j$ dass die Operation A_r^i vor der Operation B_s^j ausgeführt wird. Die Präselektion R ist somit gegeben durch $R = R^{a,b} \cup R^{A,B}$, wobei:

$$R^{a,b} = \{(h(a^i), t(A_0^i)), (h(B_{l(i)+1}^i), t(b^i)) : i \in I^Q\}$$

$$R^{A,B} = \{(h(A_r^i), t(B_s^j)) : e \in A_2 \cup A_3, e = e_r^i = f_s^j\}$$

Der resultierende Lager-Graph $G^\Pi[R]$ mit Präselektion R ist in der untenstehenden Abbildung schematisch dargestellt. Es werden zwei Operationen $i, j \in I^Q$ gezeigt, für welche im ursprünglichen th -Graphen G eine Bogen von t_i nach h_j führt. Die Illustration zeigt, wie zu den Operationen i und j die Operation a^i, b^i bzw. a^j, b^j und die Jobs A^i, B bzw. A^j, B hinzugefügt werden. Die zur Präselektion R gehörenden Bogen sind gestrichelt eingezeichnet.

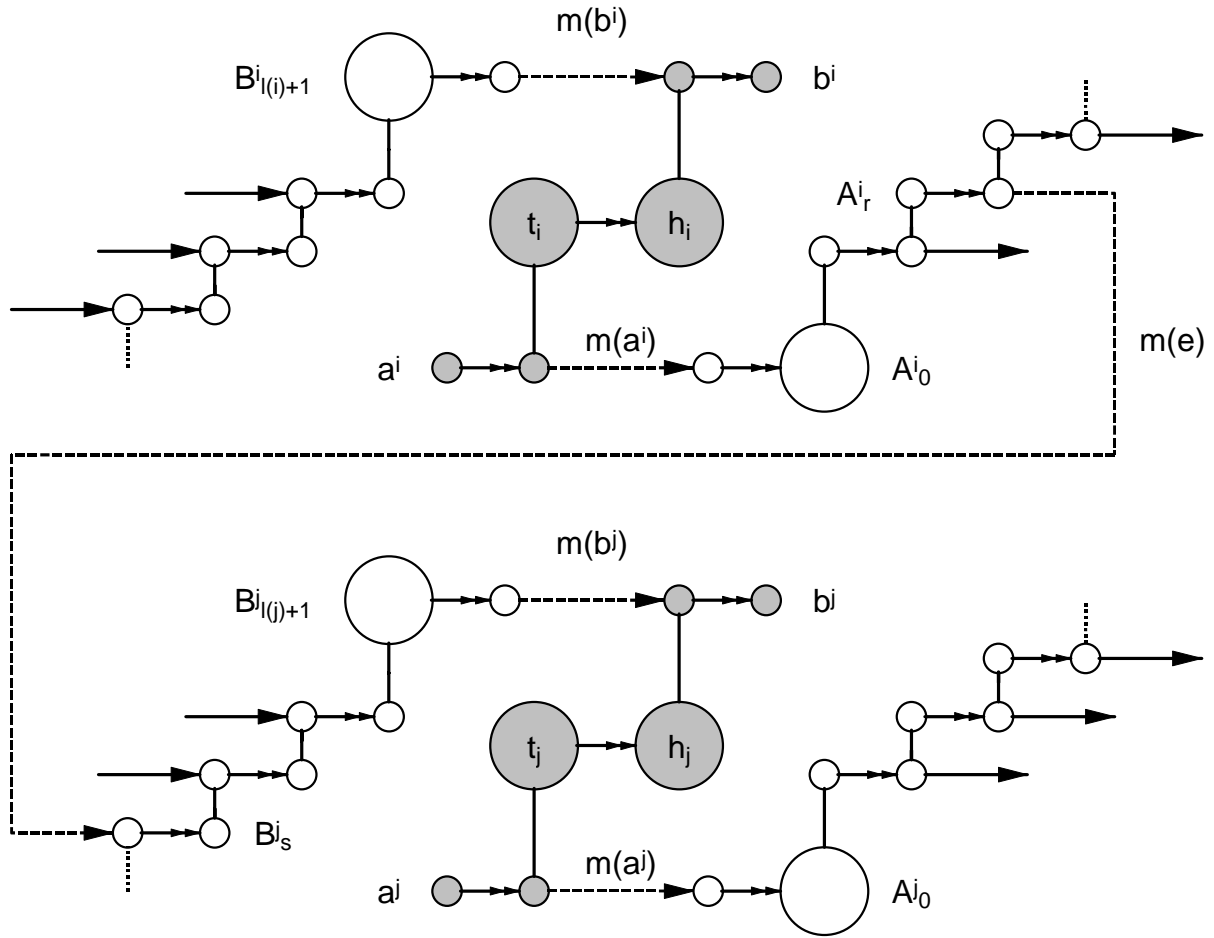


Abbildung 5.21. Konstruktion des Lager-Graphen G^Π

(4) Zeige, dass im Graphen $G^\Pi[R]$ für jeden Bogen $e \in A_2 \cup A_3$ ein eindeutiger Weg von $\text{tail}(e)$ nach $\text{head}(e)$ existiert. Sei $e \in A_2 \cup A_3$. Dann existieren gemäß Konstruktion des Graphen G zwei Operationen $i, j \in I^Q$, so dass $e = (t_i, h_j)$. Weiter existieren Indizes $1 \leq r \leq k(i)$ bzw. $1 \leq s \leq l(j)$, so dass $e = e_r^i = f_s^j$ ist. Folglich ist der Bogen $(h(A_r^i), t(B_s^j))$ in der Präselektion R enthalten. Somit existiert der folgende Weg $P(e)$ von $\text{tail}(e) = t_i$ nach $\text{head}(e) = h_j$:

$$P(e) = (t_i, h(a^i), t(A_0^i), h(A_0^i), t(A_1^i), h(A_1^i), \dots, t(A_r^i), h(A_r^i), t(B_s^j), h(B_s^j), \\ t(B_{s+1}^j), h(B_{s+1}^j), \dots, t(B_{l(j)+1}^j), h(B_{l(j)+1}^j), t(b^j), h_j)$$

(5) Sei $G^\Pi[R] = \langle V', A', E', \mathcal{E}', c' \rangle$. Gemäss Konstruktion von $G^\Pi[R]$ gilt $A_1 \cup A_\sigma \cup A_\tau \subseteq A'$. Weiter gilt, dass die Menge der disjunktiven Bogen bzw. der disjunktiven Mengen in G und $G^\Pi[R]$ identisch ist. Diese Mengen sind gegeben durch:

$$E = E' = \{(h_C(x^0), t_C(x^1)), (h_C(x^1), t_C(x^0)) : C \in Q, x^s \in C\}$$

$$\mathcal{E} = \mathcal{E}' = \{\{(h_C(x^0), t_C(x^1)), (h_C(x^1), t_C(x^0))\} : C \in Q, x^s \in C\}$$

Daraus folgt, dass jede vollständige Selektion in G eine vollständige Selektion in $G^\Pi[R]$ ist und umgekehrt jede vollständige Selektion in $G^\Pi[R]$ eine vollständige Selektion in G darstellt.

(6) Zeige, dass jede zulässige Selektion in $G^\Pi[R]$ eine zulässige Selektion in G ist. Sei S eine zulässige Selektion in $G^\Pi[R]$. S ist eine vollständige Selektion in G . Zeige, dass S positiv azyklisch ist in G . Nehme an, dass $G(S)$ einen positiven Zyklus Z enthält. Fall 1: Sei $Z \subseteq A_1 \cup A_\sigma \cup A_\tau \cup S$. Da $A_1 \cup A_\sigma \cup A_\tau \subseteq A'$ enthält auch $G^\Pi[R](S)$ den Zyklus Z . Ein Widerspruch zur Annahme, dass S zulässig ist in $G^\Pi[R]$. Fall 2: Nehme an, dass Z mindestens einen Bogen $e \in A_2 \cup A_3$ enthält. Da G gemäss Konstruktion positiv azyklisch ist, gilt weiter, dass Z mindestens einen disjunktiven Bogen $e \in S$ enthält. Somit hat Z die Form $Z = (W_1, e_1, W_2, e_2, \dots, W_q, e_q)$ mit $q \geq 1$, wobei $W_p \subseteq A_1 \cup A_\sigma \cup A_\tau$ und $e_p \in A_2 \cup A_3$ für alle $1 \leq p \leq q$. Da für jeden Bogen $e_p \in A_2 \cup A_3$ ein Weg $P(e_p)$ von $\text{tail}(e_p)$ nach $\text{head}(e_p)$ in $G^\Pi[R]$ existiert, enthält $G^\Pi[R](S)$ den geschlossenen Bogenzug $Z' = (W_1, P(e_1), \dots, W_q, P(e_q))$. Das Gewicht von Z' ist positiv, da alle Bogengewichte in $G^\Pi[R](S)$ nicht-negativ sind und Z' mindestens einen Bogen $e_p \in A_2 \cup A_3$ enthält, dessen Gewicht positiv ist. Ein Widerspruch zur Annahme, dass S zulässig ist in $G^\Pi[R]$.

(7) Zeige umgekehrt, dass jede zulässige Selektion in G eine zulässige Selektion in $G^\Pi[R]$ ist. Sei S eine zulässige Selektion in G . S ist eine vollständige Selektion in $G^\Pi[R]$. Nehme an, dass $G^\Pi[R](S)$ einen positiven Zyklus Z enthält. Fall 1: Sei $Z \subseteq A_1 \cup A_\sigma \cup A_\tau \cup S$. Da $A_1 \cup A_\sigma \cup A_\tau \subseteq A'$, enthält auch $G(S)$ den Zyklus Z . Ein Widerspruch zur Annahme, dass S zulässig ist in G . Fall 2: Bezeichne $A'' = A' - (A_1 \cup A_\sigma \cup A_\tau)$. Nehme an, dass Z mindestens einen Bogen $e \in A''$ enthält. Da $G^\Pi[R]$ gemäss Konstruktion positiv azyklisch ist, gilt weiter, dass Z mindestens einen disjunktiven Bogen $e \in S$ enthält. Somit hat Z die Form $Z = (W_1, U_1, W_2, U_2, \dots, W_q, U_q)$ mit $q \geq 1$, wobei $W_p \subseteq A''$ und $U_p \subseteq A_1 \cup A_\sigma \cup A_\tau$ für alle $1 \leq p \leq q$. Gemäss Konstruktion von $G^\Pi[R]$ existieren für alle $1 \leq p \leq q$ zwei eindeutige Operationen $i_p, j_p \subseteq I^Q$, so dass W_p einen Weg von $t(i_p)$ nach $h(j_p)$ darstellt. Daraus folgt weiter gemäss Konstruktion von $G^\Pi[R]$, dass ein Bogen $e_p \in A_2 \cup A_3$ existiert mit $\text{tail}(e_p) = t(i_p)$ und $\text{head}(e_p) = h(j_p)$. Folglich existiert in $G(S)$ die geschlossene Bogenfolge $Z' = (e_1, U_1, \dots, e_q, U_q)$. Das Gewicht von Z' ist positiv, da alle Bogengewichte in G nicht negativ sind und die Gewichte von e_p , $1 \leq p \leq q$ positiv sind. Ein Widerspruch zur Annahme, dass S positiv zulässig ist in G .

(8) Zusammenfassend wurde gezeigt, dass sich jede Instanz des SAT-Problems auf das Zulässigkeitsproblem in einem Lager-Graphen $G^\Pi[R]$ mit Präselektion R zurückführen lässt. Der Graph $G^\Pi[R]$ hat gemäss Konstruktion die Eigenschaft, dass auf jedem Prozessor höchstens zwei Operationen ausgeführt werden. Somit ist die Behauptung bewiesen. ♦

5.5.6. 1-Job-Lager-Graphen

Ein 1-Job-Lager-Graph $G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ wird ausgehend von einem Lager-Graphen $G = G^\Pi[Q]$ mit Präselektion Q konstruiert, indem ein einzelner Job J von Π ausgewählt wird und alle disjunktiven Bogen (und die entsprechenden disjunktiven Mengen), welche nicht mit einem Knoten von J inzidieren, aus G entfernt werden. G_J enthält somit nur disjunktive Bogen, welche mit einem Knoten von J inzidieren. Die disjunktiven Bogen setzen sich zusammen aus $E_J = E_J^t \cup E_J^h$, wobei E_J^t die Bogen sind, welche in einem t -Knoten von J enden, und E_J^h die Bogen, welche in einem h -Knoten von J starten. Die Bogen in E_J^t werden als t -Bogen und diejenigen in E_J^h als h -Bogen bezeichnet. Anschaulich stellen die t -Bogen Präzedenzbedingungen dar, welche besagen, dass eine Operation von J nach einer andern Operation ausgeführt werden muss, und die h -Bogen besagen, dass eine Operation von J vor einer andern Operation ausgeführt werden muss. Im folgenden werden einige strukturelle Resultate im Zusammenhang mit 1-Job-Lager-Graphen erarbeitet, welche für die Optimierung in Lager-Graphen von Bedeutung sind (vgl. Kapitel 6).

Strukturelle Resultate

In den folgenden Untersuchungen werden 1-Job-Lager-Graphen betrachtet, welche konjunktiv abgeschlossen und positiv azyklisch sind. Als erstes wird gezeigt, dass jede Selektion S , welche ausschliesslich t - bzw. h -Bogen enthält, positiv azyklisch ist. Als Konsequenz daraus ergibt sich, dass in einem konjunktiv abgeschlossenen, positiv azyklischen 1-Job-Lager-Graphen stets zwei zulässige, disjunkte Selektionen existieren. Schliesslich wird gezeigt, dass für eine Selektion S , welche ausschliesslich t - bzw. h -Bogen enthält, das konjunktive Erzeugnis positiv azyklisch ist und ebenfalls nur t - bzw. h -Bogen enthält.

Satz 5.66. Sei $H = G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph. Sei $S \subseteq E_J$ eine Selektion in H mit der Eigenschaft, dass entweder $S \subseteq E_J^t$ oder $S \subseteq E_J^h$. Dann ist S positiv azyklisch in H .

Beweis:

Nehme an, dass S nicht positiv azyklisch ist in H , d.h. dass $H(S)$ einen positiven Zyklus Z hat. Da $H(\emptyset)$ positiv azyklisch ist, muss Z mindestens einen Bogen aus S enthalten. Zhat also die Form $Z = (e_1, W_1, e_2, W_2, \dots, e_p, W_p)$ mit $p \geq 1$ wobei e_q und W_q ein Weg ist in $H(S)$ mit $q \in \{1, \dots, p\}$. Da Z ein Zyklus ist, gilt $e_q \neq e_{q'}$ für $q, q' \in \{1, \dots, p\}$ mit $q \neq q'$.

Zeige, dass $p \geq 2$ ist. Nehme an, dass $p = 1$ ist. Dann ist $Z = (e_1, W_1)$ ein positiver Zyklus in $H(e_1)$. Folglich ist $e_1 \in \chi(H)$ und für den Partner gilt $e_1' \in \phi(H)$. Da aber H konjunktiv abgeschlossen ist, gilt gemäss Definition, dass $\phi(H) = \emptyset$. Ein Widerspruch. Sei im folgenden $p \geq 2$.

Sei $e_q \in E_{i(q)}$ für $q \in \{1, \dots, p\}$ d.h. $i(q)$ ist die Operation von J die $r(q)$ te Operation von J ist.

Zeige, dass ein $q \in \{1, \dots, p\}$ existiert, so dass $r(q) \geq r(q+1)$ ist, wobei $r(p+1) = r(1)$. Nehme an, dass für alle $q \in \{1, \dots, p\}$ gilt $r(q) < r(q+1)$. Aus der Transitivität folgt der Widerspruch, dass $r(1) < r(p+1) = r(1)$ ist.

Sei nun $q \in \{1, \dots, p\}$ mit $r(q) \geq r(q+1)$. Bezeichne zur Vereinfachung $f = e_q$, $g = e_{q+1}$, $j = i(q)$, $i = i(q+1)$, $s = r(q)$ und $r = r(q+1)$. Bezeichne weiter $W = W_q$ wobei der Weg W von $head(f)$ nach $tail(g)$ führt.

Fall 1: Sei $S \subseteq E_f^t$. Betrachte zur Illustration Abbildung 5.22. Sei $f = (h_p, t_j)$ und $g = (h_k, t_i)$ wobei $i = J_p$, $j = J_s$ und $s \geq r$ ist. In $H(\emptyset)$ existiert der Weg $U = (t_i = t(J_r), h(J_r), t(J_{r+1}), h(J_{r+1}), \dots, t(J_s) = t_j)$. Folglich existiert in $H(\emptyset)$ der Weg (U, W) von $head(g) = t_i$ nach $tail(g) = h_k$. Somit enthält $H(\emptyset)$ den Zyklus $Z' = (U, W, g)$. Das Gewicht $c(Z')$ dieses Zyklus ist positiv, da $c(t(J_r), h(J_r)) > 0$ ist und alle Bogengewichte nicht-negativ sind. Somit ist $g \in \chi(H)$ und folglich $g' \in \phi(H)$. Da aber H konjunktiv abgeschlossen ist, gilt gemäß Definition, dass $\phi(H) = \emptyset$. Ein Widerspruch.

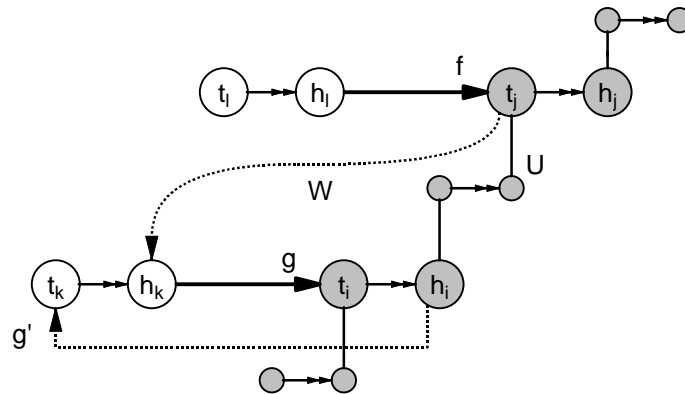


Abbildung 5.22. $H(\emptyset)$ enthält den Zyklus $Z' = (U, W, g)$

Fall 2: Sei $S \subseteq E_f^h$. Betrachte zur Illustration Abbildung 5.23. Sei $f = (h_j, t_l)$ und $g = (h_i, t_k)$ wobei $i = J_p$, $j = J_s$ und $s \geq r$ ist. In $H(\emptyset)$ existiert der Weg $U = (h_i = h(J_r), t(J_{r+1}), h(J_{r+1}), t(J_{r+2}), \dots, h(J_s) = h_j)$. Folglich existiert in $H(\emptyset)$ der Weg (W, U) von $head(f) = t_l$ nach $tail(f) = h_j$. Somit enthält $H(\emptyset)$ den Zyklus $Z' = (W, U, f)$. Das Gewicht $c(Z')$ dieses Zyklus ist positiv, da $c(t(J_s), h(J_s)) > 0$ ist und alle Bogengewichte nicht-

negativ sind. Somit ist $f \in \chi(H)$ und folglich $f' \in \phi(H)$. Da aber H konjunktiv abgeschlossen ist, gilt gemäss Definition, dass $\phi(H) = \emptyset$. Ein Widerspruch. ♦

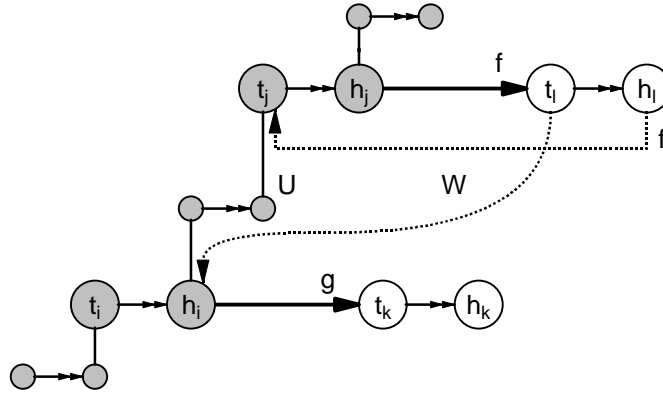


Abbildung 5.23. $H(f)$ enthält den Zyklus $Z' = (W, U, f)$

Korollar 5.67. Sei $H = G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph. Dann sind E_J^t und E_J^h zwei zulässige Selektionen in H .

Beweis:

Gemäss Satz 5.66. sind die Selektionen E_J^t und E_J^h positiv azyklisch in H . Zeige, dass E_J^t und E_J^h vollständig sind in H . Da H ein 1-Job-Shop-Graph ist, haben alle disjunktiven Mengen $D \in \mathcal{E}_J$ die Form $D = \{e, e'\}$ mit $e \in E_J^t$ und $e' \in E_J^h$. Folglich ist $D \cap E_J^t \neq \emptyset$ und $D \cap E_J^h \neq \emptyset$ für alle $D \in \mathcal{E}_J$. Somit sind die Selektionen E_J^t und E_J^h vollständig in H . ♦

Korollar 5.68. Sei $H = G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph. Sei $S \subseteq E_J$ eine Selektion in H mit der Eigenschaft, dass entweder $S \subseteq E_J^t$ oder $S \subseteq E_J^h$. Es gilt:

- (i) Falls $S \subseteq E_J^t$, dann $\phi_H^*(S) \subseteq E_J^t$, und falls $S \subseteq E_J^h$, dann $\phi_H^*(S) \subseteq E_J^h$.
- (ii) $\phi_H^*(S)$ ist positiv azyklisch in H .

Beweis:

Beweis von (i): Fall 1: Sei $S \subseteq E_J^t$. Gemäss Korollar 5.67. ist E_J^t eine zulässige Selektion in H . Da $S \subseteq E_J^t$, gilt gemäss Lemma 5.39., dass $\phi_H^*(S) \subseteq E_J^t$. Fall 2: Sei $S \subseteq E_J^h$. Analog.

Beweis von (ii): Fall 1: Sei $S \subseteq E_J^t$. Gemäss Korollar 5.67. ist E_J^t eine zulässige Selektion in H . Gemäss Beweis von (i) gilt $\phi_H^*(S) \subseteq E_J^t$. Folglich ist $\phi_H^*(S)$ positiv azyklisch in H . Fall 2: Sei $S \subseteq E_J^h$. Analog. ♦

Bedingungen für Zulässigkeit

Aus Korollar 5.67. lässt sich direkt eine notwendige und hinreichende Bedingung für die Existenz einer zulässigen Selektion in 1-Job-Lager-Graphen herleiten.

Korollar 5.69. Sei $H = G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ ein 1-Job-Lager-Graph und Q eine Präselektion in H . $H[Q]$ hat genau dann eine zulässige Selektion, wenn $\phi_H^*(Q)$ positiv azyklisch ist in H .

Beweis:

(1) Zeige einerseits: Falls $H[Q]$ eine zulässige Selektion hat, dann ist $\phi_H^*(Q)$ positiv azyklisch in H . Sei S eine zulässige Selektion in $H[Q]$. Gemäss Lemma 5.11. ist $S' = Q \cup S$ eine zulässige Selektion in H . Da $Q \subseteq S'$, gilt gemäss Lemma 5.39., dass $\phi_H^*(Q) \subseteq S'$. Da S' positiv azyklisch ist in H , muss auch $\phi_H^*(Q)$ positiv azyklisch sein in H .

(2) Zeige andererseits: Falls $\phi_H^*(Q)$ positiv azyklisch ist in H dann hat $H[Q]$ eine zulässige Selektion. Sei $\phi_H^*(Q)$ positiv azyklisch in H . Sei $K = H[\phi_H^*(Q)]$ ist ein positiv azyklischer 1-Job-Lager-Graph. Gemäss Lemma 5.31. ist K konjunktiv abgeschlossen. Sei $S = E_J^t \cap E_K$. Gemäss Korollar 5.67. ist S eine zulässige Selektion in K . Daraus folgt gemäss Lemma 5.11., dass $\phi_H^*(Q) \cup S$ eine zulässige Selektion in H ist und $(\phi_H^*(Q) \cup S) \cap E_{H[Q]}$ eine zulässige Selektion in $H[Q]$. ♦

Komplexität

Als eine Konsequenz der vorangehenden Ausführungen ergibt sich, dass das Zulässigkeitsproblem für 1-Job-Lager-Graphen polynomial lösbar ist.

Korollar 5.70. Das Zulässigkeitsproblem (ohne und mit Präselektion) für 1-Job-Lager-Graphen ist polynomial lösbar.

Beweis:

Sei $H = G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ ein 1-Job-Lager-Graph und Q eine Präselektion in H . Gemäss Korollar 5.69. hat $H[Q]$ genau dann eine zulässige Selektion, wenn $\phi_H^*(Q)$ positiv azyklisch ist in H . Gemäss Satz 5.51. ist die Berechnung von $\phi_H^*(Q)$ mit polynomialem Aufwand möglich. ♦

Konjunktive Erweiterung

Im folgenden wird gezeigt, dass in 1-Job-Lager-Graphen die Mengen der erzwungenen bzw. verbotenen Bogen identisch sind mit den Mengen der global verbotenen bzw. erzwungenen Bogen.

Satz 5.71. Sei $H = G_J = \langle V, A, E_J, \mathcal{E}_J, c \rangle$ ein 1-Job-Lager-Graph und $S \subseteq E_J$. Es gilt:

- (i) $\chi_H^*(S) = \chi_H(S)$
- (ii) $\varphi_H^*(S) = \varphi_H(S)$

Beweis:

Gemäss Lemma 5.39. ist $\chi_H^*(S) \subseteq \chi_H(S)$ und $\varphi_H^*(S) \subseteq \varphi_H(S)$. Somit muss noch gezeigt werden, dass $\chi_H(S) \subseteq \chi_H^*(S)$ und $\varphi_H(S) \subseteq \varphi_H^*(S)$.

Fall 1: Sei $\varphi_H^*(S)$ positiv zyklisch in H . Dann gilt gemäss Lemma 5.33., dass $\chi_H^*(S) = E$ und $\varphi_H^*(S) = E$. Folglich ist $\chi_H(S) \subseteq \chi_H^*(S)$ und $\varphi_H(S) \subseteq \varphi_H^*(S)$.

Fall 2: Sei $\varphi_H^*(S)$ positiv azyklisch in H . Dann gilt gemäss Lemma 5.31., dass $K = H[\varphi_H^*(S)]$ ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph ist. Sei $W^t = E_J^t \cap E_K$ und $W^h = E_J^h \cap E_K$. Gemäss Korollar 5.67. sind W^t und W^h zwei zulässige Selektionen in K . Folglich gilt gemäss Lemma 5.11., dass $U^t = \varphi_H^*(S) \cup W^t$ und $U^h = \varphi_H^*(S) \cup W^h$ zwei zulässige Selektionen in H sind. Da $S \subseteq \varphi_H^*(S)$ gilt $S \subseteq U^t$ und $S \subseteq U^h$.

(1) Zeige, dass $\chi_H(S) \subseteq \chi_H^*(S)$. Sei $e \in \chi_H(S)$. Nehme an, dass $e \notin \chi_H^*(S)$. Fall 1: Sei $e \in \varphi_H^*(S)$. Dann gilt $e \in U^t$. Somit ist U^t eine zulässige Selektion in H mit $S \subseteq U^t$ und $e \in U^t$. Daraus folgt gemäss Lemma 5.36., dass $e \in \chi_H(S)$. Ein Widerspruch. Fall 2: Sei $e \in \overline{\varphi_H^*(S)}$. Dann gilt für den Partner e' von e , dass $e' \in \varphi_H^*(S)$. Gemäss Lemma 5.29. ist $\chi_H^*(S) = \chi_H(\varphi_H^*(S))$. Da $e \notin \chi_H^*(S)$, gilt somit, dass $G(\varphi_H^*(S) \cup e)$ positiv azyklisch ist. Da aber e zusammen mit seinem Partner e' einen Zyklus bildet in H , ist $e' \notin \varphi_H^*(S)$. Ein Widerspruch. Fall 3: Sei $e \notin \varphi_H^*(S)$ und $e \notin \overline{\varphi_H^*(S)}$. Dann gilt $e \notin \mathcal{E}(\varphi_H^*(S))$. Gemäss Lemma 5.12. ist $E_K = E - \mathcal{E}(\varphi_H^*(S))$. Folglich ist $e \in E_H$. Da $E_K = W^t \cup W^h$, gilt somit $e \in W^t$ oder $e \in W^h$ und folglich $e \in U^t$ oder $e \in U^h$. Da aber $e \in \chi_H(S)$, gilt gemäss Lemma 5.36. für alle zulässigen Selektionen S' in H mit $S \subseteq S'$ dass $e \in S'$. Folglich gilt $e \in U^t$ und $e \in U^h$. Ein Widerspruch.

(2) Zeige, dass $\varphi_H(S) \subseteq \varphi_H^*(S)$. Sei $e \in \varphi_H(S)$. Nehme an, dass $e \notin \varphi_H^*(S)$. Da $e \in \varphi_H(S)$, gilt gemäss Lemma 5.36., dass $e \in S'$ für alle zulässigen Selektionen S' in H mit $S \subseteq S'$. Da $S \subseteq U^t$ und $S \subseteq U^h$, gilt somit $e \in U^t$ und $e \in U^h$. Da $e \notin \varphi_H^*(S)$, muss $e \in W^t$ und $e \in W^h$ sein. Da aber $E_J^t \cap E_J^h = \emptyset$ gilt $W^t \cap W^h = \emptyset$. Ein Widerspruch. ♦

Heuristik zur Steuerungsoptimierung

In diesem Kapitel wird eine Heuristik für das Lager-Problem, d.h. das Problem der Steuerungsoptimierung für automatisierte Kompaktlager, präsentiert. Es wird ein Verbesserungsverfahren vorgeschlagen, welches von der Problemdarstellung in einem disjunktiven Lager-Graphen ausgeht. Der verwendete Nachbarschaftsbegriff basiert im wesentlichen auf dem Prinzip der konjunktiven Erweiterung und dem Konzept der 1-Job-Lager-Graphen.

