Stochastisches Scheduling – Algorithmen und Polyedrische Methoden

Diplomarbeit

bei Prof. Dr. R. H. Möhring

Vorgelegt von Marc Uetz

am Fachbereich Mathematik der

Technischen Universität Berlin

Berlin, November 1996

Inhaltsverzeichnis

Einleitung **1**

1 Stochastisches Scheduling mit geplanten Leerzeiten **5**

1.1 Ein Beispiel 5

1.2 Stochastisches Scheduling - Das Modell 6

1.3 Strategien 10

1.3.1 Der Zusammenhang der Verteilungen 13

1.3.2 Optimalität von Strategien 13

1.3.3 Spezielle Klassen von Strategien 16

1.4 Exponentielle Schedulingprobleme und Set-Strategien 18

1.5 Scheduling mit geplanten Leerzeiten: Das Deliberate Idleness - Problem . . 24

1.5.1 Deliberate Idleness – Ein Beispiel 26

1.5.2 2-Maschinen Probleme und Deliberate Idleness 30

1.5.3 Optimalitätsresultate über statische Prioritätsstrategien 36

1.5.4 $\sum_{i \in I} w_i C_i$: Die Notwendigkeit dynamischer Strategien 38

1.5.5 $\sum_{i \in I} w_i C_i$: Deliberate Idleness vs. Greedyness 40

2 Polyedrische Methoden für Stochastisches Scheduling **47**

2.1 Polyeder und Polytope 49

2.2 Lineare Optimierung und Dualität 53

2.3 Spezielle polyedrische Strukturen 56

2.3.1 Super- und Submodularität 57
2.3.2 Super- und Submodulare Polyeder, Lineare Transformationen 59
2.3.3 Erweiterte Polymatroide und Contra-Polymatroide 60
2.3.4 Basispolytope und Optimierung 65
2.4 Der erweiterte Greedy-Algorithmus 69
2.5 Anwendungen im Stochastischen Scheduling 76
2.5.1 „Erhaltungssätze“ für stochastisches Scheduling 78
2.5.2 Die Optimalität der Ratio Rule im stochastischen Fall 80
2.5.3 Mehrmaschinenmodelle und weitere Anwendungen 83

3 Approximationsheuristiken für Stochastisches Scheduling 85

3.1 Gütegarantien für Heuristiken auf **LP**-Basis – Genereller Ansatz 86
3.2 Stochastisches Scheduling auf einer Maschine 88
3.2.1 Gültige Ungleichungen 88
3.2.2 Präzedenzrelationen 89
3.2.3 Release Dates 91
3.3 Stochastisches Scheduling auf mehreren Maschinen 94
3.3.1 Gültige Ungleichungen 95
3.3.2 Heuristiken ohne Berücksichtigung von Release Dates 100
3.3.3 Eine Heuristik unter Berücksichtigung von Release Dates 106
3.4 Zur polynomialen Lösbarkeit der **LP**-Relaxierungen 108
3.5 Zusammenfassung der Resultate 115

Symbolverzeichnis 116

Literaturverzeichnis 117

Einleitung

Bei der Durchführung von *Projekten* ist in der Regel die zeitliche Allokation einzelner *Vorgänge* mitentscheidend für den Gesamterfolg. Typischerweise entsteht ein *Planungsbedarf* bei solchen Projekten, bei denen Knappheit gewisser Ressourcen und gegebenenfalls zusätzlich Interdependenzen zwischen den einzelnen Vorgängen zu berücksichtigen sind. Ein Beispiel für ein derartiges Projekt wird in Abschnitt 1.1 gegeben.

Als Zweig des *Operations Research* beschäftigt sich das mathematische Forschungsgebiet des *Scheduling* mit einer Vielzahl unterschiedlicher Modelle zur Untersuchung derartiger Planungssituationen.

Die „klassischen" Arbeiten auf dem Gebiet des Scheduling sind [Joh 54], [Jac 55] und [Smi 56]. Sie behandeln *deterministische* Schedulingprobleme, die sich dadurch auszeichnen, daß sämtliche Parameter eines Projektes *fest* und im voraus bekannt sind. Einen ausführlichen Überblick der verschiedenen Modelle des deterministischen Scheduling liefert beispielsweise [Law 82]. Im Hinblick auf die mathematische Modellierung praktischer („*real-world*") Probleme ist jedoch im Regelfall *Unsicherheit* über die Zukunft eines Projektes zu berücksichtigen. Insbesondere sind die Dauern einzelner Vorgänge meist nicht im voraus bekannt.

Gegenstand der vorliegenden Arbeit ist ein Modell, das dieser Unsicherheit dadurch Rechnung trägt, daß die *Dauern* der einzelnen Vorgänge nur noch über eine Wahrscheinlichkeitsverteilung beschrieben werden.

Dieser *stochastische* Rahmen führt zu einer Fülle zusätzlicher Schwierigkeiten. Unter anderem wird eine neue Definition der *Optimalität* einer Lösung sowie die Einführung des Begriffs der *Strategie* zur zeitlichen Allokation der Vorgänge notwendig. In der grundlegenden Arbeit [MRW 84] werden ein Modell für stochastisches Scheduling, der Begriff der Strategie sowie Fragen zur Existenz optimaler Strategien ausführlich diskutiert. Sie bildet die Basis der Probleme, die wir im folgenden untersuchen werden.

Das Ziel bei der Untersuchung vieler Optimierungsprobleme besteht in dem Entwurf eines (möglichst *polynomialen*) Lösungsalgorithmus. Viele Probleme erweisen sich jedoch als

„zu komplex" für diese Zielsetzung, und die meisten Schedulingprobleme bilden in dieser Hinsicht keine Ausnahme.

Eine Klassifizierung von Problemen in in diesem Sinne „schwierige" und „weniger schwierige" Probleme wird im Rahmen der *Komplexitätstheorie* unternommen. Die (vermutlich) schwierigen Probleme werden in diesem Zusammenhang $\mathcal{NP}$*-schwer* genannt; für solche Probleme wurde bislang kein polynomialer Algorithmus gefunden. Zu einer Einführung in die Theorie der $\mathcal{NP}$*-Vollständigkeit* und zur Klassifikation einer Vielzahl von Optimierungsproblemen sei auf das Lehrbuch [GJ 79] verwiesen; Schedulingprobleme werden in dieser Hinsicht speziell auch in [LRKB 77] untersucht. Bei der Behandlung $\mathcal{NP}$-schwerer Probleme begnügt man sich entweder mit Optimallösungen, die nicht notwendig mit polynomialem Aufwand gefunden werden, oder mit polynomialen *Heuristiken*, die nicht notwendig eine Optimallösung erzeugen.

Wir werden in dieser Arbeit Fragen zur Optimalität im Rahmen sogenannter *exponentieller* Schedulingprobleme untersuchen. Darüber hinaus werden wir Approximationsheuristiken für einige $\mathcal{NP}$-schwere stochastische Schedulingprobleme angeben.

Die Arbeit gliedert sich in drei Kapitel, deren Inhalte hier skizziert werden sollen.

Das **erste Kapitel** besteht aus einer Einführung in das Modell des stochastischen Scheduling. Der Schwerpunkt liegt dabei auf *exponentiellen* Schedulingproblemen mit *linearen* Kostenfunktionen, sowie den Klassen der *Set-* und *Prioritätsstrategien*.

In diesem Rahmen untersuchen wir das *deliberate idleness*-Problem. Dahinter verbirgt sich das Phänomen sogenannter *geplanter Leerzeiten*, die in einigen Modellen zur Minimierung der Kosten notwendig werden.

Die Suche nach hinreichenden Bedingungen zum Ausschluß derartiger geplanter Leerzeiten, also die Frage nach der Optimalität von Prioritätsstrategien, ist ein bislang ungeklärtes Problem im Rahmen exponentieller Schedulingprobleme:

> „[...] In any case, gaining more insight into this *non idleness problem* seems to be a very involved task and certainly constitutes at present one of the most important and most challenging open problems in stochastic scheduling. [...] We are certain that such results will also give some additional justification for the regular use of priority rules in stochastic scheduling problems and queueing theory [...]." [MRW 85]

Vor dem Hintergrund dieser bis heute offenen *deliberate idleness*-Frage werden Beispiele sowie bekannte Resultate für Spezialfälle angegeben. Die Basis bilden dabei die Arbeiten [MRW 85] und [Käm 85].

In diesem Zusammenhang liefert Lemma 1.6 ein Resultat für „kleine“ Probleme[1], das zwar den Hintergrund für mehrere, in [Käm 85] untersuchte Beispiele bildet, aber letztlich zur Beantwortung der *deliberate idleness*-Frage für lineare Kostenfunktionen nicht verallgemeinert werden konnte. Der Abschluß des ersten Kapitels wird durch die Skizze des untersuchten Beweisansatzes gebildet.

Das **zweite Kapitel** beschäftigt sich zunächst mit polyedrischen Untersuchungen. Wir stellen einige Grundlagen zu *Polyedern* und zur *linearen Optimierung* dar, um dann Optimierungsprobleme über speziellen Polyedern, sogenannten *erweiterten* (*Contra*)-*Polymatroiden* zu behandeln.

Der Zusammenhang zu stochastischen Schedulingproblemen ergibt sich über die Charakterisierung ihrer Lösungsräume:
In [BN-M 93] wird gezeigt, daß sich die Lösungsräume von *multi-armed bandit problems* und *branching bandit problems* in den polyedrischen Zusammenhang erweiterter (Contra)-Polymatroide einordnen lassen. Darüber hinaus stellen sich in [Que 93] und [QS 94] *super-* und *submodulare Polyeder* bei der Charakterisierung der Lösungsräume deterministischer Schedulingprobleme als zugrundeliegende polyedrische Strukturen heraus.

Mit Satz 2.6 wird in dieser Arbeit ein direkter Beweis dafür geliefert, daß super- und submodulare Polyeder spezielle erweiterte (Contra-)Polymatroide sind. Ferner wird ein in [BN-M 93] angegebener $\mathcal{O}(n^3)$-Algorithmus zur Optimierung linearer Zielfunktionen über erweiterten Contra-Polymatroiden vorgestellt. Dieser Algorithmus verallgemeinert den in [Edm 70] angegebenen *greedy*-Ansatz für super- und submodulare Polyeder.

Mit Satz 2.14 wird gezeigt, daß sich der Lösungsraum für stochastisches Scheduling auf einer Maschine *ohne Präzedenzrelation* in diesen polyedrischen Zusammenhang einordnet. Die Voraussetzung ist dabei die stochastische Unabhängigkeit der Verteilungen der einzelnen Vorgangsdauern. Dieses Resultat dehnt ein entsprechendes Ergebnis zur Beschreibung des Lösungsraums deterministischer Schedulingprobleme in [Que 93] auf den stochastischen Fall aus. Die Basis bildet hier die Arbeit [BN-M 93], in der gezeigt wird, daß Schedulingprobleme auf einer Maschine sich als Spezialfall in das Modell der *branching bandits* einordnen.

Anhand dieser polyedrischen Untersuchung stellt sich heraus, daß wie im deterministischen Scheduling auf einer Maschine die *ratio rule* von Smith [Smi 56] auch im stochastischen Fall eine optimale Strategie ist. Sie ist damit ein Spezialfall des Algorithmus für erweiterte Contra-Polymatroide.

Auch wenn der polyedrische Ansatz zur exakten Beschreibung der Lösungsräume stochastischer Schedulingprobleme auf wenige Spezialfälle beschränkt bleibt, ist die genaue

[1] „Klein“ bedeutet hier, daß lediglich *ein* Vorgang aufgrund der Kapazitätsbeschränkung nicht sofort bearbeitet werden kann.

Untersuchung „effizienter“ polyedrischer Strukturen durchaus gerechtfertigt. Denn für den Entwurf von *polynomialen Approximationsheuristiken* für stochastische Schedulingprobleme können die vorgestellten polyedrischen Methoden im dritten Kapitel genutzt werden.

Das **dritte Kapitel** beschäftigt sich mit *Approximationsheuristiken* für $\mathcal{NP}$-schwere stochastische Schedulingprobleme. Dabei gehen wir folgendermaßen vor: Anhand der Lösung von **LP**-*Relaxierungen* für stochastische Schedulingprobleme werden Heuristiken abgeleitet und schließlich Analysen zu deren *worst case*-Verhalten vorgestellt.

Die Resultate basieren auf den in [Sch 96] und [HSSW 96] angegebenen Heuristiken für $\mathcal{NP}$-schwere deterministische Schedulingprobleme.

Wir zeigen, daß sich diese Heuristiken beim Scheduling auf *einer* Maschine analog auf den stochastischen Fall übertragen lassen, sofern die Dauerverteilungen der einzelnen Vorgänge stochastisch unabhängig sind.

Für Probleme mit *mehreren* Maschinen geben wir Ergebnisse für Heuristiken bei stochastisch unabhängig verteilten Dauern mit „kleiner“ Varianz an. „Klein“ bedeutet hier, daß die Varianz durch das Quadrat des Erwartungwertes beschränkt ist.

Speziell im Hinblick auf das *deliberate idleness*-Problem bei exponentiellen Schedulingproblemen erhalten wir zusätzlich ein allgemeines Ergebnis über die Güte der *ratio rule*.

Schließlich beschäftigen wir uns im letzten Abschnitt mit der Lösung der den Heuristiken zugrundeliegenden linearen Programmen. Wir geben einen $\mathcal{O}(n^2)$-Algorithmus an, der unter anderem auf dem *greedy*-Algorithmus für supermodulare Polyeder basiert. Anhand dieses Algorithmus können einige der angegebenen **LP**-Relaxierungen tatsächlich effizient gelöst werden.

Eine Übersicht zu den Ergebnissen des dritten Kapitels ist auf Seite 115 angegeben.

An dieser Stelle möchte ich mich bei Prof. Dr. R. H. Möhring und Dr. A. S. Schulz für viele hilfreiche Gespräche im Rahmen der Betreuung der Arbeit bedanken. Weiter danke ich sämtlichen Korrekturlesern, die sicherlich erheblich zur Verständlichkeit des Textes beigetragen haben.

Kapitel 1

Stochastisches Scheduling mit geplanten Leerzeiten

1.1 Ein Beispiel

Wir wollen mit dem folgenden Beispiel das in dieser Arbeit zugrundegelegte Modell für stochastisches Scheduling motivieren.

Ein Produktionsbetrieb bestehe aus verschiedenen Fertigungsstraßen zur Herstellung gewisser Produkte. Sobald innerhalb einer Fertigungsstraße ein Defekt auftritt, werden durch den entstehenden Produktionsausfall Umsatzeinbußen verursacht.

Zur Behebung auftretender Defekte verfügt der Betrieb über ein „Wartungsteam" sowie einen Manager zur zeitlichen Aufteilung des Teams auf die bestehenden Reparaturaufträge.

Er sieht sich bei seiner Arbeit mit den folgenden Bedingungen und Notwendigkeiten konfrontiert:

- Jeder Auftrag $1, \ldots, n$ verursacht solange *Kosten* wie er nicht fertiggestellt ist; seine *Aufgabe* besteht also in der *Minimierung einer Kostenfunktion*, die von den *Fertigstellungszeiten* aller Aufträge abhängt.

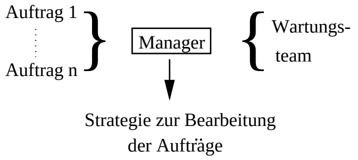


Abbildung 1.1: Planung der Auftragsbearbeitung durch den Manager

- Das Wartungsteam kann aufgrund *mangelnder Kapazität* gewisse Mengen von Aufträgen *nicht* gleichzeitig bearbeiten. Beispielsweise besteht das Team aus $m \leq n$ „identischen" Mitarbeitern, von denen jeder genau einen Auftrag zur gleichen Zeit bearbeiten kann.

- Die *Bearbeitungsdauer* der einzelnen Reparaturaufträge ist im voraus nicht genau einzuschätzen. Stattdessen liefert die mathematische Abteilung genaue Wahrscheinlichkeitsverteilungen für die Dauern der Aufträge. Insbesondere sind also zumindest *erwartete* Dauern für jeden Auftrag bekannt.

- Aufgrund von technischen Anforderungen kann es notwendig sein, gewisse zeitliche Reihenfolgebeziehungen zwischen einzelnen Aufträgen zu beachten.

- Ist ein Auftrag einmal begonnen, kann seine Bearbeitung aufgrund technischer Erfordernisse bis zur Fertigstellung nicht unterbrochen werden.

In Bezug auf die Bearbeitungsdauern ist dabei der Fall sogenannter „gedächtnisloser" Dauern typisch: Solange ein bereits begonnener Reparaturauftrag nicht fertiggestellt ist, ist die erwartete Dauer bis zu seiner Fertigstellung stets dieselbe.

Die folgenden Fragen sind für den Manager von Interesse:

- Wie ist unter den gegebenen Voraussetzungen das Ziel „Kostenminimierung" zu definieren?

- Was ist die beste „Strategie" zur zeitlichen Bearbeitung der Aufträge; existiert eine solche, und ist sie in angemessener Zeit zu ermitteln?

- Gibt es „einfache" Strategien, die zumindest „akzeptable" Kosten verursachen?

- Müssen auch Strategien berücksichtigt werden, die die Kapazität des Teams nicht immer voll ausschöpfen?

Im folgenden Abschnitt stellen wir ein Modell vor, in dessen Rahmen sich Beispiele wie das obige im Hinblick auf diese Fragen untersuchen lassen.

1.2 Stochastisches Scheduling - Das Modell

In dieser Arbeit wollen wir uns mit stochastischen Schedulingproblemen befassen, die sich im Rahmen der stochastischen dynamischen Programmierung insbesondere durch einen *endlichen* Planungshorizont auszeichnen. Eine Einführung in Modelle der stochastischen dynamischen Programmierung findet man etwa in [Ros 83].

Zur ausführlichen Erläuterung des in dieser Arbeit untersuchten Modells sei insbesondere auf den Artikel *Stochastic Scheduling Problems I – General Strategies* von R. H. Möhring, F. J. Radermacher und G. Weiss [MRW 84] hingewiesen. Aus dieser Arbeit, sowie auch deren Fortsetzung *Stochastic Scheduling Problems II – Set Strategies* [MRW 85] haben wir im wesentlichen die Notation und einige wichtige Resultate übernommen, die wir an geeigneter Stelle zitieren werden.

Das Modell, mit dem wir uns beschäftigen werden, wird durch ein Tupel $(\Theta, \mathcal{N}, P, \kappa)$ beschrieben. Wir wollen die Bedeutung dieses Tupels erläutern.

- Es bezeichnet $\Theta = (I, \prec)$ eine endliche Menge von **Vorgängen** oder **Jobs** $I = \{1, \ldots, n\}$, zusammen mit einer **strikten Halbordnung** $\prec$ auf I[1], die auch **Präzedenzrelation** genannt wird.
  Die Vorgänge $I = \{1, \ldots, n\}$ müssen in eine zeitliche Planungsreihenfolge gebracht werden. Man stelle sich beispielsweise vor, daß die Jobs unter Beachtung technisch bedingter *Reihenfolgebeziehungen* auf Maschinen abgearbeitet werden sollen. Ein wichtiges Charakteristikum der in dieser Arbeit behandelten Probleme ist die *non-preemptiveness*. Das bedeutet, daß jeder einmal gestartete Vorgang bis zu seiner Fertigstellung nicht unterbrochen werden darf.
  Die strikte Halbordnung $\prec$ beschreibt vorgeschriebene **Reihenfolgebeziehungen** der Art „Vorgang k muß vor der Bearbeitung von Vorgang l beendet sein". Solche Bedingungen treten üblicherweise in Planungssituationen auf, und wir schreiben $(k, l) \in \prec$ für jede derartige Reihenfolgebedingung.