In Abschnitt 6.1. werden grundlegende Betrachtungen im Zusammenhang mit der Optimierung in verallgemeinerten disjunktiven Graphen angeführt. Es werden Ansätze diskutiert für den Entwurf von Eröffnungs- und Verbesserungsverfahren. Insbesondere wird gezeigt, welche Konsequenzen aus der Tatsache resultieren, dass das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen NP-hart ist, im Gegensatz zu Job-Shop-Graphen. Aufgrund dieses Umstands ist der Entwurf eines geeigneten Nachbarschaftsbegriffs für das Lager-Problem wesentlich anspruchsvoller als im Falle des Job-Shop-Problems. Abschnitt 6.2. präsentiert einige Definitionen und Resultate im Zusammenhang mit verallgemeinerten disjunktiven Graphen, welche anschliessend beim Entwurf eines Nachbarschaftsbegriffs benötigt werden. In Abschnitt 6.3. wird ein Nachbarschaftsbegriff erarbeitet als Grundlage für ein iteratives Verbesserungsverfahren für das Lager-Problem. Der Nachbarschaftsbegriff basiert auf der Idee des Austausches von kritischen Bogen und verwendet das Konzept der 1-Job-Lager-Graphen. Abschnitt 6.4. illustriert diesen Nachbarschaftsbegriff anhand eines Beispiels. In Abschnitt 6.5. wird eine Heuristik für das Lager-Problem formuliert. Es handelt sich um Abstiegsverfahren auf der Basis des entworfenen Nachbarschaftsbegriffs. In Abschnitt 6.6. wird das Problem der Bestimmung einer zulässigen

Startlösung diskutiert. Das Auffinden einer zulässigen Lösung in Lager-Graphen ist NP-hart. Es wird jedoch gezeigt, dass für Lager-Probleme ohne Umlagerungen stets eine zulässige Startlösung existiert, welche in polynomialer Zeit konstruiert werden kann. In Abschnitt 6.7. werden die numerischen Resultate präsentiert.

6.1. Optimierung in disjunktiven Graphen

Optimierung in disjunktiven Graphen beschäftigt sich mit dem Problem, in einem disjunktiven Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$ eine zulässige Selektion $S \subseteq \mathbb{E}$ zu finden, welche die Länge eines längsten $\sigma\tau$ -Weges in $G(S)$ minimiert. Dieses Problem wird das disjunktive Graphenproblem genannt. In den vorangehenden Ausführungen wurde gezeigt, dass sich das Lager-Problem mit Hilfe von Lager-Graphen als disjunktives Graphenproblem formulieren lässt. Angesichts der Tatsache, dass das disjunktive Graphenproblem komplexitätstheoretisch NP-hart ist, wird ein heuristischer Ansatz zur Lösung des Lager-Problems ins Auge gefasst.

Heuristische Verfahren lassen sich allgemein aufteilen in Eröffnungsverfahren (constructive methods) und Verbesserungsverfahren (iterative improvement, local search). In einem Eröffnungsverfahren wird eine zulässige Lösung konstruiert, indem eine bestehende Teillösung schrittweise zu einer zulässigen Lösung erweitert wird. In einem Verbesserungsverfahren wird eine bestehende, zulässige Lösung sukzessive verbessert. Ausgehend von einer zulässigen Lösung x wird versucht, durch lokale Veränderungen eine bessere Lösung x' zu konstruieren. x' wird eine Nachbarlösung von x genannt, und der Übergang von x zu x' als ein Move bezeichnet. Beim Entwurf eines Verbesserungsverfahrens besteht das Hauptproblem darin, einen geeigneten Nachbarschaftsbegriff zu finden für die Übergänge zwischen den einzelnen Lösungen. Sowohl die Effektivität als auch die Effizienz eines Verbesserungsverfahrens hängen in wesentlichem Masse von der Wahl des Nachbarschaftsbegriffs ab.

Bei gegebenem Nachbarschaftsbegriff sind verschiedene Ansätze für ein Verbesserungsverfahren möglich. Zwei grundlegende generische Methoden seien hier erwähnt: Abstiegsverfahren (descent methods) und Tabu-Verfahren (taboo search). In einem Abstiegsverfahren werden nur Moves ausgeführt, welche die Zielfunktion verbessern. Eine Startlösung wird solange verbessert, bis das Verfahren in einem (bezüglich der Nachbarschaft) lokalen Optimum steckenbleibt. Bei Tabu-Verfahren werden nach gewissen Kriterien auch Moves zugelassen, welche die Zielfunktion nicht verbessern oder sogar verschlechtern, wobei mittels geeigneter Massnahmen verhindert werden muss, dass das Verfahren zyklisiert. Tabu-Verfahren sind generell aufwendiger, haben aber ein grösseres Optimierungspotential, da lokale Optima bis zu einem gewissen Grad überwunden werden können und somit ein grösserer Bereich des Lösungsraums abgesucht wird.

6.1.1. Eröffnungsverfahren

Ein Eröffnungsverfahren für ein disjunktives Graphenproblem $G = \langle V, A, E, \mathcal{E}, c \rangle$ kann folgendermaßen aussehen: Ausgehend von der leeren Selektion $S = \emptyset$ werden aufgrund bestimmter Auswahlkriterien sukzessive disjunktive Bogen zu S hinzugefügt. Die Erweiterung wird solange fortgesetzt, bis S vollständig ist in G . Falls am Ende S positiv azyklisch ist in G , bildet S eine zulässige Lösung. Bei diesem Vorgehen liegt in jedem Stadium eine partielle Selektion S vor, welche sukzessive zu einer zulässigen Selektion S' erweitert wird. Damit das Verfahren funktioniert, muss in jedem Stadium garantiert sein, dass sich die bestehende Teilselektion S zu einer zulässigen Selektion erweitern lässt. Anders ausgedrückt bedeutet dies, dass der disjunktive Subgraph $G[S]$ in jedem Stadium eine zulässige Selektion haben muss.

Eine notwendige Bedingung für die Existenz einer zulässigen Selektion in $G[S]$ ist, dass S positiv azyklisch ist in G , d.h. dass $G[S]$ positiv azyklisch ist. In Abschnitt 5.5. wurde gezeigt, dass diese Bedingung für klassische disjunktive Graphen auch hinreichend ist für die Existenz einer zulässigen Selektion in $G[S]$. Für verallgemeinerte disjunktive Graphen trifft dies jedoch nicht zu. Eine notwendige und hinreichende Bedingung ist in diesem Fall gemäß Abschnitt 5.5. gegeben durch die Forderung, dass $G[\Phi_G(S)]$ positiv azyklisch ist, wobei $\Phi_G(S)$ die durch \mathcal{E} global erzwungenen Bogen darstellt. Da aber das Zulässigkeitsproblem und folglich auch die Bestimmung von $\Phi_G(S)$ in verallgemeinerten disjunktiven Graphen NP-hart ist (vgl. Abschnitt 5.5.), kann die Frage, ob sich eine partielle Selektion S zu einer zulässigen Selektion S' erweitern lässt, kaum mit polynomialem Aufwand beantwortet werden.

Für verallgemeinerte disjunktive Graphen ist die Konstruktion einer zulässigen Selektion ein NP-hartes Problem und somit viel schwieriger als das Problem in klassischen disjunktiven Graphen. Im Zusammenhang mit dem Job-Shop-Problem wurde eine Vielzahl von Eröffnungsverfahren entwickelt, bei welchen auf der Basis von Prioritätsregeln (dispatching rules) sukzessive eine zulässige Lösung aufgebaut wird (list scheduling). Diese konstruktiven Methoden basieren auf der Tatsache, dass in Job-Shop-Graphen jede zulässige Teillösung (d.h. jede positiv azyklische Teilselektion) zu einer vollständigen Lösung erweitert werden kann. Da diese Eigenschaft für verallgemeinerte disjunktive Graphen (und insbesondere auch für Lager-Graphen) nicht zutrifft, lassen sich diese auf Prioritätsregeln basierenden Methoden nicht auf das Lager-Problem übertragen. In Abschnitt 6.6. wird das Problem der Konstruktion einer zulässigen Startlösung für das Lager-Problem diskutiert.

6.1.2. Verbesserungsverfahren

Ein Verbesserungsverfahren für ein disjunktives Graphenproblem $G = \langle V, A, E, \mathcal{E}, c \rangle$ funktioniert typischerweise wie folgt: Ausgehend von einer zulässigen Selektion S wird versucht, durch lokale Veränderungen eine bessere zulässige Selektion S' daraus zu konstruieren. S' wird eine Nachbarselektion von S genannt, und der Übergang von S zu S' als ein Move bezeichnet. Ein

Move besteht darin, gewisse Bogen aus S zu entfernen und durch andere disjunktive Bogen zu ersetzen, so dass die resultierende Selektion S' wieder zulässig, d.h. vollständig und positiv azyklisch, ist in G . Eine Nachbarselektion hat somit die Form $S' = S - U \cup U'$, wobei $U, U' \subseteq E$ ist.

Falls in G alle disjunktiven Mengen genau zwei disjunktive Bogen umfassen und jeder disjunktive Bogen in genau einer disjunktiven Menge vorkommt, d.h. falls $|D| = 2$ für alle $D \in \mathcal{E}$ und $|\mathcal{E}(e)| = 1$ für alle $e \in E$, dann wird jeder Bogen $e \in U$ ersetzt durch seinen Partner e' , d.h. es gilt $U' = \bar{U} = \mathcal{E}(U)^{\circ} - U$. Diese Bedingung muss in diesem Fall gelten, damit S' wiederum vollständig ist in G . Beachte, dass die erwähnte Eigenschaft, dass $|D| = 2$ für $D \in \mathcal{E}$ und $|\mathcal{E}(e)| = 1$ für $e \in E$, im Falle der Lager-Graphen zutrifft.

Damit eine Nachbarselektion S' besser ist als die Selektion S , muss die Länge eines längsten $\sigma\tau$ -Weges in $G(S')$ kleiner sein als in $G(S)$. Dies ist offensichtlich nur möglich, wenn für jeden längsten $\sigma\tau$ -Weg in $G(S)$ mindestens ein Bogen, welcher auf diesem Weg liegt, aus S entfernt wird. Bezeichne einen Bogen, welcher auf einem längsten $\sigma\tau$ -Weg liegt, als kritischen Bogen. Als eine Abschwächung dieser Bedingung kann gesagt werden, dass in einem Move mindestens ein kritischer Bogen ausgetauscht werden muss, damit die Nachbarlösung S' besser sein kann.

Job-Shop-Graphen

Diverse Optimierungsmethoden für das Job-Shop-Problem basieren auf dieser Idee des Austausches von kritischen disjunktiven Bogen im disjunktiven Job-Shop-Graphen. Eine ausführliche Diskussion dieses Ansatzes findet sich in dem grundlegenden Artikel von [Balas, E. (1969)]. Darin wird ein Verbesserungsverfahren beschrieben, bei welchem eine Selektion S sukzessive verbessert wird, in dem jeweils ein kritischer Bogen durch seinen Partner ersetzt wird. Eine Nachbarselektion hat demnach die Form $S' = S - e \cup e'$, wobei e einen kritischen disjunktiven Bogen in $G(S)$ darstellt und e' dem Partner von e entspricht. Beachte noch, dass der Austausch eines kritischen disjunktiven Bogen $e = (v, w)$ bedeutet, dass die beiden zu den Knoten v und w gehörenden Operationen in der Ausführungsreihenfolge vertauscht werden auf dem entsprechenden Prozessor.

Im erwähnten Artikel wird auch die Frage der Zulässigkeit der Nachbarselektion S' untersucht. Es wird gezeigt, dass in einem Job-Shop-Graphen G durch den Austausch eines kritischen Bogens e kein Zyklus entstehen kann. Daraus folgt, dass die Nachbarselektion $S' = S - e \cup e'$ zulässig ist, denn S' ist vollständig in G , da e durch seinen Partner e' ersetzt wird, und S' ist positiv azyklisch in G , da $G(S')$ azyklisch ist. Diese wichtige Eigenschaft von Job-Shop-Graphen sei an dieser Stelle nochmals nachvollzogen. Aus dem nachfolgenden Beweis ist ersichtlich, dass diese Eigenschaft aufgrund der speziellen Bogengewichtung in Job-Shop-Graphen zu Stande kommt. Die Bogengewichte in Job-Shop-Graphen haben die Eigenschaft, dass alle Bogen, welche in einem bestimmten Knoten starten, dasselbe Gewicht haben: Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein Job-Shop-Graph und $v \in V - \{\sigma, \tau\}$. Für alle Bogen $(v, w) \in A \cup E$ gilt $c_{vw} = d_v$, wobei d_v die Dauer der zum Knoten v gehörenden Operation darstellt.

Satz 6.1. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein Job-Shop-Graph. Sei $S \subseteq E$ eine zulässige Selektion in G und e ein kritischer Bogen in $G(S)$. Sei $S' = S - e \cup e'$ die Nachbarselektion, welche erhalten wird, indem e durch seinen Partner e' ausgetauscht wird. Dann ist $G(S')$ azyklisch.

Beweis:

Beachte zunächst, dass $G(S)$ azyklisch ist, da S positiv azyklisch ist in G und alle Bogen (ausser den in σ startenden Bogen) positive Gewichte haben. Sei $e = (v, w)$. Da e ein kritischer Bogen ist in $G(S)$, existiert ein längster σ -Weg W in $G(S)$, welcher die Form $W = (U_1, e, U_2)$ hat. Hierbei ist U_1 ein Weg von σ nach v und U_2 ein Weg von w nach τ in $G(S)$. Nehme nun an, dass $G(S')$ einen Zyklus Z besitzt. Da $G(S)$ azyklisch ist, muss $e' \in Z$ sein. Folglich hat Z die Form $Z = (e', U)$, wobei U ein Weg von v nach w ist in $G(S)$. Aufgrund der Struktur von G umfasst U mindestens zwei Bogen. Sei $f \in U$ der erste Bogen in U . Da f in v startet, hat f das Gewicht $c_f = d_v = c_e$. Da alle Bogengewichte positiv sind und U mindestens zwei Bogen umfasst, gilt $c(U) > c_e$. Sei $W' = (U_1, U, U_2)$. W' ist ein Weg von σ nach τ in $G(S)$. Da $c(U) > c_e$ gilt $c(W') > c(W)$. Folglich ist W kein längster Weg in $G(S)$. Ein Widerspruch. ♦

Lager-Graphen

Für Lager-Graphen trifft die oben erwähnte Eigenschaft nicht zu, dass durch den Austausch eines kritischen Bogens keine Zyklen entstehen. Es ist im Gegenteil so, dass ein solcher Austausch typischerweise einen Zyklus erzeugt. Der Grund dafür ist einerseits zu finden in den Synchronisationsbedingungen und andererseits in der topologischen Struktur des Lager-System-Graphen, aufgrund welcher die möglichen Prozessorreihenfolgen innerhalb eines Jobs bestimmt werden. Wie schon erwähnt, bedeutet der Austausch eines kritischen Bogens eine Vertauschung der entsprechenden Operationen in der Ausführungsreihenfolge. Wenn nun in einem Lager-Problem beispielsweise die Reihenfolge zweier Quergang-Operationen vertauscht wird, dann muss zwingendermassen auch die Reihenfolge der anschliessenden Aufzug-Operationen vertauscht werden (sofern die beiden Jobs denselben Aufzug verwenden). Wenn die Reihenfolge auf dem Aufzug nicht geändert wird, entsteht eine unzulässige, d.h. positiv zyklische Selektion. Dieses Beispiel veranschaulicht, dass in Lager-Graphen typischerweise Zyklen entstehen, wenn einzelne kritische Bogen ausgetauscht werden.

Somit stellt sich beim Entwurf eines Verbesserungsverfahrens für Lager-Graphen die Frage nach einem geeigneten Nachbarschaftsbegriff. Als Vorgabe besteht die Forderung, dass in einem Move mindestens ein kritischer Bogen ausgetauscht werden muss. Da durch den Austausch eines einzelnen kritischen Bogens positive Zyklen entstehen können, müssen in einem Move im allgemeinen mehrere Bogen ausgetauscht werden. Dabei stellt sich die Frage, welche zusätzlichen Bogen ausgetauscht werden müssen, damit beim Austausch eines kritischen Bogens eine zulässige Nachbarselektion resultiert.

Aus dieser Frage resultiert die folgende Problemstellung: Gegeben ist ein Lager-Graph $G = \langle V, A, E, \mathcal{E}, c \rangle$. Sei $S \subseteq E$ eine zulässige Selektion in G und $e \in S$ ein kritischer disjunktiver Bogen. Gesucht ist eine Teilmenge $U \subseteq S$ mit $e \in U$ so dass $S' = S - U \cup \bar{U}$ positiv azyklisch ist in G . Hierbei bezeichnet \bar{U} die zu U komplementäre Selektion, d.h. die Menge der Partner der Bogen aus U . Die Menge U soll hierbei nicht beliebig wählbar, sondern möglichst klein sein, damit die bestehende Selektion S nur wenig verändert wird. Allgemein wird in einem Verbesserungsverfahren angestrebt, dass eine bestehende Lösung durch einen Move möglichst wenig (d.h. möglichst "lokal") verändert wird, damit "die schon optimierten Teile" der Lösung erhalten bleiben und durch den Move nicht wieder zerstört werden.

Das Ziel der obigen Problemstellung besteht darin, eine Nachbarselektion S' zu finden, welche den kritischen Bogen e nicht enthält und sich im übrigen möglichst wenig unterscheidet von der Selektion S . Unter diesem Gesichtspunkt lässt sich das Problem, in leicht veränderter Form, wie folgt formulieren: Gesucht ist eine (möglichst kleine) Teilmenge $U \subseteq S$ mit $e \in U$ so dass der disjunktive Subgraph $G[S - U \cup e']$ eine zulässige Selektion besitzt. Falls eine Menge $U \subseteq S$ mit $e \in U$ gefunden ist, für welche $G[S - U \cup e']$ eine zulässige Selektion W besitzt, dann ist auch eine zulässige Nachbarselektion S' gefunden, welche e nicht enthält. Diese ist gegeben durch $S' = (S - U \cup e') \cup W$.

Da das Zulässigkeitsproblem für Lager-Graphen NP-hart ist, ist die Frage, ob für ein $U \subseteq E$ der Subgraph $G[S - U \cup e']$ eine zulässige Selektion besitzt, NP-hart. Daraus lässt sich erahnen, dass die Suche nach einer (möglichst kleinen) Menge $U \subseteq E$ mit $e \in U$ für welche $G[S - U \cup e']$ eine zulässige Selektion besitzt, ein anspruchsvolles Problem darstellt. Da die Konstruktion einer zulässigen Nachbarselektion dieses Problem beinhaltet, ist der Entwurf eines Nachbarschaftsbegriffs für Lager-Graphen offensichtlich wesentlich schwieriger als im Falle der Job-Shop-Graphen. An dieser Stelle sei bemerkt, dass die Suche nach einem geeigneten Nachbarschaftsbegriff für Lager-Graphen eine der zentralen Herausforderungen der vorliegenden Arbeit darstellt.

6.2. Vorbereitung

In diesem Abschnitt werden einige Definitionen und Resultate im Zusammenhang mit disjunktiven Graphen präsentiert. Diese Ausführungen werden in den folgenden Abschnitten benötigt, in welchen ein Nachbarschaftsbegriff und ein darauf basierendes lokales Verbesserungsverfahren für das Lager-Problem entwickelt wird.

Lemma 6.2. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph mit $|\mathcal{E}(e)| = 1$ für alle $e \in E$. Sei $R \subseteq E$ eine Selektion in G und $S \subseteq E_{G[R]}$ eine Selektion in $G[R]$. Es gilt:

$$(i) \quad \chi_{G[R]}(S) = \chi_G(R \cup S) - \mathcal{E}(R)^\circ$$

$$(ii) \quad \phi_{G[R]}(S) = \phi_G(R \cup S) - \mathcal{E}(R)^\circ$$

Beweis:

Beweis von (i): Gemäss Definition ist $\chi_{G[R]}(S) = \{e \in E_{G[R]} : G[R](S \cup e) \text{ ist positiv zyklisch}\}$. Gemäss Lemma 5.12. gilt $E_{G[R]} = E - \mathcal{E}(R)^\circ$. Aufgrund von Lemma 5.10. ist $G[R](S \cup e) = G(R \cup S \cup e)$.

Somit gilt $\chi_{G[R]}(S) = \{e \in E - \mathcal{E}(R)^\circ : G(R \cup S \cup e) \text{ ist positiv zyklisch}\}$. Gemäss Definition ist $\chi_G(R \cup S) = \{e \in E : G(R \cup S \cup e) \text{ ist positiv zyklisch}\}$. Daraus folgt die Behauptung.

Beweis von (ii): Gemäss Definition gilt $\phi_{G[R]}(S) = S \cup U$ und $\phi_G(R \cup S) = (R \cup S) \cup W$ wobei $U = \Phi_{G[R]}(\chi_{G[R]}(S))$ und $W = \Phi_G(\chi_G(R \cup S))$

(1) Zeige erstens, dass gilt $\phi_G(R \cup S) - \mathcal{E}(R)^\circ = S \cup (W - \mathcal{E}(R)^\circ)$. Gemäss Definition gilt $\phi_G(R \cup S) - \mathcal{E}(R)^\circ = (R \cup S \cup W) - \mathcal{E}(R)^\circ = (R - \mathcal{E}(R)^\circ) \cup (S - \mathcal{E}(R)^\circ) \cup (W - \mathcal{E}(R)^\circ)$. Da $R \subseteq \mathcal{E}(R)^\circ$, gilt $R - \mathcal{E}(R)^\circ = \emptyset$. Gemäss Lemma 5.12. ist $E_{G[R]} = E - \mathcal{E}(R)^\circ$. Da $S \subseteq E_{G[R]}$ gilt somit $S \cap \mathcal{E}(R)^\circ = \emptyset$. Folglich ist $S - \mathcal{E}(R)^\circ = S$. Daraus folgt die Behauptung.

(2) Zeige zweitens, dass gilt $U = W - \mathcal{E}(R)^\circ$.

Zeige einerseits, dass $U \subseteq W - \mathcal{E}(R)^\circ$. Sei $e \in U$. Zeige, dass $e \in W - \mathcal{E}(R)^\circ$. Da $e \in U$ gilt gemäss Definition von U , dass $e \in E_{G[R]}$ und dass ein $D \in \mathcal{E}_{G[R]}$ mit $e \in D$ existiert, so dass $e' \in \chi_{G[R]}(S)$ für alle $e' \in D - e$. Da $\mathcal{E}_{G[R]} \subseteq \mathcal{E}$ gilt $D \in \mathcal{E}$. Gemäss Beweis von (i) gilt $\chi_{G[R]}(S) = \chi_G(R \cup S) - \mathcal{E}(R)^\circ$. Somit ist gezeigt, dass ein $D \in \mathcal{E}$ mit $e \in D$ existiert, so dass $e' \in \chi_G(R \cup S)$ für alle $e' \in D - e$. Folglich ist $e \in W$. Da $e \in E_{G[R]}$ und $E_{G[R]} = E - \mathcal{E}(R)^\circ$ gilt $e \in W - \mathcal{E}(R)^\circ$.

Zeige andererseits, dass $W - \mathcal{E}(R)^\circ \subseteq U$. Sei $e \in W - \mathcal{E}(R)^\circ$. Zeige, dass $e \in U$. Da $e \in W$ existiert ein $D \in \mathcal{E}$ mit $e \in D$ so dass $e' \in \chi_G(R \cup S)$ für alle $e' \in D - e$. Zeige erstens, dass $e \in E_{G[R]}$. Da $e \in W - \mathcal{E}(R)^\circ$ ist $e \notin \mathcal{E}(R)^\circ$. Da $E_{G[R]} = E - \mathcal{E}(R)^\circ$ gilt folglich $e \in E_{G[R]}$. Zeige zweitens, dass $D \in \mathcal{E}_{G[R]}$. Wegen $e \in W - \mathcal{E}(R)^\circ$ gilt $e \notin \mathcal{E}(R)^\circ$. Folglich existiert kein $D' \in \mathcal{E}(R)$ mit $e \in D'$. Da $e \in D$ gilt somit $D \notin \mathcal{E}(R)$. Daraus folgt wegen $\mathcal{E}_{G[R]} = \mathcal{E} - \mathcal{E}(R)$ dass $D \in \mathcal{E}_{G[R]}$. Zeige drittens, dass für alle $e' \in D - e$ gilt $e' \in \chi_{G[R]}(S)$. Sei $e' \in D - e$. Da $e' \in \chi_G(R \cup S)$ und gemäss Beweis von (i) $\chi_{G[R]}(S) = \chi_G(R \cup S) - \mathcal{E}(R)^\circ$ muss gezeigt werden, dass $e' \notin \mathcal{E}(R)^\circ$. Nehme an, dass $e' \in \mathcal{E}(R)^\circ$. Dann existiert ein $D' \in \mathcal{E}(R)$ mit $e' \in D'$. Da $|\mathcal{E}(e')| = 1$ und $e' \in D$, ist $D' = D$ und folglich $D \in \mathcal{E}(R)$. Ein Widerspruch zur Tatsache, dass $D \in \mathcal{E}_{G[R]}$, d.h. $D \in \mathcal{E} - \mathcal{E}(R)$. Zusammenfassend ist somit gezeigt, dass $e \in E_{G[R]}$ und dass ein $D \in \mathcal{E}_{G[R]}$ mit $e \in D$ existiert, so dass $e' \in \chi_{G[R]}(S)$ für alle $e' \in D - e$. Folglich ist $e \in U$. ♦

Lemma 6.3. Sei $G = \langle V, A, E, \mathcal{E}, c \rangle$ ein disjunktiver Graph mit $|\mathcal{E}(e)| = 1$ für alle $e \in E$. Sei $R \subseteq E$ eine Selektion in G und $S \subseteq E_{G[R]}$ eine Selektion in $G[R]$. Falls $\phi_G^*(R \cup S) \cap \bar{R} = \emptyset$, dann gilt:

- (i) $\phi_{G[R]}^*(S) = \phi_G^*(R \cup S) - R$
- (ii) $\phi_G^*(R \cup S) = \phi_{G[R]}^*(S) \cup R$

Beweis:

Zur Erinnerung sei erwähnt, dass \bar{R} die zu R komplementäre Selektion in G darstellt, d.h. $\bar{R} = \mathcal{E}(R)^\circ - R$.

Beweis von (i): Zeige, dass $\phi_{G[R]}^r(S) = \phi_G^r(R \cup S) - R$ für alle $r \geq 0$. Der Beweis erfolgt mittels Induktion über r . (1) Induktionsbehauptung: Für alle $r \geq 0$ gilt die Aussage $\mathcal{P}(r)$ $\phi_{G[R]}^r(S) = \phi_G^r(R \cup S) - R$. (2) Induktionsverankerung: Zeige, dass $\mathcal{P}(0)$ gilt. Gemäss Definition ist $\phi_{G[R]}^0(S) = S$ und $\phi_G^0(R \cup S) = R \cup S$. Gemäss Lemma 5.12. ist $E_{G[R]} = E - \mathcal{E}(R)^\circ$. Da $S \subseteq E_{G[R]}$ und $R \subseteq \mathcal{E}(R)^\circ$, gilt $R \cap S = \emptyset$. Folglich ist $\phi_{G[R]}^0(S) = \phi_G^0(R \cup S) - R$. (3) Induktionsschritt: Zeige, dass für alle $r \geq 0$ gilt: $\mathcal{P}(r) \Rightarrow \mathcal{P}(r+1)$. Sei $r \geq 0$ und gelte $\mathcal{P}(r)$. Zeige, dass $\mathcal{P}(r+1)$ gilt.

Gemäss Definition ist $\phi_{G[R]}^{r+1}(S) = \phi_{G[R]}(\phi_{G[R]}^r(S))$. Da $|\mathcal{E}(e)| = 1$ für alle $e \in E$ gilt aufgrund von Lemma 6.2., dass $\phi_{G[R]}(\phi_{G[R]}^r(S)) = \phi_G(R \cup \phi_{G[R]}^r(S)) - \mathcal{E}(R)^\circ$. Gemäss Induktionsvoraussetzung $\mathcal{P}(r)$ ist $\phi_{G[R]}^r(S) = \phi_G^r(R \cup S) - R$. Da $R \subseteq \phi_G^r(R \cup S)$ gilt folglich $\phi_{G[R]}^{r+1}(S) = \phi_G(\phi_G^r(R \cup S)) - \mathcal{E}(R)^\circ$. Da $\phi_G(\phi_G^r(R \cup S)) = \phi_G^{r+1}(R \cup S)$ ist $\phi_{G[R]}^{r+1}(S) = \phi_G^{r+1}(R \cup S) - \mathcal{E}^\circ(R)$. Da $\mathcal{E}^\circ(R) = R \cup \bar{R}$ gilt somit $\phi_{G[R]}^{r+1}(S) = \phi_G^{r+1}(R \cup S) - R - \bar{R}$. Gemäss Voraussetzung ist $\phi_G^*(R \cup S) \cap \bar{R} = \emptyset$ und folglich auch $\phi_G^{r+1}(R \cup S) \cap \bar{R} = \emptyset$. Somit ist $\phi_{G[R]}^{r+1}(S) = \phi_G^{r+1}(R \cup S) - R$.

Beweis von (ii): Gemäss Beweis von (i) ist $\phi_{G[R]}^*(S) = \phi_G^*(R \cup S) - R$. Da $R \subseteq \phi_G^*(R \cup S)$ gilt somit $\phi_G^*(R \cup S) = \phi_{G[R]}^*(S) \cup R$ ♦

Lemma 6.4. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $S \subseteq E$ eine positiv azyklische Selektion in G . Dann gilt:

- (i) $|S \cap D| \leq 1$, für alle $D \in \mathcal{E}$
- (ii) $S \cap \bar{R} = \emptyset$, für alle $R \subseteq S$

Beweis:

Beweis von (i): Sei $D \in \mathcal{E}$. Nehme an, dass $|S \cap D| > 1$ d.h. $D \subseteq S$. Gemäss Definition von G hat D die Form $D = \{(h_i, t_j), (h_j, t_i)\}$ wobei $\{i, j\} \subseteq I_m$ und $m \in M$ umfasst weiter die konjunktiven Bogen $(t_i, h_i), (t_j, h_j) \in A_1$. Folglich existiert im Graphen $G(S)$ der 4-Zyklus

$Z = ((t_i, h_i), (h_i, t_j), (t_j, h_j), (h_j, t_i))$. Da die Gewichte aller Bogen von Z positiv sind, ist $G(S)$ positiv zyklisch. Ein Widerspruch zur Annahme, dass S positiv azyklisch ist in G .

Beweis von (ii): Sei $R \subseteq S$. Nehme an, dass $S \cap \bar{R} \neq \emptyset$. Dann existiert ein $e \in S$ mit $e \in \bar{R}$. Da $\bar{R} = \mathcal{E}^\circ(R) - R$, gilt $e \in \mathcal{E}^\circ(R)$ und $e \notin R$. Da $e \in \mathcal{E}^\circ(R)$ existiert ein $D \in \mathcal{E}(R)$ mit $e \in D$ und $D \cap R \neq \emptyset$. Da $D \cap R \neq \emptyset$ existiert ein $f \in D$ mit $f \in R$. Da $e \in \bar{R}$ $f \in R$ und $R \cap \bar{R} = \emptyset$ gilt $e \neq f$. Da $f \in R$ und $R \subseteq S$ gilt $f \in S$. Somit ist $\{e, f\} \subseteq S$ und $\{e, f\} \subseteq \bar{R}$. Daraus folgt $|D \cap S| \geq 2$. Da aber S positiv azyklisch ist in G , gilt gemäss Beweis von (i), dass $|S \cap D| \leq 1$. Ein Widerspruch. ♦

Definition 6.5. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $e \in E$ und e' der Partner von e . Bezeichne $\mathcal{J}(e) = \{J_{tail(e)}, J_{head(e)}\}$. Sei $J \in \mathcal{J}(e)$.

Die Menge der auf der Seite von e bzw. e' liegenden Bogen von J ist definiert als $E_J^e = E_J^h$ bzw. $E_J^{e'} = E_J^t$, falls $J = J_{tail(e)}$, und $E_J^e = E_J^t$ bzw. $E_J^{e'} = E_J^h$, falls $J = J_{head(e)}$.

Eine Illustration zu dieser Definition findet sich in Abschnitt 6.4. Für einen disjunktiven Bogen e werden die beiden Jobs $J_{tail(e)}$ und $J_{head(e)}$ betrachtet. $J_{tail(e)}$ ist derjenige Job, zu welchem der Startknoten von e , d.h. $tail(e)$, gehört. Beachte hierbei, dass $tail(e)$ ein h -Knoten ist, d.h. es gilt $tail(e) = h_i$ für eine Operation $i \in I$. $J_{tail(e)}$ ist also derjenige Job, zu welchem Operation i gehört. Analog ist $J_{head(e)}$ derjenige Job, zu welchem der Endknoten von e d.h. $head(e)$ gehört. Beachte hierbei, dass $head(e)$ ein t -Knoten ist, d.h. es gilt $head(e) = t_j$ für eine Operation $j \in I$. $J_{head(e)}$ ist derjenige Job, zu welchem Operation j gehört.

Ausgehend von e und J werden die Mengen E_J^e und $E_J^{e'}$ definiert. Falls $J = J_{tail(e)}$ startet e in einem h -Knoten von J , d.h. $e \in E_J^h$. Folglich ist E_J^e gegeben durch die Menge aller Bogen, welche in einem h -Knoten von J starten, d.h. $E_J^e = E_J^h$. Umgekehrt endet der Partnerbogen e' in einem t -Knoten von J , d.h. $e' \in E_J^t$. Folglich ist $E_J^{e'}$ gegeben durch die Menge aller Bogen, welche in einem t -Knoten von J enden, d.h. $E_J^{e'} = E_J^t$. Entsprechend sind diese Mengen für den Fall $J = J_{head(e)}$ definiert. In diesem Fall endet e in einem t -Knoten und e' startet in einem h -Knoten von J . Folglich gilt umgekehrt $E_J^e = E_J^t$ und $E_J^{e'} = E_J^h$. Die Mengen E_J^e und $E_J^{e'}$ bezeichnen also in beiden Fällen diejenigen Bogen, die im selben Knotentyp (d.h. t - oder h -Knoten) starten bzw. enden wie e bzw. e' . Anschaulich werden diese Bogen als diejenigen Bogen bezeichnet, welche "auf derselben Seite von J liegen wie e bzw. e' ". Beachte noch, dass gilt $E_J = E_J^t \cup E_J^h$ und $E_J^t \cap E_J^h = \emptyset$.

Definition 6.6. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $S \subseteq E$ eine Selektion in G . Sei $e \in S$ und e' der Partner von e . Sei $J \in \mathcal{J}(e)$.

Es bezeichnet $S_J = S \cap E_J$, $S_J^e = S \cap E_J^e$ und $S_J^{e'} = S \cap E_J^{e'}$ sowie $R_J = S - S_J$, $R_J^e = S - S_J^e$ und $R_J^{e'} = S - S_J^{e'}$.

Eine Illustration zu dieser Definition findet sich in Abschnitt 6.4. Ausgehend von einer zulässigen Selektion S , einem Bogen $e \in S$ und einem der beiden zu e gehörenden Jobs $J \in \mathcal{J}(e)$ werden verschiedene Teilselektionen definiert. S_J entspricht den Bogen aus S , welche mit einem Knoten von J inzidieren. S_J^e bzw. $S_J^{e'}$ bezeichnet die Bogen aus S_J , welche auf derselben Seite wie e bzw. e' liegen. Es gilt $S_J = S_J^e \cup S_J^{e'}$ und $S_J^e \cap S_J^{e'} = \emptyset$. R_J^e bzw. $R_J^{e'}$ bezeichnet alle Bogen aus S , welche nicht mit einem Knoten von J inzidieren. R_J^e bzw. $R_J^{e'}$ bezeichnet alle Bogen aus S ausser denjenigen, welche auf derselben Seite von J liegen wie e bzw. e' . Es gilt $R_J^e = S \cup S_J^{e'}$ und $R_J^{e'} = S \cup S_J^e$.

Lemma 6.7. Sei $G = G^{\Pi}[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion $S \subseteq E$ eine vollständige Selektion in G und J ein Job von G . Es gilt:

- (i) $G[R_J]$ ist ein 1-Job-Lager-Graph.
- (ii) $G[U]$ ist ein 1-Job-Lager-Graph für alle $U \subseteq E$ mit $R_J \subseteq U$.

Beweis:

Beweis von (i): Sei $G[R_J] = \langle V, A', E', \mathcal{E}', c \rangle$ Zeige, dass $\mathcal{E}' \subseteq \mathcal{E}_J$ Gemäss Definition ist $\mathcal{E}' = \mathcal{E} - \mathcal{E}(R_J)$. Da S eine vollständige Selektion ist in G gilt $\mathcal{E} = \mathcal{E}(S)$ Somit ist $\mathcal{E}' = \mathcal{E}(S) - \mathcal{E}(R_J)$. Gemäss Lemma 5.9. gilt $\mathcal{E}' = \mathcal{E}(S - \mathcal{E}^\circ(R_J))$ Da $R_J \subseteq \mathcal{E}^\circ(R_J)$ ist $S - \mathcal{E}^\circ(R_J) \subseteq S - R_J$ und folglich $\mathcal{E}' \subseteq \mathcal{E}(S - R_J)$ d.h. $\mathcal{E}' \subseteq \mathcal{E}(S_J)$ Da $S_J \subseteq E_J$ und $\mathcal{E}(E_J) = \mathcal{E}_J$ folgt daraus $\mathcal{E}' \subseteq \mathcal{E}_J$.

Beweis von (ii): Sei $G[U] = \langle V, A', E', \mathcal{E}', c \rangle$ Zeige, dass $\mathcal{E}' \subseteq \mathcal{E}_J$ Gemäss Definition ist $\mathcal{E}' = \mathcal{E} - \mathcal{E}(U)$. Wie in Beweis von (i) gezeigt wurde, gilt $\mathcal{E} - \mathcal{E}(R_J) \subseteq \mathcal{E}_J$ Da $R_J \subseteq U$ ist $\mathcal{E}(R_J) \subseteq \mathcal{E}(U)$. Folglich ist $\mathcal{E}' \subseteq \mathcal{E} - \mathcal{E}(R_J)$ und somit $\mathcal{E}' \subseteq \mathcal{E}_J$ ♦

6.3. Nachbarschaftsbegriff

Im folgenden wird ein Nachbarschaftsbegriff für die Optimierung in disjunktiven Lager-Graphen definiert. Der Nachbarschaftsbegriff basiert auf der Idee des Austausches von kritischen Bogen. Zielsetzung ist, ausgehend von einer zulässigen Selektion S , welche einen kritischen Bogen e enthält, eine benachbarte Selektion S' zu konstruieren, welche den Bogen e nicht enthält. Wie schon früher gezeigt wurde, müssen beim Austausch von kritischen Bogen in verallgemeinerten disjunktiven Graphen im allgemeinen noch weitere Bogen ausgetauscht werden, damit die resul-

tierende Nachbarlösung positiv azyklisch ist. Die Frage dabei ist, welche Bogen zusätzlich ausgetauscht werden sollen. Dieses Problem wird gelöst mit Hilfe des Konzepts der 1-Job-Lager-Graphen.

Definition 6.8. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $S \subseteq E$ eine zulässige Selektion in G . Sei $e \in S$ und e' der Partner von e . Sei $J \in \mathcal{J}(e)$. Bezeichne $U = \varphi_G^*(R_J^e \cup e')$. Durch e und J wird die folgende Nachbarselektion von S definiert:

$$S_J(e) = U \cup (S - \mathcal{E}(U)^\circ)$$

Die obige Definition zeigt, wie zu einer zulässigen Selektion S und einem Bogen $e \in S$ zwei Nachbarselektionen konstruiert werden. Zu jedem der beiden zu e gehörenden Jobs $J \in \mathcal{J}(e)$ lässt sich eine zugehörige Nachbarselektion $S_J(e)$ bilden. Das Vorgehen bei der Konstruktion lässt sich wie folgt beschreiben: Gegeben ist eine zulässige Selektion S und ein Bogen $e \in S$. Zunächst werden alle Bogen, welche auf derselben Seite von J wie e liegen, aus S entfernt, d.h. es wird die Menge $R_J^e = S - S_J^e$ gebildet. Insbesondere wird damit auch der Bogen e aus S entfernt. Anschliessend wird der Partner e' von e zu den verbleibenden Bogen hinzugefügt, d.h. es wird die Menge $R_J^e \cup e'$ gebildet. Danach wird das konjunktive Erzeugnis von $R_J^e \cup e'$ in G berechnet, d.h. es wird die Menge $U = \varphi_G^*(R_J^e \cup e')$ bestimmt. U bildet eine Teilselektion in G welche alle disjunktiven Mengen aus $\mathcal{E}(U)$ schneidet. Beachte weiter, dass der disjunktive Subgraph $G[\varphi_G^*(R_J^e \cup e')]$ einen konjunktiv abgeschlossenen 1-Job-Lager-Graph darstellt (vgl. Beweis des nächsten Satzes).

Die Menge $S - \mathcal{E}(U)^\circ$ enthält diejenigen Bogen aus S welche nicht in einer von U geschnittenen disjunktiven Menge sind. Die Bogen aus $S - \mathcal{E}(U)^\circ$ schneiden alle disjunktiven Mengen, welche nicht von U geschnitten werden (vgl. Beweis des nächsten Satzes). Die Nachbarselektion $S_J(e)$ wird gebildet durch die Vereinigung dieser beiden Mengen, d.h. $S_J(e) = U \cup (S - \mathcal{E}(U)^\circ)$. $S_J(e)$ ist eine vollständige Selektion, da jede disjunktive Menge $D \in \mathcal{E}$ entweder geschnitten wird von U oder von $S - \mathcal{E}(U)^\circ$. Eine Illustration zur Konstruktion der beiden Nachbarlösungen $S_J(e)$, $J \in \mathcal{J}(e)$, findet sich in Abschnitt 6.4.

Im nachfolgenden Satz wird gezeigt, dass die nach dem oben beschriebenen Prinzip konstruierten Nachbarselektionen $S_J(e)$, $J \in \mathcal{J}(e)$ genau dann zulässig sind in G wenn das konjunktive Erzeugnis $\varphi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G . Weiter gilt in diesem Fall, dass $S_J(e)$ den Bogen e nicht enthält. Somit ist die Zielsetzung erfüllt, einen Nachbarschaftsbegriff zu definieren, mit welchem zu einer Selektion S und einem (kritischen) Bogen $e \in S$ eine zulässige Nachbarselektion S' konstruiert werden kann, welche e nicht enthält.

Satz 6.9. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $S \subseteq E$ eine zulässige Selektion in G . Sei $e \in S$ und e' der Partner von e . Sei $J \in \mathcal{J}(e)$. Bezeichne $U = \phi_G^*(R_J^e \cup e')$. Die folgenden beiden Aussagen sind äquivalent:

- (i) U ist positiv azyklisch in G .
- (ii) $S_J(e)$ ist eine zulässige Selektion in G mit $e \notin S_J(e)$.

Beweis:

Beweis von (i) \Rightarrow (ii): Sei U positiv azyklisch in G .

(1) Bezeichne $K = G[U]$. Zeige, dass K ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph ist. Wegen $R_J \subseteq R_J^e$ und $R_J^e \subseteq U$ gilt $R_J \subseteq U$. Daraus folgt gemäss Lemma 6.7., dass K ein 1-Job-Lager-Graph ist. Da U positiv azyklisch ist in G , ist K positiv azyklisch. Aufgrund von Lemma 5.31. ist K konjunktiv abgeschlossen.

(2) Bezeichne $S' = S - \mathcal{E}(U)^\circ$. Zeige, dass S' eine zulässige Selektion ist in K . Gemäss Definition ist $\mathcal{E}_K = \mathcal{E} - \mathcal{E}(U)$. Da S eine vollständige Selektion ist, gilt $\mathcal{E} = \mathcal{E}(S)$ und folglich $\mathcal{E}_K = \mathcal{E}(S) - \mathcal{E}(U)$. Gemäss Lemma 5.9. ist somit $\mathcal{E}_K = \mathcal{E}(S - \mathcal{E}(U)^\circ) = \mathcal{E}(S')$. Zeige erstens, dass S' eine Selektion ist in K , d.h. $S' \subseteq E_K$. Da $E_K = \mathcal{E}_K^\circ = \mathcal{E}(S')^\circ$ und $S' \subseteq \mathcal{E}(S')$ gilt $S' \subseteq E_K$. Zeige zweitens, dass S' vollständig ist in K . Dies trifft zu, da $\mathcal{E}_K = \mathcal{E}(S')$. Zeige drittens, dass S' positiv azyklisch ist in K . Wegen $R_J^e \subseteq U$ und $U \subseteq \mathcal{E}(U)^\circ$ gilt $R_J^e \subseteq \mathcal{E}(U)^\circ$. Folglich ist $S' = S - R_J^e - \mathcal{E}(U)^\circ = S_J^e - \mathcal{E}(U)^\circ$. Somit ist $S' \subseteq S_J^e$ und wegen $S_J^e \subseteq E_J^e$ gilt $S' \subseteq E_J^e$. Daraus folgt gemäss Satz 5.66., dass S' positiv azyklisch ist in K .

(3) Da S' eine zulässige Selektion in $K = G[U]$ ist, gilt gemäss Lemma 5.11., dass $S_J(e) = U \cup S'$ eine zulässige Selektion in G ist.

Beweis von (ii) \Rightarrow (i) Sei $S_J(e)$ eine zulässige Selektion in G . Gemäss Definition ist $U \subseteq S_J(e)$. Da $S_J(e)$ positiv azyklisch ist in G , muss auch U positiv azyklisch sein in G . ♦

Aus der Definition der Nachbarselektionen $S_J(e)$, $J \in \mathcal{J}(e)$ ist nicht explizit ersichtlich, welche Bogen aus S tatsächlich ausgetauscht werden. Der folgende Satz beantwortet diese Frage für den Fall, wo $S_J(e)$ zulässig ist in G . Für den Fall, wo $S_J(e)$ unzulässig ist, gilt gemäss vorangehendem Satz, dass $U = \phi_G^*(R_J^e \cup e')$ positiv zyklisch ist in G . Daraus folgt gemäss Lemma 5.33., dass $U = E$ und folglich $S_J(e) = E$ ist.

Für den Fall, wo $S_J(e)$ zulässig ist in G , zeigt der untenstehende Satz folgendes: Die Nachbarselektion $S_J(e)$ wird aus S konstruiert, indem die Bogen aus \bar{W} ausgetauscht werden durch ihre Partner W . Hierbei ist die Bogenmenge $W = \phi_H^*(e')$ das konjunktive Erzeugnis des Partnerbogens e' im disjunktiven Subgraphen $H = G[\phi_G^*(R_J^e)]$. Beachte, dass $\phi_G^*(R_J^e) \subseteq H$ ist, und dass H

ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph ist (vgl. Beweis des Satzes).

Weiter wird gezeigt, dass $W \subseteq E_J^{e'}$, d.h. alle neu hinzugefügten Bogen liegen auf derselben Seite von J wie e' . Anschaulich bedeutet dies, dass bei einem Move die Operationen von Job J alle "in dieselbe Richtung" verschoben werden bezüglich der Ausführungsreihenfolge auf den entsprechenden Prozessoren. Falls e' ein t -Bogen von J ist, d.h. $e' \in E_J^t$ dann werden die Operationen "nach rechts" verschoben, d.h. in der Reihenfolge später ausgeführt, und falls $e' \in E_J^h$ werden sie "nach links" verschoben, d.h. in der Reihenfolge früher ausgeführt. Weiter wird aus diesem Satz ersichtlich, dass die Ausführungsreihenfolgen unter den Operationen, welche nicht zum Job J gehören, nicht verändert werden.

Satz 6.10. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $S \subseteq E$ eine zulässige Selektion in G . Sei $e \in S$ und e' der Partner von e . Sei $J \in \mathcal{J}(e)$. Bezeichne $H = G[\varphi_G^*(R_J^e)]$. Sei $W = \varphi_H^*(e')$ und $\bar{W} = \mathcal{E}(W)^\circ - W$. Falls $S_J(e)$ zulässig ist in G , dann gilt:

- (i) $S_J(e) = S - \bar{W} \cup W$
- (ii) $W \subseteq E_J^{e'}$ und $\bar{W} \subseteq E_J^e$
- (iii) $W \cap S = \emptyset$ und $\bar{W} \subseteq S$

Beweis:

Sei $S_J(e)$ zulässig in G . Dann gilt gemäss Satz 6.9., dass $U = \varphi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G .

Beweis von (i): Zeige zunächst, dass $W = \varphi_H^*(e')$ definiert ist. Dies ist genau dann der Fall, wenn e' eine Selektion in H ist, d.h. $e' \in E_H$. Nehme an, dass $e' \notin E_H$. Gemäss Lemma 5.12. ist $E_H = E - \mathcal{E}(\varphi_G^*(R_J^e))^\circ$. Da $e' \notin E_H$ muss $e' \in \mathcal{E}(\varphi_G^*(R_J^e))^\circ$ sein. Also existiert ein $D \in \mathcal{E}(\varphi_G^*(R_J^e))$ mit $e' \in D$. Da $|\mathcal{E}(e')| = 1$ ist $D = \{e, e'\}$. Da $D \in \mathcal{E}(\varphi_G^*(R_J^e))$ gilt $D \cap \varphi_G^*(R_J^e) \neq \emptyset$. Fall 1: Sei $e \in \varphi_G^*(R_J^e)$. Aufgrund der Monotonie gilt $\varphi_G^*(R_J^e) \subseteq \varphi_G^*(R_J^e \cup e')$ und somit $e \in U$. Da auch $e' \in U$ gilt $|D \cap U| = 2$. Da aber U positiv azyklisch ist in G , gilt gemäss Lemma 6.4., dass $|D \cap U| \leq 1$. Ein Widerspruch. Fall 2: Sei $e' \in \varphi_G^*(R_J^e)$. Da $R_J^e \subseteq S$ und S zulässig ist in G , gilt gemäss Lemma 5.39., dass $\varphi_G^*(R_J^e) \subseteq S$. Folglich ist $e' \in S$. Da auch $e \in S$, gilt $|D \cap S| = 2$. Da aber S positiv azyklisch ist in G gilt gemäss Lemma 6.4., dass $|D \cap S| \leq 1$. Ein Widerspruch.

(4) Sei $S' = S - \bar{W} \cup W$. Zeige, dass $S' = \varphi_G^*(R_J^e) \cup W \cup (S - \mathcal{E}(\varphi_G^*(R_J^e \cup e'))^\circ)$ zulässig ist in G und $R_J^e \subseteq S'$ gilt gemäss Lemma 5.39., dass $\varphi_G^*(R_J^e) \subseteq S$. Somit ist $S = \varphi_G^*(R_J^e) \cup (S - \varphi_G^*(R_J^e))$ und folglich $S' = (\varphi_G^*(R_J^e) - \bar{W}) \cup (S - \varphi_G^*(R_J^e) - \bar{W}) \cup W$. Da $\bar{W} \subseteq E_H$ und $E_H = E - \mathcal{E}(\varphi_G^*(R_J^e))^\circ$ gilt wegen $\varphi_G^*(R_J^e) \subseteq \mathcal{E}(\varphi_G^*(R_J^e))^\circ$ dass $\varphi_G^*(R_J^e) \cap \bar{W} = \emptyset$. Somit gilt $S' = \varphi_G^*(R_J^e) \cup W \cup (S - \varphi_G^*(R_J^e) - \bar{W})$ und weiter

$S' = \varphi_G^*(R_J^e) \cup W \cup (S - \varphi_G^*(R_J^e) - \bar{W} - W)$. Da S zulässig ist in G und $\varphi_G^*(R_J^e) \subseteq S$ gilt gemäss Lemma 6.4., dass $S \cap \varphi_G^*(R_J^e) = \emptyset$. Somit gilt $S' = \varphi_G^*(R_J^e) \cup W \cup (S - \varphi_G^*(R_J^e) - \varphi_G^*(R_J^e) - \bar{W} - W)$, d.h. $S' = \varphi_G^*(R_J^e) \cup W \cup (S - (\mathcal{E}(\varphi_G^*(R_J^e)) \cup \mathcal{E}(W)^\circ))$. Gemäss Lemma 5.8. ist $\mathcal{E}(\varphi_G^*(R_J^e)) \cup \mathcal{E}(W)^\circ = \mathcal{E}(\varphi_G^*(R_J^e) \cup W)^\circ$, woraus die Behauptung folgt.

(5) Zeige, dass $S' = S_J(e)$. Da $R_J^e \subseteq \varphi_G^*(R_J^e \cup e')$ gilt gemäss Lemma 5.32., dass $\varphi_G^*(R_J^e \cup e') = \varphi_G^*(\varphi_G^*(R_J^e) \cup e')$. Da $U = \varphi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G und $\varphi_G^*(R_J^e) \subseteq \varphi_G^*(\varphi_G^*(R_J^e) \cup e')$, gilt gemäss Lemma 6.4., dass $\varphi_G^*(\varphi_G^*(R_J^e) \cup e') \cap \varphi_G^*(R_J^e) = \emptyset$. Daraus folgt gemäss Lemma 6.3., dass $\varphi_G^*(\varphi_G^*(R_J^e) \cup e') = \varphi_{G[\varphi_G^*(R_J^e)]}^*(e') \cup \varphi_G^*(R_J^e)$. D.h. $U = W \cup \varphi_G^*(R_J^e)$. Daraus folgt aufgrund des vorangehenden Absatzes, dass $S' = U \cup (S - \mathcal{E}(U)^\circ)$, d.h. $S' = S_J(e)$.

Beweis von (ii): (1) Zeige, dass $H = G[\varphi_G^*(R_J^e)]$ ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph ist. Wegen $R_J \subseteq R_J^e$ und $R_J^e \subseteq \varphi_G^*(R_J^e)$ gilt $R_J \subseteq \varphi_G^*(R_J^e)$. Daraus folgt gemäss Lemma 6.7., dass H ein 1-Job-Lager-Graph ist. Da S eine zulässige Selektion ist in G und $R_J^e \subseteq S$, gilt $\varphi_G^*(R_J^e) \subseteq S$. Da $\varphi_G^*(R_J^e)$ positiv azyklisch ist in G , muss auch $\varphi_G^*(R_J^e)$ positiv azyklisch sein in G . Somit ist H positiv azyklisch. Aufgrund von Lemma 5.31. ist H konjunktiv abgeschlossen.

(2) Zeige, dass $W \subseteq E_J^{e'}$. Gemäss Definition ist $W = \varphi_H^*(e')$. Da $e' \in E_J^{e'}$ und H ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph ist, gilt gemäss Korollar 5.68., dass $\varphi_H^*(e') \subseteq E_J^{e'}$, d.h. $W \subseteq E_J^{e'}$. Da $W \subseteq E_J^{e'}$ gilt $\bar{W} \subseteq E_J^e$.

Beweis von (iii): Zeige, dass $W \cap S = \emptyset$. Da S eine zulässige Selektion ist in G und $R_J^e \subseteq S$ gilt gemäss Lemma 5.39., dass $\varphi_G^*(R_J^e) \subseteq S$. Wegen $S = R_J^e \cup S_J^e$ und $R_J^e \subseteq \varphi_G^*(R_J^e)$ ist $S = \varphi_G^*(R_J^e) \cup S_J^e$. Zeige erstens, dass $\varphi_G^*(R_J^e) \cap W = \emptyset$. Da $E_H = E - \mathcal{E}(\varphi_G^*(R_J^e))$ und $W \subseteq E_H$ gilt $W \cap \mathcal{E}(\varphi_G^*(R_J^e))^\circ = \emptyset$. Da $\varphi_G^*(R_J^e) \subseteq \mathcal{E}(\varphi_G^*(R_J^e))$ gilt somit $\varphi_G^*(R_J^e) \cap W = \emptyset$. Zeige zweitens, dass $S_J^e \cap W = \emptyset$. Wie eben gezeigt wurde, ist $W \subseteq E_J^{e'}$. Da $S_J^e \subseteq E_J^e$ und $E_J^e \cap E_J^{e'} = \emptyset$ gilt $S_J^e \cap W = \emptyset$.

(3) Zeige, dass $\bar{W} \subseteq S$. Nehme an, es existiert ein $f \in \bar{W}$ mit $f \notin S$. Da $f \in \bar{W}$ und $\bar{W} = \mathcal{E}(W)^\circ$, W gilt $f \in \mathcal{E}(W)^\circ$. Folglich existiert ein $D \in \mathcal{E}(W)$ mit $f \in D$. Sei $D = \{f, f'\}$. Da $D \in \mathcal{E}(W)$ gilt $D \cap W \neq \emptyset$. Da $f \in \bar{W}$ und $W \cap \bar{W} = \emptyset$ ist $f' \in W$. Somit muss $f' \in S$ sein. Da $S \cap W = \emptyset$ gilt $f' \notin S$. Da S vollständig ist in G , gilt $S \cap D \neq \emptyset$. Folglich muss $f \in S$ sein. Ein Widerspruch. ♦

In Satz 6.9. wurde gezeigt, dass eine Nachbarselektion $S_J(e)$, $J \in \mathcal{J}(e)$ genau dann zulässig ist, wenn $U = \varphi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G . Aus dieser Bedingung ist nicht explizit ersichtlich, unter welchen Umständen eine Nachbarselektion tatsächlich zulässig ist. Der folgende Satz klärt diese Frage, indem gezeigt wird, dass diese Bedingung äquivalent ist zur Bedingung, dass $e \notin \varphi_G^*(R_J)$.

R_J ist die Menge aller Bogen in S , welche nicht mit einem Knoten von J inzidieren, d.h. $R_J = S - S_J$. $\varphi_G^*(R_J)$ ist das konjunktive Erzeugnis dieser Bogen, wobei gilt, dass $\varphi_G^*(R_J) \subseteq S$ da $R_J \subseteq S$ und S zulässig ist in G . Anschaulich ausgedrückt beschreibt $\varphi_G^*(R_J)$ die "äusseren Rahmenbedingungen", die gelten, wenn der Job J aus dem durch S beschriebenen Ablaufplan herausgelöst und wieder neu eingefügt wird. Die "äusseren Rahmenbedingungen" ergeben sich einerseits aus den durch die Präselektion Q vorgegebenen konjunktiven Präzedenzen und andererseits aus den Präzedenzen $R_J \subseteq S$, welche bei der Konstruktion von $S_J(e)$ nicht verändert werden.

Der folgende Satz besagt, dass die Nachbarlösung $S_J(e)$ genau dann zulässig ist, wenn der Bogen e nicht durch die "äusseren Rahmenbedingungen" $\varphi_G^*(R_J)$ erzwungen wird. Falls e durch die "äusseren Rahmenbedingungen" erzwungen wird, d.h. $e \in \varphi_G^*(R_J)$ dann kann offensichtlich keine zulässige Nachbarselektion konstruiert werden, welche e nicht enthält, ohne dass die "äusseren Rahmenbedingungen" verändert werden.

Satz 6.11. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ ein Lager-Graph mit Präselektion Q . Sei $S \subseteq E$ eine zulässige Selektion in G . Sei $e \in S$ und e' der Partner von e . Die folgenden beiden Aussagen sind äquivalent:

- (i) $e \notin \varphi_G^*(R_J)$
- (ii) $S_J(e)$ ist eine zulässige Selektion in G

Beweis:

Beweis von (i) \Rightarrow (ii) Sei $e \notin \varphi_G^*(R_J)$

(1) Bezeichne $K = G[\varphi_G^*(R_J)]$. Zeige, dass K ein konjunktiv abgeschlossener, positiv azyklischer 1-Job-Lager-Graph ist. Da $R_J \subseteq \varphi_G^*(R_J)$ ist K gemäss Lemma 6.7. ein 1-Job-Lager-Graph. Da S zulässig ist in G und $R_J \subseteq S$ gilt aufgrund von Lemma 5.39., dass $\varphi_G^*(R_J) \subseteq S$. Da S positiv azyklisch ist in G , ist auch $\varphi_G^*(R_J)$ positiv azyklisch in G . Somit ist K positiv azyklisch. Gemäss Lemma 5.31. ist K konjunktiv abgeschlossen.

(2) Sei $W = E_J^{e'} \cap E_K$. Zeige, dass W eine zulässige Selektion ist in K . W ist eine Selektion in K da $W \subseteq E_K$. Da $W \subseteq E_J^{e'}$ ist W gemäss Satz 5.66. positiv azyklisch in K . W ist vollständig in K da $|D \cap E_J^{e'}| = 1$ für alle $D \in \mathcal{E}_K$.

(3) Sei $S' = \varphi_G^*(R_J) \cup W$. Zeige, dass S' eine zulässige Selektion ist in G . Da W zulässig ist in $K = G[\varphi_G^*(R_J)]$, folgt aufgrund von Lemma 5.11. die Behauptung.

(4) Zeige, dass $S_J^{e'} \subseteq S'$. Nehme an, dass ein $f \in S_J^{e'}$ existiert mit $f \notin S'$. Sei $D = \{f, f'\}$ die disjunktive Menge, welche f enthält. Da S' vollständig ist in G , gilt $D \cap S' \neq \emptyset$. Da $f \notin S'$ muss gelten $f' \in S'$. Fall 1: Sei $f' \in \varphi_G^*(R_J)$. Da $R_J \subseteq S$ und S zulässig ist in G , gilt gemäss Lemma

5.39., dass $\phi_G^*(R_J) \subseteq S$ und folglich $f' \in S$. Da $f \in S_J^{e'}$ ist $f \in S$ und somit $|\{f, f'\} \cap S| = 2$. Da aber S positiv azyklisch ist in G , gilt gemäss Lemma 6.4., dass $|\{f, f'\} \cap S| \leq 1$. Ein Widerspruch. Fall 2: Sei $f' \in W$. Da $W \subseteq E_J^{e'}$ ist $f' \in E_J^{e'}$. Folglich muss $f \in E_J^e$ sein. Da $E_J^e \cap E_J^{e'} = \emptyset$ gilt $f \notin E_J^{e'}$. Da aber $f \in S_J^{e'}$ ist, gilt $f \in E_J^{e'}$. Ein Widerspruch. Somit ist gezeigt, dass $S_J^{e'} \subseteq S'$.