- Zusätzlich zu den Reihenfolgebeziehungen sind sogenannte **Betriebsmittelbeschränkungen** zu beachten. Diese beschreiben die Tatsache, daß gewisse Mengen N von Jobs, die zwar bezüglich $\prec$ eine Antikette[2] bilden, nicht gleichzeitig in Arbeit sein dürfen. Man geht dabei davon aus, daß die Bearbeitung der Vorgänge Ressourcen bindet (etwa werden Maschinen belegt), und dadurch die gleichzeitige Bearbeitung der Vorgänge zusätzlich durch die Bindung dieser Ressourcen restringiert ist. Die Familie aller Mengen N, die aus diesem Grunde nicht gleichzeitig in Arbeit sein können, wird durch $\mathcal{N}$ bezeichnet, und wird auch die Familie der **verbotenen Mengen** genannt. Dabei bedeutet $N \in \mathcal{N}$, daß die Vorgänge jeder Menge $M \subseteq I$ mit $N \subseteq M$ nicht gleichzeitig in Arbeit sein können. Daher reicht es aus, $\mathcal{N}$ durch eine Familie inklusionsminimaler Mengen zu repräsentieren.
  Einen wichtigen Spezialfall bilden die sogenannten m-**Maschinen-Probleme**, bei denen $m \in \mathbb{N}$ identische Maschinen zur Bearbeitung der Jobs zur Verfügung stehen, und demnach maximal m Vorgänge gleichzeitig in Arbeit sein können. In diesem

[1] Eine strikte Halbordnung ist eine transitive und azyklische binäre Relation.

[2] Das bedeutet $(k, l) \notin \prec$ für alle $k, l \in N$; also dürfen diese Vorgänge bezüglich $\prec$ gleichzeitig bearbeitet werden.

Fall kann man $\mathcal{N}$ durch die Familie $\mathcal{N} = \{S \subseteq I \mid |S| = m+1, S$ Antikette bzgl. $\prec\}$ repräsentieren.

- Im stochastischen Scheduling geschieht die Bindung der Ressourcen durch die in Arbeit befindlichen Jobs für eine im vorhinein unbekannte Dauer[3].
  Die einzelnen **Vorgangsdauern** werden durch **Zufallsvariablen** $P_i : \Omega \longrightarrow \mathbb{R}_+$ über einem passenden Wahrscheinlichkeitsraum $(\Omega, \mathcal{F}, \mathbb{P})$ beschrieben. Es sei dann $\mathbb{P}^P := \mathbb{P}^{(P_1,\ldots,P_n)}$ die gemeinsame Verteilung der Vorgangsdauern. $\mathbb{P}^P$ ist eine Verteilung über $(\mathbb{R}_+^n, \mathcal{B}_+^n)$, wenn $\mathcal{B}_+^n$ die Borelmengen des $\mathbb{R}_+^n$ bezeichnet. Jede mögliche Realisierung der Vorgangsdauern werde durch $p = (p_1, \ldots, p_n) \in \mathbb{R}_+^n$ bezeichnet (P, bzw. p für *processing times*).
  Wir setzen voraus, daß die Erwartungswerte $E(P_i)$ für alle $i \in I$ existieren, also alle Vorgänge eine endliche erwartete Dauer besitzen.
  Wir schreiben im folgenden für das Bildmaß $\mathbb{P}^P$ der Kürze halber ebenfalls P, so daß P sowohl die Zufallsvariable als auch das zugehörige Bildmaß bezeichnet; Verwechslungen sind nicht zu erwarten.
  In dieser Arbeit behandeln wir den Fall stochastisch unabhängig verteilter Vorgangsdauern P_i, und häufig werden wir speziell exponentialverteilte Dauern voraussetzen. Im allgemeinen sind jedoch durchaus stochastisch abhängige Verteilungen erlaubt, man vergleiche hierzu etwa [MRW 84, Beispiel 3.1.3].

- Die Aufgabe des Scheduling ist die **Minimierung einer Kostenfunktion** κ durch die Angabe einer „optimalen Vorgehensweise" (Strategie) zur zeitlichen Planung der Vorgänge. In dieser Arbeit ist κ stets eine Funktion in den Fertigstellungszeiten der einzelnen Vorgänge. Wir haben also

  $$\kappa : \mathbb{R}_+^n \longrightarrow \mathbb{R}_+,$$

  wobei $\kappa(C_1, \ldots, C_n)$ die anfallenden **Kosten** oder **Projektkosten** sind, wenn C_i den Fertigstellungstermin von Vorgang i bezeichnet (C für *completion time*). Jeder einzelne Vorgang trägt demnach so lange zu den Projektkosten bei, wie er noch nicht fertiggestellt ist. Dies läßt sich dadurch motivieren, daß ein Vorgang erst im Termin seiner Fertigstellung für weiteren Nutzen zur Verfügung steht, und bis zu diesem Zeitpunkt Kosten verursacht (wie es etwa im Beispiel des ersten Abschnitts der Fall war).

  Es werden an κ die Bedingungen der *Regularität*[4], der *universellen Meßbarkeit* sowie der *linearen Majorisierbarkeit* gestellt. Die Forderung der Regularität macht man sich durch die Überlegung plausibel, daß eine spätere Beendigung *aller* Vorgänge

[3] Dies unterscheidet das stochastische Scheduling von seiner deterministischen Version, bei der die Dauern aller Vorgänge in vorhinein bekannt sind. Das stochastische Modell umfaßt unter Berücksichtigung von Einpunktverteilungen jedoch auch das deterministische Scheduling.

[4] D. h. für $C' \geq C \in \mathbb{R}_+^n$ gilt auch $\kappa(C') \geq \kappa(C)$.

keine Kostenvorteile bringen sollte (auch wenn andere Kostenfunktionen durchaus denkbar sind).
Die universelle Meßbarkeit sowie lineare Majorisierbarkeit spielen eine wichtige Rolle in Bezug auf die Wohldefiniertheit der Verteilung der Projektkosten (P-f.-ü.), und die Existenz und Endlichkeit des Erwartungswertes für die Projektkosten. Meßbarkeitsprobleme sollen nicht Thema dieser Arbeit sein, daher verweisen wir an dieser Stelle auf die Diskussion der Problematik in [MRW 84, Abschnitt 4]. Da wir uns ohnehin auf *lineare* Kostenfunktionen konzentrieren werden, sind beide Bedingungen in dieser Arbeit stets erfüllt.

Von besonderem Interesse sind Kostenfunktionen, bei denen sich die Gesamtkosten *additiv* aus den Kosten zwischen den einzelnen Aktionszeitpunkten zusammensetzen.

Definition 1.1 (Additive Kostenfunktion) *Eine Kostenfunktion $\kappa : \mathbb{R}_+^n \longrightarrow \mathbb{R}$ für ein Schedulingproblem heißt additiv, wenn es eine Abbildung $g : 2^I \longrightarrow \mathbb{R}$ gibt, so daß für jede Reihenfolge $C_{i_1} \leq \cdots \leq C_{i_n}$ der Fertigstellungszeiten der Jobs gilt:*

$$\begin{aligned}\kappa(C_1, \ldots, C_n) \quad = \quad & g(I)(C_{i_1} - 0) \\ & + g(I - \{i_1\})(C_{i_2} - C_{i_1}) \\ & + g(I - \{i_1, i_2\})(C_{i_3} - C_{i_2}) \\ & + \cdots \\ & + g(I - \{i_1, \ldots, i_{n-1}\})(C_{i_n} - C_{i_{n-1}}).\end{aligned}$$

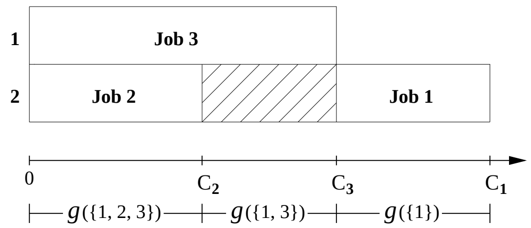


Abbildung 1.2: Gantt-Diagramm eines 2-Maschinen Terminplans

Man kann demnach $g(S)$ als die „laufenden" Kosten des Projektes auffassen, wenn genau alle Jobs aus $S \subseteq I$ noch nicht beendet sind. Die Funktion g wird auch *Kostenrate* zu κ genannt. Abbildung 1.2 zeigt ein 2-Maschinen Schedulingproblem mit zugehörigen Kostenraten im sogenannten *Gantt-Diagramm*.

Bekannte Kostenfunktionen wie die Gesamtprojektdauer (auch *makespan* oder $C_{\max} = \max\{C_i \mid i \in I\}$) sowie die gewichtete Summe der Fertigstellungszeiten

$\sum_{i\in I} w_i C_i$ mit $w_i \geq 0$ für alle $i \in I$, sind additiv mit zugehörigen Kostenraten

$$g(S) := \begin{cases} 1 & : \quad \text{falls } S \neq \emptyset, \\ 0 & : \quad \text{sonst,} \end{cases}$$

beziehungsweise

$$g(S) := \sum_{i\in S} w_i \text{ für alle } S \subseteq I.$$

In dieser Arbeit werden wir lineare Kostenfunktionen, also die gewichtete Summe der Fertigstellungszeiten untersuchen. Im letzten Kapitel werden wir zusätzlich die Hinzunahme sogenannter *release dates* berücksichtigen. Dabei bedeutet ein *release date* $r_i \geq 0$, daß Vorgang i frühstens zum Zeitpunkt r_i gestartet werden darf.

Schließlich müssen wir klären, was im stochastischen Scheduling eine *Strategie* zur zeitlichen Planung der Vorgänge ist.

1.3 Strategien

Eine Strategie entscheidet, wann welche Vorgänge gestartet werden sollen. Start- und Fertigstellungszeitpunkte der Vorgänge werden auch *Aktionszeitpunkte* genannt, und die zu einem Zeitpunkt getroffene Entscheidung zur weiteren Planung nennt man die von einer Strategie gewählte *Aktion*.

Dabei müssen sowohl die Reihenfolgebeziehungen $\prec$, als auch die Betriebsmittelbeschränkungen $\mathcal{N}$ und gegebenenfalls auch *release dates* berücksichtigt werden.

Die Aktion einer Strategie hängt stets vom augenblicklichen *Zustand* des Projekts ab, und der Zustand wird zu jedem Zeitpunkt t durch die bedingte Verteilung der Dauern bei t bestimmt. Dadurch wird der Zustand bei gegebener Verteilung der Dauern letztlich durch die Menge der bereits fertiggestellten Vorgänge und deren Startzeiten und Dauern, die Menge der noch andauernden Vorgänge und deren Startzeiten, sowie die Zeit t bestimmt.

Die Aktion selbst besteht dabei einerseits aus einer Startmenge S_t und andererseits aus einem vorläufigen nächsten Aktionszeitpunkt $t^{next} > t$. Die nächste Aktion einer Strategie findet zum Zeitpunkt $\min\{t^{next}, \text{Zeitpunkt der nächsten Vorgangsbeendigung}\}$ statt. Abbildung 1.3 zeigt die Aktion einer Strategie zu einem Zeitpunkt t.

Jede Strategie erzeugt schließlich einen *Terminplan* oder *Schedule* für das Projekt, in dem Start- und Fertigstellungszeiten der Jobs festgehalten werden. Aufgrund der *non-preemtiveness* des Modells kann man eine Strategie daher auch als eine Abbildung

$$\Pi : \mathbb{R}^n_+ \longrightarrow \mathbb{R}^n_+$$

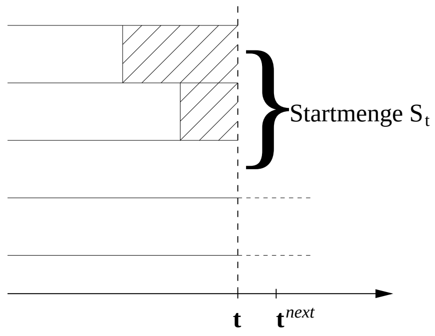


Abbildung 1.3: Die Aktion einer Strategie

auffassen, die jeder möglichen Realisierung der Vorgangsdauern einen Vektor von Startzeiten zuordnet, also

$$\mathbb{R}^n_+ \ni (p_1, \ldots, p_n) = p \overset{\Pi}{\longmapsto} \Pi(p) = ((\Pi(p))_1, \ldots, (\Pi(p))_n) \in \mathbb{R}^n_+.$$

Dabei sei $(\Pi(p))_i$ die Startzeit von Vorgang i unter Strategie Π. Wir haben demnach den folgenden Zusammenhang zwischen jeder Realisierung der Vorgangsdauern, Fertigstellungszeiten und Projektkosten unter einer Strategie Π:

$$\mathbb{R}^n_+ \ni p \overset{\Pi}{\longmapsto} \Pi(p) \overset{id+p}{\longmapsto} C \overset{\kappa}{\longmapsto} \kappa(C) \in \mathbb{R}_+.$$

Der stochastische Charakter des Modells – insbesondere die ex ante fehlende Information über die tatsächlichen Vorgangsdauern – führt dazu, daß erst während des Planungsprozesses „aktuelle“ Informationen über Vorgangsdauern und Fertigstellungszeiten entstehen. Diese Informationen können offenbar erst im Zeitpunkt ihres Entstehens für weitere Aktionen einer Strategie antizipiert werden, so daß die Aktion einer Strategie in diesem Sinne auf der „t-Vergangenheit“, zusammen mit der Verteilung für die Dauern beruht. Um diesem *dynamischen* Charakter einer Strategie in deren Definition als Abbildung Rechnung tragen zu können, benötigen wir zunächst den Begriff der *t-Ähnlichkeit* von Dauervektoren:

Definition 1.2 (t-Ähnlichkeit) *Es seien $\Pi : \mathbb{R}^n_+ \longrightarrow \mathbb{R}^n_+$, $t \in \mathbb{R}_+$ und $p \in \mathbb{R}^n_+$. Dann heißt $p' \in \mathbb{R}^n_+$ t-ähnlich zu p bezüglich Π, wenn gilt:*

(1) Ist $(\Pi(p))_i + p_i \leq t$, so folgt $p_i = p'_i$,

(2) Ist $(\Pi(p))_i \leq t < (\Pi(p))_i + p_i$, so folgt $p'_i > t - (\Pi(p))_i$.

Die t-Ähnlichkeit zweier Realisierungen der Vorgangsdauern bezüglich Π beschreibt die „Nichtunterscheidbarkeit“ dieser beiden Realisierungen aus der Sicht von Π zu einem Zeitpunkt t. Ist nämlich ein Dauervektor p' t-ähnlich zu p bezüglich Π, so sind beide

Dauervektoren für Strategie Π deshalb nicht unterscheidbar, da die von Π zur Zeit t gewonnene Information über den Zustand des Projektes sowohl auf p, als auch auf p' zutrifft. Um nun dem dynamischen Charakter einer Strategie gerecht zu werden, fordert man, daß eine Strategie in nicht unterscheidbaren Situationen stets dieselben Vorgänge startet.

Definition 1.3 (Strategie) *Für ein stochastisches Schedulingproblem* $(\Theta, \mathcal{N}, P, \kappa)$ *ist eine* Strategie $\Pi : \mathbb{R}^n_+ \longrightarrow \mathbb{R}^n_+$ *zur Planung der Vorgänge* $I = \{1, \ldots, n\}$ *durch folgende Bedingungen charakterisiert.*

(STO 1) *Es gilt* $(\Pi(p))_i + p_i \leq (\Pi(p))_j$ *für alle* $(i,j) \in \prec$ *und alle möglichen Realisierungen* $p \in \mathbb{R}^n_+$.

(STO 2) *Es gilt* $N \not\subseteq \{i \in I \mid (\Pi(p))_i \leq t < (\Pi(p))_i + p_i\}$ *für alle* $t \in \mathbb{R}_+$, *alle* $N \in \mathcal{N}$ *und alle* $p \in \mathbb{R}^n_+$.

(STO 3) *Gilt für* $t \in \mathbb{R}_+$ *und* $p \in \mathbb{R}^n_+$ $(\Pi(p))_i = t$ *für einen Vorgang* $i \in I$, *dann ist auch* $(\Pi(p'))_i = t$ *für alle* $p' \in \mathbb{R}^n_+$, *die in Bezug auf* Π t*-ähnlich zu* p *sind.*

(STO 4) Π *ist universell meßbar und für alle* $p \in \mathbb{R}^n_+$ *und alle* $t < \max_{i \in I}(\Pi(p))_i + p_i$ *gilt* $\{i \in I \mid (\Pi(p))_i \leq t < (\Pi(p))_i + p_i\} \neq \emptyset$, *d. h. unter* Π *ist zu jedem Zeitpunkt* t *vor Beendigung des Projekts wenigstens ein Vorgang in Arbeit.*

Die Bedingungen (STO 1) und (STO 2) garantieren lediglich, daß für jede beliebige Realisierung der Vorgangsdauern der von einer Strategie erzeugte Terminplan die Reihenfolgebeziehungen $\prec$ und die durch $\mathcal{N}$ beschriebenen Betriebsmittelbeschränkungen respektiert. Gelegentlich wird eine solche Strategie auch *zulässig* genannt.

Bedingung (STO 3) sichert, daß eine Strategie zu einem Zeitpunkt t in nicht unterscheidbaren Situationen, d. h. im Falle von t-Ähnlichkeit von Dauerrealisierungen, stets dieselben Vorgänge startet. Man überzeugt sich leicht, daß aus t-Ähnlichkeit auch t'-Ähnlichkeit für alle $t' < t$ folgt. Daher ist der von Π bis zum Zeitpunkt t erzeugte Terminplan für (in Bezug auf Π) t-ähnliche Dauervektoren stets identisch. Das bedeutet, daß die Aktion einer Strategie, abgesehen von der im vorhinein bekannten Verteilung der Dauern, tatsächlich stets auf der „t-Vergangenheit“ beruht, denn Π startet dieselben Vorgänge bei gleicher Vergangenheit. Damit beschreibt (STO 3) den dynamischen Charakter von Strategien. Zu einer ausführlicheren Behandlung der Bedingung (STO 3) bei der Definition von Strategien weisen wir auf den Anhang zu [MRW 84] hin.

Die Forderung der universellen Meßbarkeit in (STO 4) ist von entscheidender Bedeutung für den Beweis von Existenzsätzen optimaler Strategien; es sei an dieser Stelle nochmals auf [MRW 84, Abschnitt 4] verwiesen. Die in (STO 4) geforderte Vermeidung „totaler

Leerzeiten" zusammen mit den weiteren Anforderungen an P und κ garantiert die Endlichkeit der erwarteten Projektkosten unter einer Strategie Π. Da wir ohnehin an einer Kostenminimierung einer *regulären* Kostenfunktion interessiert sind, ist diese Forderung zwar plausibel, aber nicht notwendig[5].
Unter Berücksichtigung von *release dates* kann diese Bedingung offenbar nicht immer eingehalten werden, und an geeigneter Stelle werden wir daher auch totale Leerzeiten zulassen.

1.3.1 Der Zusammenhang der Verteilungen

Unter den gegebenen Voraussetzungen haben wir ausgehend von der Dauerverteilung P und einer beliebigen Strategie Π eine Bildverteilung P^{Π} der Startzeiten der Jobs. Weiter ist dann $P^{\Pi+P}$ die entsprechende Verteilung der Fertigstellungszeiten der Jobs, und schließlich ist $P^{\kappa\circ(\Pi+P)}$ die Verteilung der Projektkosten unter Strategie Π (Hier bezeichnet P einmal die Verteilung der Dauern und einmal die entsprechende Zufallsvariable).