(5) Zeige, dass $e' \in S'$. Gemäss Voraussetzung ist $e \notin \phi_G^*(R_J)$. Da $W \subseteq E_J^{e'}$ gilt $e \notin W$ und folglich $e \notin S'$. Da S' vollständig ist in G muss somit $e' \in S'$ sein.

(6) Zeige, dass $S_J(e)$ zulässig ist in G . Gemäss Satz 6.9. ist dies genau dann der Fall, wenn $U = \phi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G . Zeige, dass $\phi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G . Wie gezeigt wurde, ist $S_J^{e'} \cup e' \subseteq S'$. Da $R_J \subseteq \phi_G^*(R_J)$ gilt somit $R_J \cup S_J^{e'} \cup e' \subseteq S'$. h. $R_J^e \cup e' \subseteq S'$. Da S' zulässig ist in G , gilt gemäss Lemma 5.39., dass $\phi_G^*(R_J^e \cup e') \subseteq S'$. Da S' positiv azyklisch ist in G , ist auch $\phi_G^*(R_J^e \cup e')$ positiv azyklisch in G .

Beweis von (ii) \Rightarrow (i) Sei $S_J(e)$ zulässig in G . Gemäss Satz 6.9. ist dies genau dann der Fall, wenn $U = \phi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G . Da $R_J \subseteq R_J^e$ gilt aufgrund der Monotonie, dass $\phi_G^*(R_J) \subseteq \phi_G^*(R_J^e \cup e')$. Da $\phi_G^*(R_J^e \cup e')$ positiv azyklisch ist in G ist auch $\phi_G^*(R_J)$ positiv azyklisch in G . ♦

6.4. Beispiel zum Nachbarschaftsbegriff

In diesem Abschnitt werden die in den vorangehenden Abschnitten eingeführten Definitionen und Konzepte anhand eines Beispiels illustriert. Es wird schrittweise gezeigt, wie in einem Lager-Graphen zu einer gegebenen Selektion S und einem Bogen $e \in S$ die beiden zugehörigen Nachbarselektionen $S_J(e)$, $J \in \mathcal{J}(e)$ konstruiert werden.

Zu den folgenden Abbildungen ist zu bemerken, dass disjunktive Bogen, welche in einem bestimmten Graphen erzwungen werden, mit Sternen (*) gekennzeichnet werden, wobei ein k -stufig erzwungener Bogen k Sterne erhält.

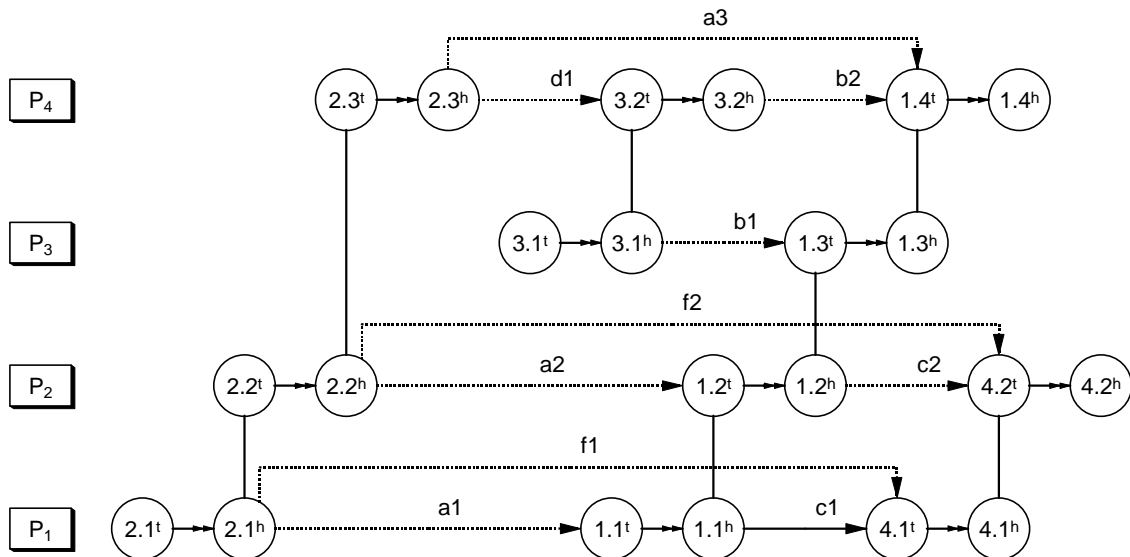
Das verwendete Beispiel ist ein Lager-Problem $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^S)$ mit 4 Prozessoren $M = \{P_1, \dots, P_4\}$ und 4 Jobs $\mathcal{J} = \{J_1, \dots, J_4\}$. Die Operationen der verschiedenen Jobs werden der Reihe nach auf folgenden Prozessoren ausgeführt:

Job:	Operation 1:	Operation 2:	Operation 3:	Operation 4:
J_1	P_1	P_2	P_3	P_4
J_2	P_1	P_2	P_4	
J_3	P_3	P_4		
J_4	P_1	P_2		

Abbildung 6.1. Prozessorreihenfolge der verschiedenen Jobs

Die Zeitparameter d, d^t, d^h, d^S sind für die folgenden Ausführungen nicht relevant und werden deshalb weggelassen.

Sei G^Π der Lager-Graph zu obigen Lager-Modell Π . Nehme an, dass als zusätzliche Nebenbedingung in Π gefordert wird, dass die Operation J_{11} vor J_{41} ausgeführt werden muss. Dies führt zu einer Präselektion Q in G^Π , welche einen Bogen vom h -Knoten von J_{11} zum t -Knoten von J_{41} enthält, d.h. $Q = \{(1.1^h, 4.1^t)\}$. Sei $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, d \rangle$ der entsprechende Lager-Graph mit Präselektion Q . Die untenstehende Abbildung zeigt den Graphen G . Der in der Präselektion Q enthaltene Bogen ist mit $c_1 = (1.1^h, 4.1^t)$ bezeichnet. Zur besseren Uebersicht ist für jede disjunktive Menge nur einer der beiden disjunktiven Bogen eingezeichnet.

Abbildung 6.2. Lager-Graph $G = G^\Pi[Q]$ mit Präselektion $Q = \{c_1\}$

Für die folgenden Illustrationen wird als Beispiel die zulässige Selektion $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$ gewählt. Die untenstehende Abbildung zeigt diese Selektion in G .

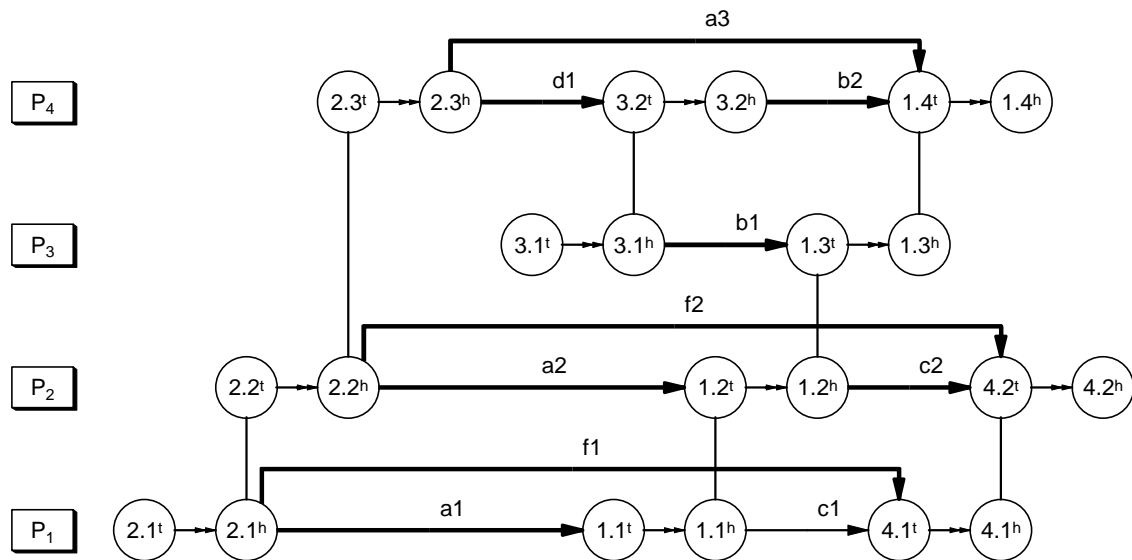


Abbildung 6.3. Zulässige Selektion $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$ in G

6.4.1. Illustration zu Definition 6.5.

Sei $e = a_1$ und $e' = a_1'$. Dann ist $\mathcal{J}(e) = \{J_{tail(e)}, J_{head(e)}\} = \{J_2, J_1\}$. Sei $J = J_{head(e)}$. Die Bogenmengen E_J^e und $E_J^{e'}$ sind gegeben durch $E_J^e = E_J^t = \{a_1, a_2, a_3, b_1, b_2, c_2\}$ und $E_J^{e'} = E_J^h = \{a_1', a_2', a_3', b_1', b_2', c_2\}$. Anschaulich ausgedrückt entspricht E_J^e den disjunktiven Bogen, welche "auf derselben Seite von J wie e liegen", und $E_J^{e'}$ denjenigen Bogen, welche "auf derselben Seite von J wie e' liegen".

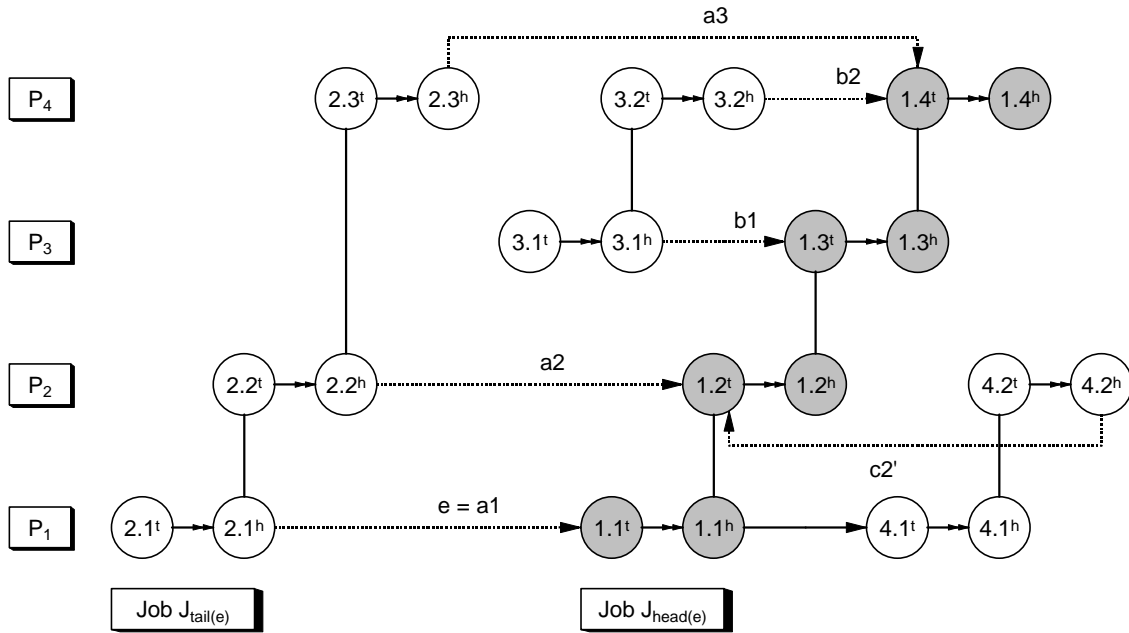


Abbildung 6.4. Disjunktive Bogenmenge $E_J^e = E_J^t = \{a_1, a_2, a_3, b_1, b_2, c_2'\}$ für $J = J_{\text{head}(e)}$

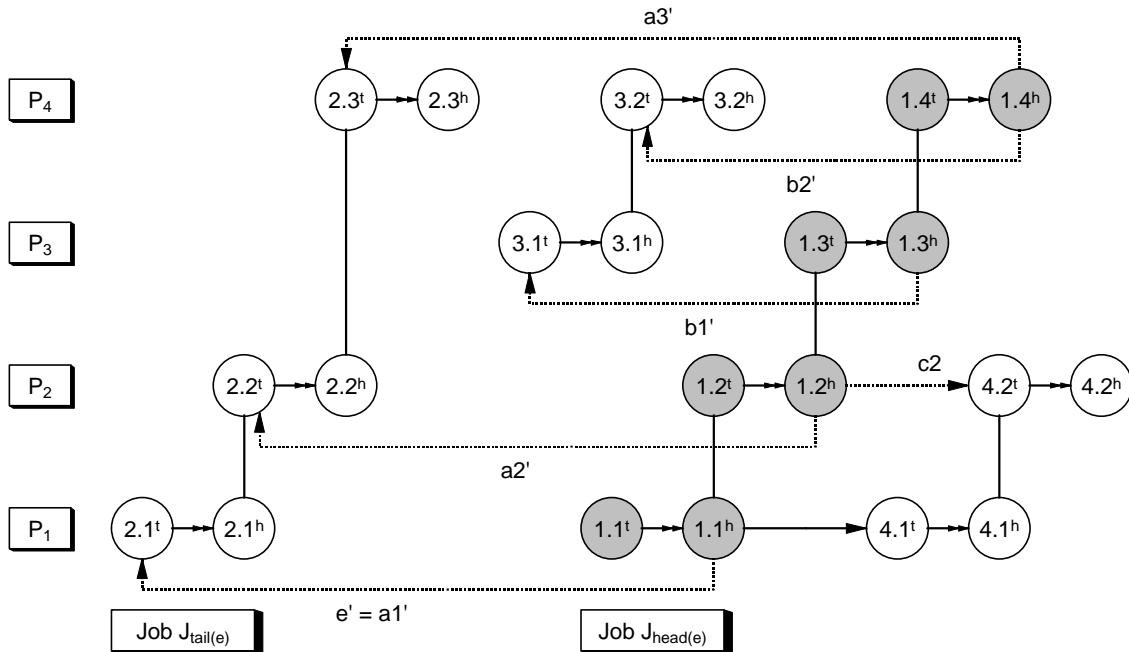


Abbildung 6.5. Disjunktive Bogenmenge $E_J^{e'} = E_J^h = \{a_1', a_2', a_3', b_1', b_2', c_2\}$ für $J = J_{\text{head}(e)}$

6.4.2. Illustration zu Definition 6.6.

Betrachte die zulässige Selektion $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$. Sei $e = a_1$ und $e' = a_1'$. Sei $J = J_{head(e)} = J_1$. Die Teilselektion S_J ist gegeben durch $S_J = \{a_1, a_2, a_3, b_1, b_2, c_2\}$. S_J entspricht der Menge aller Bogen, welche mit einem Knoten von J inzidieren. Die Teilselektionen S_J^e und $S_J^{e'}$ sind gegeben durch $S_J^e = \{a_1, a_2, a_3, b_1, b_2\}$ und $S_J^{e'} = \{c_2\}$. Anschaulich gesprochen ist S_J^e bzw. $S_J^{e'}$ die Menge der disjunktiven Bogen von S welche "auf der selben Seite von J wie e bzw. e' liegen".

Die Teilselektionen R_J , R_J^e und $R_J^{e'}$ sind gegeben durch $R_J = S - S_J = \{d_1, f_1, f_2\}$, $R_J^e = S - S_J^e = \{c_2, d_1, f_1, f_2\}$ und $R_J^{e'} = S - S_J^{e'} = \{a_1, a_2, a_3, b_1, b_2, d_1, f_1, f_2\}$.

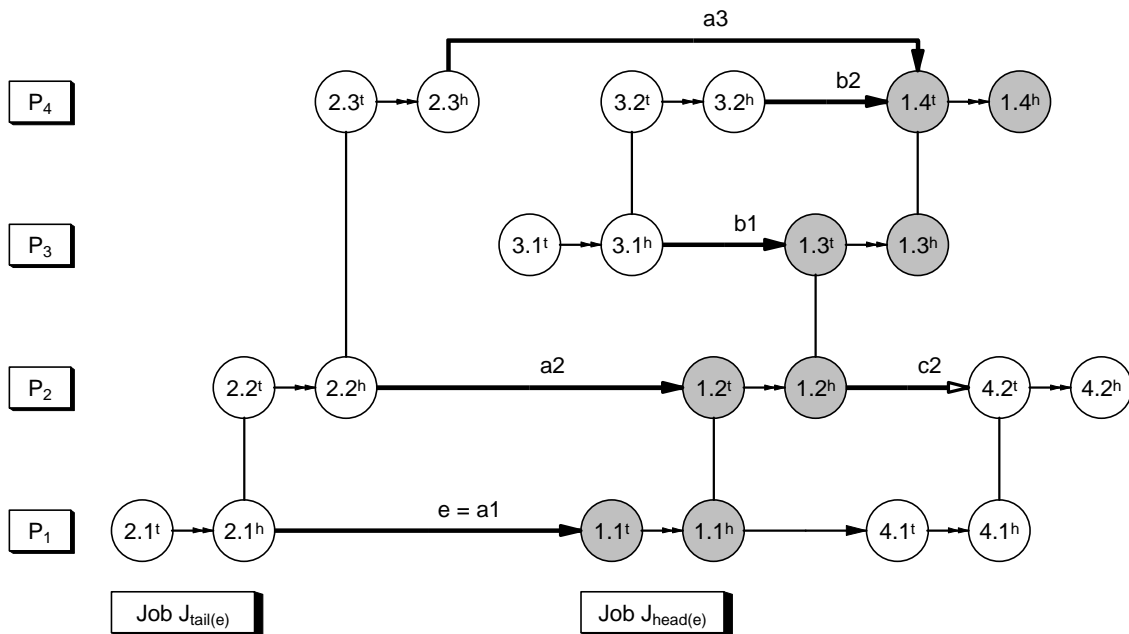


Abbildung 6.6. Die Teilselektionen $S_J^e = \{a_1, a_2, a_3, b_1, b_2\}$ und $S_J^{e'} = \{c_2\}$ für $J = J_{head(e)}$.

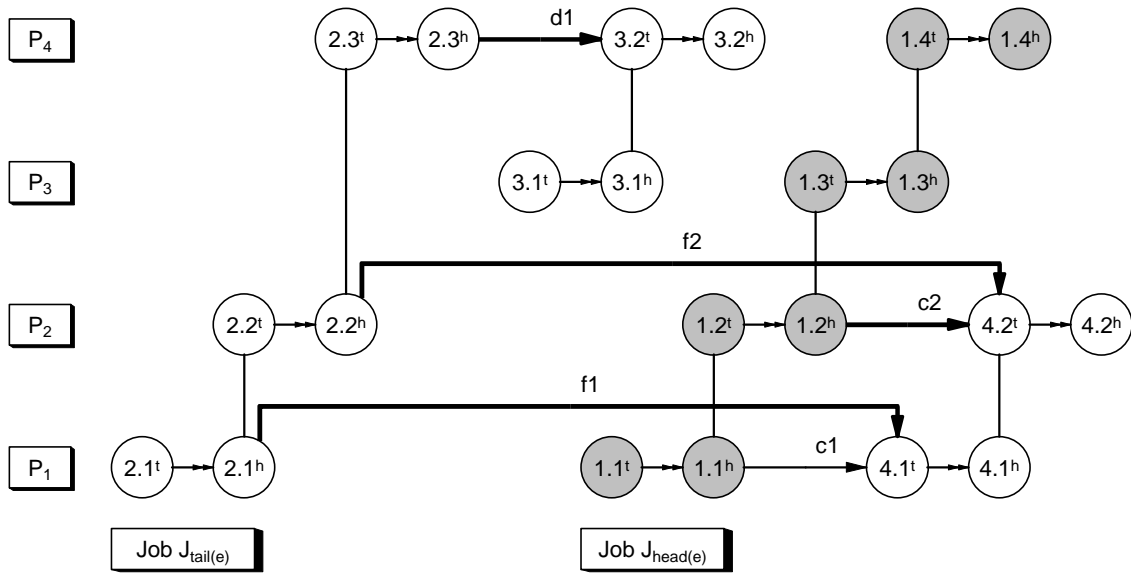


Abbildung 6.7. Die Teilselektion $R_J^e = S - S_J^e = \{c_2, d_1, f_1, f_2\}$ für $J = J_{head(e)}$.

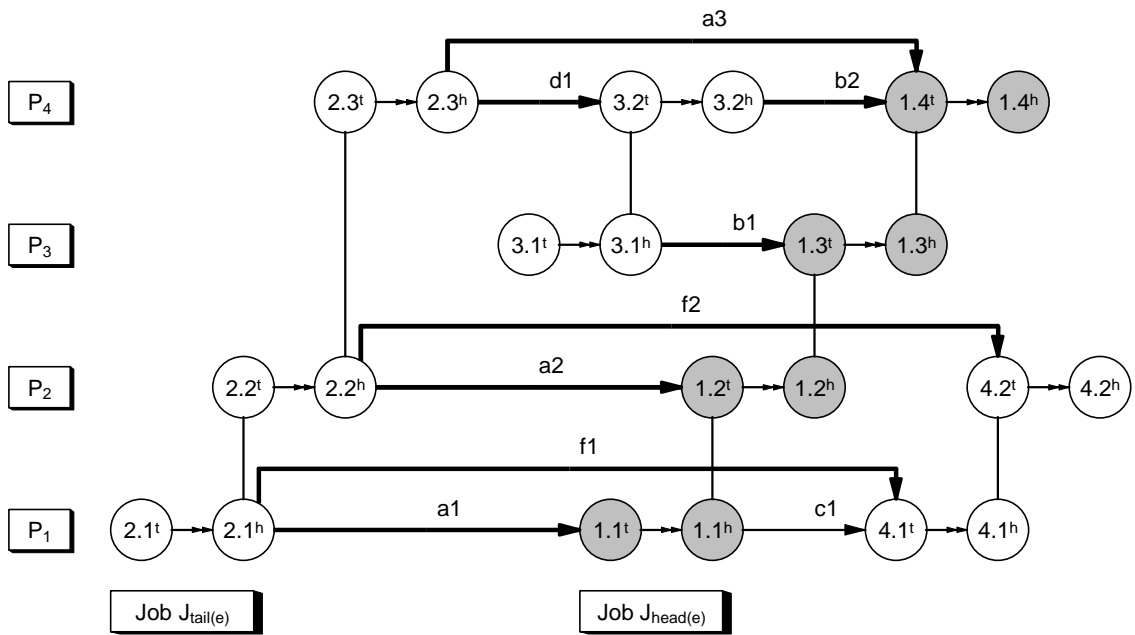


Abbildung 6.8. Die Teilselektion $R_J^{e'} = S - S_J^{e'} = \{a_1, a_2, a_3, b_1, b_2, d_1, f_1, f_2\}$ für $J = J_{head(e)}$.

6.4.3. Konstruktion einer Nachbarselektion, Fall 1

Betrachte die zulässige Selektion $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$. Sei $e = a_1$ und $e' = a_1'$. In diesem Abschnitt wird gezeigt, wie die Nachbarselektion $S_J(e)$ konstruiert wird für den Fall $J = J_{head(e)} = J_1$. Gemäss Definition ist $S_J(e) = U \cup (S - \mathcal{E}^\circ(U))$ wobei $U = \Phi_G^*(R_J^e \cup e')$.

(1) Bestimme $U = \Phi_G^*(R_J^e \cup e')$. U ist das konjunktive Erzeugnis der partiellen Selektion $R_J^e \cup e'$ in G . Es gilt $R_J^e \cup e' = \{c_2, d_1, f_1, f_2\} \cup \{a_1'\}$. Das konjunktive Erzeugnis $\Phi_G^*(R_J^e \cup e')$ wird wie folgt berechnet:

$$\Phi_G^0(R_J^e \cup e') = R_J^e \cup e' = \{a_1', c_2, d_1, f_1, f_2\}$$

$$\Phi_G^1(R_J^e \cup e') = \Phi_G^0(R_J^e \cup e') \cup \{a_2'\}$$

$$\Phi_G^2(R_J^e \cup e') = \Phi_G^1(R_J^e \cup e') \cup \{b_1'\}$$

$$\Phi_G^3(R_J^e \cup e') = \Phi_G^2(R_J^e \cup e') \cup \{b_2'\}$$

$$\Phi_G^*(R_J^e \cup e') = \Phi_G^3(R_J^e \cup e')$$

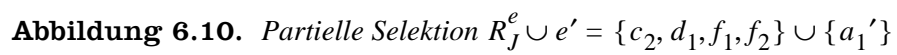
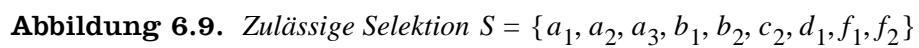
Somit gilt $U = \Phi_G^*(R_J^e \cup e') = \{a_1', a_2', b_1', b_2', c_2, d_1, f_1, f_2\}$

(2) Bestimme die restliche Selektion $S - \mathcal{E}^\circ(U)$. Die Menge $\mathcal{E}^\circ(U) = \bigcup_{D \in \mathcal{E}(U)} D$ umfasst alle Bogen, welche in einer disjunktiven Menge sind, die von U geschnitten wird. Es gilt:

$$\mathcal{E}^\circ(U) = \{a_1, a_1', a_2, a_2', b_1, b_1', b_2, b_2', c_2, c_2', d_1, d_1', f_1, f_1', f_2, f_2'\}$$

Da $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$ folgt somit $S - \mathcal{E}^\circ(U) = \{a_3\}$

(3) Die Nachbarselektion $S_J(e) = U \cup (S - \mathcal{E}^\circ(U))$ ist folglich gegeben als $S_J(e) = \{a_1', a_2', a_3, b_1', b_2', c_2, d_1, f_1, f_2\}$. Die Nachbarselektion $S_J(e)$ unterscheidet sich offensichtlich dadurch von S , dass die Bogen $\{a_1, a_2, b_1, b_2\}$ aus S entfernt wurden und durch die Bogen $\{a_1', a_2', b_1', b_2'\}$ ersetzt wurden. Anschaulich bedeutet diese Änderung, dass die Operationen 1.1, 1.2, 1.3 und 1.4 auf den zugeordneten Prozessoren P_1 , P_2 , P_3 und P_4 in der Ausführungsreihenfolge "um eine Position nach links" gerückt sind. Oder anders ausgedrückt, die Positionen der Operationen 1.1 und 2.1, 1.2 und 2.2, 1.3 und 3. sowie 1.4 und 3. wurden in der Ausführungsreihenfolge vertauscht. Die folgenden Abbildungen zeigen die verschiedenen Schritte im disjunktiven Graphen G .



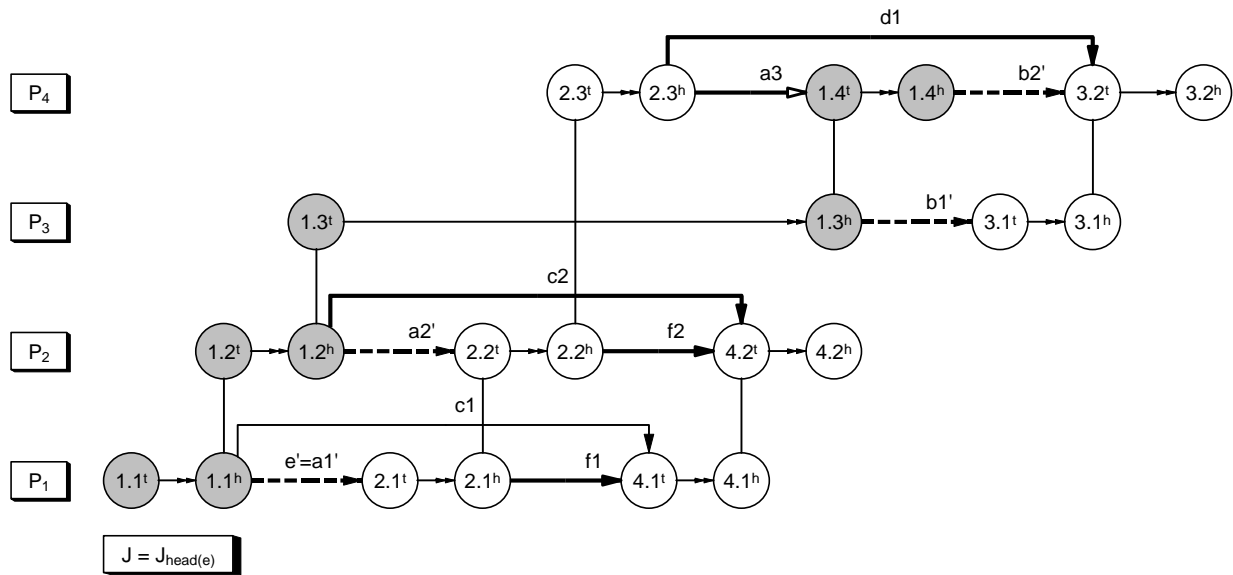


Abbildung 6.13. Nachbarschelektion $S_J(e) = \{a_1', a_2', a_3, b_1', b_2', c_2, d_1, f_1, f_2\}$ nach Umordnung

Gantt-Diagramme

Untenstehend wird der Uebergang von der Selektion S zur Nachbarschelektion $S_J(e)$ im Gantt-Diagramm gezeigt. Die Zeiten für die verschiedenen Aktionen sind zufällig gewählt. Die angeführten Ablaufpläne entsprechen den semi-aktiven Plänen bezüglich der durch S bzw. $S_J(e)$ definierten Bearbeitungsreihenfolgen, d.h. alle Teiloperationen werden zum frühestmöglichen Zeitpunkt gestartet.

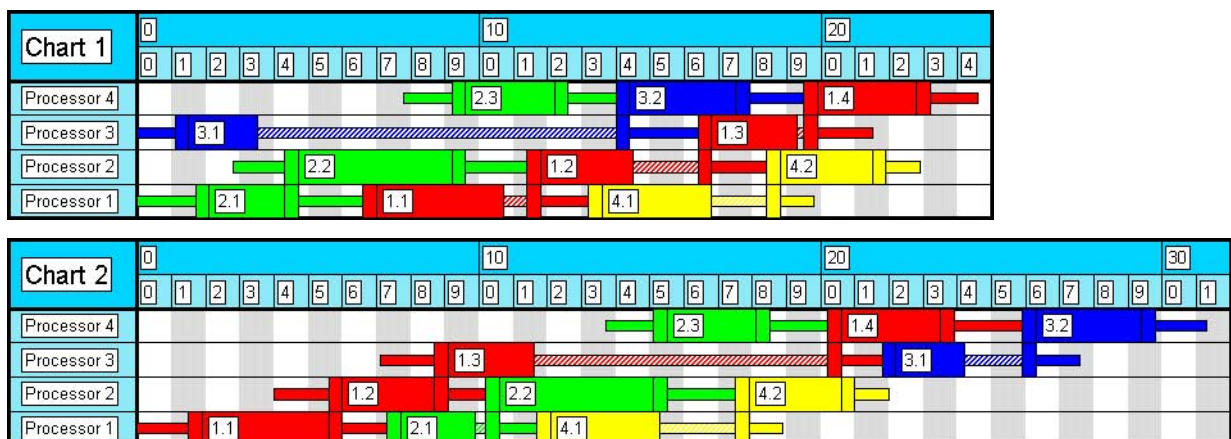


Abbildung 6.14. Gantt-Diagramme zu den Selektionen S und $S_J(e)$

6.4.4. Konstruktion einer Nachbarselektion, Fall 2

Betrachte die zulässige Selektion $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$. Sei $e = a_1$ und $e' = a_1'$. In diesem Abschnitt wird gezeigt, wie die Nachbarselektion $S_J(e)$ konstruiert wird für den Fall $J = J_{\text{tail}(e)} = J_2$. Gemäss Definition ist $S_J(e) = U \cup (S - \mathcal{E}^\circ(U))$ wobei $U = \Phi_G^*(R_J^e \cup e')$.