Wir wollen die folgenden (intuitiveren) Bezeichnungen einführen:
$\mathcal{S}^{\Pi} := P^{\Pi}$ sei die Verteilung der Startzeiten, $\mathcal{C}^{\Pi} := P^{\Pi+P}$ sei die Verteilung der Fertigstellungszeiten und $\kappa^{\Pi} := P^{\kappa\circ(\Pi+P)}$ schließlich die Verteilung der Projektkosten unter einer Strategie Π. Der Zusammenhang der Verteilungen ist durch

$$P \xrightarrow{\Pi} \mathcal{S}^{\Pi} \xrightarrow{id+P} \mathcal{C}^{\Pi} \xrightarrow{\kappa} \kappa^{\Pi}$$

gegeben. Wir werden wieder die Bezeichnungen $\mathcal{S}^{\Pi}$, $\mathcal{C}^{\Pi}$ und κ^{Π} sowohl für die Bildverteilungen, als auch für die entsprechenden Zufallsvariablen verwenden; die jeweilige Bedeutung dürfte stets aus dem Zusammenhang klar werden.

Wir können unter den Anforderungen an Π und κ (Vermeidung totaler Leerzeiten von Π; lineare Majorisierbarkeit von κ; universelle Meßbarkeit) die Endlichkeit der Erwartungswerte für die Zufallsvariablen $\mathcal{S}^{\Pi}$, $\mathcal{C}^{\Pi}$ und κ^{Π} folgern. Dabei ist der Erwartungswert der n-dimensionalen Zufallsvariablen $\mathcal{S}^{\Pi}$ und $\mathcal{C}^{\Pi}$ durch den Vektor der Erwartungswerte der zugehörigen eindimensionalen Randverteilungen definiert.

1.3.2 Optimalität von Strategien

Wie wir bereits gesehen haben, führt der stochastische Rahmen des Modells zu einer Planung unter unvollständiger Information über die genauen Dauern der Vorgänge; einer entscheidungstheoretischen *Risikosituation*. Je nach tatsächlicher Realisierung der Vorgangsdauern können sich Aktionen einer Strategie als günstig oder ungünstig im Hinblick

[5]Eine Strategie, die totale Leerzeiten läßt, wird im Hinblick auf Kostenminimierung von einer Strategie übertroffen, die *keine* totalen Leerzeiten läßt, vergleiche [MRW 84, Theorem B].

auf die Projektkosten erweisen. In diesem Sinne ist eine Strategie, die für *jede* beliebige Dauerrealisierung minimale Projektkosten erzeugt, nicht zu erwarten. Eine solche Strategie müßte die *Verteilung* κ^Π der Projektkosten *punktweise* minimieren.

Um dieses Problem zu umgehen, gehen wir von einer (subjektiven) *Präferenzordnung* auf der Klasse $\varsigma = \{\kappa^\Pi \mid \Pi \text{ Strategie}\}$ aller Verteilungen der Projektkosten aus. Unser Ziel ist dann, eine Strategie möglichst hoher Präferenz zu finden. Unter Beachtung eines Resultats der Nutzenoptimierung[6] ist dann die Existenz einer stetigen, monoton wachsenden und beschränkten Nutzenfunktion $u : \mathbb{R}_+ \longrightarrow \mathbb{R}_+$ garantiert, so daß

$$\kappa^{\Pi_1} \text{ hat höhere Präferenz als } \kappa^{\Pi_2} \iff E((u \circ \kappa)^{\Pi_1}) \;\leq\; E((u \circ \kappa)^{\Pi_2})$$

gilt. Die Entscheidung für eine Strategie Π^{opt} maximaler Präferenz entspricht dann gerade einer Strategie mit maximalem erwarteten Nutzen $E((u \circ \kappa)^{\Pi^{opt}})$.

Wir werden im folgenden κ durch eine derartige Nutzenfunktion transformiert voraussetzen, wobei offenbar alle an κ gestellten Forderungen dadurch erhalten bleiben. In diesem Rahmen ist es plausibel, sich mit der Optimierung des *Erwartungswertes* der Projektkosten zu begnügen und nach derjenigen Strategie zu suchen, die einen minimalen Erwartungswert der Projektkosten $E(\kappa^\Pi)$ liefert, falls eine solche existiert[7].

Für eine feste Realisierung p der Vorgangsdauern sind die Projektkosten einer Strategie durch $\kappa(\Pi(p) + p) = \kappa(\underbrace{(\Pi(p))_1 + p_1}_{C_1}, \dots, \underbrace{(\Pi(p))_n + p_n}_{C_n})$ gegeben. Wir suchen nach einer Strategie Π mit möglichst geringen erwarteten Kosten

$$E(\kappa^\Pi) = \int_{\mathbb{R}_+} \kappa \, d\kappa^\Pi(\kappa) = \int_{\mathbb{R}_+^n} \kappa(\Pi(p) + p) \, dP(p).$$

Definition 1.4 (Optimale Strategie) *Der Optimalwert eines stochastischen Schedulingproblems wird bei gegebener Verteilung P der Vorgangsdauern durch*

$$\rho(\kappa) := \inf\{E(\kappa^\Pi) \mid \Pi \textit{ Strategie}\}$$

definiert. Eine Strategie Π^{opt} heißt optimal, falls $E(\kappa^{\Pi^{opt}}) = \rho(\kappa)$ gilt.

Beschränkt man sich auf eine Klasse σ spezieller Strategien, definieren wir entsprechend

$$\rho_\sigma(\kappa) := \inf\{E(\kappa^\Pi) \mid \Pi \in \sigma\}.$$

[6] Das Resultat geht auf J. von Neumann und O. Morgenstern zurück, vergleiche [RSZ 79, Satz 1.27].

[7] Optimalitätsaussagen, die auf Ordnungen zwischen den *Verteilungen* κ^Π beruhen, findet man etwa in [Käm 85, Kapitel 4].

Eine optimale Strategie in der Klasse *aller* Strategien braucht im allgemeinen nicht zu existieren, wie etwa [MRW 84, Beispiel 4.2.7] zeigt. Ist aber eine der Bedingungen

(i) κ ist unterhalb halbstetig[8]

(ii) P ist endlich diskret,

(iii) P hat eine (Lebesgue-)Dichte,

erfüllt, so existiert immer eine Strategie Π^{opt}, die das Infimum $\rho(\kappa)$ in der Klasse *aller* Strategien annimmt; vergleiche [MRW 84, Theorem 4.2.6].

Wenn wir uns also mit *linearen* Kostenfunktionen, d. h. der gewichteten Summe der Fertigstellungszeiten beschäftigen, ist wegen (i) die Existenz optimaler Strategien stets garantiert. Wir wollen das folgende Lemma für *lineare* Kostenfunktionen festhalten.

Lemma 1.1 *Für lineare und reguläre Kostenfunktionen, d. h. $\kappa = \sum_{i\in I} w_i C_i$ mit $w_i \geq 0$ für alle $i \in I$, gilt für eine beliebige Strategie Π*

$$E(\kappa^{\Pi}) = \sum_{i\in I} w_i\, E(\mathcal{C}_i^{\Pi}),$$

wobei $E(\mathcal{C}_i^{\Pi})$ die unter Π erwartete Fertigstellungszeit von Vorgang i ist.

Beweis. Die Behauptung ergibt sich folgendermaßen:

$$\begin{aligned} E(\kappa^{\Pi}) &= \int_{\mathbb{R}_+^n} \kappa(\Pi(p)+p)\, dP(p) \\ &= \int_{\mathbb{R}_+^n} \sum_{i\in I} w_i\, ((\Pi(p))_i + p_i)\, dP(p) \\ &= \sum_{i\in I} w_i \underbrace{\int_{\mathbb{R}_+^n} (\Pi(p))_i + p_i\, dP(p).}_{=E(\mathcal{C}_i^{\Pi})} \end{aligned}$$

■

Wir halten nochmals fest, daß ein stochastisches Schedulingproblem durch die Angabe einer optimalen *Strategie* gelöst wird, und der *Optimalwert* keineswegs den minimal erreichbaren Projektkosten entspricht. Der Optimalwert als Infimum der unter allen Strategien *erwarteten Projektkosten* kann natürlich bei entsprechenden Realisierungen der Vorgangsdauern unter einer optimalen Strategie sowohl über-, als auch unterschritten werden. Weiter ist mit der polynomialen Lösbarkeit[9] eines stochastischen Schedulingproblems gemeint, daß die Ermittlung einer optimalen Strategie in polynomialer Zeit möglich ist.

[8] D. h. für alle $C \in \mathbb{R}_+^n$ und alle Folgen $(C^k)_{k\in\mathbb{N}}$ mit $\lim_{k\to\infty} C^k = C$ gilt $\liminf_{k\to\infty} \kappa(C^k) \geq \kappa(C)$.

[9] Zur Komplexitätstheorie vergleiche insbesondere [GJ 79].

1.3.3 Spezielle Klassen von Strategien

Elementare Strategien

Eine Strategie heißt elementar, wenn Vorgänge lediglich zum Zeitpunkt 0 oder zum Fertigstellungszeitpunkt eines anderen Vorgangs gestartet werden. Elementare Strategien sind daher durch $t^{next} = \infty$ für jeden Aktionszeitpunkt charakterisiert.

Beschränkt man ein beliebiges stochastisches Schedulingproblem auf die Klasse ELE der elementaren Strategien, so existiert innerhalb dieser Klasse stets eine optimale Strategie Π^{opt}, also mit $\rho_{ELE}(\kappa) = E(\kappa^{\Pi^{opt}})$; vergleiche [MRW 84, Theorem 4.2.8].

Set-Strategien

Eine Strategie heißt Set-Strategie, wenn sie

(i) elementar ist, und

(ii) die Aktion zu jedem Aktionszeitpunkt ausschließlich von den *Mengen* der bislang beendeten sowie der noch andauernden Vorgänge abhängt.

Die Aktion einer Set-Strategie ist also unabhängig von der tatsächlichen Zeit t und den realisierten Dauern der beendeten und noch andauernden Vorgänge. Die Klasse aller Set-Strategien wollen wir mit SET bezeichnen.

Aufgrund der endlichen Anzahl von Vorgängen folgt aus (i) und (ii) sofort die Endlichkeit der Klasse aller Set-Strategien: Denn aus der Sicht von Set-Strategien gibt es stets nur endlich viele verschiedene Zustände, in denen sich ein Projekt befinden kann. Aufgrund der endlichen Anzahl möglicher Startmengen zu jedem Zeitpunkt folgt die Behauptung. Daher existiert stets eine optimale Set-Strategie Π^{opt} innerhalb der Klasse SET mit $\rho_{SET}(\kappa) = E(\kappa^{\Pi^{opt}})$.

Der *Zustand* des Projektes ist aus der Sicht von Set-Strategien bereits eindeutig durch die *Mengen* bereits beendeter und noch andauernder Jobs bestimmt. Wir können uns eine Set-Strategie daher durch einen Entscheidungsbaum veranschaulichen:
Jede Set-Strategie beinhaltet eine nichtleere Startmenge $S_0 \subseteq I$. Dies sei die Wurzel des Baumes. Für jeden der nächstmöglichen *Zustände* j des Projektes im Moment t_1 der ersten Vorgangsbeendigung hält die Set-Strategie eine (zulässige) Startmenge S_j^1 bereit.

In dieser Weise fahren wir fort – für jeden möglichen Zustand i des Projektes zu t_2 gibt es eine Startmenge $S_{j,i}^2$. Dabei hängt der Zustand i zu t_2 offenbar auch vom Zustand j zu t_1 ab. So erhalten wir letztlich einen Baum (mit maximaler Tiefe $|I|$), in dem jeder Weg

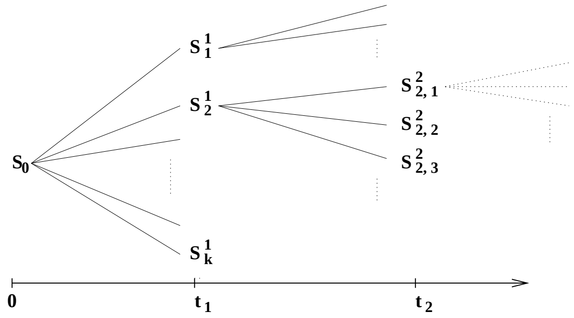


Abbildung 1.4: Darstellung einer Set-Strategie

eine mögliche Reihenfolge der Vorgangsbeendigungen zusammen mit den entsprechenden Startmengen der Strategie darstellt. Abbildung 1.4 zeigt das zugrundeliegende Prinzip.

Set-Strategien verzichten gewissermaßen auf alle Informationen, die die tatsächlichen Vorgangsdauern betreffen. Wie wir im nächsten Abschnitt sehen werden, entsteht dadurch unter geeigneten Voraussetzungen an die Vorgangsdauern und die Kostenfunktion kein Nachteil für die erwarteten Projektkosten, so daß dann auch optimale Set-Strategien in der Klasse *aller* Strategien existieren. Etwa [MRW 84, Beispiel 4.2.7] zeigt allerdings, daß ohne weitere Einschränkungen die Nutzung zeitlicher Informationen von fundamentaler Bedeutung für die Optimalität einer Strategie ist.

Prioritätsstrategien

Wir sprechen von einer Prioritätsstrategie, wenn zu jedem Aktionszeitpunkt t eine lineare Ordnung (Prioritätsliste) $L(t)$ der noch zu bearbeitenden Vorgänge vorliegt, und unter Berücksichtigung der Reihenfolgebeziehungen $\prec$ und Betriebsmittelbeschränkungen $\mathcal{N}$ *maximal* viele Vorgänge eingeplant werden. Dabei werden Vorgänge höherer Priorität – wenn möglich – vorgezogen.

Diese Strategien sind wegen ihrer einfachen Handhabung und Implementierbarkeit von Interesse, auch wenn Optimalität von Prioritätsstrategien innerhalb der Klasse aller Strategien nur in einigen Fällen auftritt.

Ist die Prioritätsliste $L(t)$ zu jedem Zeitpunkt dieselbe (abgesehen von der Entfernung bereits gestarteter Vorgänge), heißt die Prioritätsstrategie *statisch*. Andernfalls nennen wir sie *dynamisch*.

Jede Prioritätsstrategie ist *elementar*. Diese Tatsache folgt aus der inhärenten maximal möglichen Kapazitätsauslastung, der *greedyness* dieser Strategien: Zu jedem Aktionszeitpunkt werden maximal viele Jobs bezüglich $\prec$ und $\mathcal{N}$ gestartet. Daher kann frühstens zur nächsten Vorgangsbeendigung ein neuer Vorgang gestartet werden.

Eine statische Prioritätsstrategie wird eindeutig durch eine Permutation π auf der Vorgangsmenge I beschrieben, und wir bezeichnen eine Strategie, die nach der (statischen) Liste $\pi_1 < \pi_2 < \cdots < \pi_n$ vorgeht durch L_π (für „Listscheduling gemäß π"). Bekannte statische Prioritätsstrategien sind etwa LEPT und SEPT (longest bzw. shortest expected processing time first) sowie die *ratio rule* von Smith [Smi 56]; vergleiche Abschnitt 1.5.3 dieser Arbeit.

Set-Prioritätsstrategien

Jede Prioritätsstrategie, die gleichzeitig Set-Strategie ist, wollen wir *Set-Prioritätsstrategie* nennen.

Eine statische Prioritätsstrategie ist stets eine Set-Strategie, denn die Aufstellung der Prioritätsliste erfolgt zwangsläufig ohne Berücksichtigung zeitlicher Informationen, da sie bereits zum Zeitpunkt 0 festgelegt werden muß. Die Informationen, die hierzu verwendet werden können, sind lediglich die ex ante bekannten Informationen über Dauerverteilung P und Zielfunktion κ, wie beispielsweise die erwarteten Dauern $E(P_i)$ oder Gewichte w_i.

Eine dynamische Prioritätsstrategie ist nur dann gleichzeitig eine Set-Strategie, wenn die Prioritätsliste $L(t)$ jederzeit ausschließlich anhand der Informationen über die *Mengen* bereits beendeter und noch andauernder Vorgänge aufgestellt wird.

Umgekehrt ist eine Set-Strategie Π bereits eine Set-Prioritätsstrategie, wenn Π stets maximal viele Vorgänge in Bezug auf $\prec$ und $\mathcal{N}$ einplant (man nennt Π dann *greedy*):
Denn Π startet zu jedem beliebigen Aktionszeitpunkt t_k eine Jobmenge S_{t_k}. Wegen der *greedyness* kann zur Zeit t_k keine Menge $S' \supset S_{t_k}$ gestartet werden. Eine Prioritätsstrategie, die für alle Aktionszeitpunkte t_k den Jobs aus S_{t_k} die höchste Priorität gibt, führt dann stets zu demselben Ergebnis. Wir halten zusammenfassend fest [10]:

Bemerkung 1.1 *Eine Set-Strategie ist genau dann eine Set-Prioritätsstrategie, wenn sie greedy ist, also niemals Kapazitäten ungenutzt läßt, wenn dies in Bezug auf $\prec$ und $\mathcal{N}$ nicht notwendig ist.*

1.4 Exponentielle Schedulingprobleme und Set-Strategien

Zunächst benötigen wir einige Fakten über die Exponentialverteilung, die man etwa in [Ros 85, Kapitel 5] findet.

[10] Vergleiche auch [MRW 85, Bemerkung 3.1.4]; zur *greedyness* siehe insbesondere Abschnitt 1.5.

Definition 1.5 (Exponentialverteilung) *Eine reellwertige Zufallsvariable X heißt exponentialverteilt mit Parameter $\lambda > 0$, wenn X die Dichtefunktion*

$$f(x) := \begin{cases} 0 & : \quad \textit{falls } x < 0, \\ \lambda\, e^{-\lambda x} & : \quad \textit{falls } x \geq 0 \end{cases}$$

besitzt. Man schreibt auch $X \sim \exp(\lambda)$.

Es gilt für eine exponentialverteilte Zufallsvariable $X \sim \exp(\lambda)$:

(i) $E(X) = \dfrac{1}{\lambda}$,

(ii) $\mathrm{Var}(X) = \dfrac{1}{\lambda^2}$,

(iii) $W(X \leq t) = 1 - e^{-\lambda t}$ für alle $t \geq 0$,

wenn wir mit $W(\cdot)$ die Wahrscheinlichkeit des Ereignisses „ $\cdot$ “ für einen passenden Wahrscheinlichkeitsraum bezeichnen. Eine typische Eigenschaft exponentialverteilter Zufallsvariablen, die *Gedächtnislosigkeit*, ist bereits eine hinreichende Charakterisierung der Exponentialverteilung, und es gilt[11]:

Lemma 1.2 *Sei X eine stetige, reellwertige Zufallsvariable deren Dichte auf $\mathbb{R}_-$ verschwindet, dann sind äquivalent:*

(1) $X \sim \exp(\lambda)$,

(2) X *ist* gedächtnislos, *d. h.* $W(X \geq s + t \mid X \geq s) = W(X \geq t)$ *für alle* $s, t \geq 0$.

Insbesondere gilt für eine gedächtnislose Zufallsvariable damit auch

$$E(X \mid X \geq s) = s + E(X) \qquad \text{für alle } \; s \geq 0.$$

Im Hinblick auf gedächtnislose Vorgangsdauern im Scheduling bedeutet das, daß die erwartete Dauer bis zur Fertigstellung eines laufenden Vorgangs unabhängig von seiner bisherigen Bearbeitungsdauer ist.

Weiter benötigen wir:

Lemma 1.3 *Für exponentialverteilte Zufallsgrößen $X_i \sim exp(\lambda_i)$, $i = 1, \ldots, n$ gilt*

(1) $E(\min_{i=1,\ldots,n} X_i) \;=\; \dfrac{1}{\lambda_1 + \cdots + \lambda_n}$,

[11] Vergleiche [Ros 85, Abschnitt 5.2].

(2) $W(X_k = \min_{i=1,\dots,n} X_i) \;=\; \dfrac{\lambda_k}{\lambda_1 + \dots + \lambda_n}$ *für alle* $k \in \{1, \dots, n\}$.