(1) Bestimme $U = \Phi_G^*(R_J^e \cup e')$. U ist das konjunktive Erzeugnis der partiellen Selektion $R_J^e \cup e'$ in G . Es gilt $R_J^e \cup e' = \{b_1, b_2, c_2\} \cup \{a_1'\}$. Das konjunktive Erzeugnis $\Phi_G^*(R_J^e \cup e')$ wird wie folgt berechnet:

$$\Phi_G^0(R_J^e \cup e') = R_J^e \cup e' = \{a_1', b_1, b_2, c_2\}$$

$$\Phi_G^1(R_J^e \cup e') = \Phi_G^0(R_J^e \cup e') \cup \{a_2'\}$$

$$\Phi_G^2(R_J^e \cup e') = \Phi_G^1(R_J^e \cup e') \cup \{d_1'\}$$

$$\Phi_G^*(R_J^e \cup e') = \Phi_G^2(R_J^e \cup e')$$

Somit ist $U = \Phi_G^*(R_J^e \cup e') = \{a_1', a_2', b_1, b_2, c_2, d_1'\}$

(2) Bestimme die restliche Selektion $S - \mathcal{E}^\circ(U)$. Die Menge $\mathcal{E}^\circ(U) = \bigcup_{D \in \mathcal{E}(U)} D$ umfasst alle Bogen, welche in einer disjunktiven Menge sind, die von U geschnitten wird. Es gilt:

$$\mathcal{E}^\circ(U) = \{a_1, a_1', a_2, a_2', b_1, b_1', b_2, b_2', c_2, c_2', d_1, d_1'\}$$

Da $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$ folgt somit $S - \mathcal{E}^\circ(U) = \{a_3, f_1, f_2\}$

(3) Die Nachbarselektion $S_J(e) = U \cup (S - \mathcal{E}^\circ(U))$ ist folglich gegeben als $S_J(e) = \{a_1', a_2', a_3, b_1, b_2, c_2, d_1', f_1, f_2\}$. $S_J(e)$ unterscheidet sich offensichtlich dadurch von S , dass die Bogen $\{a_1, a_2, d_1\}$ aus S entfernt wurden und durch die Bogen $\{a_1', a_2', d_1'\}$ ersetzt wurden. Anschaulich bedeutet diese Änderung, dass die Operationen 2.1, 2.2 und 2.3 auf den zugeordneten Prozessoren P_1 , P_2 und P_4 in der Ausführungsreihenfolge "um eine Position nach rechts" gerückt sind. Oder anders ausgedrückt, die Positionen der Operationen 2.1 und 1.1, 2.2 und 1.2 sowie 2.3 und 3.2 wurden in der Ausführungsreihenfolge vertauscht.

Die folgenden Abbildungen zeigen die verschiedenen Schritte im disjunktiven Graphen G .

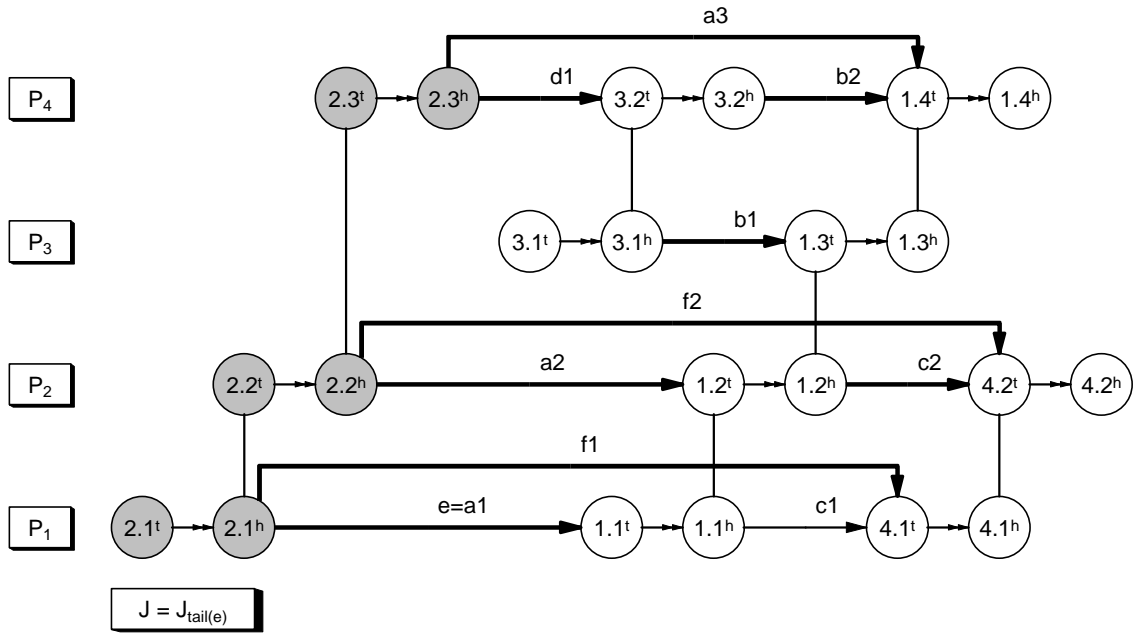


Abbildung 6.15. Zulässige Selektion $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$

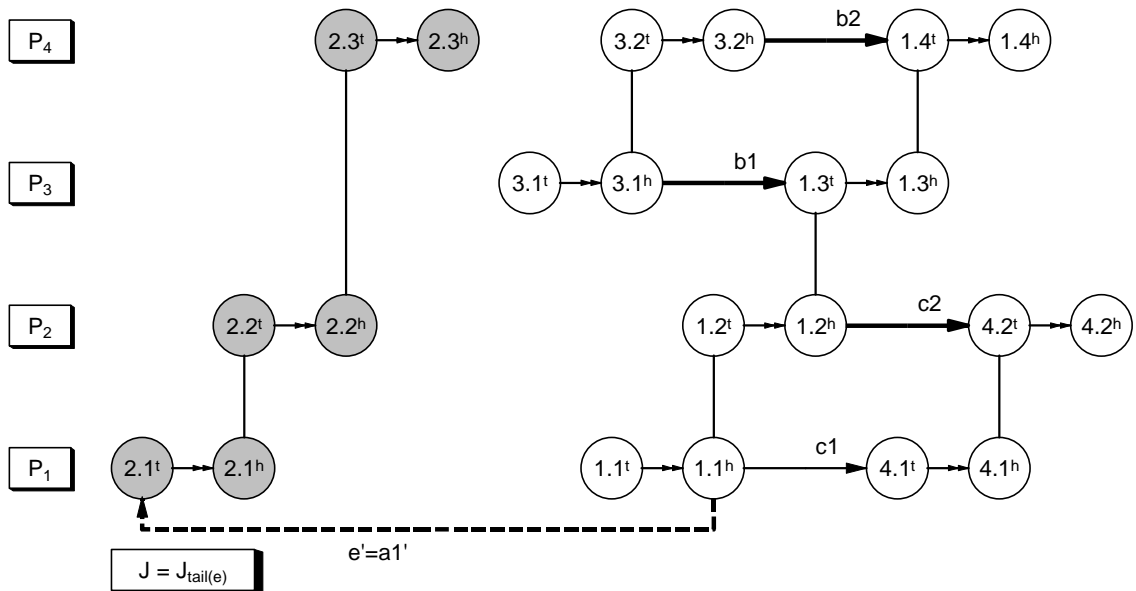


Abbildung 6.16. Partielle Selektion $R_J^e \cup e' = \{b_1, b_2, c_2\} \cup \{a_1'\}$

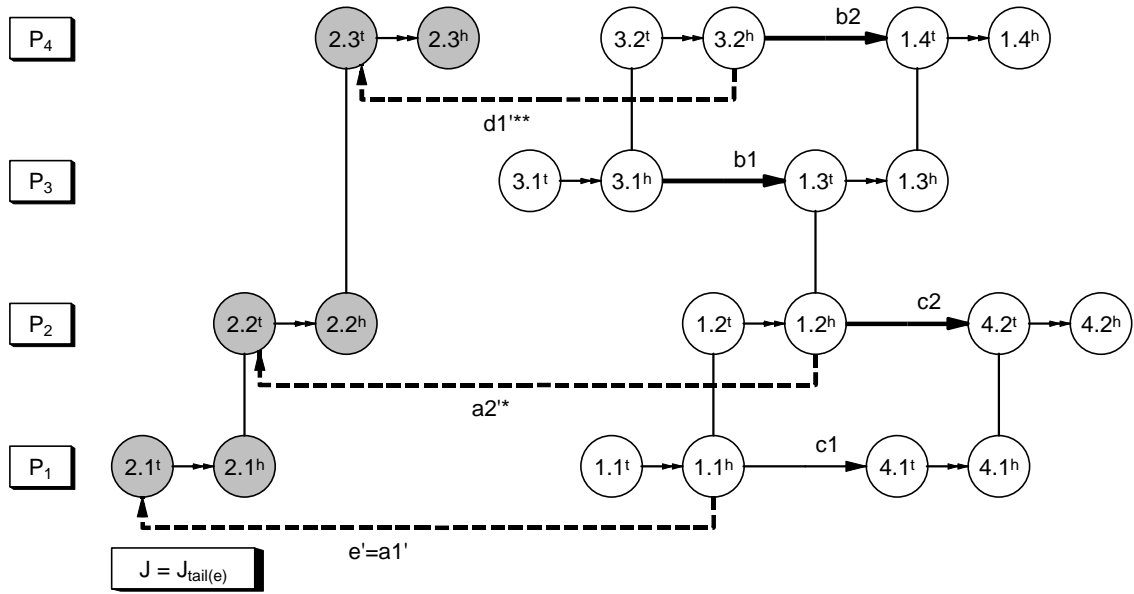


Abbildung 6.17. Konjunktives Erzeugnis $U = \varphi_G^*(R_J^e \cup e') = \{a_1', a_2', b_1, b_2, c_2, d_1'\}$

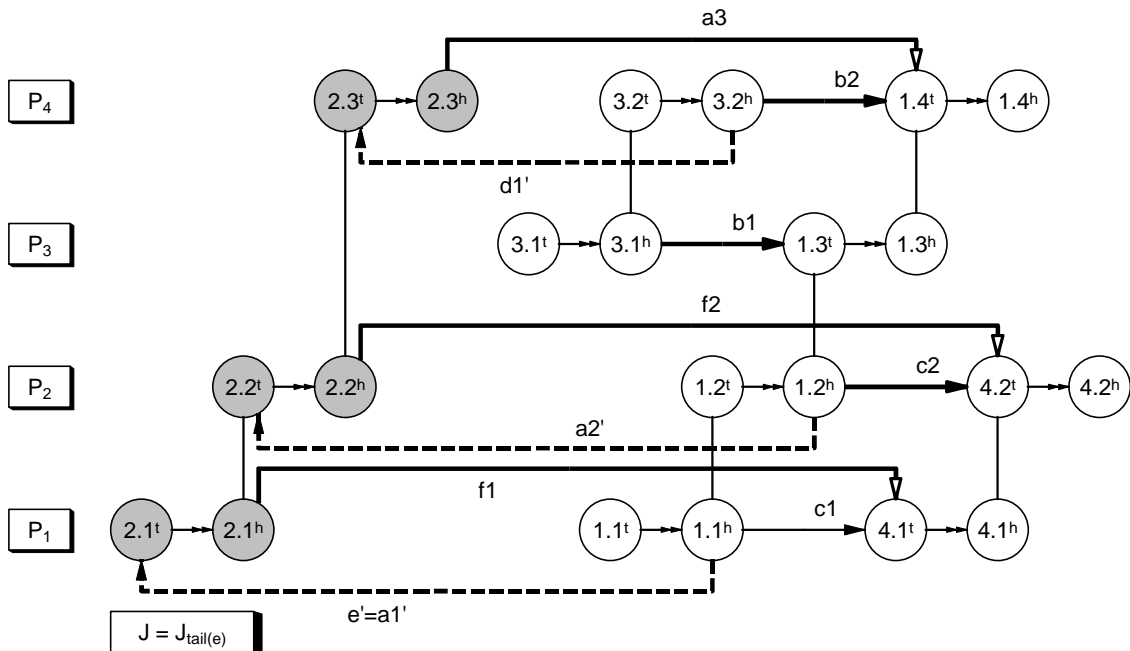


Abbildung 6.18. Nachbarselektion $S_J(e) = U \cup (S - \mathcal{E}^\circ(U))$ mit $S - \mathcal{E}^\circ(U) = \{a_3, f_1, f_2\}$

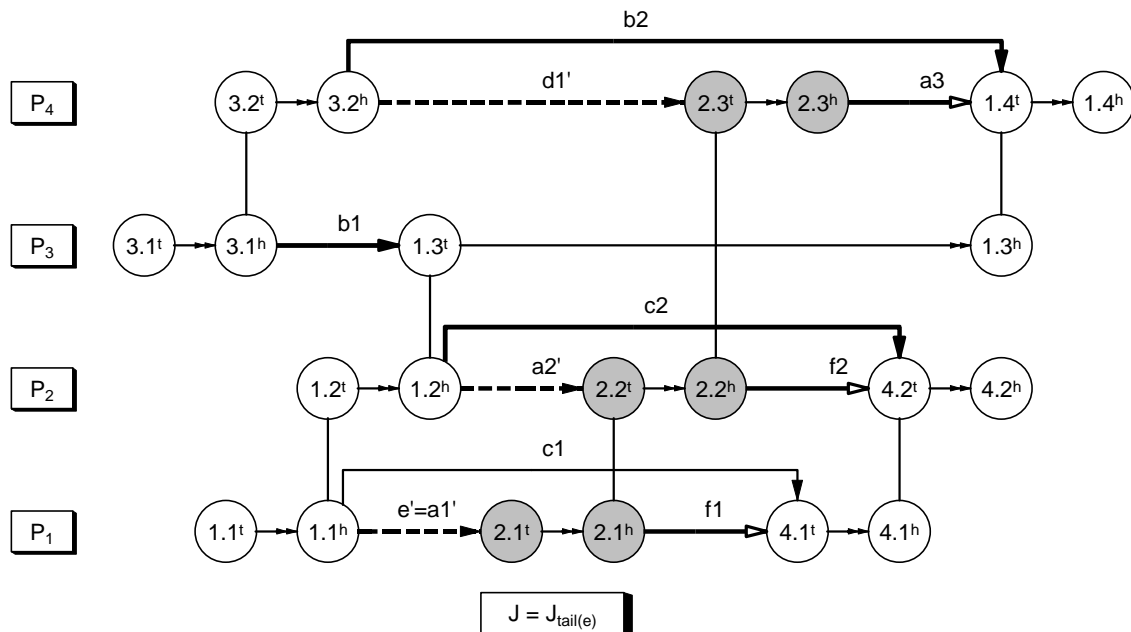


Abbildung 6.19. Nachbarsелеktion $S_J(e) = \{a_1', a_2', a_3, b_1, b_2, c_2, d_1', f_1, f_2\}$ nach Umordnung

Gantt-Diagramme

Untenstehend wird der Uebergang von der Selektion S zur Nachbarselsektion $S_J(e)$ im Gantt-Diagramm gezeigt. Die Zeiten für die verschiedenen Aktionen sind zufällig gewählt. Die angeführten Ablaufpläne entsprechen den semi-aktiven Plänen bezüglich der durch S bzw. $S_J(e)$ definierten Bearbeitungsreihenfolgen, d.h. alle Teiloperationen werden zum frühestmöglichen Zeitpunkt gestartet.

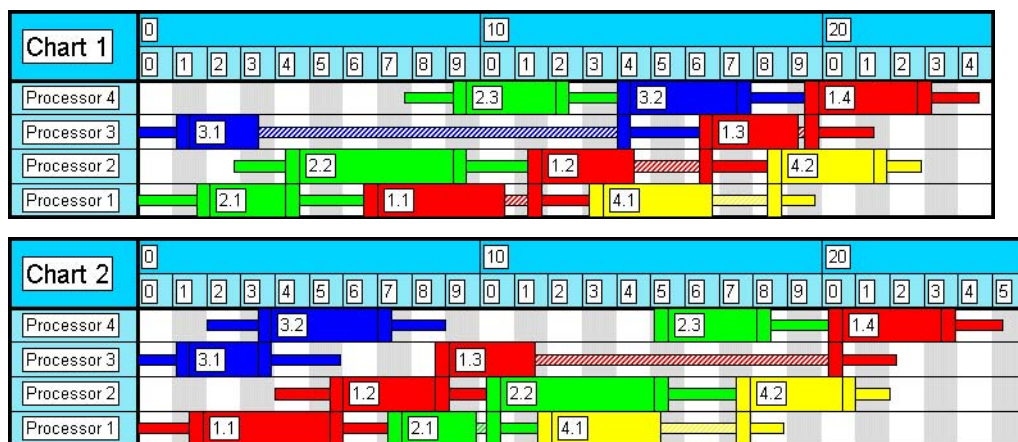


Abbildung 6.20. Gantt-Diagramme zu den Selektionen S und $S_J(e)$

6.4.5. Illustration zu Satz 6.10.

Falls $S_J(e)$ zulässig ist in G gilt gemäss Satz 6.10., dass $S_J(e) = S - \bar{W} \cup W$, wobei $H = G[\phi_G^*(R_J^e)]$ und $W = \phi_H^*(e')$. Weiter besagt der Satz, dass $W \cap S = \emptyset$ und $\bar{W} \subseteq S$ sowie $W \subseteq E_J^{e'}$ und $\bar{W} \subseteq E_J^e$. Dies bedeutet, dass W genau die Bogen darstellt, welche zu S hinzugefügt werden, und \bar{W} genau diejenigen, welche aus S entfernt werden. $W \subseteq E_J^{e'}$ bedeutet, dass die neu zu S hinzukommenden Bogen alle "auf der selben Seite von J wie e' liegen", und die aus S entfernten Bogen "auf derselben Seite von J wie e ".

Betrachte das vorangehende Beispiel mit $S = \{a_1, a_2, a_3, b_1, b_2, c_2, d_1, f_1, f_2\}$ und $e = b_1$ bzw. $e' = a_1'$. Für die Illustration wird der Fall $J = J_{head(e)}$ d.h. $J = J_1$ gewählt. Wie oben gesehen wurde, ist $S_J(e)$ zulässig in G . Somit ist W definiert, und es gilt $S_J(e) = S \cup W - \bar{W}$.

(1) Bestimme den disjunktiven Subgraphen $H = G[\phi_G^*(R_J^e)]$. Es gilt $R_J^e = \{c_2, d_1, f_1, f_2\}$. Berechne das konjunktive Erzeugnis $\phi_G^*(R_J^e)$. In diesem Beispiel (aber nicht allgemein!) gilt, dass $\phi_G^*(R_J^e) = R_J^e$. Folglich ist $H = G[R_J^e]$.

(2) Berechne das konjunktive Erzeugnis $W = \phi_H^*(e')$. Es gilt:

$$\begin{aligned}\phi_H^0(e') &= \{a_1'\} \\ \phi_H^1(e') &= \phi_H^0(e') \cup \{a_2'\} \\ \phi_H^2(e') &= \phi_H^1(e') \cup \{b_1'\} \\ \phi_H^3(e') &= \phi_H^2(e') \cup \{b_2'\} \\ \phi_H^*(e') &= \phi_H^3(e')\end{aligned}$$

Somit ist $W = \phi_H^*(e') = \{a_1', a_2', b_1', b_2'\}$.

(3) Bestimme die komplementäre Menge $\bar{W} = \mathcal{E}^\circ(W) - W$. Es gilt $\mathcal{E}^\circ(W) = \{a_1, a_1', a_2, a_2', b_1, b_1', b_2, b_2'\}$ und folglich $\bar{W} = \{a_1, a_2, b_1, b_2\}$.

(4) Die Nachbarselektion $S_J(e) = S \cup W - \bar{W}$ ist somit gegeben als $S_J(e) = \{a_1', a_2', a_3, b_1', b_2', c_2, d_1, f_1, f_2\}$. Offensichtlich ist $W \cap S = \emptyset$ und $\bar{W} \subseteq S$ d.h. es gilt $S_J(e) - S = W$ und $S - S_J(e) = \bar{W}$.

Die folgenden Abbildungen zeigen die verschiedenen Schritte im disjunktiven Graphen G .

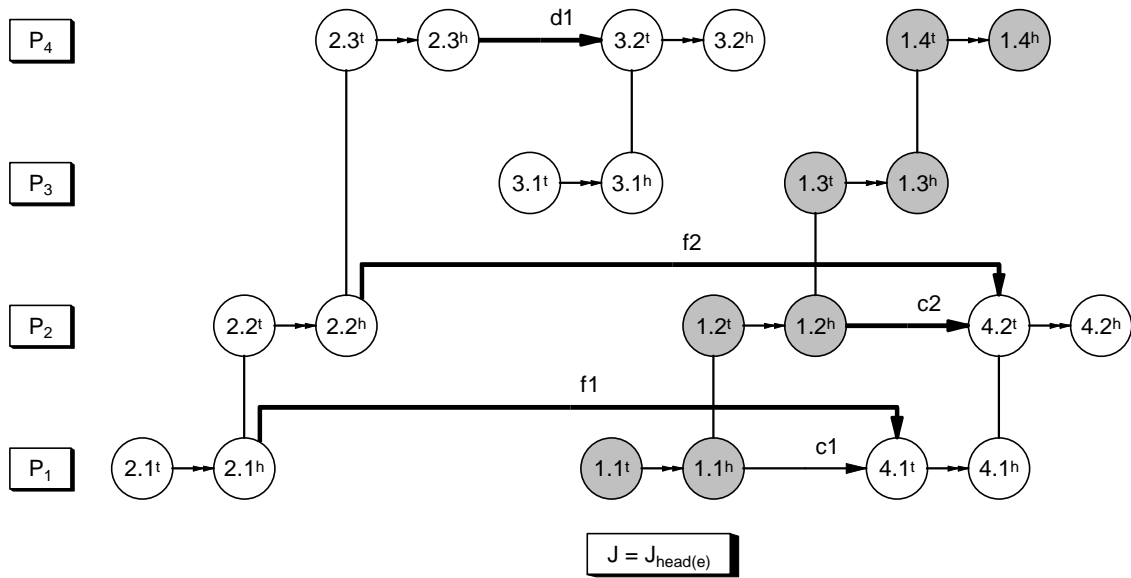


Abbildung 6.21. Disjunktiver Graph $H = G[\varphi_G^*(R_J^\ell)]$ mit $\varphi_G^*(R_J^\ell) = \{c_2, d_1, f_1, f_2\}$

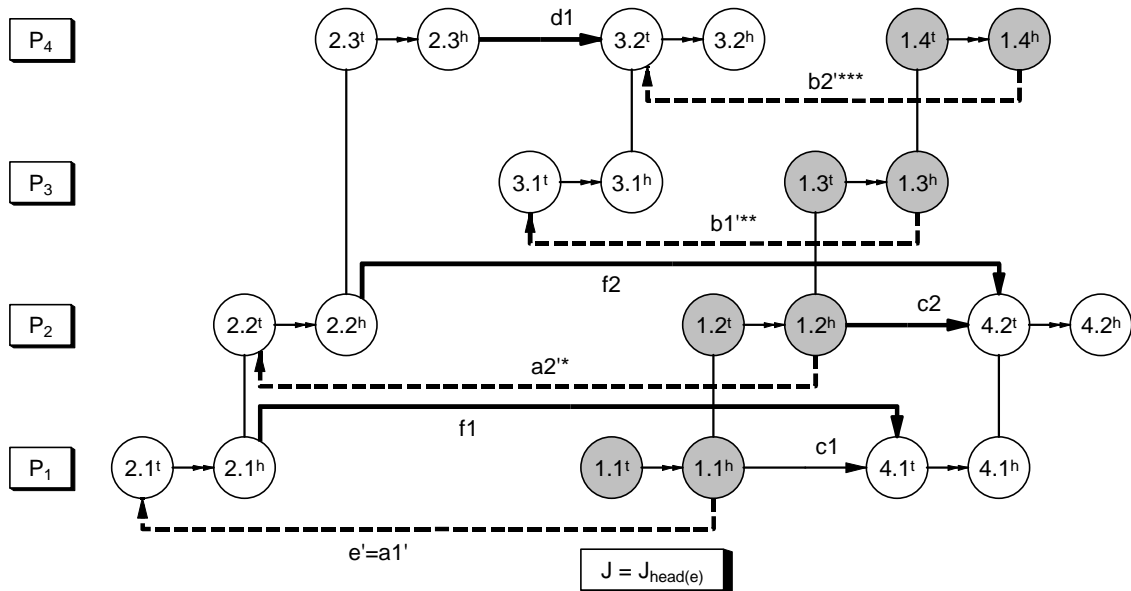


Abbildung 6.22. Konjunktives Erzeugnis $W = \varphi_H^*(e') = \{a_1', a_2', b_1', b_2'\}$ in H

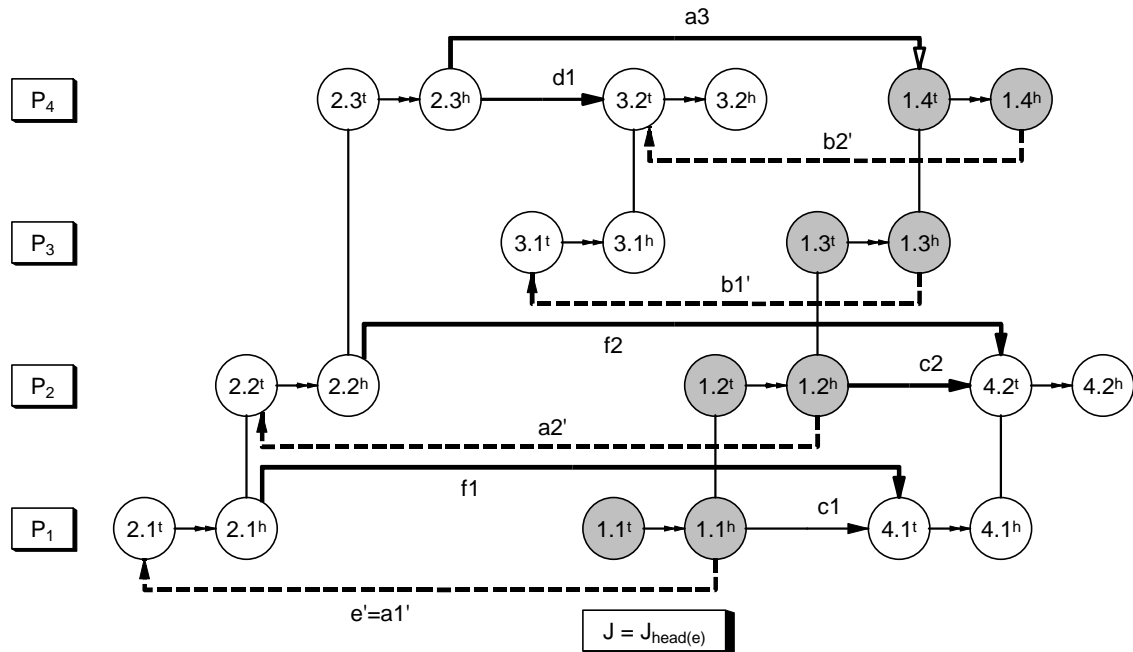


Abbildung 6.23. Nachbarselektion $S_J(e) = S - \bar{W} \cup W = \{a_1', a_2', a_3, b_1', b_2', c_2, d_1, f_1, f_2\}$

6.5. Heuristik

In diesem Abschnitt wird ein iteratives Verbesserungsverfahren für das Lager-Problem, d.h. das Problem der optimierten Steuerung in automatisierten Kompaktlagern, präsentiert. Das Verfahren basiert auf dem vorgängig beschriebenen Nachbarschaftsbegriff $S_J(e)$. Es handelt sich um ein einfaches Abstiegsverfahren, bei welchem nur Nachbarlösungen akzeptiert werden, welche die Zielfunktion verbessern. Anstelle des beschriebenen Abstiegsverfahrens wäre auch eine Tabu-Methode denkbar, welche auf dem beschriebenen Nachbarschaftsbegriff basiert. Dieser Ansatz wurde im Rahmen der vorliegenden Arbeit jedoch nicht weiter verfolgt.

Das Verbesserungsverfahren basiert auf der Idee des Austausches von kritischen disjunktiven Bogen im Lager-Graphen $G = \langle V, A, E, \mathcal{E}, c \rangle$. In jeder Iteration wird ausgehend von einer zulässigen Selektion $S \subseteq E$ und einem kritischen Bogen $e \in S$ eine der beiden Nachbarselektionen $S_J(e)$, $J \in \mathcal{J}(e)$, berechnet. Falls $S_J(e)$ zulässig ist in G und die Länge eines längsten Weges in $G[S_J(e)]$ kleiner ist als in $G[S]$, wird die Nachbarlösung akzeptiert, andernfalls wird die ursprüngliche Selektion S beibehalten.

Im folgenden wird für eine zulässige Selektion $S \subseteq E$ die Länge eines längsten $\sigma\tau$ -Weges in $G(S)$ bezeichnet als $\lambda_G(S)$ und die Menge der kritischen Bogen in $G(S)$ als $\Lambda_G(S)$.

Algorithmus:

Gegeben: Ein Lager-Graph $G = G^\Pi[Q] = \langle V, A, E, \mathcal{E}, c \rangle$ mit Präselektion Q und eine zulässige Selektion $S^0 \subseteq E$ in G .

Gesucht: Eine Selektion S^* in G , für welche $\lambda_G(S^*)$ minimal ist.