Beweis. Wir zeigen nur die erste Behauptung, die zweite zeigt man analog. Sei zunächst $Z := \min\{X_1, X_2\}$, dann gilt mittels Konditionieren:

$$\begin{aligned} W(Z \le t) &= \int_0^\infty W(Z \le t | X_2 = x)\lambda_2\, e^{-\lambda_2 x}\, dx \\ &= \int_0^t W(Z \le t | X_2 = x)\lambda_2\, e^{-\lambda_2 x}\, dx \\ &\quad + \int_t^\infty W(Z \le t | X_2 = x)\lambda_2\, e^{-\lambda_2 x}\, dx \\ &= \int_0^t \lambda_2\, e^{-\lambda_2 x}\, dx \;+\; \int_t^\infty (1 - e^{-\lambda_1 t})\lambda_2\, e^{-\lambda_2 x}\, dx \\ &= 1 - e^{-\lambda_2 t} + (1 - e^{-\lambda_1 t})e^{-\lambda_2 t} \\ &= 1 - e^{-(\lambda_1 + \lambda_2)\, t}, \end{aligned}$$

also ist Z exponentialverteilt mit Parameter $\lambda_1 + \lambda_2$. Daher gilt $E(Z) = \frac{1}{\lambda_1 + \lambda_2}$. Die Behauptung folgt nun per Induktion. ∎

Diese Tatsachen sind zur Berechnung der erwarteten Kosten einer Strategie in sogenannten exponentiellen Schedulingproblemen überaus nützlich.

Definition 1.6 (Exponentielles Schedulingproblem) *Ein stochastisches Schedulingproblem* $(\Theta, \mathcal{N}, P, \kappa)$ *nennen wir exponentielles Schedulingproblem, falls*

(1) $P = \bigotimes_{i \in I} P_i$ *mit* $P_i \sim \exp(\lambda_i)$ *und* $\lambda_i > 0$ *für alle* $i \in I$ *gilt,*

(2) und die Kostenfunktion κ *additiv ist.*

In exponentiellen Schedulingproblemen ist die Existenz optimaler Strategien offenbar garantiert, denn P hat eine (Lebesgue-)Dichte. Es gilt sogar[12]:

Satz 1.1 (Optimalität von Set-Strategien) *Ist* $(\Theta, \mathcal{N}, P, \kappa)$ *ein exponentielles Schedulingproblem, so existiert eine optimale Set-Strategie in der Klasse aller Strategien, d. h. es gilt* $\rho_{SET}(\kappa) = \rho(\kappa)$.

Wir werden uns daher zur Untersuchung exponentieller Schedulingprobleme auf die *endliche* Klasse der Set-Strategien beschränken. Nach Satz 1.1 sind in exponentiellen Schedulingproblemen Informationen über die Zeit, sowie die Dauern bereits beendeter und noch andauernder Jobs für jede Strategie wertlos.

[12] Dieses Resultat ist ein Spezialfall von [MRW 85, Theorem 4.2.1].

Tatsächlich muß die Kostenfunktion nicht notwendig additiv sein, sondern es reicht zum Beweis des Resultates aus, wenn κ die sogenannte *shift property* hat. Diese Eigenschaft einer Kostenfunktion beinhaltet, daß die Kosten, die ab einem beliebigen Zeitpunkt t anfallen, sich von den Gesamtkosten lediglich durch eine *affine Transformation* unterscheiden[13].

Die Optimalität von Set-Strategien für exponentielle Schedulingprobleme liegt in der Gedächtnislosigkeit und stochastischen Unabhängigkeit der Vorgangsdauern begründet: Die Information über die Dauern bereits vor t beendeter und laufender Vorgänge ist ohne Informationsgehalt für zu erwartende Dauern laufender und noch nicht gestarteter Vorgänge, also das „Subproblem ab t“. Ohne Fertigstellung eines Vorgangs, und damit einer Änderung der *Mengen* der beendeten und laufenden Vorgänge, stellt sich kein Informationsgewinn über die Dauerverteilung des Subproblems ab t ein. Die *shift property* sichert dann, daß sich diese Eigenschaft der Dauerverteilung in „ausreichender” Weise auf die Kostenfunktion κ fortpflanzt.

Wir wollen hier lediglich festhalten, daß *additive* Kostenfunktionen diese Eigenschaft besitzen[14],denn die Kosten für das „Subproblem ab t“ unterscheiden sich von den Gesamtkosten lediglich durch einen additiven Term.

Eine Rekursionsformel zur Berechnung von $E(\kappa^{\Pi})$

Die angegebenen Eigenschaften der Exponentialverteilung zusammen mit der Additivität der Kostenfunktionen in exponentiellen Schedulingproblemen erlauben die Berechnung der erwarteten Kosten $E(\kappa^{\Pi})$ über eine einfache Rekursionsformel.

Dazu seien die Dauerverteilung P, die Vorgangsmenge I und eine (Set-)Strategie Π gegeben. Betrachten wir das Subproblem ab dem Zeitpunkt der ersten Fertigstellung eines Vorgangs, so zeigt Th. Kämpke [Käm 85, Lemma A.2], daß das resultierende Problem wieder ein exponentielles Schedulingproblem ist, und die gemeinsame (Rest-)Dauerverteilung der noch andauernden Vorgänge das Produkt der ursprünglichen Dauerverteilungen ist. Daher kann man dieses Subproblem derart auffassen, daß die Menge der noch andauernden Vorgänge nun als vorgeschriebene Startmenge gegeben ist.

Setzen wir diese Argumentation fort, liegt nach einer beliebigen Anzahl von Fertigstellungen stets wieder ein exponentielles Schedulingproblem mit entsprechender vorgeschriebener Startmenge vor.

Wir führen die folgenden Notationen ein:
Zu einem beliebigen Aktionszeitpunkt t seien

[13] Zur genauen Definition der *shift property*, sowie einer eingehenden analytischen Untersuchung, vergleiche [MRW 85, Abschnitt 4.1]

[14] Vergleiche auch [MRW 85, Lemma 4.1.1].

- $G \subseteq I$ die Menge der bereits *vor* t gestarteten Vorgänge, und
- $L \subseteq G$ die Menge derjenigen Vorgänge, die bereits gestartet, aber bis einschließlich t noch nicht beendet sind, und
- $R := I - G$ die Menge der Vorgänge, die vor t noch nicht gestartet wurden

(G für „gestartete", L für „laufende" Vorgänge und R für den „Rest"). Korrekterweise müßten wir definieren:
$G_t(\Pi, p) := \{i \in I \mid (\Pi(p))_i < t\}$, und $L_t(\Pi, p) := \{i \in G_t(\Pi, p) \mid (\Pi(p))_i + p_i > t\}$, denn natürlich hängen diese Mengen sowohl von der Strategie Π als auch von der Realisierung der Vorgangsdauern ab. Da Verwechslungen ausgeschlossen sind, schreiben wir der Kürze halber einfach G, L bzw. R.

Den Erwartungswert von Π für das Subproblem ab t bezeichnen wir durch

$$E(\kappa^{\Pi}(R \cup L \,|L))$$

(Lies: $R \cup L$ sind noch zu bearbeiten; L ist vorgeschriebene Startmenge). Zur Berechnung der zugehörigen Kosten sind die Fertigstellungszeiten der bereits beendeten Vorgänge lediglich auf 0 zu setzten. Entsprechend sei der Optimalwert $\rho(\kappa(R \cup L \mid L))$ für ein solches Subproblem definiert.
Weiter schreiben wir für den Fall $L = \emptyset$ der Kürze halber

$$\rho(\kappa(R)) \qquad \text{bzw.} \qquad E(\kappa^{\Pi}(R)).$$

Es gilt mit diesen Bezeichnungen[15]:

Lemma 1.4 *Es sei $(\Theta, \mathcal{N}, P, \kappa)$ ein exponentielles Schedulingproblem mit zugehöriger Kostenrate $g : 2^I \longrightarrow \mathbb{R}_+$, $P = \bigotimes_{i \in I} \exp(\lambda_i)$ und $\lambda_i > 0$ für alle $i \in I$. Sei weiter $S_0 \subseteq I$ die Startmenge einer beliebigen Set-Strategie Π zum Zeitpunkt 0. Dann gilt*

$$E(\kappa^{\Pi}) \;=\; \frac{g(I)}{\sum\limits_{i \in S_0} \lambda_i} \;+\; \sum_{j \in S_0} \frac{\lambda_j}{\sum\limits_{i \in S_0} \lambda_i}\; E\Big(\kappa^{\Pi}(I - \{j\} \mid S_0 - \{j\})\Big). \tag{1.1}$$

Diese Rekursion ist folgendermaßen zu interpretieren:
Bis zur ersten Vorgangsbeendigung sind alle Jobs aus S_0 in Arbeit. Die erwartete Dauer bis zur ersten Vorgangsbeendigung ist demnach $E(\min\{P_i \mid i \in S_0\}) = 1/\ \sum_{i \in S_0} \lambda_i$. Während dieser Zeit fallen Kosten in einer Höhe $g(I)$ an.

Weiter endet jeder Vorgang $j \in S_0$ mit einer Wahrscheinlichkeit $\lambda_j/\ \sum_{i \in S_0} \lambda_i$ zuerst, und es entsteht ein Subproblem, bei dem j bereits fertiggestellt ist. Wegen der Gedächtnislosigkeit der Vorgangsdauern kann man die laufenden Vorgänge $S_0 - \{j\}$ als vorgeschriebene

[15] Vergleiche etwa [Käm 85, Lemma 1.12].

Startmenge für ein (exponentielles) Subproblem auf der noch zu bearbeitenden Vorgangsmenge $I - \{j\}$ auffassen.

Beweis zu Lemma 1.4. Vorerst halten wir fest, daß die Wahrscheinlichkeit für ein gleichzeitiges Ende mehrerer Vorgänge aufgrund der Stetigkeit der Dauerverteilungen verschwindet. Es gilt daher $\sum_{i\in S_0} W(P_j = \min\{P_i \mid i \in S_0\}) = 1$. Mit dem Satz von der totalen Wahrscheinlichkeit folgt dann mit $S_0 = \{1, \ldots, k\}$

$$E(\kappa^\Pi) = \sum_{j=1}^{k} E\Big(\kappa^\Pi \mid P_j = \min\{P_i \mid i \in S_0\}\Big) \underbrace{W(P_j = \min\{P_i \mid i \in S_0\})}_{=\frac{\lambda_j}{\lambda_1+\cdots+\lambda_k}}.$$

Wegen Additivität von κ und der Gedächtnislosigkeit aller P_i gilt

$$E\Big(\kappa^\Pi \mid P_j = \min\{P_i \mid i \in S_0\}\Big) = E\Big(\kappa^\Pi(I - \{j\} \mid S_0 - \{j\})\Big) + \frac{g(I)}{\lambda_1 + \cdots + \lambda_k},$$

und insgesamt folgt

$$\begin{aligned} E(\kappa^\Pi) &= \sum_{j=1}^{k}\left(E\Big(\kappa^\Pi(I - \{j\} \mid S_0 - \{j\})\Big) + \frac{g(I)}{\lambda_1 + \cdots + \lambda_k}\right)\frac{\lambda_j}{\lambda_1 + \cdots + \lambda_k} \\ &= \frac{g(I)}{\lambda_1 + \cdots + \lambda_k} + \sum_{j=1}^{k}\frac{\lambda_j}{\lambda_1 + \cdots + \lambda_k} E\Big(\kappa^\Pi(I - \{j\} \mid S_0 - \{j\})\Big). \end{aligned}$$

■

Die Berechnung der erwarteten Kosten einer Strategie anhand dieser Rekursion wird *Ablaufbaumverfahren* genannt. Es muß dabei jede mögliche Reihenfolge der Vorgangsbeendigungen (ausgehend von den entsprechenden Aktionen einer Strategie) berücksichtigt werden. Wir wollen zur Verdeutlichung ein Beispiel angeben:

Beispiel 1.1 Wir betrachten ein 2-Maschinen Problem mit 3 Jobs:

- $I = \{1, 2, 3\}$ und $\mathcal{N} = \{\{1, 2, 3\}\}$,
- $P_i \sim exp(\lambda_i)$, $\lambda_i > 0$ für alle $i \in \{1, 2, 3\}$,
- $\kappa = \sum_{i\in I} w_i C_i$ mit Gewichten $w_i \in \mathbb{R}_+$,
- $\prec = \{(1, 2)\}$, d. h. Job 2 muß auf das Ende von Job 1 warten.

Die zugehörige Kostenrate ist $g(S) = \sum_{i\in S} w_i$ für $S \subseteq \{1, 2, 3\}$.

Wir berechnen die erwarteten Kosten der statischen Prioritätsstrategie L_π mit $\pi = (1, 2, 3)$ (L_π verfährt nach der Prioritätsliste $1 < 2 < 3$).

Wegen $\prec$ werden Vorgänge 1 und 3 zu Beginn gestartet. Vorgang 2 wird gestartet, sobald Vorgang 1 beendet ist. Daher gibt es nur drei verschiedene Möglichkeiten für die Reihenfolge der Fertigstellungszeiten der Jobs, die in Abbildung 1.5 dargestellt sind.

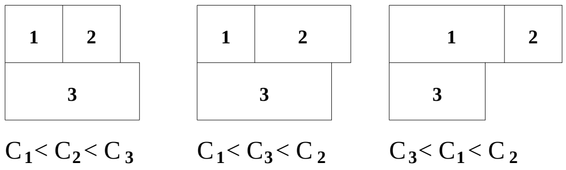


Abbildung 1.5: Gantt-Diagramm verschiedener Realisierungen unter L_π

Unter Berücksichtigung der zugehörigen Wahrscheinlichkeiten berechnen sich die erwarteten Kosten $E(\kappa^{L_\pi})$ anhand der Rekursion (1.1) wie folgt:

$$\underbrace{\frac{w_1+w_2+w_3}{\lambda_1+\lambda_3}}_{\text{Jobs 1 und 3 starten}} + \underbrace{\frac{\lambda_1}{\lambda_1+\lambda_3}}_{\text{W(Job 1 endet)}} \left(\underbrace{\left(\frac{w_2+w_3}{\lambda_2+\lambda_3} \right)}_{\text{Job 2 startet}} + \left(\begin{array}{c} \underbrace{\frac{\lambda_2}{\lambda_2+\lambda_3}}_{\text{W(Job 2 endet)}} \quad \underbrace{\left(\frac{w_3}{\lambda_3} \right)}_{\text{3 läuft noch}} \\ + \underbrace{\frac{\lambda_3}{\lambda_2+\lambda_3}}_{\text{W(Job 3 endet)}} \quad \underbrace{\left(\frac{w_2}{\lambda_2} \right)}_{\text{2 läuft noch}} \end{array} \right) \right)$$

$$+ \underbrace{\frac{\lambda_3}{\lambda_1+\lambda_3}}_{\text{W(Job 3 endet)}} \left(\underbrace{\left(\frac{w_1+w_2}{\lambda_1} \right)}_{\text{Nur Job 1 läuft}} + \underbrace{\frac{\lambda_1}{\lambda_1}}_{\text{W(Job 1 endet)}} \underbrace{\left(\frac{w_2}{\lambda_2} \right)}_{\text{2 startet}} \right),$$

und wir erhalten letztlich

$$E(\kappa^{L_\pi}) = \frac{w_1}{\lambda_1} + \frac{w_3}{\lambda_3} + w_2\left(\frac{1}{\lambda_2} + \frac{1}{\lambda_1}\right).$$

Dieses Ergebnis ist unter Beachtung von Lemma 1.1 nicht überraschend, denn es entspricht genau der Summe

$$\sum_{i\in I} w_i \, E(\mathcal{C}_i^{\Pi}).$$

Man überzeugt sich durch folgende Überlegung von dieser Tatsache: Vorgänge 1 bzw. 3 haben die erwartete Fertigstellungszeit $1/\ \lambda_1$ bzw. $1/\ \lambda_3$, da sie beide zu Beginn starten. Da Vorgang 2 in jedem Fall nach der Fertigstellung von Vorgang 1 gestartet wird, ist seine zu erwartende Startzeit $1/\ \lambda_1$, und daher seine zu erwartende Fertigstellungszeit $1/\ \lambda_1\ +\ 1/\ \lambda_2$.

1.5 Scheduling mit geplanten Leerzeiten: Das Deliberate Idleness - Problem

Wir sprechen von einer Leerzeit oder *idle time*, wenn zeitweilig keine Vorgänge auf einer Maschine in Arbeit sind. Aufgrund von Reihenfolgebedingungen oder Betriebsmittelbeschränkungen können solche *idle times* durch die Modellparameter erzwungen werden.

Darüber hinaus kann der im Laufe der Bearbeitung auftretende Informationsgewinn über tatsächlich realisierte Vorgangsdauern zu dem Phänomen der *geplanten Leerzeiten* oder *deliberate idle times* führen. Dieses Phänomen bezeichnet die irrational scheinende Verhaltensweise einer Strategie, „freiwillig“ auf verfügbare Kapazitäten zu verzichten, auch wenn dies in Bezug auf $\prec$ und $\mathcal{N}$ nicht nötig wäre.

In Abschnitt 1.5.1 zeigen wir, daß es bereits für exponentielle Schedulingprobleme auf 3 Maschinen additive Kostenfunktionen gibt, deren Optimalwert ausschließlich von Strategien *mit* geplanter Leerzeit erzielt wird. Für dasselbe Modell erweisen sich andererseits die statische Prioritätsstrategien LEPT bzw. SEPT als optimal zur Minimierung der Gesamtprojektdauer bzw. der Summe der Abschlußtermine $\sum_{i\in I} C_i$[16].

Vor diesem Hintergrund stellen wir uns die Frage, unter welchen Anforderungen an die Kostenfunktion solche Phänomene bei exponentiellen Schedulingproblemen auf m Maschinen ausgeschlossen werden können. Nach Satz 1.1 können wir uns auf die Klasse der Set-Strategien beschränken.

Das *deliberate idleness*-Problem ist nun die Frage nach passenden Anforderungen an die Kostenfunktion, die die Optimalität von Set-Prioritätsstrategien in der Klasse aller Strategien garantieren.

Wir wollen kurz andeuten, welcher *Vorteil* im Hinblick auf die Kostenfunktion überhaupt durch den freiwilligen Verzicht auf Kapazitäten entstehen kann:
Im Laufe der Zeit erhöht sich der Informationsstand über die Dauern der bereits gestarteten Jobs. Diese Information kann für weitere Aktionen einer Strategie wertvoll sein. Daher kann sich „Warten“, also eine geplante Leerzeit, durch die Möglichkeit einer „besseren“ Aktion zu einem späteren Zeitpunkt auszahlen. Man bedenke, daß eine Entscheidung zu einem früheren Zeitpunkt unter weniger Information stattfindet, und ein einmal gestarteter Vorgang eine Maschine für eine *unbekannte* Dauer belegt.

Es besteht die Vermutung, daß *Konvexität* der Kostenfunktion, oder äquivalent *Submodularität*[17] der zu κ gehörigen Kostenrate $g : 2^I \longrightarrow \mathbb{R}_+$ eine hinreichende Bedingung zum Ausschluß von *deliberate idleness*-Phänomenen ist[18].

Die in Lemma 2.4 angegebene Charakterisierung submodularer Funktionen kann diese Vermutung motivieren: Offenbar verschieben geplante Leerzeiten die Bearbeitung gewisser Jobs zeitlich nach hinten. Wenn dadurch zusätzliche Kosten entstehen, die stets so groß sind, daß sie den Vorteil des Informationsgewinns durch Warten und etwaiges „besseres Planen“ überkompensieren, so lohnen sich geplante Leerzeiten nicht. Submodularität der Kostenrate hat zur Folge, daß die gleichzeitige Bearbeitung möglichst vieler Jobs

[16] Vergleiche Sätze 1.3 und 1.4.
[17] Vergleiche Abschnitt 2.3.1 zur Definition und Charakterisierung submodularer Funktionen.
[18] Vergleiche auch die Ausführungen in [MRW 85, Abschnitt 5].

tatsächlich eine Kostenersparnis bringt, denn $(g(S \cup \{k\}) - g(S))$ ist monoton nichtsteigend in S für alle k; vergleiche Lemma 2.4. Da die *greedyness* einer Strategie die gleichzeitige Bearbeitung möglichst vieler Jobs impliziert, könnte die Submodularität der Kostenrate in der Tat hinreichend für den Ausschluß von *deliberate idleness*-Phänomenen sein.