0. Setze $S = S^0$. Berechne $\lambda = \lambda_G(S)$ und die Menge der kritischen Bogen $\Lambda = \Lambda_G(S)$.
1. Falls $\Lambda = \emptyset$, dann setze $S^* = S$ Stop.
Wähle einen kritischen Bogen $e \in \Lambda$ und setze $\Lambda = \Lambda - e$.
2. Bestimme eine der beiden Nachbarselektionen $S' = S_J(e)$, $J \in \mathcal{J}(e)$.
Falls S' unzulässig ist in G , gehe zu Schritt 1.
3. Berechne $\lambda' = \lambda_G(S')$ und die Menge der kritischen Bogen $\Lambda' = \Lambda_G(S')$.
Falls $\lambda' < \lambda$, dann setze $S = S'$, $\lambda = \lambda'$ und $\Lambda = \Lambda'$.
Gehe zu Schritt 1.

6.6. Zulässige Startlösung

In den vorangehenden Abschnitten wurde ein iteratives Verbesserungsverfahren für Lager-Graphen entwickelt. Bei diesem Verfahren wird vorausgesetzt, dass eine zulässige Startlösung gegeben ist. In diesem Abschnitt wird das Problem der Bestimmung einer zulässigen Startlösung diskutiert. In Abschnitt 4.3. und Abschnitt 4.4. wurde gezeigt, wie sich das Steuerungsproblem in einem automatisierten Kompaktlager als disjunktives Graphenproblem in einem Lager-Graphen G mit Präselektion Q formulieren lässt. Die Präselektion Q repräsentiert hierbei die Restriktionen, welche durch die räumlichen Nebenbedingungen in den Lagerkorridoren zu Stande kommen. Das Auffinden einer zulässigen Startlösung entspricht somit dem Problem, im disjunktiven Graphen $G[Q]$ eine zulässige Selektion zu finden. In Satz 5.65. wurde gezeigt, dass das Zulässigkeitsproblem mit Präselektion für disjunktive Graphen NP-hart ist.

Betrachte nun Lager-Probleme, bei welchen nur Ein- und Auslagerungen, aber keine Umlagerungen ausgeführt werden. In diesem Fall lässt sich zeigen, dass die spezielle Präselektion Q , welche aus den räumlichen Nebenbedingungen resultiert, bestimmte strukturelle Eigenschaften aufweist, auf Grund welcher das Zulässigkeitsproblem polynomial lösbar ist. Im nachfolgenden Satz wird gezeigt, dass bei Lager-Problemen ohne Umlagerungen genau dann eine zulässige Selektion in $G[Q]$ existiert, wenn Q positiv azyklisch ist in G . Anschaulich bedeutet dies, dass eine zuläs-

sige Selektion existiert, wenn die aus den räumlichen Nebenbedingungen resultierenden Präferenzbedingungen konsistent sind.

Für allgemeine Lager-Probleme, welche auch Umlagerungen umfassen, konnte kein polynomiales Verfahren gefunden werden zur Bestimmung einer zulässigen Startlösung. Es besteht die starke Vermutung, dass das Problem NP-hart ist. Tatsächlich wurde in Satz 5.65. bewiesen, dass das Zulässigkeitsproblem für Lager-Graphen bei Betrachtung beliebiger Präselektionen NP-hart ist. Es erscheint unwahrscheinlich, dass die spezielle Struktur der Präselektionen, welche aus den räumlichen Nebenbedingungen resultieren, diese Komplexitätseigenschaft verändern. Das Problem der zulässigen Startlösung für allgemeine Lager-Graphen mit Umlagerungen wurde im Rahmen dieser Arbeit nicht weiter untersucht. Es scheint jedoch so, dass in der Praxis beim Einsatz geeigneter Umlagerungsstrategien das Problem der Zulässigkeit eine untergeordnete Rolle spielt und praxisrelevante Fälle, wo keine zulässige Lösung gefunden werden kann, eher selten sind.

Satz 6.12. Sei $\Pi = (M, I, \mathcal{J}, \mu, d, d^t, d^h, d^S)$ ein Lager-Modell ohne Umlagerungen und $G = G^\Pi = \langle V, A, E, \mathcal{E}, c \rangle$ der zugehörige Lager-Graph. Sei $Q \subseteq E$ die Präselektion in G , welche aus den räumlichen Nebenbedingungen resultiert. $G[Q]$ hat genau dann eine zulässige Selektion, wenn Q positiv azyklisch ist in G .

Beweis:

Zeige einerseits: Falls $G[Q]$ eine zulässige Selektion hat, dann muss Q positiv azyklisch sein in G . Sei S eine zulässige Selektion in $G[Q]$. Gemäss Lemma 5.11. ist $Q \cup S$ eine zulässige Selektion in G . Folglich ist Q positiv azyklisch in G . Zeige andererseits: Falls Q positiv azyklisch ist in G , dann existiert eine zulässige Selektion in $G[Q]$. Sei Q positiv azyklisch in G . Q ist gemäss Abschnitt 4.4. gegeben durch $Q = \bigcup_{m \in M^C} (Q_m^{in} \cup Q_m^{out})$, wobei

$$Q_m^{in} = \{(h_i, t_j) : i \in I_m - j, j \in I_m^{in}, p_j \in P_i\}, \quad m \in M^C$$

$$Q_m^{out} = \{(h_i, t_j) : i \in I_m^{out}, j \in I_m - i, p_i \in P_j\}, \quad m \in M^C$$

Da Q positiv azyklisch ist in G , existiert eine topologische Sortierung in $G(Q)$, d.h. eine Nummerierung der Knoten $f: V \rightarrow \{1, \dots, |V|\}$ mit $f(v) \neq f(w)$ für $v, w \in V$ $v \neq w$ so dass $f(v) < f(w)$ für alle Bogen $(v, w) \in A \cup Q$ von $G(Q)$. Da jeder Job $J \in \mathcal{J}$ ein Ein- oder Auslagerungsjob ist, hat er genau eine Ein- oder Auslagerungsoperation i auf einem Korridor-Prozessor $m \in M^C$. Durch die topologische Nummer $f(t_i)$ des t -Knotens von i kann somit jedem Job $J \in \mathcal{J}$ eine eindeutige Nummer zugeordnet, welche bezeichnet wird als $n(J) = f(t_i)$. Sei $(h_i, t_j) \in Q$ und bezeichne J_i bzw. J_j den Job, welcher Operation i bzw. j enthält. Da f eine topologische Nummerierung in $G(Q)$ ist, gilt $f(h_i) < f(t_j)$. Da $G(Q)$ den konjunktiven Bogen (t_i, h_j) enthält, gilt $f(t_i) < f(h_j)$ und folglich $f(t_i) < f(t_j)$. Da $f(t_i) = n(J_i)$ und $f(t_j) = n(J_j)$ gilt $n(J_i) < n(J_j)$. Somit

ist gezeigt, dass für alle Jobs $J \in \mathcal{J}$ eine (paarweise verschiedene) Nummer $n(J)$ existiert, so dass $n(J_v) < n(J_w)$ für alle Bogen $(v, w) \in Q$. Ordne nun die Jobs $J \in \mathcal{J}$ in aufsteigender Reihenfolge ihrer Nummer $n(J)$ und nummeriere sie neu in aufsteigender Reihenfolge. Daraus resultiert eine Numerierung $n' : \mathcal{J} \rightarrow \{1, \dots, |\mathcal{J}|\}$ der Jobs mit der Eigenschaft, dass $n'(J_v) < n'(J_w)$ für alle Bogen $(v, w) \in Q$. Daraus folgt gemäss Satz 5.64., dass $G[Q]$ eine zulässige Selektion besitzt. ♦

Der vorangehende Beweis liefert ein einfaches Verfahren, wie in einem Lager-Graphen $G[Q]$ ohne Umlagerungen eine zulässige Startlösung gefunden werden kann, wobei Q die aus den räumlichen Nebenbedingungen resultierende Präselektion darstellt: Falls $G(Q)$ positiv zyklisch ist, existiert keine zulässige Selektion. Andernfalls kann eine topologische Numerierung der Knoten von $G(Q)$ bestimmt werden. Die Nummern der t -Knoten (oder auch der h -Knoten) der Ein- bzw. Auslagerungsoperationen der Jobs auf den Korridorprozessoren bestimmen eine eindeutige Reihenfolge der Jobs. Eine zulässige Selektion kann konstruiert werden, indem die Jobs in dieser Reihenfolge abgearbeitet werden.

6.7. Numerische Resultate

In diesem Abschnitt werden numerische Testergebnisse für das Lager-Problem präsentiert. Die Resultate basieren auf der Problemdarstellung in disjunktiven Lager-Graphen und dem Verbesserungsverfahren, welches in diesem Kapitel entwickelt wurde.

Implementation

Das Modell wurde in der Programmiersprache C++ implementiert. Aufgrund der hohen Rechenanforderungen bei der Evaluation von Nachbarschaftslösungen wurde dem Entwurf von effizienten Datenstrukturen besondere Beachtung geschenkt. Für die Darstellung der disjunktiven Lager-Graphen wurden spezielle Datenstrukturen entwickelt, welche auf einer Kombination von dynamischen Adjazenzlisten und Adjazenzmatrix-Darstellung basieren. Dadurch wurde erreicht, dass alle relevanten Graphen-Operationen in minimaler Zeit ausgeführt werden können. Der Test auf Knoten- und Bogenzugehörigkeit sowie das Einfügen und Entfernen von Knoten und Bogen ist mit konstantem Aufwand $O(1)$ möglich. Die Enumeration von Knoten- und Bogenmengen benötigt linearen Aufwand $O(n)$, wobei n die Kardinalität der Enumerationsmenge bezeichnet. Da diese Datenstrukturen sehr speicherintensiv sind, können sie nur bis zu einer gewissen Problemgrösse eingesetzt werden. Die numerischen Tests haben gezeigt, dass auf einem Rechner mit 128 MB Arbeitsspeicher Probleme bis zur Obergrenze von ungefähr 400 Jobs bearbeitet werden können. Aus diesen Gründen wurden alternative Datenstrukturen implementiert, welche auf einer Mengendarstellung mittels geordneter Suchbäume basieren. Diese Datenstrukturen benöti-

gen ein Minimum an Speicherplatz, sind jedoch weniger effizient. Für die oben beschriebenen Graphen-Operationen resultiert ein logarithmischer Aufwand $O(\log(n))$. Eine generische Implementation erlaubt wahlweise das Umschalten zwischen diesen beiden Datenstrukturen. Die Tests wurden unter Windows NT auf einem Rechner mit Prozessor Intel Pentium III, 800 MHz, und 128 MB Arbeitsspeicher durchgeführt.

Zur Visualisierung der Ablaufpläne (Gantt-Diagramme) wurde ein eigenständiges Programm entwickelt. Dieses wurde mit der Programmiersprache VBA 5.0 unter Microsoft Excel implementiert und erzeugt Ablaufpläne in Form von Excel-Grafiken. Das Programm erlaubt eine flexible und graphisch hochstehende Darstellung von Gantt-Diagrammen und kann sowohl interaktiv als auch im Batch-Modus bedient werden.

Testparameter

Für das Verbesserungsverfahren wurde als Nachbarschaftsbegriff von den beiden möglichen Nachbarselektionen $N_J(e)$, $J \in \mathcal{J}(e)$ die Selektion für $J = J_{head(e)}$ gewählt (vgl. Abschnitt 6.5.). Numerische Tests haben gezeigt, dass mit dieser Nachbarschaft im Durchschnitt leicht bessere Resultate erzielt werden. Für die Wahl der kritischen Bogen, welche ausgetauscht werden, wurden verschiedene Strategien geprüft. Als erfolgreichste Strategie stellte sich eine zufällige Wahl der kritischen Bogen heraus. Die verschiedenen Prozessoren-Parameter wurden nach eigenem Ermessen gewählt.

Testdaten

Es wurden insgesamt vier verschiedene Lagerkonfigurationen getestet. Die folgende Abbildung gibt einen Ueberblick über die Konfigurationen.

Konfiguration	Sektoren	Ebenen	Korridore pro Sektor	Plätze pro Korridor	Lagerplätze	Prozessoren
A	3	3	5	20	900	65
B	5	3	5	20	1500	105
C	4	4	10	20	3200	190
D	6	4	20	20	9600	522

Abbildung 6.24. Lagerkonfigurationen für numerische Tests

Für jede Konfiguration wurden fünf Testserien mit unterschiedlicher Anzahl Jobs (d.h. Paletten bzw. Lagerbewegungen) betrachtet. Als Lagerbewegungen wurden Einlagerungen gewählt. Die Testserien umfassen 25, 50, 100, 150 und 200 Paletten. Für jede Lagerkonfiguration und jede Serie wurden zwanzig zufällig generierte Beispiele geprüft. In den Beispielen wurden auch Lagerbewegungen betrachtet, welche aus mehr als drei Operationen bestehen, d.h. bei welchen einer

oder mehrere Sektoren traversiert werden. Die Wahl der Durchfahrtskorridore erfolgte für jedes Beispiel nach dem Zufallsprinzip. Für die Konfigurationen *A* und *B* wurden jeweils 70% dreistufige und 30% fünfstufige Lagerbewegungen generiert. Eine dreistufige Einlagerung umfasst drei Operationen und benützt der Reihe nach einen Korridorwagen, einen Quergangswagen und einen Aufzug. Eine fünfstufige Einlagerung umfasst fünf Operationen und entspricht der Prozessorabfolge Korridorwagen, Quergangswagen, Korridorwagen (Durchfahrt), Quergangswagen, Aufzug. Für die Konfigurationen *C* und *D* wurden 50% dreistufige, 30% fünfstufige und 20% siebenstufige Lagerbewegungen generiert.

Als Startlösung wurde jeweils ein Ablaufplan gewählt, bei welchem die Jobs in zufälliger Abfolge der Reihe nach abgearbeitet werden, unter Berücksichtigung der räumlichen Nebenbedingungen in den Korridoren (vgl. Abschnitt 6.6.)

Testresultate

Die folgende Tabelle zeigt eine Uebersicht über die Testresultate. Als Mass für die erzielte Verbesserung der Ablaufpläne wird der Optimierungsfaktor (Ratio) angegeben. Dieser zeigt an, um wieviel Prozent der Makespan der Startlösung verbessert wurde, d.h.

$$\text{Ratio} = \frac{\text{Initial Makespan} - \text{Final Makespan}}{\text{Initial Makespan}}$$

Im weiteren werden die folgenden Bezeichnungen verwendet:

<i>Config</i>	Lagerkonfiguration
<i>Pal</i>	Anzahl Paletten (bzw. Jobs, Lagerbewegungen) der Testserie
<i>Proc</i>	Anzahl Prozessoren der Lagerkonfiguration
<i>Op</i>	Anzahl Operationen der Testserie
<i>Moves</i>	Anzahl Verbesserungsschritte im Optimierungsverfahren, d.h. Anzahl Uebergänge zu einer besseren Nachbarselektion
<i>Test Moves</i>	Anzahl Nachbarselektionen, welche im Optimierungsverfahren getestet wurden
<i>Sec</i>	Rechenzeit in CPU-Sekunden
<i>Dev Ratio</i>	Standardabweichung des Optimierungsfaktors über alle Beispiele der Testserie
<i>Min Ratio</i>	Minimum des Optimierungsfaktors über alle Beispiele der Testserie
<i>Max Ratio</i>	Maximum des Optimierungsfaktors über alle Beispiele der Testserie
<i>Ratio</i>	Mittlerer Optimierungsfaktor über alle Beispiele der Testserie

<i>Config</i>	<i>Pal</i>	<i>Proc</i>	<i>Op</i>	<i>Moves</i>	<i>Test Moves</i>	<i>Sec</i>	<i>Dev Ratio</i>	<i>Min Ratio</i>	<i>Max Ratio</i>	<i>Ratio</i>
<i>A</i>	25	65	90	7	28	0.6	0.12	0.06	0.48	0.29
	50	65	180	27	217	12.8	0.09	0.17	0.51	0.41
	100	65	360	60	742	161.7	0.07	0.34	0.51	0.42
	150	65	540	74	1206	169.0	0.04	0.33	0.46	0.40
	200	65	720	107	2078	1081.1	0.05	0.33	0.50	0.42
<i>B</i>	25	105	90	7	16	0.1	0.14	0.00	0.53	0.29
	50	105	180	13	57	4.1	0.08	0.22	0.47	0.38
	100	105	360	43	286	194.0	0.07	0.30	0.58	0.43
	150	105	540	61	713	552.6	0.11	0.29	0.55	0.42
	200	105	720	81	1194	811.0	0.05	0.30	0.50	0.42
<i>C</i>	25	190	110	7	25	0.2	0.11	0.05	0.48	0.27
	50	190	220	16	90	1.8	0.10	0.14	0.50	0.33
	100	190	440	42	397	26.9	0.10	0.16	0.54	0.40
	150	190	660	69	811	114.6	0.06	0.29	0.51	0.42
	200	190	880	96	1227	1093.5	0.07	0.32	0.54	0.44
<i>D</i>	25	522	110	5	17	0.1	0.10	0.08	0.44	0.24
	50	522	220	15	69	1.5	0.13	0.05	0.56	0.34
	100	522	440	32	228	16.2	0.11	0.22	0.54	0.39
	150	522	660	54	530	75.4	0.11	0.21	0.57	0.41
	200	522	880	58	794	203.8	0.06	0.25	0.49	0.37

Abbildung 6.25. Testresultate

Besprechung der Testresultate

Der mittlere Optimierungsfaktor liegt für alle Testserien zwischen 24% und 44%. Für Testserien mit mehr als 50 Paletten beträgt er praktisch durchwegs 40% oder mehr. Der minimale Optimierungsfaktor bewegt sich im Bereich von 0% bis 34% und der maximale Optimierungsfaktor im Bereich von 46% bis 58%. Für grössere Beispiele verkleinert sich die Differenz zwischen dem minimalen und maximalen Optimierungsfaktor tendenziell.

Offensichtlich haben Beispiele mit wenig Paletten ein kleineres Optimierungspotential als grössere Beispiele. Dies lässt sich folgendermassen erklären: Durch die Anzahl der Paletten (bzw. Lagerbewegungen) wird die Anzahl der Operationen bestimmt. Wenn das Verhältnis zwischen den Operationen und den vorhandenen Prozessoren klein ist, werden auf jedem Prozessor nur wenige Operationen ausgeführt. Dadurch bestehen nur wenig Möglichkeiten, den Ablaufplan durch eine Aenderung der Operationenreihenfolgen auf den Prozessoren zu verbessern. Zu beachten ist bei diesen Überlegungen, dass sich die Operationen nicht gleichmässig auf die Prozessoren verteilen. Aufgrund der Lager-Architektur werden auf den Korridorprozessoren nur wenig

Operationen ausgeführt (ausgenommen die Durchfahrtskorridore), auf den Quergangswagen werden tendenziell mehr und auf den Aufzügen am meisten Operationen ausgeführt.

Weiter ist erkennbar, dass das Optimierungsverfahren stabiler wird, je grösser das Verhältnis der Operationen zu den Prozessoren ist. Die Zunahme der Stabilität lässt sich einerseits erkennen an den abnehmenden Werten der Standardabweichung und andererseits an der Abnahme der Differenz zwischen dem maximalen und minimalen Optimierungsfaktor. Dieser Effekt lässt sich ebenfalls mit dem oben beschriebenen Zusammenhang erklären: Wenn nur sehr wenige Operationen auf jedem Prozessor ausgeführt werden, existieren wenig Veränderungsmöglichkeiten für den Ablaufplan. Folglich wirkt sich die Qualität der Startlösung stärker aus als bei grösseren Problemen, wo die Startlösung mehr verändert wird.

Sowohl die Anzahl der Moves als auch die Anzahl der getesteten Moves nimmt zu, je grösser die Beispiele werden. Dies hängt ebenfalls damit zusammen, dass die Verbesserungsmöglichkeiten um so grösser werden, je mehr Operationen pro Prozessor ausgeführt werden. In Beispielen, wo das Verhältnis der Operationen zu den Prozessoren relativ gross ist (vgl. Serie A und B mit 150 und 200 Paletten), zeigt sich eine grosse Differenz zwischen den getesteten und den tatsächlich ausgeführten Moves. Dies hängt damit zusammen, dass die Anzahl der kritischen Bogen wächst, je mehr Operationen pro Prozessor ausgeführt werden.

Die benötigte Rechenzeit ist in wesentlichem Masse abhängig von der Anzahl der getesteten Moves. Sie wächst auch mit zunehmender Grösse des disjunktiven Graphen, da der Aufwand für die Evaluation einer Nachbarselektion von Grösse des disjunktiven Graphen abhängig ist.

Zusammenfassend lässt sich feststellen, dass die Rechenresultate ein beträchtliches Optimierungspotential des Verbesserungsverfahrens aufzeigen. Dies deutet darauf hin, dass mit dem Nachbarschaftsbegriff wichtige strukturelle Eigenschaften der disjunktiven Lager-Graphen erfasst werden konnten.

Beispiele im Anhang

Im Anhang zu dieser Arbeit werden die Ablaufpläne von zwei Testbeispielen gezeigt. Als erstes Beispiel wird ein Problem aus Serie A mit 50 Paletten gewählt, bei welchem ein Optimierungsfaktor von 51% erzielt wird. Das zweite Beispiel stammt aus Serie B und umfasst hundert Paletten. Der Optimierungsfaktor beträgt 58%.

Für jedes Beispiel wird der anfängliche und der optimierte Ablaufplan gezeigt. Aufgrund der Grösse der Ablaufpläne werden beim zweiten Beispiel nicht alle Korridorprozessoren aufgeführt. Da die Korridorprozessoren nur wenige Operationen ausführen, geht dadurch keine wesentliche Information verloren.

Zu den Gantt-Diagrammen sei bemerkt, dass die gezeigten Ablaufpläne semi-aktiv sind, d.h. die Operationen werden zum frühestmöglichen Zeitpunkt ausgeführt in Bezug auf die gegebenen

Bearbeitungsreihenfolgen. Eine Ausnahme bilden die anfänglichen Leerfahrten (Setup), welche aus Gründen der Uebersichtlichkeit zum spätestmöglichen Zeitpunkt ausgeführt werden. Die Operationen eines Jobs (d.h. einer Lagerbewegung) sind durch dieselbe Farbe gekennzeichnet. Die i -te Operation eines Jobs n ist mit $n.i$ beschriftet. Leerfahrten sind als schmale Balken eingezeichnet. Wartezeiten in beladenem Zustand (vor einer Palettenübergabe bzw. -übernahme) sind durch schmale, schräg schraffierte Balken dargestellt.

Konklusion

Automatisierte Kompaktlager bilden in der modernen Lagertechnik eine platzsparende Alternative zu herkömmlichen Zeilenregallagern. Sie verfügen über ein vollautomatisiertes, mehrstufiges Transportsystem bestehend aus Aufzügen, Quergangwagen und Kanalfahrzeugen. Die grosse Zahl der individuell gesteuerten Förderprozessoren, die Architektur des Transportsystems und die Synchronisationsbedingungen beim Lasttransfer zwischen den Prozessoren führen zu einer Komplexität des Steuerungssystems, welche mit traditionellen Ansätzen kaum bewältigbar ist. Praktische Erfahrungen haben gezeigt, dass ein Mangel besteht an entscheidungsunterstützenden Methoden bei der Konzeption und Dimensionierung der Lager sowie beim Entwurf der Lagersteuerung. In dieser Arbeit wurden mathematische Modelle entwickelt, welche einen Beitrag zur Behebung dieses Mangels leisten. Es wurden zwei Problemstellungen bearbeitet: Erstens die Problematik der Leistungsanalyse, welche speziell in der Entwurfsphase bei der Evaluation verschiedener Design-Alternativen eine wichtige Rolle spielt. Zweitens die Optimierung der Steuerung auf der Stufe der Ablaufpläne.

Für die Leistungsanalyse wurde ein lineares Modell entwickelt, das auf einer abstrahierten Netzwerk-Darstellung des Lagersystems basiert, in welchem die Lagerbewegungen als Mehrgrüterflüsse dargestellt werden können. Durch Flussoptimierung lässt sich die Prozessorenbelastung bei optimaler Allokationsstrategie abschätzen, woraus eine obere Schranke für die Lagerleistung resultiert. Gegenüber simulationsbasierten Methoden hat das Modell den wesentlichen Vorteil, dass es steuerungsunabhängig ist und nur auf Inputdaten beruht, welche in der Entwurfsphase tatsächlich zur Verfügung stehen. Im Gegensatz zu anderen Ansätzen stellen die berechneten Leistungszahlen verlässliche Leistungsgrenzen dar, welche nicht aufgrund von vereinfachenden Annahmen mit Unsicherheit behaftet sind. Mit Hilfe dieser Leistungszahlen können Design-Alternativen miteinander verglichen werden hinsichtlich der zu erwartenden Lagerlei-

stung. Weiter liefert das Modell Informationen über die Verteilung der Belastung auf die verschiedenen Prozessoren, wodurch die Engpass-Ressourcen des Systems erkannt werden können. Daraus lassen sich Rückschlüsse auf die Dimensionierung des Lagers und die Anordnung der Transproteleinrichtungen machen. Schliesslich bietet das Modell aufgrund der optimierten Ressourcenallokation Unterstützung beim Entwurf von optimierten Lager-, Routen- und Reorganisationsstrategien. Zusammenfassend kann gesagt werden, dass das Modell in verschiedener Hinsicht wertvolle Beiträge leisten kann beim Entwurf automatisierter Kompaktlager.

Da das Modell auf einer abstrahierten Sichtweise der Lagerprozesse in Form von Prozessoren, Operationen und Ressourcenallokations-Ueberlegungen basiert, dürfte es auch auf andere Verarbeitungssysteme mit ähnlichen Eigenschaften anwendbar sein. Zunächst existieren im Bereich der Lagertechnik diverse moderne Typen von dynamischen Kompaktlagern, welche eine ähnliche Architektur haben wie die betrachteten Lagersysteme. Allgemein ist die Anwendbarkeit des Modells denkbar bei Verarbeitungssystemen, in welchen für die Bearbeitung der Jobs verschiedene Prozessalternativen bzw. Prozessrouten bestehen. Darunter ist zu verstehen, dass die Bearbeitung der Objekte eine Folge von Prozessschritten (bzw. Operationen) umfasst, wobei eine Wahlmöglichkeit besteht, auf welchen Prozessoren die Bearbeitungsschritte ausgeführt werden. Im vorliegenden Modell werden die Prozessschritte durch die Bogen des System-Graphen repräsentiert und die Prozessalternativen entsprechen den verschiedenen Routen von einer Quelle zu einer Senke. Die Wahl der Prozessroute, d.h. die Allokation der Operationen auf den Prozessoren, erfolgt durch die Festlegung der Flüsse im System-Graphen. Zusammenfassend deuten diese Ueberlegungen darauf hin, dass für das entworfene Modell ein interessantes Anwendungspotential besteht sowohl innerhalb der Lagertechnik als auch in anderen Anwendungsbereichen.

Als zweite Problemstellung wurde die Optimierung der Steuerung automatisierter Kompaktlager auf der Stufe der Ablaufpläne betrachtet. Es wurde ein Optimierungsmodell entwickelt, welches auf einer Beschreibung der zulässigen Ablaufpläne mit Hilfe von konjunktiven und disjunktiven Präzedenzbedingungen basiert. Es wurde gezeigt, wie sich dieses Modell als Optimierungsproblem in verallgemeinerten disjunktiven Graphen darstellen lässt. Die Verallgemeinerung gegenüber klassischen disjunktiven Graphen besteht darin, dass die disjunktiven Mengen Bogen mit unterschiedlichen Extremitäten umfassen können. Die strukturellen Eigenschaften verallgemeinerter disjunktiver Graphen wurden ausführlich untersucht, wobei als wichtiges Resultat gezeigt wurde, dass das Zulässigkeitsproblem für verallgemeinerte disjunktive Graphen und insbesondere auch für Lager-Graphen NP-hart ist. Eine Konsequenz dieser Tatsache ist, dass herkömmliche Lösungsansätze aus dem Bereich der Maschinen-Ablaufplanung und insbesondere des Job-Shop-Problems nicht auf das Lager-Problem übertragbar sind. Die Entwicklung eines Nachbarschaftsbegriffs für ein Verbesserungsverfahren erwies sich als anspruchsvoll und stellte eine der zentralen Herausforderungen dieser Arbeit dar. Es konnte ein geeigneter Nachbarschaftsbegriff entworfen werden, welcher einerseits auf dem Prinzip der erzwungenen und verbotenen Bogen beruht und andererseits auf dem Konzept der 1-Job-Lager-Graphen. Die Idee für diesen Nachbar-

schaftsbegriff entstammte dem Studium der Struktur der möglichen Zyklen in disjunktiven Lager-Graphen und insbesondere in konjunktiv abgeschlossenen, positiv azyklischen 1-Job-Lager-Graphen. Mit Hilfe dieses Nachbarschaftsbegriffs wurde ein iteratives Abstiegsverfahren entwickelt, welches auf dem Austausch von kritischen Bogenmengen beruht. Die numerischen Resultate zeigen ein beträchtliches Optimierungspotential dieses Verfahrens, was darauf hindeutet, dass mit dem Nachbarschaftsbegriff wesentliche strukturelle Eigenschaften der disjunktiven Lager-Graphen auf adäquate Weise erfasst werden konnten.

Die weiterführende Forschung zu dieser Arbeit zielt in zwei Richtungen. Einerseits wird versucht, das bestehende Abstiegsverfahren zu verbessern, insbesondere durch eine Erweiterung der Suchstrategie (beispielsweise Tabu Search oder Simulated Annealing) und durch Verallgemeinerung des Nachbarschaftsbegriffs. Andererseits wird die Anwendbarkeit der entwickelten Ansätze auf weitere Probleme untersucht.

Allgemein scheinen verallgemeinerte disjunktive Graphen einen interessanten Modellierungsansatz darzustellen, mit welchem gewisse Maschinenbelegungsprobleme auf adäquate Weise dargestellt werden können. Die Optimierung in verallgemeinerten disjunktiven Graphen ist anspruchsvoll, wie die Erfahrungen dieser Arbeit zeigen. Es ist unwahrscheinlich, dass eine allgemeine Lösungsmethode für disjunktive Graphenprobleme gefunden werden kann. Der Entwurf eines Optimierungsverfahrens für disjunktive Graphen bedingt typischerweise eine tiefgehende Analyse der spezifischen Struktureigenschaften der betrachteten Graphen. Es gibt jedoch Hinweise dafür, dass der im Zusammenhang mit dem Lager-Problem entworfene Nachbarschaftsbegriff bis zu einem gewissen Mass auf andere disjunktive Graphenprobleme übertragbar ist. Weitere Forschungsarbeit in dieser Richtung scheint lohnenswert.

Zum Abschluss dieser Dissertation möchte ich festhalten, dass diese Arbeit in zweifacher Hinsicht eine Herausforderung und Bereicherung für mich darstellte. Einerseits beinhaltete sie die Analyse und Lösung von praxisrelevanten, anspruchsvollen Problemen im Bereich der Lagertechnik. Andererseits ergaben sich aus der Modellierung und Optimierung dieser Probleme theoretische, mathematische Fragestellungen, welche auf einer abstrahierten Ebene von eigenständigem Interesse sind. Diese Kombination und gegenseitige Inspiration von praxisbezogener Problemlösung und theoretischer mathematischer Forschungsarbeit bildet für mich einen faszinierenden Aspekt des Operations Research.

DEFINITIONEN UND NOTATIONEN

Begriffe aus der Graphentheorie

Ein gerichteter, gewichteter Graph ist gegeben durch $G = \langle V, E, c \rangle$ mit der Menge der Knoten, $E \subseteq V \times V$ ist die Menge der Bogen und $c \in \mathbb{R}^E$ stellt die Bogengewichte dar. Sei $e = (v, w) \in E$ ein Bogen von G . v und w heißen die Extremitäten von e , v stellt den Start- und w den Endknoten von e dar. Der Bogen e inzidiert mit v und w und führt von v nach w . Der Startknoten wird als $\text{tail}(e) = v$ und der Endknoten als $\text{head}(e) = w$ bezeichnet. Ein Bogen $e = (v, v)$ mit identischem Start- und Endknoten wird Schlinge genannt. Die Menge aller Bogen, welche aus einem Knoten $v \in V$ hinausführen bzw. in diesen hineinführen, wird bezeichnet mit $\delta^+(v) = \{(v, w) \in E\}$ bzw. $\delta^-(v) = \{(u, v) \in E\}$.

Ein Subgraph von G ist ein Graph $G' = \langle V', E', c' \rangle$ mit $V' \subseteq V$ und $E' \subseteq E$ wobei gelten muss, dass alle Extremitäten der Bogen aus E' in V' enthalten sind, d.h. $\{(v, w) \in E'\} \subseteq V' \times V'$. $c' \in \mathbb{R}^{E'}$ stellt dieselben Bogengewichte dar wie in G , d.h. $c'_e = c_e$ für alle $e \in E'$. Häufig wird an Stelle des reduzierten Vektors c' kurzerhand c geschrieben, d.h. $G' = \langle V', E', c \rangle$. Der durch eine Knotenteilmenge $V' \subseteq V$ induzierte Subgraph $G' = \langle V', E', c' \rangle$ enthält alle Bogen aus E welche beide Extremitäten in V' haben, d.h. $E' = \{(v, w) \in E : v, w \in V'\}$.

Ein Bogenzug in G ist eine alternierende Folge von Knoten und Bogen $P = (v_0, e_1, v_1, \dots, e_r, v_r)$ mit $r \geq 0$ sowie $v_i \in V$ für alle $i = 0, \dots, r$ und $e_i \in E$ für alle $i = 1, \dots, r$. Hierbei muss gelten, dass jeder Bogen vom vorangehenden zum nachfolgenden Knoten in P führt, d.h. $e_i = (v_{i-1}, v_i)$ für alle $i = 1, \dots, r$. P ist geschlossen, falls $r \geq 1$ und $v_r = v_0$, andernfalls ist P offen. P ist elementar, falls P offen ist und $v_i \neq v_j$ für alle $0 \leq i < j \leq r$ oder falls P geschlossen ist und $v_i \neq v_j$ für alle $0 \leq i < j \leq r-1$. Ein Weg ist ein elementarer Bogenzug. Ein Zyklus ist ein geschlossener, elementarer Bogenzug. Ein r -Bogenzug bzw. r -Weg, $r \geq 0$, ist ein Bogenzug bzw. Weg mit r Bogen. Ein r -Zyklus, $r \geq 1$, ist ein Zyklus mit r Bogen. Ein 1-Zyklus entspricht einer Schlinge.

Ein Bogenzug $P = (v_0, e_1, v_1, \dots, e_r, v_r)$ wird manchmal als Bogenfolge oder Knotenfolge dargestellt, d.h. $P = (e_1, e_2, \dots, e_r)$ bzw. $P = (v_0, v_1, \dots, v_r)$. Die Schreibweise $P \subseteq V$ bedeutet, dass $e_i \in E$ für alle $1 \leq i \leq r$. Analog bedeutet $P \subseteq V$ dass $v_i \in V$ für alle $0 \leq i \leq r$. Die Schreibweise $v \in P$ bzw. $e \in P$ drückt aus, dass v ein Knoten bzw. e ein Bogen von P ist, d.h. es existiert ein j , so dass $v = v_j$ bzw. $e = e_j$ ist. Sei $0 \leq j < k \leq r$ und $W = (v_j, e_{j+1}, v_{j+1}, \dots, e_k, v_k)$ ein Teilbogenzug von P . Dann wird bisweilen geschrieben $P = (v_0, e_1, v_1, \dots, e_j, v_j, W, v_k, e_{k+1}, v_{k+1}, \dots, e_r, v_r)$.

Die Länge (bzw. das Gewicht) eines Bogenzuges ist gegeben durch die Summe seiner Bogengewichte, d.h. $c(P) = \sum_{e \in P} c_e$. Ein Bogenzug mit positiver bzw. negativer Länge $c(P) > 0$ bzw.

$c(P) < 0$ wird ein positiver bzw. negativer Bogenzug genannt. Falls ein Graph einen (positiven bzw. negativen) Zyklus enthält, heisst er (positiv bzw. negativ) zyklisch, andernfalls (positiv bzw. negativ) azyklisch.

Weitere Bezeichnungen

Ein Vektor $c \in \mathbb{R}^E$ und eine Funktion $c : E \rightarrow \mathbb{R}$ werden häufig gleichwertig behandelt und die Schreibweisen $c_e = c(e)$, $e \in E$ als identisch betrachtet. Dasselbe gilt für einen Vektor $c \in \mathbb{R}^{I \times J}$ und eine Funktion $c : I \times J \rightarrow \mathbb{R}$ sowie die Schreibweisen $c_{ij} = c_{i,j} = c(i,j) = c((i,j))$. Ein Vektor $c \in \mathbb{R}^E$ wird manchmal als $(c_e \in \mathbb{R} : e \in E)$ dargestellt. Für eine Summe der Form $\sum_{e \in E} c_e$ wird auch $\sum \{c_e : e \in E\}$ geschrieben. Die Potenzmenge, d.h. die Familie aller Teilmengen, einer Menge I wird bezeichnet als $Pot(I)$.

LITERATURVERZEICHNIS

- Abdel-Hamid Abdel-Aziz, A. (1994), *Combinatorial Optimization Problems Arising in the Design and Management of an Automatic Storage System*, Dissertation, Fachbereich Mathematik, Technische Universität Berlin, Deutschland
- Adams, J., Balas, E., Zawack, D. (1988), *The Shifting Bottleneck Procedure for Job Shop Scheduling*, Management Science, Band 34, Nr. 3, S. 391 - 401
- Aggteleky, B. (1990), *Lagerplanung und logistische Optimierung, Prioritäten der neuzeitlichen Lagerplanung*, Fördertechnik, Nr. 3, S. 22 - 25, in [Kapoun, J. (1993 - 1997)]
- Andenmatten, K. (1992), *Leistungserfassung für ein mechanisiertes, computergesteuertes Blocklagersystem mit Maschinenbelegungsmodellen*, Diplomarbeit am Institut für Operations Research bei H. Gröflin, ETH Zürich, Schweiz
- Baker, K. R. (1974), *Introduction to Sequencing and Scheduling*, John Wiley & Sons, New York
- Balas, E. (1966), *Finding a Minimaximal Path in a Disjunctive PERT Network*, Théorie des Graphes, Journées internationales d'études, Rome, Juillet 1966, Dunod, Paris -Gordon and Beach, New York, 1967, S. 21 - 30
- Balas, E. (1967), *Discrete Programming by the Filter Method*, Operations Research, Nr. 15, S. 517 - 546
- Balas, E. (1969), *Machine Sequencing via Disjunctive Graphs: An Implicit Enumeration Algorithm*, Operations Research, Nr. 17, S. 941 - 957
- Balas, E. (1974), *Disjunctive Programming: Cutting Planes from Logical Conditions*, Nonlinear Programming 2, Academic Press, New York, S. 279 - 312
- Balas, E. (1979), *Disjunctive Programming*, Annals of Discrete Mathematics, Nr. 5, S. 3 - 51
- Beschaffung (1991), *Lagertechnik von der Stange*, Beschaffung, Nr. 1, S. 56 - 57, in [Kapoun, J. (1993 - 1996)]
- Beschaffung (1991a), *Das Lager als logistische Funktion*, Beschaffung, Nr. 11, S. 62 - 63, in [Kapoun, J. (1993 - 1997)]
- Birkholz, F. (1996), *Im Blocklager liegen Reserven, Layoutoptimierung durch Simulation*, Fördertechnik, Nr. 10, S. 28 - 30, in [Kapoun, J. (1993 - 1997)]
- Blazewicz, J., Ecker, K., Pesch, E., Schmidt, G., Weglarz, J. (1996), *Scheduling Computer and Manufacturing Processes*, Springer, Heidelberg
- Blazewicz, J. Pesch, E., Sterna, M. (2000), *Extension of Disjunctive Graph Model for Job Shop Scheduling Problem*, Institute of Computing Science, Poznan University of Technology, Poznan, Polen
- Blumenschein, K. (1991), *Fliesslager in der Logistikkette*, Automatisierte Lagersysteme '91, Anforderungen - Anpassung - Neustrukturierung, VDI Berichte 909, Verein Deutscher Ingenieure, VDI Verlag, Düsseldorf, Deutschland, S. 105 - 130
- Böckmann, H., Truskiewitz, G. (1991), *Lagersysteme im Technikvergleich*, F+H Fördern und Heben 41, Nr. 10, S. 769 - 770

- Bösl, R. (1989), *Materialfluss- und Lagersysteme mit hocheffizienter Automatisierungstechnik*, Automatisierte Lagersysteme '90, Anforderungen - Anpassung - Modernisierung, VDI Berichte 776, Verein Deutscher Ingenieure, VDI Verlag, Düsseldorf, Deutschland, S. 57 - 69
- Budde, R., Schwarz, W. (1988), *Kennzahlen zur Lager- und Fördertechnik*, Lager- und Transportlogistik, Verlag Industrielle Organisation des Betriebswirtschaftlichen Instituts der ETH Zürich, Schweiz, S. 87 - 98
- Carlier, J. (1982), *The One-Machine Sequencing Problem*, European Journal of Operations Research, Nr. 11, S. 42 - 47
- Chase, R. B., Aquilano, N. J. (1995), *Production and Operations Management*, 7th Edition, Irwin, Chicago
- Conway, R. W., Maxwell, W. L., Miller L. W. (1967), *Theory of Scheduling*, 1th Edition, Addison-Wesley, Massachusetts
- Cormier, G., Gunn, E. A. (1992), *A Review of Warehouse Models*, European Journal of Operational Research, Nr. 58, S. 3 - 13
- Corsten, H. (1996), *Produktionswirtschaft*, 6. Auflage, R. Oldenbourg Verlag, München
- Davis, M. D., Weyuker, E. J. (1983), *Computability, Complexity and Languages, Fundamentals of Theoretical Computer Science*, Academic Press, San Diego
- Dechter, R., Meiri, I., Pearl, J. (1991), *Temporal Constraint Networks*, Artificial Intelligence, Nr. 49, S. 61 - 95
- Dilworth, J. B. (1993), *Production and Operations Management: Manufacturing and Services*, 5th Edition, McGraw-Hill
- Engeler, E., Läuchli, P. (1988), *Berechnungstheorie für Informatiker*, Teubner, Stuttgart
- Esquirol, P., Huguet, M. J., Lopez, P. (1992), *Time-Bound-On-Node Graph for Task Scheduling*, LAAS Report Nr. 92303, October 1992, Centre National de la Recherche Scientifique, Laboratoire d'Automatique et d'Analyse des Systèmes, Toulouse, France
- Favarger, M., Bamert, A., Rupper, P. (1982), *Blocklager oder Reihenlager?* Lagerlogistik, Verlag Industrielle Organisation des Betriebswirtschaftlichen Instituts der ETH Zürich, Schweiz, S. 85 - 96
- Fischer, J. F., Thompson, G. L. (1963), *Industrial Scheduling*, Prentice-Hall, Englewood Cliffs, NJ
- FM (1990), *"Das schnelle automatische Blocklager" arbeitet nach dem Prinzip "Ware zum Mann"*, FM, Oktober, S. 36, in [Kapoun, J. (1993 - 1996)]
- Frank, P. (1997), *Der Lagersteuerrechner macht's möglich*, Technica, Nr. 4, S. 21 - 25
- Frauchiger, R. (1989), *Integrierte Lagerstrukturplanung mit einer Heuristik zur Minimierung des Kommissionierweges*, Dissertation ETH Nr. 8868, Eidgenössisch Technische Hochschule Zürich, Schweiz
- Garey, M. R., Johnson D. S. (1979), *Computers and Intractability, A Guide to the Theory of NP-Completeness*, W. H. Freeman and Company, New York
- Gillies, D. W., Liu, J. W.-S. (1995), *Scheduling Tasks with AND/OR Precedence Constraints*, SIAM J. Comput., Band 24, Nr. 4, S. 797 - 810
- Gödicke, D. (1991), *Dynamisches Kompaktlagersystem für die Warenverteilung mit hoher Leistung*, Automatisierte Lagersysteme '91, Anforderungen - Anpassung - Neustrukturierung, VDI Berichte 909, Verein Deutscher Ingenieure, VDI Verlag, Düsseldorf, S. 93 - 103

- Grafe, W. (1992), *Entwicklung eines neuen Hochleistungs-Lagersystems*, Fördertechnik, Nr. 7, S. 20 - 23, in [Kapoun, J. (1993 - 1997)]
- Graham, R. L., Lawler, E. L., Lenstra, J. K., Rinnooy Kan, A. H. G. (1979), *Optimization and Approximation in Deterministic Sequencing and Scheduling: A Survey*, Annals of Discrete Mathematics, Nr. 5, S. 287 - 326
- Gremm, F. (1989), *Neue Lagerstrukturen für den automatischen Betrieb*, FM, Juni, S. 14 - 18, in [Kapoun, J. (1993 - 1997)]
- Greenberg, H. H. (1968), *A Branch-Bound Solution to the General Scheduling Problem*, Operations Research, Jan - Feb 1968, S. 353 - 361
- Gysi, R. (1982), *Was ist Logistik?* Lagerlogistik, Verlag Industrielle Organisation des Betriebswirtschaftlichen Instituts der ETH Zürich, Schweiz, S. 9 - 16
- Heiner, H. (1991), *Entwicklung in der Lagerlogistik*, in Management Zeitschrift, 60, Nr. 11, S. 75 - 80, in [Kapoun, J. (1993 - 1997)]
- Helbig, S. (1994), *Duality in Disjunctive Programming via Vector Optimization*, Mathematical Programming, Nr. 65, S. 21 - 41
- Hollenstein, E. (1996), *Das dynamische Blocklager - wirtschaftlich und flexibel*, Einkauf/ Materialwirtschaft/Logistik, Nr. 1/2, S. 8 - 11, in [Kapoun, J. (1993 - 1997)]
- Hollenstein, E. (1997), *Technische und ökonomische Anforderungsprofile an Lagersysteme*, Einkauf/ Materialwirtschaft/Logistik, Nr. 1/2, Januar/Februar, S. 16 - 17, in [Kapoun, J. (1993 - 1997)]
- Huguet, M.-J. (1994), *Approche par Contraintes pour l'Aide à la Décision et à la Coopération en Gestion de Production*, Thèse 344, Institut National des Sciences Appliquées de Toulouse, France
- Hürlimann, T. (2001), *Reference Manual for the LPL Modeling Language, Version 4.42*, Working Paper 01-19, June, Department of Informatics, University of Fribourg, Switzerland
- Ingold, Th. (1992), *Steuerung eines Verarbeitungssystems mit Verschiebungsbrücken*, Diplomarbeit am Institut für Operations Research bei H. Gröflin, ETH Zürich, Schweiz
- Jansen, R. (1990/91), *Die Bedeutung der Lagertechnik innerhalb der Logistik*, ?, S. 6 - 9, in [Kapoun, J. (1993 - 1996)]
- Jewell, W. S. (1967), *A Simple Proof of: $L = \lambda W$* , Operations Research, S. 1109 - 1116
- Jorichs, H. (1988), *Ermittlung von Planungsdaten zur Lagersystembestimmung*, Fördertechnik, Nr. 1, S. 23 - 27, in [Kapoun, J. (1993 - 1997)]
- Kahrman, V. (1989), *So weit wie nötig, so eng wie möglich, Vom Blockstapel zum dynamischen Blocklager*, Lagertechnik '89, S. 64 - 66, in [Kapoun, J. (1993 - 1996)]
- Kapoun, J. (1993 - 1996), *Moderne Lagertechnik der 90er Jahre, Ausgewählte deutsche und schweizerische Artikel und Firmeninformationen über innovative und leistungsstarke Lager verschiedener Typen und Einsatzbestimmungen*, Private Monographie, Lausanne und Fribourg, Schweiz
- Kapoun, J. (1993 - 1997), *Konzipierung, Gestaltung und Realisierung komplexer Lager: Systematische Vorgehensverfahren, Methoden, Instrumente, wichtige Faktoren und Aspekte, praktische Beispiele, Dargestellt in ausgewählten schweizerischen und deutschen Fachbeiträgen*, Private Monographie, Lausanne und Fribourg, Schweiz
- Kistler, J. (1996), *Die Lager werden kleiner, kompakter und raffinierter*, Technica, Nr. 17, S. 33 - 39

- Klinkert, A. (1992), *Optimierte Steuerung des Auslagerungsprozesses in Blocklagern bei notwendigen Umlagerungen*, Diplomarbeit am Institut für Operations Research bei H. Gröflin, ETH Zürich, Schweiz
- Lenstra, J. K., Rinnooy Kan, A. H. G., Brucker, P. (1977), *Computational Complexity of Discrete Optimization Problems*, Annals of Discrete Mathematics, Nr. 4, S. 121 - 140
- Little, J. D. C. (1960), *A Proof for the Queuing Formula $L = \lambda W$* , Operations Research, S. 383 - 387
- Logistik im Unternehmen (1989), *Dynamisches Konzept für Reifenlager realisiert*, Logistik im Unternehmen, September, S. 82 - 87, in [Kapoun, J. (1993 - 1996)]
- Logistik im Unternehmen (1991), *Automatische Lager helfen die Produktionslogistik optimieren, Lagerlogistik*, Logistik im Unternehmen, Mai, S. 92 - 93, in [Kapoun, J. (1993 - 1996)]
- Ludwig, S., Malloth, C. (1990), *Leistungserfassung für ein mechanisiertes, computergesteuertes Blocklagersystem*, Semesterarbeit am Institut für Operations Research bei H. Gröflin, ETH Zürich, Schweiz
- Martin, H. (1979), *Materialfluss- und Lagerplanung, Planungstechnische Grundlagen, Materialflusssysteme, Lager- und Verteilsysteme*, Springer-Verlag, Berlin
- Martin, H. (1990), *Grundlagen der Lagerlogistik*, Fördertechnik, Nr. 6, S. 28 - 35, in [Kapoun, J. (1993 - 1997)]
- Martin, H. (1991), *Palettenregale - Komponente des Lagersystems*, JJ, S. 11 - 17 in [Kapoun, J. (1993 - 1996)]
- Morton, Th. E., Pentico, D. W. (1993), *Heuristic Scheduling Systems*, John Wiley & Sons, New York
- Nemhauser, G. L., Wolsey, L. A. (1988), *Integer and Combinatorial Optimization*, 1st Edition, John Wiley & Sons, New York
- Neue Verpackung, (1990), *Neues Blocklager sorgt für Ordnung*, Neue Verpackung, Nr. 11, S. 76 - 77, in [Kapoun, J. (1993 - 1996)]
- Ostermann, H. (1989), *Energiekosten bei automatisierten Materialfluss- und Lagersystemen, Materialfluss- und Lagersysteme mit hocheffizienter Automatisierungstechnik*, Automatisierte Lagersysteme '90, Anforderungen - Anpassung - Modernisierung, VDI Berichte 776, Verein Deutscher Ingenieure, VDI Verlag, Düsseldorf, S. 169 - 184
- Ovacik, I. M., Uzsoy, R. (1997), *Decomposition Methods for Complex Factory Scheduling Problems*, Kluwer Academic Publishers, Dordrecht
- Papadimitriou, Ch. H. (1994), *Computational Complexity*, Addison Wesley, Massachusetts
- Parrott, R. A. (1992), *Automated Space Planning for Warehouse Design*, Proceedings of the 12th International Conference on Automation in Warehousing, New Orleans, Louisiana, USA, Institute of Industrial Engineers, Norcross, USA
- Pieper-Musiol, R. (1988), *Systematik zum Finden des bestgeeigneten Kommissioniersystems*, Lager- und Transportlogistik, Verlag Industrielle Organisation des Betriebswirtschaftlichen Instituts der ETH Zürich, Schweiz, S. 153 - 161
- Pinson, E. (1995), *The Job Shop Scheduling Problem: A Concise Survey and Some Recent Developments*, Scheduling Theory and its Applications, Edited by Chrétienne, Ph., Coffman, E. G. Jr., Lenstra, J. K., Liu, Z., John Wiley & Sons, Chichester, Kapitel 13, S. 277 - 293
- Pörsch, M. (1988), *Lagertypen im Nutzwertvergleich*, Lager- und Transportlogistik, Verlag Industrielle Organisation des Betriebswirtschaftlichen Instituts der ETH Zürich, Schweiz, S. 123 - 152

- Reichert, H. (1989), *Schöne Aussichten, Trends bei Automatiklagern*, Techno-Tip, Jg. 19, Nr. 7, Juli, S. 44 - 46, in [Kapoun, J. (1993 - 1996)]
- Rinza, T. (1996), *Lagerkapazität optimal nutzen*, Zeitschrift für Logistik, Nr. 3/96, S. 12 - 14, in [Kapoun, J. (1993 - 1996)]
- Roy, B., Sussman, B. (1964), *Les problèmes d'ordonnancement avec contraintes disjonctives*, SEMA, Note D.S. No. 9 bis
- Schardt, W. (1991), *Automatisches Kompaktlager für die Just-in-time-Auslieferung eines Zulieferbetriebes*, Automatisierte Lagersysteme '91, Anforderungen - Anpassung - Neustrukturierung, VDI Berichte 909, Verein Deutscher Ingenieure, VDI Verlag, Düsseldorf, Deutschland, S. 131 -146
- Scheid, W.-M. (1996), *Richtlinien allein sind keine Garantie, Kleines Vademekum der Lagerlogistik*, Fördertechnik, Nr. 3, S. 12 - 15, in [Kapoun, J. (1993 - 1996)]
- Schneeweiss, Ch. (1993), *Einführung in die Produktionswirtschaft*, 5. Auflage, Springer-Verlag, Berlin
- Schneider, S. (1988), *Kompaktierung im VLSI-Design*, Diplomarbeit am Institut für Operations Reserach bei H. Gröflin, ETH Zürich, Schweiz
- Schulte-Zurhausen, M. (1982), *Planung von Blocklagersystemen*, Dissertation, Technische Hochschule Aachen, Beuth Verlag GmbH, Berlin, Köln, Deutschland
- Schulze, L. (1990), *Das Lager - Leistungsfaktor oder notwendiges Übel?, Rationalisierungspotentiale durch integrierte Lagerlogistik*, Zeitschrift für Logistik, S. 5 - 16, in [Kapoun, J. (1993 - 1997)]
- Schulze, L., Martin, H. (1990), *Materialfluss ABC, Fachwortverzeichnis Lagern*, Jungheinrich Journal, Jungheinrich GmbH, Schinhuettstr. 12, CH-5036 Oberentfelden und 4, rue du Simplon, CH-1800 Vevey, Schweiz, in [Kapoun, J. (1993 - 1996)]
- Schulze, L. (1995), *Logistik und Lagersysteme*, Fördertechnik, Nr. 6, S. 3 - 12, in [Kapoun, J. (1993 - 1997)]
- Stidham Jr., S. (1972), $L = \lambda W$: A Discounted Analogue and a New Proof, Operations Research, Band 20, Nr. 6, S. 1115 - 1125
- Tanaev, V. S., Sotskov, Y. N., Strusevich, V. A. (1994), *Scheduling Theory, Multi-Stage Systems*, Kluwer Academic Publishers, Dordrecht
- Tempelmeier, H. (1992), *Material-Logistik*, Springer-Verlag, Berlin
- Transport, Förder- und Lagertechnik (1987), *Erfahrungen mit automatischen Lagern*, Transport, Förder- und Lagertechnik, Nr. 11, S. 9 - 12, in [Kapoun, J. (1993 - 1997)]
- Tübergen, F. (1982), *Wirtschaftliches Lagern und Kommissionieren*, Lagerlogistik, Verlag Industrielle Organisation des Betriebswirtschaftlichen Instituts der ETH Zürich, Schweiz, S. 114 - 120
- VDI (1996), *Spielzeitermittlung von automatischen Kanallager-Systemen, VDI-Richtlinie 3561, Blatt 4, Entwurf*, VDI-Handbuch Materialfluss und Fördertechnik, Band 5, Verein Deutscher Ingenieure, VDI-Gesellschaft Fördertechnik Materialfluss Logistik, zu beziehen bei Beuth Verlag GmbH, Berlin, Deutschland
- Vogt, G. (1991), *Intelligente Lager*, Materialfluss, November, S. 70 - 72, in [Kapoun, J. (1993 - 1997)]
- Vogt, G. (1996), *Lagerplanung*, Sonderpublikation der Zeitschrift MATERIALFLUSS, Verlag Moderne Industrie AG, D-86895 Landsberg

- Weber, R. (1990), *Neue Wege zum "alten" Blocklager*, Transport-, Förder- und Lagertechnik, Nr. 1/2, S. 14 - 15, in [Kapoun, J. (1993 - 1996)]
- Wenz, R. (1990), *Umdenken ist erforderlich*, Fördertechnik, Nr. 2, S. 24 - 28, in [Kapoun, J. (1993 - 1997)]
- Westfal, U. (1990), *Moderne Lagersysteme in Industrie und Handel*, Logistik im Unternehmen, Januar/Februar, S. 54 - 56, in [Kapoun, J. (1993 - 1996)]
- White Jr., K. P., Rogers, R. V. (1990), *Job-Shop Scheduling: Limits of the Binary Disjunctive Formulation*, International Journal of Productions Research, Band 28, Nr. 12, S. 2187 - 2200

DANKSAGUNG

In der Zeit der Entstehung dieser Arbeit haben mich viele Personen auf fachlicher und menschlicher Ebene unterstützt. Dafür möchte ich mich bei allen herzlich bedanken.

Als erster Stelle möchte ich meine grosse Dankbarkeit gegenüber Herrn Professor H. Gröflin zum Ausdruck bringen. Er hat mir die Möglichkeit geboten, in seinem Forschungsteam zu arbeiten und unter seiner Leitung diese Dissertation zu verfassen. Durch ihn erfuhr ich eine fundierte Ausbildung auf dem Gebiet des Operations Research und entdeckte meine Faszination an diesem Fach. Seine Intuition eröffnete mir die Türen zu dieser Forschungsarbeit. Mit viel Enthusiasmus, Scharfsinn, Geduld und Vertrauen hat er mich bei meiner Forschung unterstützt und geleitet. Ohne sein Wissen, seine Erfahrung und seine Inspiration wäre diese Arbeit nicht zu Stande gekommen.

Weiter bedanke ich mich bei Herrn Professor Th. M. Liebling für die Übernahme des Koreferats, für sein Interesse an dieser Arbeit, seine wohlwollende Unterstützung und seine wertvollen fachlichen Kommentare.

Ich habe es sehr geschätzt, am Institut für Informatik arbeiten zu können und möchte mich bei allen Mitarbeitern für die Zusammenarbeit bedanken. Insbesondere richtet sich mein Dank an Thomas Ingold, Tony Hürlimann und Marino Widmer für ihre vielfältige Unterstützung, die hilfreichen Diskussionen sowie den Austausch auf fachlicher und persönlicher Ebene.

Ich möchte diese Gelegenheit auch nutzen, meine Dankbarkeit gegenüber Géraldine und Kurt Sager auszudrücken, welche mich bei meinen Aufenthalten in Fribourg über all die Jahre hinweg mit ihrer Herzlichkeit und Gastfreundschaft beschenkt haben.

Tiefe Dankbarkeit verbindet mich mit meinen Eltern im Bewusstsein der wunderbaren Möglichkeiten, welche sie mir in meinem Leben eröffnet haben.

Schliesslich danke ich von Herzen meiner Frau Priska und meinem Sohn Jonas für ihre Präsenz, Lebendigkeit, Geduld und Liebe, mit welcher sie mich auf meinem Weg hierher begleitet haben.