Diese Vermutung wird auch die *deliberate idleness*-Vermutung genannt.

In diesem Zusammenhang wäre eine Aussage über die gewichtete Summe der Abschlußtermine $\kappa = \sum_{i \in I} w_i C_i$, die aufgrund ihrer Linearität insbesondere konvex ist, bereits ein Fortschritt. Jedoch ist das *deliberate idleness*-Problem bislang noch nicht einmal für diesen Spezialfall vollständig geklärt. Die Schwierigkeit liegt in der genauen Quantifizierung der Vor- und Nachteile von Prioritätsstrategien gegenüber Strategien mit mit geplanten Leerzeiten.

In diesem Abschnitt sollen einige Überlegungen zur *deliberate idleness*-Frage, insbesondere für den Spezialfall $\kappa = \sum_{i \in I} w_i C_i$ vorgestellt werden. Auch wenn keine befriedigende Antwort gefunden wurde, wird in jedem Fall deutlich, wo das Problem bei der Quantifizierung der Vor- und Nachteile von Prioritätsstrategien liegt.

1.5.1 Deliberate Idleness – Ein Beispiel

Wir untersuchen ein Beispiel für ein exponentielles Schedulingproblem auf 3 Maschinen, in dem eine optimale Strategie *nicht* auf geplante Leerzeiten verzichten kann.

Wir betrachten ein 3-Maschinen Problem mit 6 Vorgängen[19]:

- $I = \{1, \ldots, 6\}$,
- $\prec = \emptyset$
- $\mathcal{N} = \{S \subseteq I \mid |S| = 4\}$,
- $P = \bigotimes_{i=1}^{6} \exp(\lambda_i)$ mit $\lambda_1 = \lambda_2 = x > 0$ und $\lambda_3 = \cdots = \lambda_6 = 1$.

Die Kostenfunktion ist additiv, und die zugehörige Kostenrate $g : 2^I \longrightarrow \mathbb{R}_+$ sei folgendermaßen definiert:

$$g(S) := \begin{cases} c & \text{falls} \quad \{1,2\} \subseteq S, \\ b & \text{falls} \quad \left\{ \begin{array}{ll} 1 \in S \text{ und } 2 \notin S \text{ und } \{3,4\} \cap S \neq \emptyset & \text{oder} \\ 2 \in S \text{ und } 1 \notin S \text{ und } \{5,6\} \cap S \neq \emptyset & \end{array} \right\}, \\ a & \text{falls} \quad |\{1,2\} \cap S| = 1 \text{ und nicht die obigen Fälle }, \\ 0 & \text{falls} \quad |\{1,2\} \cap S| = 0. \end{cases}$$

[19] Das Beispiel ist eine leicht variierte Version von [MRW 85, Beispiel 4.2.5].

Dabei seien $a, b, c \in \mathbb{R}_+$ beliebig. Wegen Regularität von κ muß g nichtfallend sein. Daher muß $a \leq b \leq c$ gelten[20].

Die Kostenfunktion ist so aufgebaut, daß am meisten Kosten (in einer Höhe von c) anfallen, solange die Jobs 1 *und* 2 noch nicht beendet sind. Sind Jobs 1 und 2 *beide* beendet, fallen keine Kosten mehr an. Ist *genau ein* Job aus der Menge $\{1, 2\}$ noch nicht beendet, sind die Kosten daran gekoppelt, welche weiteren Jobs noch nicht beendet sind. Job 1 ist also teurer (Kosten $b \geq a$), wenn die Jobs 3 und/oder 4 ebenfalls noch nicht beendet sind. Bei Job 2 gilt dasselbe mit den Jobs 5 und/oder 6.

Wir suchen nach einer optimalen Strategie für dieses Problem. Sei nun $0 \leq a < b < c$ beliebig, und sei $c \gg b$ so groß gewählt, daß die einzig erfolgversprechenden Strategien solche sind, die Jobs 1 *und* 2 zu Beginn starten.

Wir wollen kurz begründen, weshalb eine solche Wahl möglich ist:

Solange beide der Vorgänge 1 und 2 noch nicht beendet sind, fallen Kosten in der Höhe c an. Wenn wir beide Vorgänge zu Beginn starten, ist die erwartete Dauer bis zum Ende einer der beiden Vorgänge bekanntlich $\frac{1}{\lambda_1+\lambda_2}$. Der Parameter c geht demnach gerade durch den Term $\frac{c}{\lambda_1+\lambda_2}$ in die erwarteten Kosten ein. Starten wir nur *einen* der beiden Vorgänge (Vorgang 1) zu Beginn, ist die erwartete Dauer bis zu seinem Ende $\frac{1}{\lambda_1}$. In diesem Fall geht c durch den Term $\frac{c}{\lambda_1}$ in die erwarteten Kosten ein. Für beliebig großes c ist wegen $\lambda_1, \lambda_2 > 0$ auch die Differenz $\frac{c}{\lambda_1} - \frac{c}{\lambda_1+\lambda_2}$ beliebig groß.

Gehen wir davon aus, daß die Parameter a und b *fest* gewählt sind, können wir daher c stets so wählen, daß die erwarteten Kosten *jeder* Strategie, die Jobs 1 und 2 zu Beginn startet unter den Kosten einer Strategie liegen, die nur einen der beiden Jobs zu Beginn startet.

Ferner können Strategien, die *keinen* der Jobs zu Beginn starten mit denselben Überlegungen ausgeschlossen werden.

Aus Symmetriegründen kommen nur noch die folgenden drei Strategien in Frage:

> Strategie Π_1:
> Starte Jobs 1 und 2 und lasse eine Maschine bis zur Fertigstellung des ersten Vorgangs frei. Dann fahre nach den folgenden Prioritätslisten fort:
>
> $$\begin{cases} 5 < 6 < 4 < 3, \text{ falls Job 1 zuerst endet,} \\ 3 < 4 < 5 < 6, \text{ falls Job 2 zuerst endet.} \end{cases}$$

[20] Elementare Überlegungen zeigen, daß dies die einzig notwendige Bedingung an a, b und c ist.

Strategie Π_2:
Starte Jobs 1, 2 und 3. Fahre nach den folgenden Prioritätslisten fort:

$$\begin{cases} 5 < 6 < 4, \text{ falls Job 1 zuerst endet,} \\ 4 < 5 < 6, \text{ falls Job 2 zuerst endet,} \\ 4 < 5 < 6, \text{ falls Job 3 zuerst endet.} \end{cases}$$

Strategie Π_3:
Starte Jobs 1, 2 und 3. Fahre nach den folgenden Prioritätslisten fort:

$$\begin{cases} 5 < 6 < 4, \text{ falls Job 1 zuerst endet,} \\ 4 < 5 < 6, \text{ falls Job 2 zuerst endet,} \\ 5 < 6 < 4, \text{ falls Job 3 zuerst endet.} \end{cases}$$

Alle Strategien sind offenbar Set-Strategien. Dabei sind nur Strategien Π_2 und Π_3 auch Set-Prioritätsstrategien, denn sie verhalten sich *greedy*. Alle Strategien versuchen „möglichst schnell die Kostenrate zu senken“.

Die Kostenrate ist derart gestaltet, daß das Ende einer der Vorgänge 1 und 2 die vollständige Information über die „beste“ weitere Vorgehensweise liefert, d. h. 3 und 4 *oder* 5 und 6 sind – falls möglich – als nächste zu starten.

Strategie Π_1 „wartet“ nach dem Start von 1 und 2 auf diese Information, und hat nach der ersten Vorgangsbeendigung zwei Maschinen für die beiden „richtigen“ Jobs zur Verfügung.

Strategien Π_2 und Π_3 entscheiden sich zum Zeitpunkt 0 *greedy* zum Start von Vorgang 3[21]. Bei diesen Strategien kann nach der ersten Vorgangsbeendigung daher nur *ein* neuer Job gestartet werden. Endet etwa Job 1 als erster, erweist sich diese Entscheidung als „Fehlentscheidung“.

Der Unterschied zwischen Strategien Π_2 und Π_3 sind die unterschiedlichen Aktionen für den Fall, daß Vorgang 3 als erster endet. Wiederum aus Symmetriegründen reicht es, nur diese beiden Strategien zu untersuchen.

Wir können nun die erwarteten Kosten $E(\kappa^{\Pi_k})$ für alle Strategien Π_k anhand der Rekursion (1.1) berechnen. Es ergeben sich nach elementarer, aber aufwendiger Rechnung folgende Resultate:

$$E(\kappa^{\Pi_1}) - E(\kappa^{\Pi_2}) = \frac{(a-b)(2x^3 - 10x^2 - 13x - 6)}{(2+x)^2(1+2x)^4}, \tag{1.2}$$

$$E(\kappa^{\Pi_2}) - E(\kappa^{\Pi_3}) = \frac{(a-b)x}{(2+x)(1+2x)^3}. \tag{1.3}$$

[21] Aus Symmetriegründen brauchen wir andere Alternativen nicht zu untersuchen.

Dieses Resultat ist folgendermaßen zu interpretieren:
Da wir $a < b$ vorausgesetzt hatten, gilt $a - b < 0$. (1.3) besagt dann, daß $E(\kappa^{\Pi_2}) < E(\kappa^{\Pi_3})$ für alle $x > 0$ ist. Strategie Π_3 ist also für eine beliebige Wahl von x Strategie Π_2 unterlegen. Daher braucht Π_3 nicht weiter berücksichtigt zu werden[22].

Welche der beiden Strategien Π_1 und Π_2 nun die *optimale* Strategie ist, hängt von x ab. Die einzig reelle Nullstelle des Zählerpolynoms in (1.2) liegt bei $x_0 \approx 6,14$, und es gilt

$$E(\kappa^{\Pi_1}) < E(\kappa^{\Pi_2}) \quad \text{genau dann, wenn } x > 6,14 \text{ ist.}$$

Wir erinnern daran, daß $E(P_1) = E(P_2) = \frac{1}{x}$ gilt. Strategie Π_1 ist also Strategie Π_2 genau dann überlegen, wenn die erwarteten Dauern der Vorgänge 1 und 2 in diesem Sinne „klein genug" sind.

Damit folgt insbesondere unter geeigneter Wahl von x:

Bemerkung 1.2 *Bereits für exponentielle 3-Maschinen Probleme ohne Präzedenzrelation existieren additive Kostenfunktionen, deren Optimalwert nicht durch Prioritätsstrategien erzielt werden kann.*

Wir wollen uns anhand einer Realisierung der Vorgangsdauern schematisch überlegen, wie der Vorteil von Strategie Π_1 gegenüber Strategie Π_2 zustandekommen kann.

Wir betrachten die folgende Realisierung der Vorgangsdauern:

$$p_1 = \tfrac{1}{2},\ p_2 = 2,\ p_3 = p_4 = p_5 = p_6 = 1$$

Abbildung 1.6 zeigt die Ergebnisse der beiden Strategien mit zugehörigen Kostenraten.

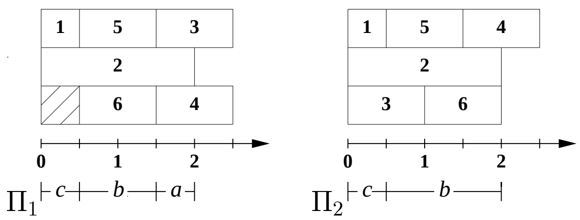


Abbildung 1.6: Gantt-Diagramm für Strategien Π_1 und Π_2

An der Abbildung lesen wir ab:
Π_1 verursacht Kosten in einer Höhe von $\frac{1}{2}c + b + \frac{1}{2}a$. Die Kosten von Strategie Π_2 betragen $\frac{1}{2}c + 1\frac{1}{2}b$. Damit beträgt die Differenz genau $\frac{1}{2}(b - a)$ zu Ungunsten von Strategie Π_2.

[22] Daß Π_3 stets Π_2 unterlegen ist, kann man folgendermaßen motivieren: Die Strategien unterscheiden sich nur durch ihre Reaktion auf das frühzeitige Ende von Vorgang 3. Beide entscheiden sich zum Zeitpunkt 0 zum Start von Job 3, sie „hoffen" implizit auf das frühzeitige Ende von Job 2 (dann war die Wahl richtig). Endet Job 3 als erster, setzt Π_2 dieselbe Vorgehensweise fort, und startet Job 4. Strategie Π_3 allerdings „wechselt" die Vorgehensweise, und startet einen der Jobs 5 und 6. Die „Konsequenz" von Strategie Π_2 zahlt sich durch einen geringeren Erwartungswert für die Kosten aus.

Es gibt natürlich ebenfalls Realisierungen für P, in denen Strategie Π_2 ein besseres Ergebnis erzielt. Wir wollen es mit der Feststellung bewenden lassen, daß die Vorteile von Π_1 in der Vermeidung einer „Fehlentscheidung" liegen, die Vorteile von Π_2 dagegen in der *greedyness* begründet sind.

Im Hinblick auf die *deliberate idleness*-Vermutung sei noch folgende Bemerkung angebracht:

Bemerkung 1.3 *Die oben definierte Kostenrate g ist nicht submodular, denn es gilt*

$$\underbrace{g(\{1,3\})}_{=b} \not\leq \underbrace{g(\{1\}) + g(\{3\})}_{=a}.$$

Es läßt sich erkennen, daß für die Submodularität der Kostenrate g jedenfalls $a = b$ gelten muß[23].

Unter Beachtung von (1.2) und (1.3) haben dann *alle* Strategien stets dieselben erwarteten Kosten. Insbesondere wird das Optimum dann bereits durch die Prioritätsstrategien Π_2 und Π_3 erreicht.

Dieses Beispiel zeigt damit einerseits, daß geplante Leerzeiten bereits bei exponentiellen Schedulingproblemen auf 3 Maschinen ohne Reihenfolgebeziehungen notwendig sein können.

Es nährt andererseits zusätzlich die Vermutung, daß Submodularität der Kostenrate eine ausreichende Forderung zur Unterdrückung solcher Phänomene sein könnte.

Im nächsten Abschnitt werden wir sehen, daß derartige Phänomene bei Problemen mit nur zwei Maschinen nicht auftreten können.

1.5.2 2-Maschinen Probleme und Deliberate Idleness

Wir betrachten exponentielle 2-Maschinen Probleme ohne Präzedenzrelation. Wir haben also $(\Theta, \mathcal{N}, P, \kappa)$ mit

$$\begin{aligned} \prec &= \emptyset, \\ \mathcal{N} &= \{S \subseteq I \mid |S| = 3\}, \\ P &= \bigotimes_{i \in I} \exp(\lambda_i). \end{aligned} \tag{1.4}$$

Ferner sei g die Kostenrate zur additiven Kostenfunktion κ. Unter diesen Voraussetzungen lohnt sich der freiwillige Verzicht auf verfügbare Maschinenkapazität nicht, und es gilt:

[23] g ist genau dann submodular, wenn $a = b$ und $2a \geq c$ gilt. Man überlegt sich, daß abgesehen von dem trivialen Fall $a = b = c$ auch unter diesen zusätzlichen Bedingungen wieder nur Strategien in Frage kommen, die Jobs 1 und 2 zu Beginn starten.

Satz 1.2 *Unter den Bedingungen (1.4) existiert unter allen (Set-)Strategien immer eine optimale Set-Prioritätsstrategie.*
Ist ferner g *oder* $\rho(\kappa(\cdot)) : 2^I \longrightarrow \mathbb{R}_+$ *streng isoton, so ist jede optimale Strategie bereits eine Set-Prioritätsstrategie.*

Beweis. Wir gehen folgendermaßen vor:
Ausgehend von einer beliebigen *optimalen (Set-)Strategie* Π, werden wir eine Set-Prioritätsstrategie Π' konstruieren, so daß $E(\kappa^{\Pi'}) \leq E(\kappa^{\Pi})$ gilt.

Wir benötigen dazu die folgende Aussage, die wir im Anschluß beweisen werden[24]:

> Die Abbildung $\rho(\kappa(\cdot)) : 2^I \longrightarrow \mathbb{R}_+$ ist monoton nichtfallend,
> d. h. $\rho(\kappa(S)) \leq \rho(\kappa(T))$ für alle $S \subseteq T$.

Wir gehen in zwei Schritten vor:
Im ersten Schritt nehmen wir an, daß Π zum Zeitpunkt 0 eine Maschine leer läßt. Wir ersetzen Π durch eine Strategie Π_1, die frühstens zur ersten Vorgangsbeendigung eine Maschine leer läßt und zeigen, daß $E(\kappa^{\Pi_1}) \leq E(\kappa^{\Pi})$ gilt.
Der zweite Schritt besteht lediglich aus der induktiven Fortsetzung des Argumentes.

Schritt 1: Angenommen Π lasse zum Zeitpunkt 0 eine Maschine leer. O. B. d. A. sei k der Job, den Π zum Zeitpunkt 0 startet. Da Π eine Strategie ist, ist k eindeutig bestimmt. Die nächste Aktion findet statt sobald k beendet ist. Da der *Zustand* des Projektes nach Beendigung von k offenbar stets derselbe ist, ist die entsprechende Aktion von Π ebenfalls stets dieselbe. Da Π keine totalen Leerzeiten zuläßt wird wenigstens *ein* Job zu diesem Zeitpunkt gestartet. Sei i dieser Job. Ob Π auf der zweiten Maschine einen weiteren Job startet oder nicht spielt keine Rolle, und wir schreiben dafür „$*$“. Ferner schreiben wir für die von Π gewählte Startmenge nach Fertigstellung von i stellvertretend „$**$“. Abbildung 1.7 zeigt das Verhalten von Π.

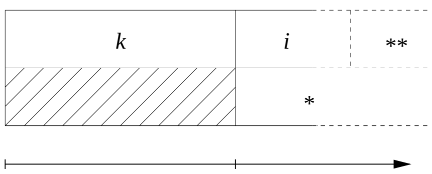


Abbildung 1.7: Strategie Π

Wir definieren eine (Set-)Strategie Π_1 durch:

- Starte Vorgänge k *und* i zum Zeitpunkt 0.

[24] Zur Erinnerung: $\rho(\kappa(R))$ ist der Optimalwert eines Schedulingproblems unter Einschränkung auf eine Vorgangsmenge $R \subseteq I$.

- Falls k zuerst endet, fahre wie Π fort (Startmenge „$*$")
- Falls i zuerst endet, lasse bis zum Ende von Job k eine Maschine leer. Sobald auch k beendet ist, fahre wie Π fort (Startmenge „$* \cup **$").

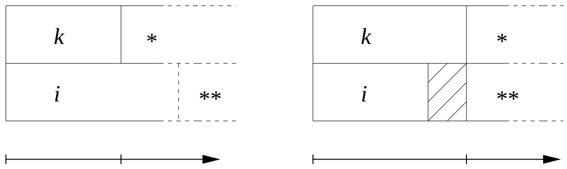


Abbildung 1.8: Strategie Π_1

Abbildung 1.8 zeigt das Verhalten von Π_1. Zur *Wohldefiniertheit* von Π_1 müssen wir überprüfen, ob die Zustände des Projektes, zu denen wir gemäß Π fortfahren auch unter Π selbst auftreten können. Nur dann ist die Vorschrift „fahre wie Π fort" auch zulässig.

Endet k zuerst, ist der Zustand durch „k ist beendet, i läuft noch" gegeben. Diese Situation kann offenbar auch unter Π eintreten.
Endet i zuerst, wartet Π_1 noch das Ende von k ab. Der Zustand ist dann durch „i und k sind beendet" gegeben. Auch diese Situation kann unter Π eintreten. Daher ist Π_1 tatsächlich wohldefiniert.

Es gilt nun unter Benutzung der Rekursion (1.1)

$$E(\kappa^{\Pi}) = \frac{g(I)}{\lambda_k} + \rho\Big(\kappa(I - \{k\})\Big),$$

denn nach Voraussetzung war Π optimal, und daher ist Π auch optimal für das Subproblem auf $I - \{k\}$. Für Π_1 liefert (1.1):

$$\begin{aligned} E(\kappa^{\Pi_1}) = \frac{g(I)}{\lambda_k + \lambda_i} \quad &+ \quad \frac{\lambda_k}{\lambda_k + \lambda_i} \underbrace{E\Big(\kappa^{\Pi}(I - \{k\} \mid \{i\})\Big)}_{=\rho(\kappa(I-\{k\}))} \\ &+ \quad \frac{\lambda_i}{\lambda_k + \lambda_i} \left(\frac{g(I - \{i\})}{\lambda_k} + \underbrace{E\Big(\kappa^{\Pi}(I - \{k, i\})\Big)}_{=\rho(\kappa(I-\{k,i\}))} \right). \end{aligned}$$

Diese Darstellung ergibt sich, da Π_1 an entsprechender Stelle gemäß der (optimalen) Strategie Π verfährt. Demnach erhalten wir für $E(\kappa^{\Pi_1}) - E(\kappa^{\Pi})$ folgenden Term:

$$\begin{aligned} \frac{1}{\lambda_k + \lambda_i} \quad & \Bigg(g(I) \ + \ \lambda_k \, \rho(\kappa(I - \{k\})) \ + \ \frac{\lambda_i}{\lambda_k} g(I - \{i\}) \ + \ \lambda_i \, \rho(\kappa(I - \{k, i\})) \\ & - \frac{\lambda_k + \lambda_i}{\lambda_k} g(I) \ - \ (\lambda_k + \lambda_i) \, \rho(\kappa(I - \{k\})) \Bigg). \end{aligned}$$

Wir kürzen und erhalten letztlich:

$$
\begin{aligned}
& E(\kappa^{\Pi_1}) - E(\kappa^{\Pi}) \\
= \quad & \frac{1}{\lambda_k + \lambda_i} \left(\frac{\lambda_i}{\lambda_k} \underbrace{\Big(g(I - \{i\}) - g(I)\Big)}_{\leq 0} + \lambda_i \underbrace{\Big(\rho(\kappa(I - \{k, i\})) - \rho(\kappa(I - \{k\}))\Big)}_{\leq 0} \right) \\
\leq \quad & 0.
\end{aligned}
$$

Dabei gilt die letzte Abschätzung, da sowohl g als auch $\rho(\kappa(\cdot))$ monoton nichtfallend sind. Also ist Π_1 tatsächlich nicht schlechter als Π.

Insbesondere ergibt sich für den Fall strenger Isotonie von g oder $\rho(\kappa(\cdot))$ der Widerspruch, daß Π nicht optimal gewesen sein kann. Da wir dieselbe Konstruktion auch für spätere Aktionszeitpunkte durchführen können, gilt daher die zweite Behauptung des Satzes.

Schritt 2: Π_1 ist optimal und läßt zu Beginn keine Maschine leer. Ausgehend von Π_1 können wir genau wie in Schritt 1 eine Strategie Π_2 konstruieren, die auch nach der ersten Vorgangsbeendigung *keine* Maschine leer läßt. Π_2 startet ebenfalls die Vorgänge k und i und für den Fall, daß Π_1 eine Maschine leer läßt, wiederholen wir die obige Konstruktion. Auf diese Weise erhalten wir sukzessive eine endliche Folge von Strategien mit

$$E(\kappa^{\Pi}) \geq E(\kappa^{\Pi_1}) \geq E(\kappa^{\Pi_2}) \geq \cdots \geq E(\kappa^{\Pi_j}),$$

und Π_j ist schließlich eine Strategie ohne geplante Leerzeiten. Damit haben wir eine Set-Prioritätsstrategie gefunden, die mindestens so gut ist wie Π. ■

Worin liegt der Unterschied zwischen zwei und mehr Maschinen ?

Wir wollen an dieser Stelle kurz darauf eingehen, wieso derselbe Beweis bei *mehr* als zwei Maschinen nicht greift.
Der wesentliche Unterschied bei nur zwei Maschinen liegt darin, daß im Falle einer geplanten Leerzeit durch eine optimale Strategie gleichzeitig *genau ein* Vorgang in Arbeit ist. Daher ist der Zustand zum Zeitpunkt der nächsten Vorgangsbeendigung bereits im voraus bekannt (er wird durch das Ende des Vorganges bestimmt). Die entsprechende Startmenge für den nächsten Aktionszeitpunkt ist ebenfalls im voraus bekannt.

Wie wir gesehen haben, können wir daher eine Leerzeit durch simple „Vorverlegung" eines (eindeutig bestimmten) Startvorganges vermeiden, ohne die erwarteten Kosten zu erhöhen.

Bei mehr als zwei Maschinen hingegen ist eine derartige *eindeutige* Bestimmung eines solchen Startvorganges nicht möglich:
Nehmen wir an, eine optimale Strategie lasse eine Maschine leer. Im allgemeinen sind

zur gleichen Zeit *mehrere* Vorgänge in Arbeit. Es besteht daher Unsicherheit über die nächste Vorgangsbeendigung und den nächsten Zustand. Es existiert kein Vorgang, der „zum nächsten Aktionszeitpunkt ohnehin gestartet würde“ (einen solchen Vorgang könnten wir ohne Kostennachteil wie oben vorverlegen). Abbildung 1.9 zeigt den Sachverhalt schematisch bei drei Maschinen.

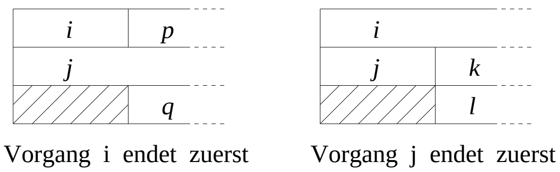


Abbildung 1.9: Verschiedene Startmengen je nach Vorgangsbeendigung

Wenn wir nun eine geplante Leerzeit durch Vorverlegung eines Vorgangs vermeiden wollen, entsteht das Problem die Erwartungswerte der Strategien auf verschiedenen Subproblemen gegeneinander abzuschätzen. Im letzten Abschnitt dieses Kapitels werden wir den Versuch einer solchen Abschätzung skizzieren.

Wir müssen noch das folgende Lemma beweisen, dessen Aussage für beliebige m-Maschinen Probleme gilt.

Lemma 1.5 *Wir betrachten ein exponentielles Schedulingproblem auf m Maschinen ohne Präzedenzrelation: $((I,\emptyset), \{S \subseteq I \mid |S| = m+1\}, \bigotimes_{i\in I} exp(\lambda_i), \kappa)$. Sei g die zugehörige Kostenrate. Dann ist die Abbildung*

$$\rho(\kappa(\cdot)) : 2^I \longrightarrow \mathbb{R}_+$$

monoton nichtfallend.

Diese Tatsache ist nicht erstaunlich, da sie lediglich besagt, daß weniger Vorgänge keine höheren Kosten verursachen[25].

Beweis. Es ist

$$\rho(\kappa(S)) \leq \rho(\kappa(T)) \quad \text{für alle } S \subseteq T \subseteq I$$

zu zeigen. Wir gehen folgendermaßen vor: Sei $S \subseteq T$ beliebig. Ausgehend von einer optimalen Set-Strategie Π^T für das Problem auf der Vorgangsmenge T definieren wir eine Set-Strategie Π^S auf der kleineren Vorgangsmenge S mit $E(\kappa^{\Pi^S}(S)) \leq E(\kappa^{\Pi^T}(T))$. Damit folgt wegen $\rho(\kappa(S)) \leq E(\kappa^{\Pi^S}(S))$ die Behauptung.

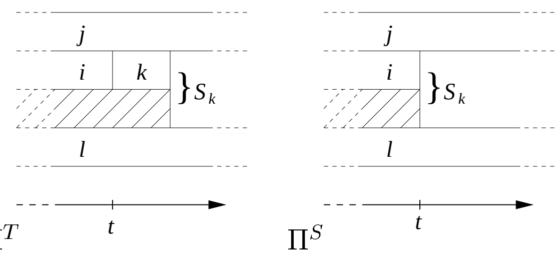
Es sei Π^S analog zu Π^T definiert, solange kein Job aus $T-S$ gestartet wird. Wir betrachten nun den frühstmöglichen Aktionszeitpunkt t, zu dem Π^T einen Job $k \in T-S$ startet (Es

[25]Dennoch bedarf die Behauptung eines Beweises, denn Anomalien und Instabilitäten sind im Scheduling keine Besonderheit.

sei an die Darstellung von Set-Strategien durch Entscheidungsbäume erinnert; vergleiche Abbildung 1.4).

Für den Fall, daß unter Π^T Vorgang k zuerst endet, hält Π^T eine (zulässige) Startmenge $S_k \subseteq T$ bereit. Π^S starte anstatt von k die Startmenge S_k bereits zum Zeitpunkt t.

Daß die so gewählte Startmenge zulässig ist, sieht man daran, daß der Zustand unter Π^S zum Zeitpunkt t genau dem Zustand unter Π^T *nach* Beendigung von Vorgang k entspricht[26]. Abbildung 1.10 zeigt zur Verdeutlichung der Konstruktion die beiden Strategien zum Zeitpunkt t.



Π^T Π^S

Abbildung 1.10: Strategien Π^T und Π^S

Wir definieren Π^S also sukzessive dadurch, daß für *jeden* Job $k \in T - S$ stattdessen die Menge S_k gestartet wird. Falls S_k wieder einen Job $m \in T - S$ enthält, ersetzen wir diesen durch S_m etc.. Die Wohldefiniertheit von Π^S folgt daraus, daß alle Zustände, die bei dem Subproblem auf S unter Π^S auftreten können, auch unter Π^T auftreten können.

Es gilt nun per Definition von Π^S für *jede* Realisierung p der Vorgangsdauern $(\Pi^S(p))_i \leq (\Pi^T(p))$ für alle $i \in I$[27],und damit folgt

$$(\Pi^S(p))_i + p_i \leq (\Pi^T(p))_i + p_i \text{ für alle } i \in I.$$

Wir setzen dabei für alle Jobs, die nicht im jeweiligen Subproblem enthalten sind Dauer, Start- und Fertigstellungszeit auf 0.

Das bedeutet, daß bezüglich jeder Realisierung p der Vorgangsdauern alle Jobs unter Π^S nicht später fertiggestellt werden als unter Π^T. Aufgrund der Regularität der Kostenfunktion folgt dann

$$\kappa(\Pi^S(p) + p) \leq \kappa(\Pi^T(p) + p) \text{ für alle } p \in \mathbb{R}^n_+.$$

Damit gilt auch $E(\kappa^{\Pi^S}) \leq E(\kappa^{\Pi^T})$, was zu zeigen war. ■

[26] Diese Tatsache folgt aus der Gedächtnislosigkeit der Dauerverteilungen und daraus, daß wir nur Set-Strategien betrachten.

[27] Zur Erinnerung: $(\Pi(p))_i$ ist die Startzeit von Vorgang i unter Strategie Π bei Realisierung p der Dauern.

1.5.3 Optimalitätsresultate über statische Prioritätsstrategien

In diesem Abschnitt wollen wir einige Resultate zur Optimalität *statischer* Prioritätsstrategien angeben. Aufgrund der *greedyness* von statischen Prioritätsstrategien wird dadurch insbesondere die Frage nach dem Auftreten von *deliberate idleness*-Phänomenen negativ beantwortet.

Sätze 1.3 und 1.4 und deren Beweise findet man etwa in [Käm 85] oder [Wei 82].

Wir betrachten exponentielle m-Maschinen Probleme ohne Präzedenzrelation. Wir haben also

$$((I,\emptyset),\{S\subseteq I \mid |S|=m+1\},\bigotimes_{i\in I}\exp(\lambda_i),\kappa). \tag{1.5}$$

Das erste Resultat liefert eine Aussage über die Optimalität einer Prioritätsstrategie zur Minimierung der Gesamtprojektdauer[28]:

Satz 1.3 (Optimalität von LEPT für $\kappa = C_{\max}$) *Unter den Voraussetzungen (1.5) ist die durch LEPT (d. h. longest expected processing time first) induzierte statische Prioritätsstrategie optimal unter allen (Set-)Strategien zur Minimierung der Gesamtprojektdauer.*

Für die Summe der Fertigstellungszeiten gilt:

Satz 1.4 (Optimalität von SEPT für $\kappa = \sum_{i\in I} C_i$) *Unter den Voraussetzungen (1.5) ist die durch SEPT (d. h. shortest expected processing time first) induzierte statische Prioritätsstrategie optimal unter allen (Set-)Strategien zur Minimierung der Summe der Fertigstellungszeiten.*

Für die *gewichtete* Summe der Fertigstellungszeiten, also $\kappa = \sum_{i\in I} w_i C_i$ geben wir ein Resultat für den Fall einer Maschine an.

Wir nennen w_i die *absoluten Kosten* von Vorgang i und den Quotienten $\frac{w_i}{E(P_i)}$ die *relativen Kosten* von Vorgang i. Im Einmaschinenfall erweist sich diejenige statische Prioritätsstrategie als optimal, die gemäß der Regel „Jobs mit maximalen relativen Kosten zuerst" verfährt. Dieses Ergebnis wurde bereits 1956 von Smith [Smi 56] durch ein Austauschargument für *deterministisches* Scheduling bewiesen. Aus diesem Grund wird die zugehörige Prioritätsstrategie auch Smith's *ratio rule* genannt.

[28]Dieses Resultat ist aus einem komplexitätstheoretischen Grund von besonderem Interesse: Das zugehörige *deterministische* Problem bei beliebigen, festen Dauern der Jobs ist bekanntlich $\mathcal{NP}$-schwer; vergleiche [GJ 79]. Erstaunlicherweise ist die stochastische Version bei exponentialverteilten Dauern polynomial „lösbar".

Das Austauschargument kann auch auf den stochastischen Fall übertragen werden, und es gilt[29]:

Satz 1.5 (Optimalität der Ratio Rule für m=1 und $\kappa = \sum_{i\in I} w_i C_i$) *Für das Problem $((I,\emptyset), \{S \subseteq I \mid |S| = 2\}, \bigotimes_{i\in I} P_i, \sum_{i\in I} w_i C_i)$ mit stochastisch unabhängig verteilten Dauern P_i definieren wir eine Permutation π auf I durch*

$$\frac{w_{\pi_1}}{E(P_{\pi_1})} \geq \frac{w_{\pi_2}}{E(P_{\pi_2})} \geq \cdots \geq \frac{w_{\pi_n}}{E(P_{\pi_n})}.$$

Dann ist die statische Prioritätsstrategie L_π optimal unter allen (Set-)Strategien.

Wir benötigen für das nächste Resultat den Begriff der Verträglichkeit absoluter und relativer Kosten.

Definition 1.7 (Verträglichkeit relativer und absoluter Kosten) *Für ein exponentielles Schedulingproblem mit Kostenfunktion $\kappa = \sum_{i\in I} w_i C_i$ nennen wir absolute und relative Kosten verträglich, wenn eine Permutation π auf I existiert so daß folgende Bedingungen gleichzeitig erfüllt sind:*

(1) $w_{\pi_1} \geq w_{\pi_2} \geq \cdots \geq w_{\pi_n}$,

(2) $\frac{w_{\pi_1}}{E(P_{\pi_1})} \geq \frac{w_{\pi_2}}{E(P_{\pi_2})} \geq \cdots \geq \frac{w_{\pi_n}}{E(P_{\pi_n})}$.

Es gilt damit[30]:

Satz 1.6 (Optimalität der Ratio Rule für beliebiges m und $\kappa = \sum_{i\in I} w_i C_i$) *Unter den Voraussetzungen (1.5) mit $\kappa = \sum_{i\in I} w_i C_i$ seien absolute und relative Kosten verträglich. Wir definieren eine Permutation π auf I, so daß gleichzeitig*

$$\frac{w_{\pi_1}}{E(P_{\pi_1})} \geq \frac{w_{\pi_2}}{E(P_{\pi_2})} \geq \cdots \geq \frac{w_{\pi_n}}{E(P_{\pi_n})}$$

und

$$w_{\pi_1} \geq w_{\pi_2} \geq \cdots \geq w_{\pi_n}$$

erfüllt sind. Dann ist die statische Prioritätsstrategie L_π optimal unter allen (Set-)-Strategien.

Diese Resultate zeigen, daß die Voraussetzungen (1.5) in einigen Fällen bereits die Einschränkung auf die Klasse der statischen Prioritätsstrategien erlauben.

Für die gewichtete Summe der Fertigstellungszeiten benötigen wir allerdings bei mehr als einer Maschine die zusätzliche Bedingung der Verträglichkeit absoluter und relativer Kosten. Im folgenden Abschnitt untersuchen wir das Problem *ohne* diese zusätzliche Bedingung.

[29]Ein Beweis zu Satz 1.5 wird in Satz 2.14 dieser Arbeit angegeben.
[30]Vergleiche [Käm 85, Satz 3.2].

1.5.4 $\sum_{i \in I} w_i C_i$: Die Notwendigkeit dynamischer Strategien

Vor dem Hintergrund der bisher vorgestellten Beispiele und Resultate stellt sich die Frage, welche Klasse von (Set-)Strategien für exponentielle Probleme ohne Präzedenzrelation hinreichend zur Minimierung der gewichteten Summe der Fertigstellungszeiten ist.

Wie wir gesehen haben, wird das Optimum unter zusätzlichen Voraussetzungen bereits durch eine statische Prioritätsstrategie erreicht. Der *deliberate idleness*-Vermutung folgend nimmt man

$$\rho_{SET-PRIOR}(\kappa) = \rho(\kappa)$$

an. Daß die Klasse der *statischen* Prioritätsstrategien nicht reichhaltig genug ist, soll in diesem Abschnitt gezeigt werden.

Wir betrachten exponentielle m-Maschinen Probleme mit der gewichteten Summe der Fertigstellungszeiten, also

$$((I, \emptyset), \{S \subseteq I \mid |S| = m+1\}, \bigotimes_{i \in I} \exp(\lambda_i), \sum_{i \in I} w_i C_i) \tag{1.6}$$

für beliebiges $m \geq 2$. Dann gilt:

Bemerkung 1.4 *Sind unter den Voraussetzungen (1.6) absolute und relative Kosten nicht verträglich, so ist die Klasse der statischen Prioritätsstrategien nicht mehr hinreichend, um in der Klasse aller (Set-)Strategien den Optimalwert zu erreichen.*

Diese Tatsache wird durch das folgende Beispiel gezeigt.

Beispiel 1.2 (Die Notwendigkeit dynamischer Strategien) Wir betrachten ein 2-Maschinen Problem mit vier Vorgängen exponentialverteilter Dauern[31]. Die Parameter seien:

$$w_1 = 10^6, \quad w_2 = 10^6, \quad w_3 = 2, \quad w_4 = 10,$$
$$\lambda_1 = x \geq 1, \quad \lambda_2 = 1, \quad \lambda_3 = 10, \quad \lambda_4 = 1.$$

Dann gilt mit $E(P_i) = \frac{1}{\lambda_i}$:

$$\begin{array}{ccccccc} \frac{w_1}{E(P_1)} & \geq & \frac{w_2}{E(P_2)} & > & \frac{w_3}{E(P_3)} & > & \frac{w_4}{E(P_4)}, \\ w_1 & \geq & w_2 & > & w_3 & < & w_4. \end{array}$$

Damit sind relative und absolute Kosten *nicht* verträglich. Eine statische Prioritätsliste gemäß absoluter Kosten wäre $1 < 2 < 4 < 3$, gemäß relativer Kosten hingegen $1 < 2 < 3 < 4$. Aufgrund der Wahl der Gewichte w_i kommen nur solche Strategien in Frage, die Jobs 1 und 2 zu Beginn starten.