CURRICULUM VITAE

Personalien

Name: Klinkert
Vorname: Andreas
Geburtsdatum: 14. Mai 1961
Heimatort: Lütisburg SG

Ausbildung

1974 - 1980 Kantonsschule Freudenberg Zürich
1980 Eidgenössische Maturität, Typus A
1983 - 1984 Seminar für Pädagogische Grundausbildung (SPG) des Kantons Zürich
1984 - 1985 Real- und Oberschullehrerseminar (ROS) des Kantons Zürich
1985 - 1992 Studium der Informatik an der Abteilung IIIc der Eidgenössisch Technischen Hochschule (ETH) in Zürich
1992 Diplom als Informatik-Ingenieur ETH
1993, 1995 Nachdiplomstudium (Cours postgrade) in Diskreter Optimierung, Ecole Polytechnique Fédérale (EPF) de Lausanne und Université de Grenoble, France

Berufstätigkeit

1980 - 1983 Freiberufliche Tätigkeit im handwerklichen Bereich
1992 Projektarbeit im Bereich Akustische Spracherkennung
1992 - 1998 Assistent am Institut für Informatik der Universität Freiburg, Schweiz, bei Prof. Dr. H. Gröflin, Fachgruppe Manufacturing Systems, Fachgebiet Operations Research and Decision Support
1999 Freier Mitarbeiter am Institut für Informatik der Universität Freiburg, Schweiz
2000 - 2001 Selbständige Projektstätigkeit in den Bereichen Akustische Spracherkennung und Bilddatenbanken

ANHANG

Beispiele von Ablaufplänen

Beispiel 1

- Lagerdimensionen: 3 Sektoren, 3 Ebenen, 5 Korridore/Sektor, 20 Lagerplätze/Korridor
- Jobs: 50 Einlagerungen
- Makespan: Startlösung 1831, optimierte Lösung 897
- Optimierungsfaktor: 51%

Beispiel 2

- Lagerdimensionen: 5 Sektoren, 3 Ebenen, 5 Korridore/Sektor, 20 Lagerplätze/Korridor
- Jobs: 100 Einlagerungen
- Makespan: Startlösung 3313, optimierte Lösung 1378
- Optimierungsfaktor: 58%



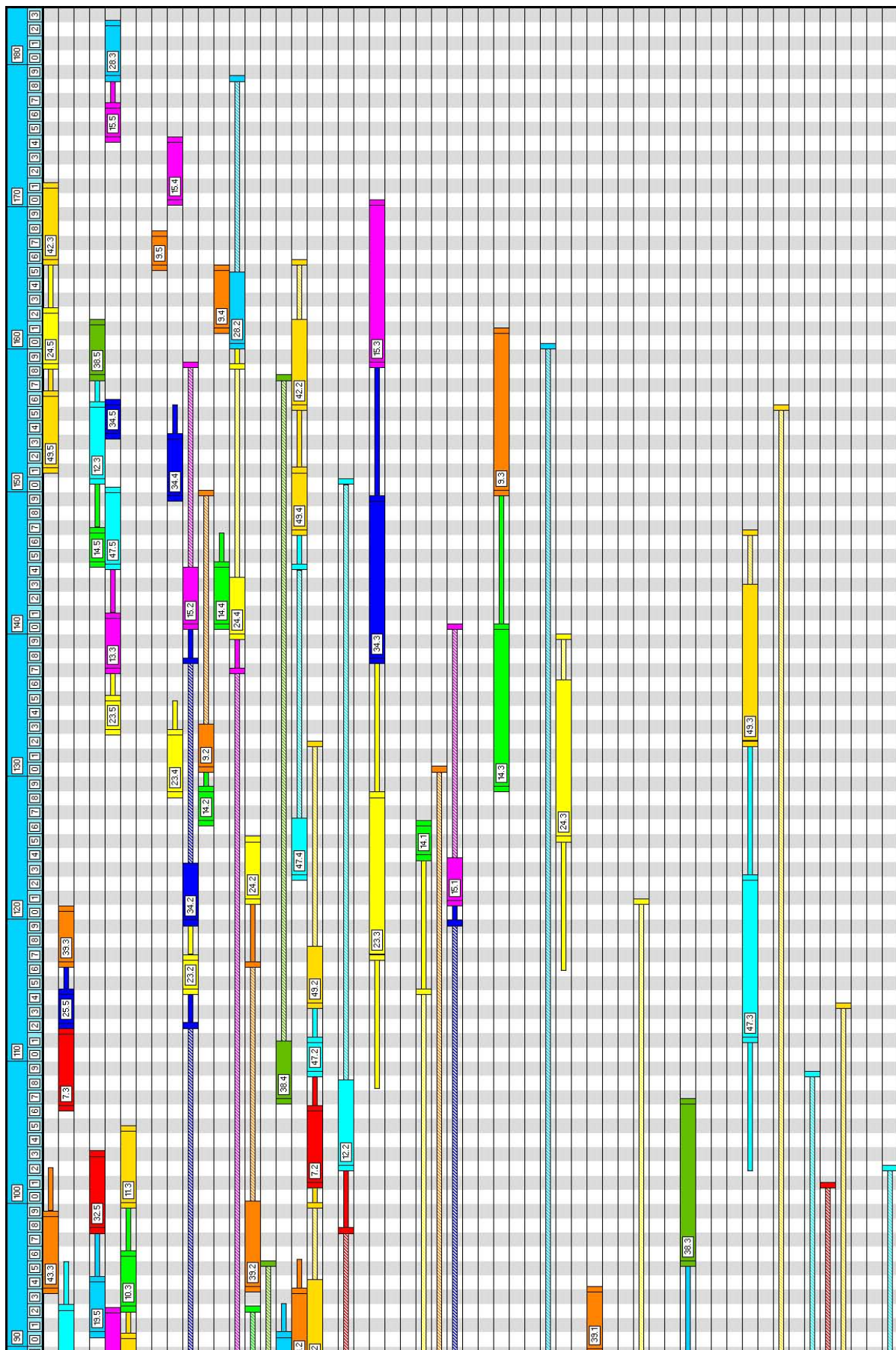


Abbildung A.2. Beispiel 1: Startlösung, Teil 2 (Makespan 1831)

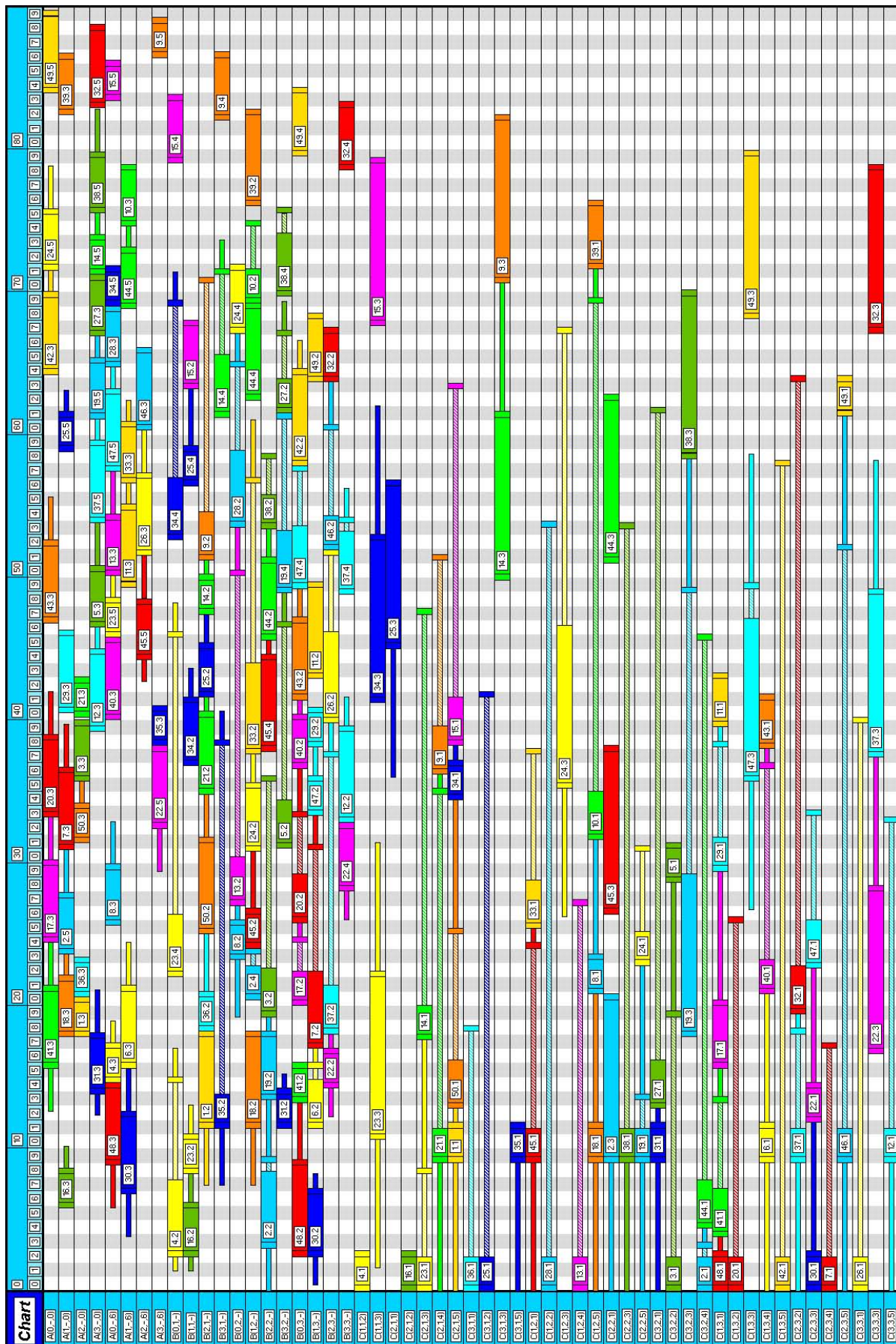


Abbildung A.3. Beispiel 1: Optimierte Lösung (Makespan 897)

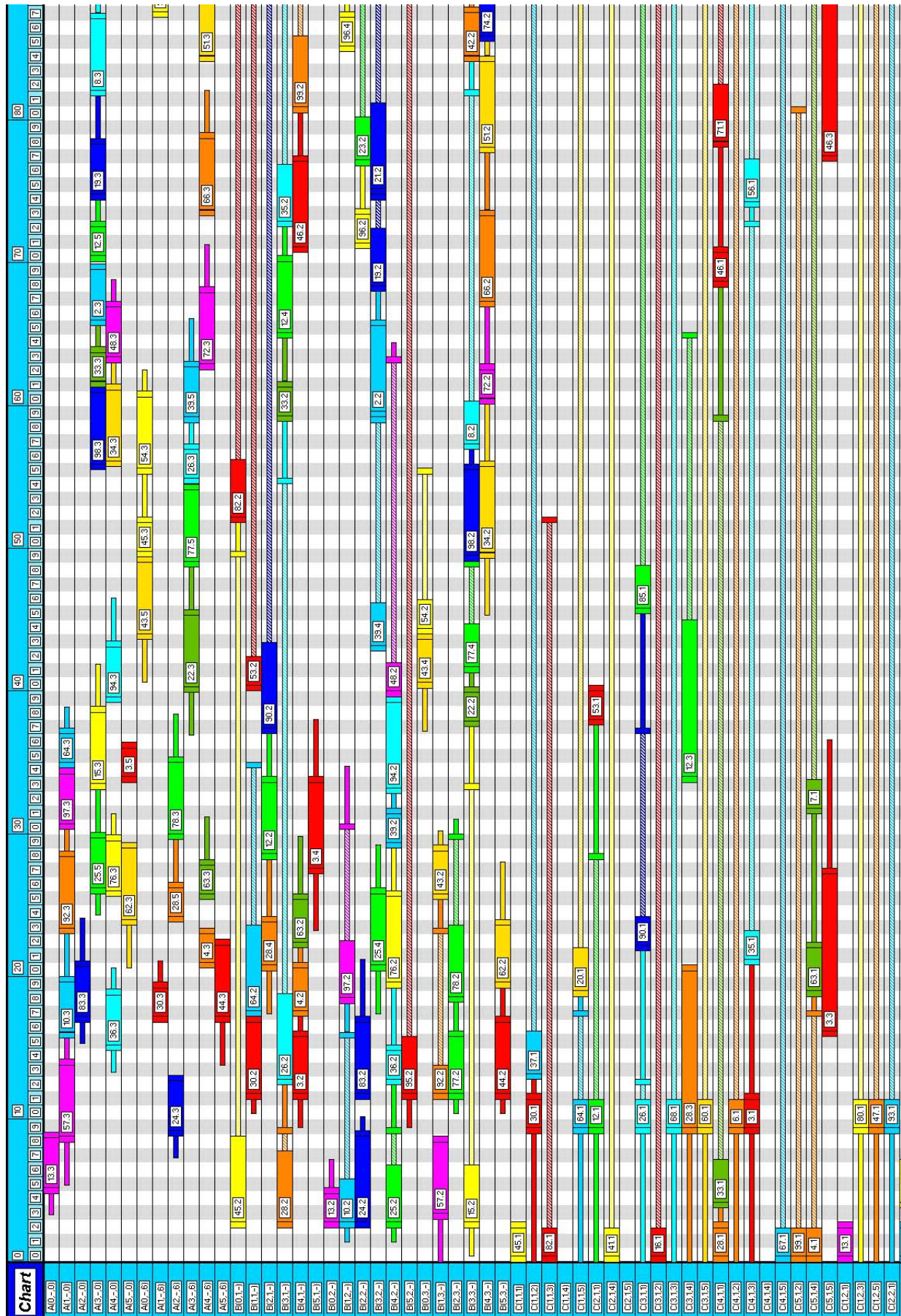


Abbildung A.4. Beispiel 2: Startlösung, Teil 1 (Makespan 3313)

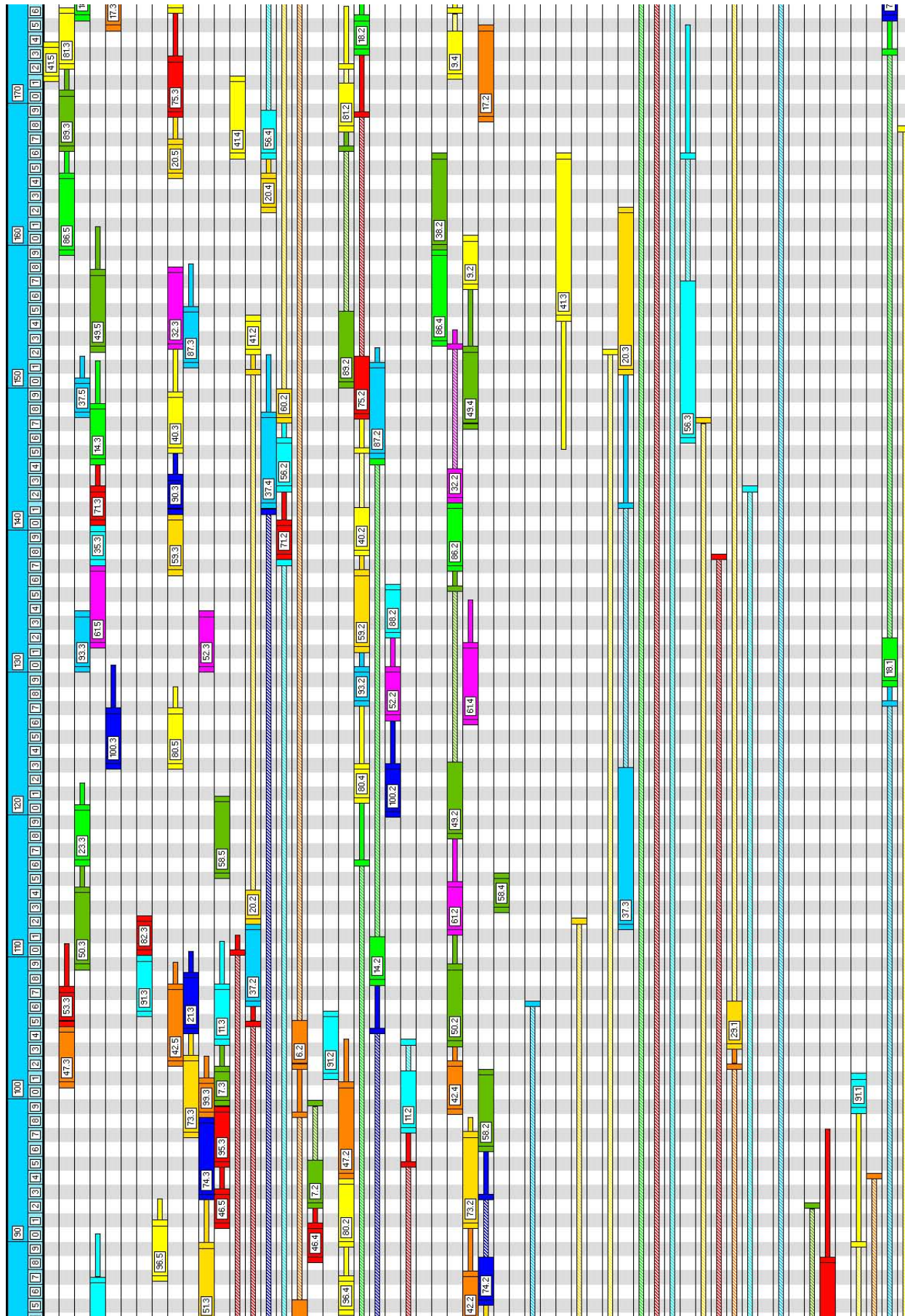


Abbildung A.5. Beispiel 2: Startlösung, Teil 2 (Makespan 3313)

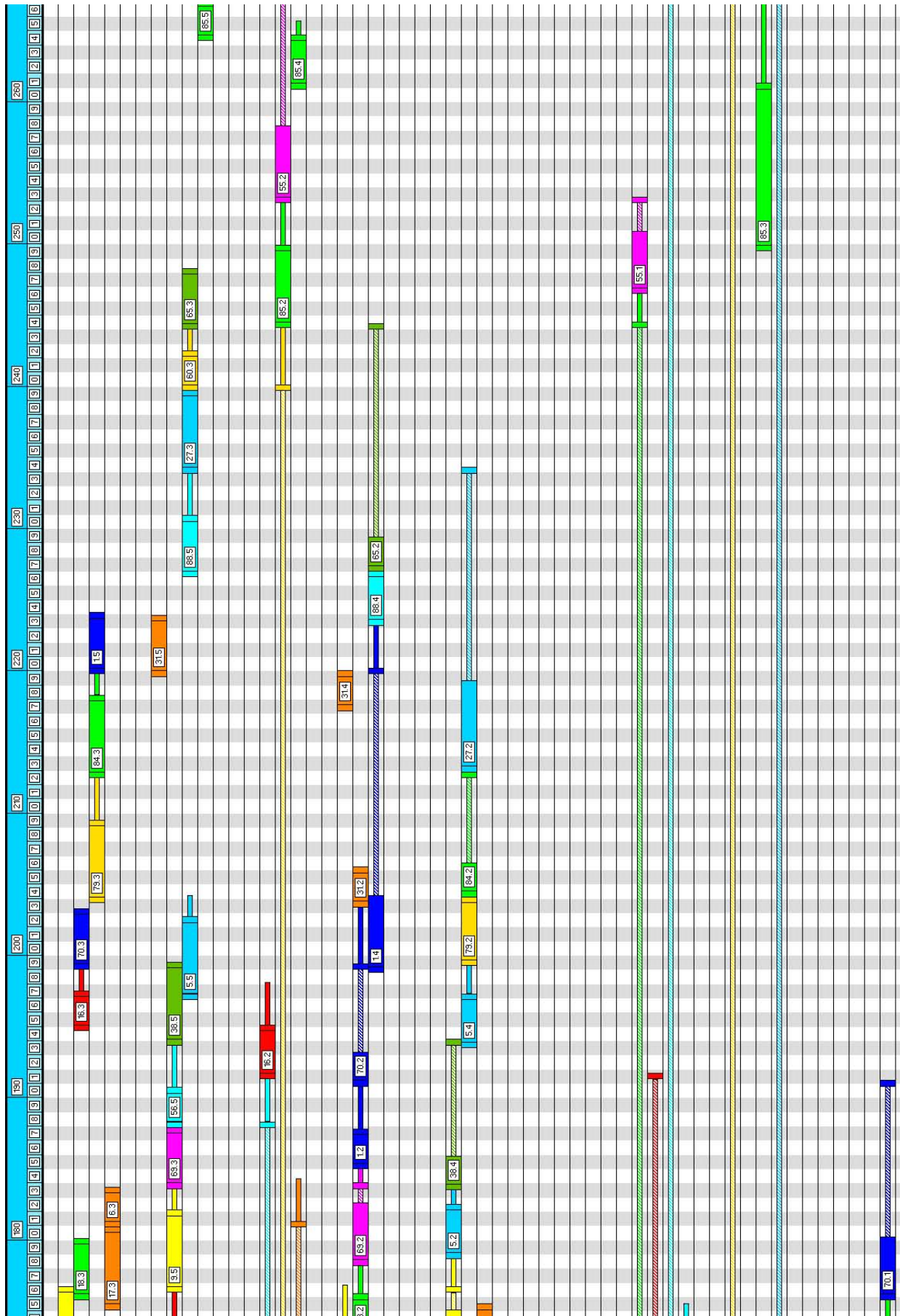


Abbildung A.6. Beispiel 2: Startlösung, Teil 3 (Makespan 3313)

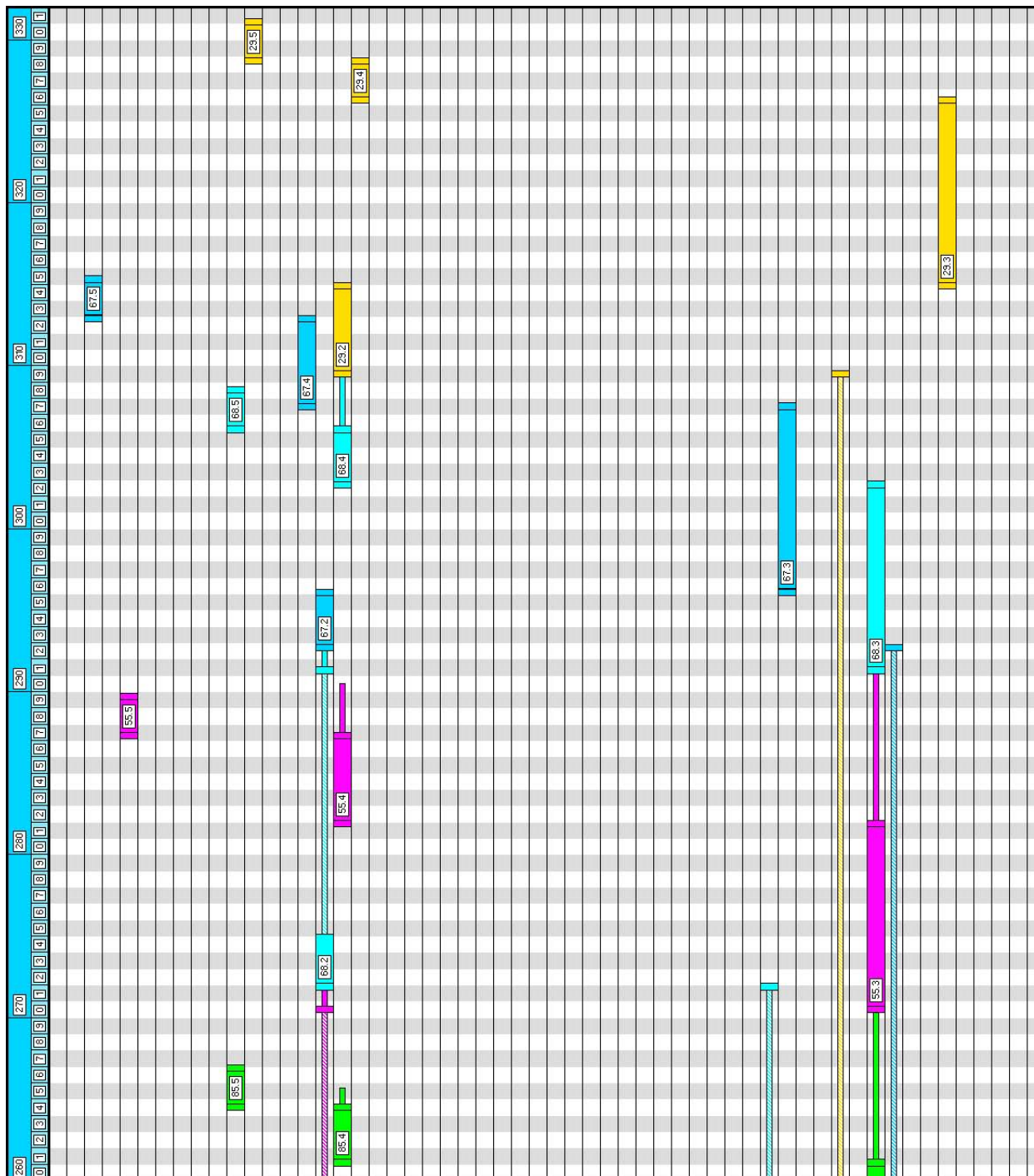
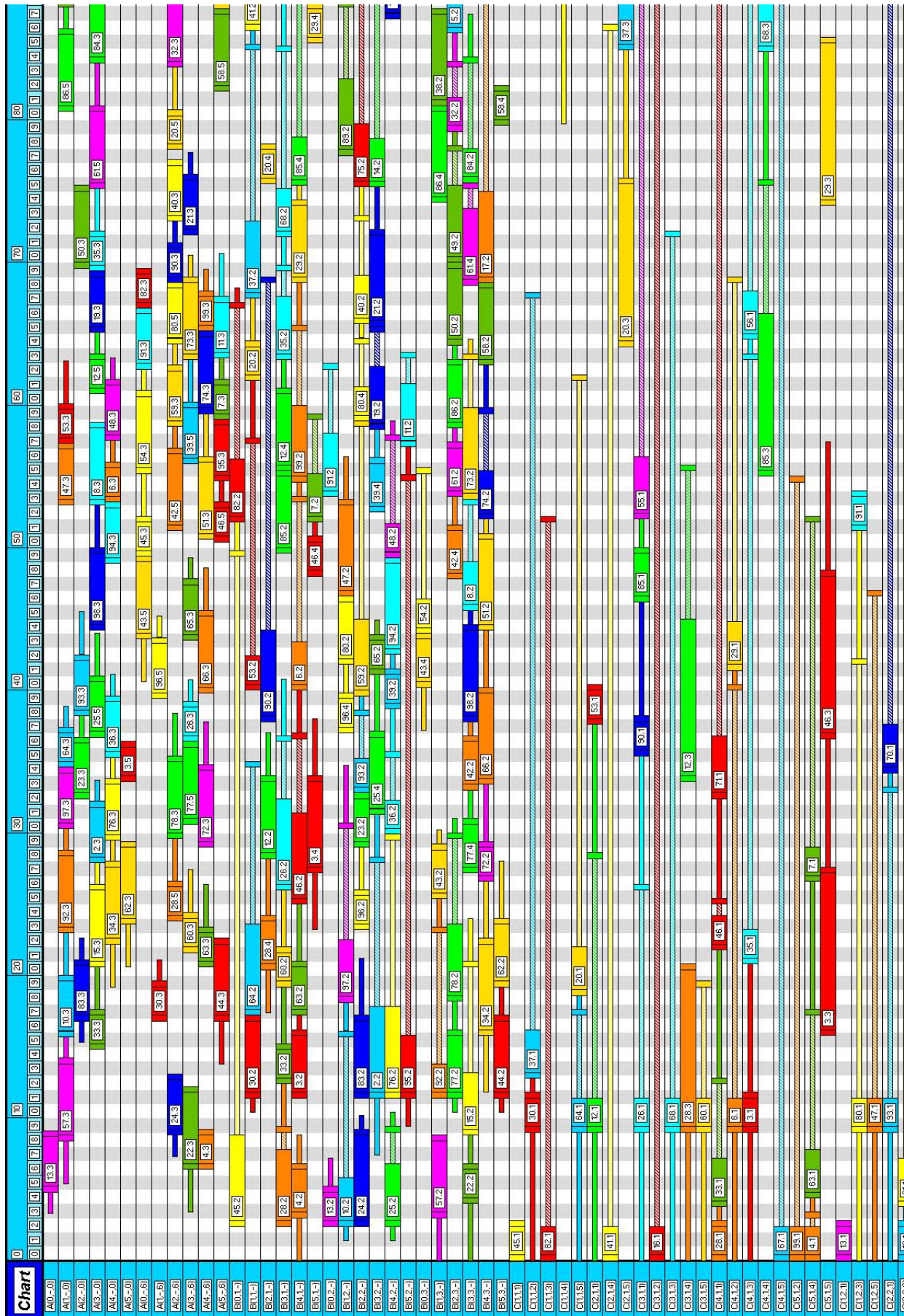


Abbildung A.7. Beispiel 2: Startlösung, Teil 4 (Makespan 3313)



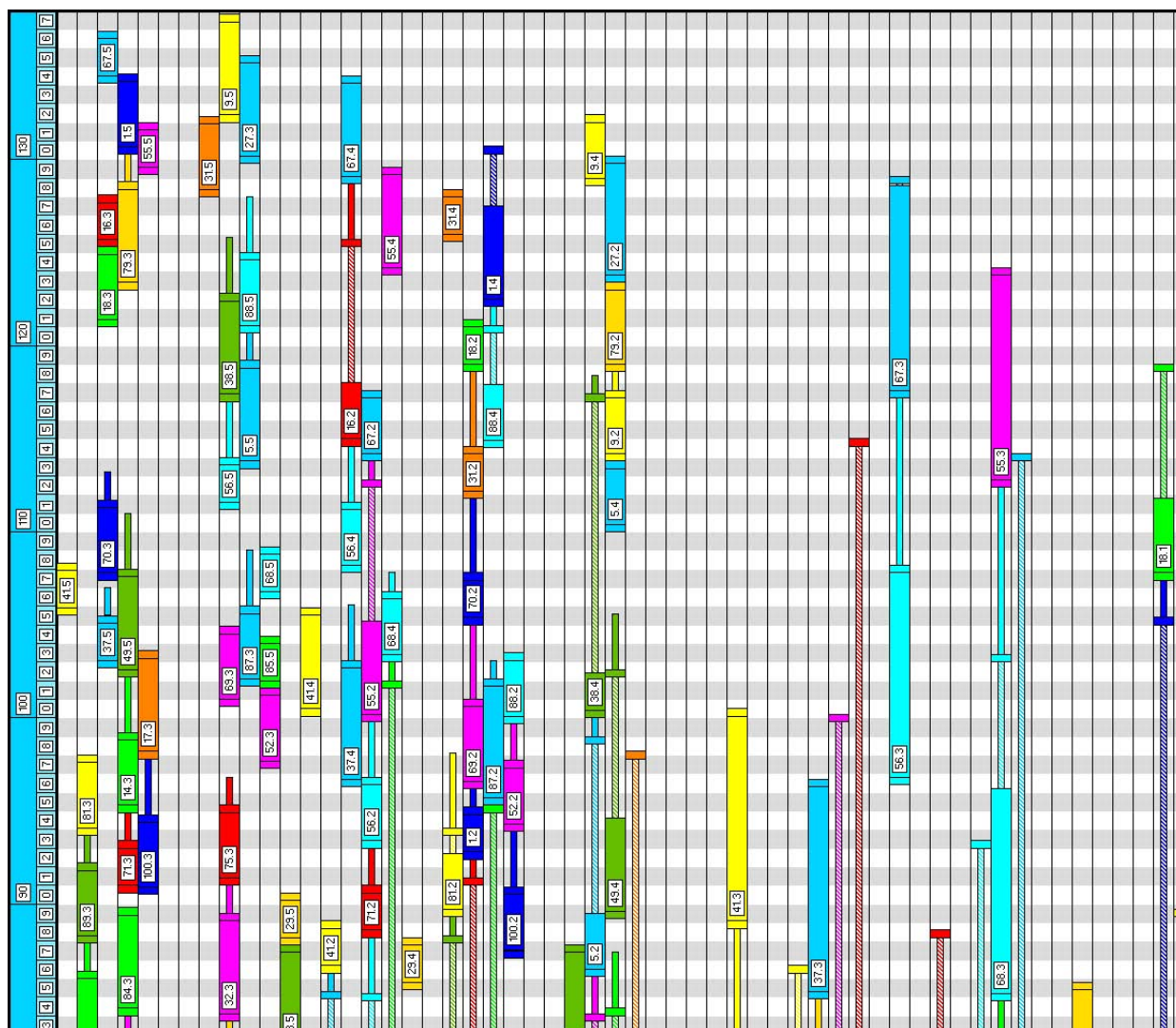


Abbildung A.9. Beispiel 2: Optimierte Lösung, Teil 2 (Makespan 1378)