[31] Das Beispiel ist eine Erweiterung von [Käm 85, Beispiel 3.5].

Wir wissen aus Satz 1.2 bereits, daß das Optimum durch eine (Set-)Prioritätsstrategie erreicht wird. Es kommen aus Symmetriegründen genau vier Strategien in Frage:

Strategie Π_1:
Starte Jobs 1 und 2. Dann fahre nach den folgenden Prioritätslisten fort:

$$\begin{cases} 3 < 4, \text{ falls Job 1 zuerst endet,} \\ 4 < 3, \text{ falls Job 2 zuerst endet.} \end{cases}$$

Strategie Π_2:
Plane nach der statischen Prioritätsliste $1 < 2 < 3 < 4$ (*ratio rule*).

Strategie Π_3:
Plane nach der statischen Prioritätsliste $1 < 2 < 4 < 3$.

Strategie Π_4:
Starte Jobs 1 und 2. Dann fahre nach den folgenden Prioritätslisten fort:

$$\begin{cases} 4 < 3, \text{ falls Job 1 zuerst endet,} \\ 3 < 4, \text{ falls Job 2 zuerst endet.} \end{cases}$$

Wir können die Erwartungswerte der einzelnen Strategien nun mit (1.1) bzw. anhand von Lemma 1.1 über $E(\kappa^\Pi) = \sum_{i \in I} w_i \, E(\mathcal{C}^\Pi)$ berechnen. Es ergibt sich:

$$\begin{aligned} E(\kappa^{\Pi_1}) - E(\kappa^{\Pi_2}) &= \frac{2(5-4x)}{(1+x)^2(10+x)} \\ E(\kappa^{\Pi_1}) - E(\kappa^{\Pi_3}) &= \frac{-x}{11(1+x)} \\ E(\kappa^{\Pi_2}) - E(\kappa^{\Pi_4}) &= \frac{-x}{11(1+x)} \end{aligned}$$

Damit ist Π_1 genau dann optimal, wenn $x \geq \frac{5}{4}$ gilt. Andernfalls ist Π_2 optimal.
Insbesondere ist für $x > \frac{5}{4}$ die *dynamische* Prioritätsstrategie Π_1 die einzig optimale Strategie. ■

Beispiel 1.2 beschreibt zwei Phänomene:
Wie wir bereits bemerkt hatten, kann man zur Minimierung der gewichteten Summe der Fertigstellungszeiten ohne Verträglichkeit absoluter und relativer Kosten auf dynamische Strategien nicht verzichten.

Weiter können wir feststellen: Der Unterschied zwischen Strategien Π_1 und Π_2 liegt lediglich in der Planungsreihenfolge der Vorgänge 3 und 4 für den Fall, daß Vorgang 2 als

erster endet. Die Optimalität einer der beiden Strategien ist in diesem Beispiel von der erwarteten Dauer von Vorgang 1 abhängig.
Damit muß die Entscheidung einer *optimalen* Strategie über die als nächstes zu startenden Vorgänge auch von Informationen über *noch nicht beendete Vorgänge* abhängen. Wir folgern:

Bemerkung 1.5 *Unter den Voraussetzungen (1.6) muß eine optimale Strategie im allgemeinen auch die zur Verfügung stehenden Informationen über noch andauernde Vorgänge nutzen.*

1.5.5 $\sum\limits_{i \in I} w_i C_i$: Deliberate Idleness vs. Greedyness

Zum Abschluß des ersten Kapitels werden wir uns explizit mit der Quantifizierung von Vorteilen von *deliberate idleness*, bzw. *greedyness* unter den Voraussetzungen (1.6) beschäftigen.

Wir werden eine Prioritätsregel für die zeitlich „letzte" (echte) Entscheidung einer Strategie angeben, die für *jede* optimale Strategie bindend ist. Diese Regel liefert nachträglich den Hintergrund für die Optimalität einer der beiden Strategien Π_1 bzw. Π_2 aus Beispiel 1.2.

Die Frage nach einer Regel zur Konstruktion einer optimalen Prioritätsstrategie bleibt allerdings unbeantwortet (wie auch die *deliberate idleness*-Frage für $\sum_{i \in I} w_i C_i$ im Falle der Nichtverträglichkeit relativer und absoluter Kosten).

Wir erinnern an die Bezeichnungen:

- $G \subseteq I$ für die Menge der bereits *vor* t gestarteten Vorgänge,
- $L \subseteq G$ für die Menge derjenigen Vorgänge, die bereits gestartet, aber bis einschließlich t noch nicht beendet sind, und
- $R = I - G$ für die Menge der Vorgänge, die vor t noch nicht gestartet wurden.

Die erwarteten Kosten für ein Subproblem ab t unter einer Strategie Π sind durch $E(\kappa^{\Pi}(L \cup R \mid L))$ gegeben. Für eine *optimale* Strategie Π^{opt} muß für jeden Aktionszeitpunkt $E(\kappa^{\Pi^{opt}}(L \cup R \mid L)) = \rho(\kappa(L \cup R \mid L))$ gelten, denn eine optimale Strategie ist auf jedem der induzierten Subprobleme ebenfalls optimal.

Wenn wir noch weitere Bedingungen oder Einschränkungen betrachten, bezeichnen wir den Optimalwert durch

$$\rho(\ \kappa(L \cup R \mid L) \quad \mid \quad \textit{Bedingung}\ ).$$

Zusammen mit Lemma 1.1 erhalten wir

$$E(\kappa^{\Pi}(L \cup R \mid L)) \;=\; \sum_{i \in L \cup R} w_i \left(E(\mathcal{C}_i^{\Pi}) - t \right).$$

Wir werden im folgenden der Kürze halber $t \equiv 0$ voraussetzten.

Bemerkung 1.6 *Sei ein stochastisches Schedulingproblem unter den Voraussetzungen (1.6) gegeben. Dann ist eine Strategie nur dann optimal, wenn zu jedem Aktionszeitpunkt t mit $|R| + |L| = m$ alle Vorgänge $i \in R$ gestartet werden.*

Diese Tatsache ist trivial:
$|R| + |L| = m$ besagt, daß noch genau so viele Vorgänge auf ihre Bearbeitung warten, wie Maschinen frei sind. Die *einzige* Strategie minimaler Kosten ist dann, alle Vorgänge zu starten. Der Vollständigkeit halber erwähnen wir

$$\rho(\ \kappa(R \cup L \mid L) \quad | \quad |R| + |L| = m\ ) \quad = \quad \sum_{i \in R \cup L} \frac{w_i}{\lambda_i}.$$

Das nächste Lemma bezieht sich auf den Fall, daß genau ein Vorgang mehr auf seine Bearbeitung wartet, als Maschinen zur Verfügung stehen (Es gilt dann $|R| + |L| = m + 1$). Sei o. B. d. A. $R = \{1, \ldots, k\}$, und so indiziert, daß

$$\frac{w_1}{\sum\limits_{i \in L \cup (R - \{1\})} \lambda_i} \quad \geq \quad \frac{w_2}{\sum\limits_{i \in L \cup (R - \{2\})} \lambda_i} \quad \geq \quad \cdots \geq \quad \frac{w_k}{\sum\limits_{i \in L \cup (R - \{k\})} \lambda_i} \tag{1.7}$$

gilt.

Lemma 1.6 *Sei ein stochastisches Schedulingproblem unter den Voraussetzungen (1.6) gegeben. Dann ist eine Strategie nur dann optimal, wenn zu jedem Aktionszeitpunkt t mit $|R| + |L| = m + 1$ entsprechend der Prioritätsliste (1.7) maximal viele Vorgänge (also $|R| - 1$ Stück) gestartet werden.*

Eine wesentliche Aussage des Lemmas ist die *greedyness jeder* optimalen Strategie „am Ende“ des Planungshorizontes.

Beweis. Wir betrachten eine *beliebige* Strategie Π eingeschränkt auf das Subproblem ab t. Wir untersuchen, wann Π minimale erwartete (Rest-)Kosten hat. Eine optimale Strategie Π^{opt} für das gesamte Problem darf für das Subproblem ab t dann nicht *mehr* Kosten als Π verursachen.

Π starte zum Zeitpunkt t eine beliebige Startmenge $S \subset R$ (Es gilt $0 \leq |S| \leq |R| - 1$). Wir wollen uns nun überlegen, daß Π – um minimale Kosten zu haben – im nächsten Aktionszeitpunkt t' alle restlichen Jobs $R - S$ starten muß.

Wir betrachten den Zeitpunkt t' der nächsten Vorgangsbeendigung. Für die zugehörigen Mengen L', bzw. R' gilt: $|R'| = |R| - |S|$ und $|L'| = |L| + |S| - 1$ (denn genau ein Vorgang wird zu t' beendet). Damit folgt

$$|R'| = |R| - |S| = (m - |L| + 1) - (|L'| - |L| + 1) = m - |L'|.$$

Wenn Π also minimale Kosten haben soll, müssen zum Zeitpunkt t' aufgrund von Bemerkung 1.6 *alle* Vorgänge aus R' gestartet werden.

Damit erhalten wir für die erwarteten Fertigstellungszeiten der einzelnen Vorgänge:

- Für alle $i \in L \cup S$ gilt
$$E(\mathcal{C}_i^{\Pi}) = \frac{1}{\lambda_i} \quad \forall\, i \in L \cup S,$$
denn diese Vorgänge starten (bzw. laufen bereits) zum Zeitpunkt t.

- Für alle $i \in R - S$ ist die erwartete *Startzeit* genau $1/\sum_{i \in L \cup S} \lambda_i$, denn diese Vorgänge starten nach den bisherigen Überlegungen alle zum Zeitpunkt der ersten Vorgangsbeendigung, also zum Zeitpunkt $\min\{E(P_i) \mid i \in L \cup S\} = 1/\sum_{i \in L \cup S} \lambda_i$. Damit erhalten wir für die erwartete Fertigstellungszeit für diese Vorgänge:
$$E(\mathcal{C}_i^{\Pi}) = \frac{1}{\sum_{i \in L \cup S} \lambda_i} \quad + \quad \frac{1}{\lambda_i} \quad \forall\, i \in R - S.$$

Die erwarteten Kosten für Π betragen demnach:

$$E\Big(\kappa^{\Pi}(L \cup R \mid L) \quad \mid \quad |R| + |L| = m + 1\Big) = \sum_{i \in L \cup R} \frac{w_i}{\lambda_i} \quad + \quad \sum_{i \in R - S} \frac{w_i}{\sum_{j \in L \cup S} \lambda_j}.$$

Also ist Π genau dann optimal, wenn die Startmenge S derart gewählt wird, daß der zweite Summand minimal ist.

Offenbar ist der zweite Summand *streng monoton fallend* in S. Die erwarteten Kosten von Π sind demnach um so geringer, je größer die Startmenge S ist, also je mehr Vorgänge bereits zum Zeitpunkt t gestartet werden. Unter Beachtung von (1.7) sind die Kosten genau dann minimal, wenn $S = \{1, \ldots, k-1\}$ gilt, und wir erhalten:

$$\rho\Big(\kappa(L \cup R \mid L) \quad \mid \quad |R| + |L| = m + 1\Big) = \sum_{i \in L \cup R} \frac{w_i}{\lambda_i} \quad + \quad \frac{w_k}{\sum_{j \in L \cup (R - \{k\})} \lambda_j}.$$

Eine Strategie, die zum Zeitpunkt t *nicht* maximal viele Vorgänge startet, oder die *nicht* die Vorgänge maximaler Priorität bezüglich der Werte $w_i/\sum_{i \in L \cup (R - \{i\})} \lambda_i$ vorzieht kann demnach nicht optimal sein. Das war die Behauptung. ■

Die Interpretation des Resultates ist sehr anschaulich:
In einer Situation wie der oben untersuchten muß sich eine Strategie für (wenigstens) einen „Wartevorgang“ entscheiden, da nicht genügend Maschinen zum Start *aller* restlichen Vorgänge frei sind.

Abgesehen von der Tatsache, daß es sich nicht lohnt mehr als einen Vorgang warten zu lassen, muß man sich für denjenigen Vorgang $i \in R$ entscheiden, der als Wartevorgang die geringsten Kosten verursacht.

Beispiel 1.2 (Fortsetzung) Wir sind nun in der Lage, das Ergebnis von Beispiel 1.2 genauer zu begründen.

Die beiden Startvorgänge sind Vorgänge 1 und 2. Sobald einer der beiden Vorgänge beendet ist, ist genau eine Maschine belegt, und es sind noch zwei Jobs zu starten (3 und 4). Wir haben also $R = \{3,4\}$ und $L = \{1\}$ bzw. $L = \{2\}$, und es gilt $|R| + |L| = m + 1$. Wir betrachten die beiden Alternativen:

Job 1 endet zuerst Dann ist $L = \{2\}$, und wir erhalten für die Prioritätsliste (1.7):

$$\frac{\overbrace{w_3}^{=2}}{\underbrace{\lambda_2 + \lambda_4}_{=2}} = 1 \quad > \quad \frac{10}{11} = \frac{\overbrace{w_4}^{=10}}{\underbrace{\lambda_2 + \lambda_3}_{=11}}.$$

Damit ist in diesem Fall die Prioritätsliste $3 < 4$ optimal, im Einklang mit den beiden Strategien Π_1 und Π_2.

Job 2 endet zuerst Dann ist $L = \{1\}$, und wir erhalten:

$$\frac{\overbrace{w_3}^{=2}}{\underbrace{\lambda_1 + \lambda_4}_{=x+1}} = \frac{2}{x+1} \quad \text{und} \quad \frac{\overbrace{w_4}^{=10}}{\underbrace{\lambda_1 + \lambda_3}_{=x+10}} = \frac{10}{x+10}.$$

Und

$$\frac{2}{x+1} \leq \frac{10}{x+10} \quad \text{genau dann, wenn} \quad x \geq \frac{5}{4}.$$

Damit ist die Prioritätsliste $4 < 3$ optimal, wenn $x \geq \frac{5}{4}$ gilt, und $3 < 4$ entsprechend für $x \leq \frac{5}{4}$. Da Strategie Π_1 nach $4 < 3$ verfährt, und Π_2 nach $3 < 4$, ist auch dieses Resultat im Einklang mit dem vorher erzielten Ergebnis.

Die beste Strategie startet also (nach dem Start der Vorgänge 1 und 2) gerade denjenigen Vorgang als nächstes, der als Wartevorgang höhere Kosten verursacht. ■

Ein Versuch zur Konstruktion einer optimalen Prioritätsstrategie

Sei ein Schedulingproblem unter den Voraussetzungen (1.6) gegeben. Wir unternehmen nun den Versuch, eine *beliebige optimale (Set-)Strategie* Π durch eine Prioritätsstrategie Π' zu ersetzen.

Die Idee ist eine Rückwärtsinduktion über $p := |G - L|$, die Anzahl der bereits beendeten Vorgänge.

Die Induktionsbehauptung besteht darin, daß für *jedes* Subproblem mit $p = |G-L|$ bereits beendeten Vorgängen stets eine optimale Prioritätsstrategie existiert.

Den *Induktionsanfang* liefert Lemma 1.6:
Es gilt

$$|G - L| \geq |I| - (m+1) \iff |R| + |L| \leq m + 1.$$

Für $|R| + |L| = m + 1$ liefert die Prioritätsstrategie gemäß (1.7) das gewünschte. Also ist die Behauptung für $|G - L| \geq |I| - (m+1)$ richtig.

Der *Induktionsschritt*:
Nehmen wir also an, die Behauptung gelte für alle $|G - L| \geq p$ mit beliebigem $1 \leq p \leq |I| - m + 1$. Wir müssen zeigen daß eine Prioritätsstrategie für jedes beliebiges Subproblem mit $|G - L| = p - 1$ existiert.

Sei also t ein beliebiger Aktionszeitpunkt mit $|G - L| = p - 1$. Ferner sei Π eine *optimale* Strategie für das Subproblem ab t, und $S \subseteq R$ die von Π gewählte Startmenge zu t.

Den Zeitpunkt der nächsten Vorgangsbeendigung bezeichnen wir (unabhängig vom nächsten Zustand des Projektes) durch t'.

Nehmen wir nun an, Π lasse zum Zeitpunkt t genau eine Maschine leer[32].

Dann seien o. B. d. A. $L = \{1, \ldots, l\}$, $S = \{l+1, \ldots, s\}$ und $R - S = \{s+1, \ldots, r\}$. Die erwarteten (Rest-)Kosten von Π sind nach Bemerkung 1.1 durch folgenden Ausdruck gegeben:

$$\begin{aligned} E(\kappa^{\Pi}(L \cup R \mid L)) &= \frac{w_1 + \cdots + w_r}{\lambda_1 + \cdots + \lambda_s} \\ &+ \sum_{j=1}^{s} \frac{\lambda_j}{\lambda_1 + \cdots + \lambda_s} \, \rho\Big(\kappa(L \cup R - \{j\} \mid L \cup S - \{j\})\Big) \end{aligned}$$

Die Darstellung folgt aus der Optimalität von Π, denn Π ist auch optimal auf allen induzierten Subproblemen.

[32] Die Verallgemeinerung auf mehrere freie Maschinen bereitet höchstens technischen Aufwand – wir könnten wieder mehrere Konstruktionsschritte derselben Art anwenden

Möchten wir nun Π durch eine Prioritätsstrategie Π' ersetzen, müssen wir genau einen Vorgang *mehr* starten als Π. Wir definieren Π' zum Zeitpunkt t durch die Startmenge $S \cup \{k\}$ für einen beliebigen Vorgang $k \in R - S$.

Wir wissen bereits aufgrund der Induktionsvoraussetzung, daß sowohl unter Π, als auch unter Π' jedes der Subprobleme ab t' optimal durch eine Prioritätsstrategie gelöst wird. Wir definieren Π' auf jedem der Subprobleme nun durch eine derartige Prioritätsstrategie.

Für die erwarteten Kosten unter Π' erhält man:

$$\begin{aligned} E(\kappa^{\Pi'}(L \cup R \mid L)) \quad &= \quad \frac{w_1 + \cdots + w_r}{\lambda_1 + \cdots + \lambda_s + \lambda_k} \\ &+ \sum_{j=1}^{s} \frac{\lambda_j}{\lambda_1 + \cdots + \lambda_s + \lambda_k} \, \rho\Big(\kappa(L \cup R - \{j\} \mid L \cup S + \{k\} - \{j\})\Big) \\ &+ \frac{\lambda_k}{\lambda_1 + \cdots + \lambda_s + \lambda_k} \, \rho\Big(\kappa(L \cup R - \{k\} \mid L \cup S)\Big). \end{aligned}$$

Unser nächstes Ziel ist eine Abschätzung der beiden Strategien gegeneinander. Dazu benötigen wir Abschätzungen für die unter Π' auftretenden Subprobleme:

Lemma 1.7 *Es gilt unter der Voraussetzung (1.6)*

$$\rho\Big(\kappa(L \cup R - \{k\} \mid L \cup S)\Big) \leq \rho\Big(\kappa(L \cup R \mid L)\Big) \;-\; \frac{w_k}{\lambda_k}\,.$$

Beweis. Wir wissen nach Lemma 1.1, daß jeder Vorgang über den Term $w_i\, E(\mathcal{C}_i^{\Pi})$ zu den erwarteten Kosten einer beliebigen Strategie Π beiträgt.
Da die erwartete Fertigstellungszeit für *jeden* Vorgang i wenigstens $\frac{1}{\lambda_i}$ beträgt (für den Fall, daß er sofort gestartet wird), trägt jeder Vorgang wenigstens durch den Term $\frac{w_i}{\lambda_i}$ zu den erwarteten Kosten bei. Daher bewirkt die „Entfernung“ eines Vorgangs *wenigstens* eine Kostenreduzierung in dieser Höhe. Da ferner bekannt ist, daß die optimale Strategie Π die Startmenge S wählt, folgt $\rho\Big(\kappa(L \cup R \mid L)\Big) = \rho\Big(\kappa(L \cup R \mid L \cup S)\Big)$.
Damit gilt die Behauptung. ■

Mit etwas mehr Aufwand läßt sich folgendes zeigen:

$$\rho\Big(\kappa(L \cup R - \{k\} \mid L \cup S)\Big) \leq \rho\Big(\kappa(L \cup R \mid L)\Big) \;-\; \Big(\frac{w_k}{\lambda_k} \;+\; \frac{w_k}{\lambda_1 + \cdots + \lambda_s}\Big). \tag{1.8}$$

Diese Tatsache folgt letztlich daraus, daß die optimale Strategie Π Vorgang k nicht zu Beginn, sondern frühstens zum Zeitpunkt der nächsten Vorgangsbeendigung, also zu $1\,/(\lambda_1 + \cdots + \lambda_s)$ startet. Auf den Beweis wollen wir der Kürze halber verzichten.

Als nächstes ist eine Abschätzung der Art

$$\begin{aligned} &\rho\Big(\kappa(L \cup R - \{j\} \mid L \cup S + \{k\} - \{j\})\Big) \\ &\quad \leq \rho\Big(\kappa(L \cup R - \{j\} \mid L \cup S - \{j\})\Big) \quad + \quad \Delta_{j,k} \end{aligned}$$

für alle $j \in L \cup S$ wünschenswert. Dabei sind die Parameter $\Delta_{j,k} \in \mathbb{R}_+$. Sie beschreiben den Nachteil durch den zusätzlich vorgeschriebenen Startvorgang k, der auf den einzelnen Subproblemen (nach Beendigung von j) durch die *greedyness* von Strategie Π' gegenüber Π entstehen kann.

Die Bestimmung einer oberen Schranke für $\Delta_{j,k}$ ist nicht gelungen.

Wir wollen jedoch angeben, was die Differenz der erwarteten Kosten beider Strategien mit den unbestimmten Parametern $\Delta_{j,k}$ ergibt:

$$
\begin{aligned}
& E(\kappa^{\Pi'}(L \cup R \mid L)) - E(\kappa^{\Pi}(L \cup R \mid L)) \\
\leq \quad & \frac{\lambda_1 \Delta_{1,k} + \cdots + \lambda_s \Delta_{s,k}}{\lambda_1 + \cdots + \lambda_s + \lambda_k} \quad - \quad \frac{w_k}{\lambda_1 + \cdots + \lambda_s} \\
\leq 0 \ ?
\end{aligned}
$$

Hätte man die Gewißheit, daß Π Vorgang k in jedem Fall zum Zeitpunkt t' startet, so gälte offenbar $\Delta_{j,k} = 0$, und Π' hätte geringere erwartete Kosten als Π.

Zwar ist per Induktionsannahme bekannt, daß die optimale Strategie zum Zeitpunkt t' (o. B. d. A.) gemäß einer Prioritätsliste vorgeht, aber die Prioritäten hängen davon ab, *welcher* Vorgang zu t' endet. Schon die Diskussion beim Vergleich zwischen zwei und mehr Maschinen in Abschnitt 1.5.2 hatte dies deutlich gemacht.

Wir verweisen an dieser Stelle auf die Resultate des letzten Kapitels dieser Arbeit, wo für m-Maschinen-Probleme zumindest Gütegarantien für statische Prioritätsstrategien nachgewiesen werden.

Kapitel 2

Polyedrische Methoden für Stochastisches Scheduling

Eine vielversprechende Möglichkeit zur Untersuchung stochastischer (und deterministischer) Schedulingprobleme besteht darin, zulässige Lösungen des Problems mit Vektoren im endlichdimensionalen euklidischen Raum zu identifizieren, und strukturelle Eigenschaften des Lösungsraums zu nutzen. Gelingt der Nachweis *polyedrischer* Strukturen des Lösungsraums, können zur Bearbeitung der Schedulingprobleme Methoden der linearen Programmierung genutzt werden.

Vielversprechend ist der Ansatz insbesondere dann, wenn spezielle Eigenschaften des Lösungsraums die Anwendung von besonders einfachen *greedy*-Lösungsansätzen erlauben. Solche Eigenschaften bieten *Polymatroide* und deren Verwandte[1], welche in diesem Kapitel speziell im Hinblick auf den Nutzen für stochastisches Scheduling untersucht werden.

Prinzipiell ist die Vorgehensweise bei der polyedrischen Untersuchung von Schedulingproblemen die folgende:

(1) Formuliere das Schedulingproblem möglichst derart, daß die Zielfunktion κ eine *lineare* Funktion in einer *endlichen* Anzahl von Variablen $x_1, \ldots, x_n$ ist.

(2) Identifiziere alle zulässigen Lösungen $(x_1, \ldots, x_n)$ des Schedulingproblems als Teilmenge des $\mathbb{R}^n$. Ist die konvexe Hülle dieses Lösungsraums ein *Polyeder* $\mathcal{P}$, suche eine *äußere* Beschreibung dieses Polyeders. Falls eine solche lineare Beschreibung von Zielfunktion oder Lösungsraum nicht gelingt, bestimme möglichst eine **LP**-Relaxierung für das ursprüngliche Problem.

[1] Polymatroidiale Strukturen werden unter anderem in [Edm 70], [Lov 82], [Fuj 91], [BGT 92], [BN-M 93], [Que 93] und [QS 94] untersucht.

(3) Löse das lineare Optimierungsproblem $\min\{\kappa(x) \mid x \in \mathcal{P}\}$, bzw. die entsprechende **LP**-Relaxierung.

(4) Bestimme anhand der Lösung des linearen Optimierungsproblems aus (3) eine optimale, oder zumindest eine approximative Lösung für das Schedulingproblem.

Tatsächlich ist eine *vollständige* algebraische Beschreibung der Lösungsräume von Schedulingproblemen in den wenigsten Fällen zu erwarten. Die Möglichkeit, über **LP**-Relaxierungen zumindest (gute) Approximationsheuristiken zu gewinnen, rechtfertigt jedoch in vielen Fällen den polyedrischen Ansatz.

Eine approximative Lösung ist nur dann auch von praktischem Interesse, wenn sie zumindest „schnell" gefunden werden kann. Um die zugehörigen linearen Programme effizient, das heißt in polynomialer Zeit lösen zu können, ist demnach die Beschäftigung mit speziellen polyedrischen Strukturen durchaus von Interesse. Die Behandlung einer in diesem Sinne „effizienten" polyedrischen Struktur bildet den Hauptteil dieses Kapitels.

Wir wollen noch einige grundsätzliche Bemerkungen zu polyedrischen Methoden machen: Von fundamentaler Bedeutung für den Erfolg eines derartigen Ansatzes ist stets die Modellformulierung – einschließlich der Wahl der Variablen, mit denen wir das Problem beschreiben, und die letztlich die Zielfunktion κ determinieren.

Zur Nutzung von Methoden der linearen Programmierung ist die *Linearität* der Zielfunktion entscheidend. Von der Festlegung der Variablen hängt daher einerseits ab, *welche* Zielfunktionen sich als lineare Funktionen darstellen lassen. Andererseits wird durch die Wahl der Modellformulierung festgelegt, ob und wie sich verschiedene Nebenbedingungen für Schedulingprobleme derart integrieren lassen, daß sie ebenfalls *lineare* Ungleichungen für den Lösungsraum darstellen.

Beispielsweise denke man an die gewichtete Summe der Fertigstellungszeiten. Hier liegt es auf der Hand, das Problem durch die Fertigstellungszeiten C_i zu beschreiben[2]. Möchten wir etwa zusätzlich *release dates* r_i für jeden Vorgang i einführen, fügen wir lediglich die *linearen* Ungleichungen $C_i \geq r_i + p_i$ zur Beschreibung des Lösungsraums hinzu (Dabei bezeichnet p_i die Dauer von Vorgang i).

Für den Fall des *deterministischen* Scheduling bieten M. Queyranne und A. S. Schulz in [QS 94] eine ausführliche Übersicht zu polyedrischen Charakterisierungen verschiedener Lösungsräume für Schedulingprobleme.

Weiter findet man bei D. Bertsimas and J. Niño-Mora in [BN-M 93] polyedrische Charakterisierungen von Lösungsräumen sogenannter *multi-armed bandit problems* sowie *bran-*

[2] In diesem Zusammenhang werden Variablen, bei denen C_i die Fertigstellungszeit von Vorgang i beschreibt, auch *natural date variables* genannt; vergleiche [QS 94].

ching bandit problems zur Modellierung verschiedener Zielfunktionen zur Gewinnmaximierung.

Wir werden uns auch in diesem Kapitel auf die Charakterisierung von Schedulingproblemen durch die Fertigstellungszeiten der Vorgänge zur Modellierung der Zielfunktion $\sum_{i\in I} w_i\, C_i$ beschränken; durch andere Ansätze sind durchaus weitere Resultate denkbar.

Das zweite Kapitel ist folgendermaßen gegliedert:
Die ersten zwei Abschnitte enthalten die nötigen Grundlagen und Notationen zu Polyedern und Polytopen, sowie zur linearen Optimierung und Dualitätstheorie.
Es werden im dritten Abschnitt spezielle Polyeder, nämlich *super- und submodulare Polyeder*, bzw. *Polymatroide*, und deren Verallgemeinerungen vorgestellt. Wir zeigen, daß sich diese Polyeder als Spezialfall in die durch P. P. Bhattacharya et al. [BGT 92] eingeführten *erweiterten Polymatroide und Contra-Polymatroide* einordnen lassen. Weiter werden wir auf die Definition und Bedeutung der zugehörigen *Basispolytope* eingehen.
Im vierten Abschnitt wollen wir uns konkret mit der Optimierung linearer Zielfunktionen über erweiterten (Contra-)Polymatroiden beschäftigen: Es wird gezeigt, daß für diese Polyeder ein von Bhattacharya et al. vorgeschlagener *erweiterter Greedy-Ansatz* zum Erfolg führt. Diesen Algorithmus werden wir eingehend erläutern.
Schließlich wird im letzten Abschnitt die Anwendung der polyedrischen Methoden auf das im ersten Kapitel vorgestellte Modell für stochastisches Scheduling untersucht. Wir werden sogenannte *Erhaltungssätze*[3] für stochastische Schedulingprobleme angeben, die garantieren, daß der zugehörige Lösungsraum die gewünschten polyedrischen Eigenschaften hat. Es wird sich herausstellen, daß die Erhaltungssätze bereits die Optimalität einer statischen Prioritätsstrategie implizieren.
Ferner zeigen wir, daß die *ratio rule* von Smith [Smi 56] im stochastischen Scheduling auf einer Maschine bei stochastisch unabhängigen Dauerverteilungen und $\prec = \emptyset$ eine optimale Strategie ist, die sich als Spezialfall in den erweiterten *greedy*-Ansatz für Contra-Polymatroide einordnet.

2.1 Polyeder und Polytope

In diesem Abschnitt werden die Begriffe und Notationen erläutert, die für den Zugang zu erweiterten Polymatroiden unverzichtbar sind. Dieser Abschnitt ist daher nicht als Einführung in die Theorie der Polyeder zu verstehen, sondern als Bereitstellung des nötigen Handwerkszeugs für unsere Zwecke. Es sei, insbesondere für fehlende Beweise, auf ein Lehrbuch von G. L. Nemhauser und A. L. Wolsey [NW 88, Abschnitt I.4] oder etwa den Übersichtsartikel [GT 89] hingewiesen.

[3] Vergleiche insbesondere [BGT 92], [BN-M 93] und [GG 95].

Wir wollen die Begriffe *lineare, konvexe, konische und affine* Kombination von Vektoren, sowie die *lineare, konvexe, konische und affine Hülle* endlich vieler Punkte des $\mathbb{R}^n$ als bekannt voraussetzen. Die *Dimension* einer beliebigen Menge $S \subseteq \mathbb{R}^n$ ist die Dimension von $\mathrm{aff}(S)$, der affinen Hülle von S. Eine Menge $S \subseteq \mathbb{R}^n$ heißt *volldimensional*, falls $\dim(S) = n$ gilt.

Definition 2.1 (Polyeder) *Eine Menge $\mathcal{P} \subseteq \mathbb{R}^n$ heißt Polyeder, falls $\mathcal{P}$ der Durchschnitt endlich vieler, abgeschlossener Halbräume der Art $\mathcal{H} = \{x \in \mathbb{R}^n \mid a^t x \geq \beta\}$ mit $a \in \mathbb{R}^n$ und $\beta \in \mathbb{R}$ ist.*

Mit passender Matrix $A \in \mathbb{R}^{m \times n}$ und einem Vektor $b \in \mathbb{R}^m$ kann man jedes Polyeder in der Form $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ schreiben. Das System $Ax \geq b$ kann natürlich auch *implizite Gleichungen* enthalten, wenn eine Ungleichung $A_{i,\cdot}\, x \geq b_i$ ohnehin mit Gleichheit erfüllt ist für alle $x \in \mathcal{P}$ und eine Zeile $A_{i,\cdot}$ von A. Ein System von Gleichungen und Ungleichungen heißt *irredundant*, falls keine Gleichung oder Ungleichung bereits durch andere Gleichungen oder Ungleichungen impliziert wird. Die Beschreibung eines Polyeders durch Ungleichungen wird die *äußere* Beschreibung eines Polyeders genannt
Eine *innere* Beschreibung liefert der folgende Satz, der auf Minkowski zurückgeht[4].

Satz 2.1 *Eine Menge $\mathcal{P} \in \mathbb{R}^n$ ist genau dann ein Polyeder, wenn es endliche Mengen $V, R \subset \mathbb{R}^n$ gibt, so daß $\mathcal{P} = \mathrm{conv}(V) + \mathrm{cone}(R)$ gilt.*

Sei $V = \{v_1, \ldots, v_s\}$ und $R = \{r_1, \ldots, r_t\}$, dann sind nach Satz 2.1 alle Punkte x eines Polyeders $\mathcal{P}$ gegeben durch

$$x = \sum_{i=1}^{s} \lambda_i v_i \; + \; \sum_{j=1}^{t} \mu_j r_j \tag{2.1}$$

mit $\lambda_i, \mu_j \geq 0$ für alle $i = 1, \ldots, s$, $j = 1, \ldots, t$ sowie $\sum_{i=1}^{s} \lambda_i = 1$.

Definition 2.2 (Rezessionskegel) *Der Rezessionskegel $\mathrm{rec}(\mathcal{P})$ eines Polyeders $\mathcal{P}$ ist die Menge aller Vektoren $y \in \mathbb{R}^n$, für die ein $x \in \mathcal{P}$ existiert, so daß $x + \lambda y \in \mathcal{P}$ gilt für alle $\lambda \geq 0$. Die Menge $\{x + \lambda y \mid \lambda \geq 0\}$ für ein $y \neq 0$ wird auch* Strahl *genannt.*

Es ist bekannt, daß $\mathrm{rec}(\mathcal{P}) = \mathrm{cone}(R) = \{x \in \mathbb{R}^n \mid Ax \geq 0\}$ gilt, wenn $\mathcal{P}$ die äußere Darstellung $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ bzw. die innere Darstellung $\mathcal{P} = \mathrm{conv}(V) + \mathrm{cone}(R)$ besitzt.

Wir nennen ein Polyeder $\mathcal{P}$ *beschränkt*, falls $\mathcal{P}$ keinen *Strahl* enthält, oder äquivalent, es ein $s \in \mathbb{R}^n_+$ gibt, so daß $-s \leq x \leq s$ für alle $x \in \mathcal{P}$ gilt.

[4] Vergleiche [NW 88, Abschnitt I.4.4, Theorem 4.8].

Definition 2.3 (Polytop) *Ein beschränktes Polyeder heißt Polytop.*

Ausgehend von Satz 2.1 läßt sich ein Polytop äquivalent auch als Konvexkombination endlich vieler Punkte des $\mathbb{R}^n$ definieren. Damit ist ein nichtleeres Polyeder genau dann ein Polytop, wenn $\text{rec}(\mathcal{P}) = \text{cone}(R) = \{0\}$ gilt.

Definition 2.4 (Gültige Ungleichungen, Seitenflächen und Facetten) *Eine Ungleichung $a^t x \geq \beta$ heißt gültig für ein Polyeder $\mathcal{P}$, wenn $\mathcal{P} \subseteq \{x \in \mathbb{R}^n \mid a^t x \geq \beta\}$ gilt, $\mathcal{P}$ also in dem durch die Ungleichung definierten Halbraum enthalten ist. Für eine gültige Ungleichung $a^t x \geq \beta$ nennt man die Menge $F = \{x \in \mathcal{P} \mid a^t x = \beta\}$ eine Seitenfläche von $\mathcal{P}$. Triviale Seitenflächen sind die leere Menge und $\mathcal{P}$ selbst. Die nichttrivialen Seitenflächen maximaler Dimension, also mit $dim(F) = dim(\mathcal{P}) - 1$, werden die Facetten von $\mathcal{P}$ genannt.*

Ist ein Polyeder $\mathcal{P}$ durch eine irredundante äußere Beschreibung durch Gleichungen und Ungleichungen gegeben, ist bekannt, daß ein eineindeutiger Zusammenhang zwischen der Menge der Facetten von $\mathcal{P}$ und der Menge der Ungleichungen in der Beschreibung von $\mathcal{P}$ besteht; vergleiche etwa [NW 88, Abschnitt I.4.3]. Für Optimierungszwecke sind ferner die nichttrivialen Seitenflächen minimaler Dimension, die Ecken, von Bedeutung.

Definition 2.5 (Ecke) *Es sei $\mathcal{P} \subseteq \mathbb{R}^n$ ein Polyeder, dann nennt man $x \in \mathcal{P}$ eine Ecke von $\mathcal{P}$, wenn x sich nicht als echte Konvexkombination zweier Punkte $y, z \in \mathcal{P}$ mit $y \neq z$ darstellen läßt, d. h. mit $x = \lambda y + (1 - \lambda) z$, $0 \leq \lambda \leq 1$ ist entweder $\lambda = 0$ oder $\lambda = 1$.*

Verschiedene Charakterisierungen von Ecken werden durch den folgenden Satz gegeben[5].

Satz 2.2 *Seien $\mathcal{P} \subseteq \mathbb{R}^n$ ein Polyeder und $x \in \mathcal{P}$, dann sind äquivalent:*

(1) x ist Ecke von $\mathcal{P}$.

(2) $F := \{x\}$ ist eine nulldimensionale Seitenfläche von $\mathcal{P}$, d. h. es gibt eine gültige Ungleichung $a^t x \geq \beta$ für $\mathcal{P}$, so daß $F = \{x\} = \{y \in \mathcal{P} \mid a^t y = \beta\}$.

(3) Es gibt einen Vektor $w \in \mathbb{R}^n$, so daß x die einzige Optimallösung des linearen Optimierungsproblems $\min\{w^t y \mid y \in \mathcal{P}\}$ ist.

Ein Polyeder, das zumindest eine Ecke besitzt, heißt *spitz*. Im allgemeinen braucht ein Polyeder nicht spitz zu sein; beispielsweise ist ein Halbraum ein nichtspitzes Polyeder.

[5] Vergleiche [NW 88, Abschnitt I.4.4]

Ausgehend von Satz 2.1 erhält man für Polyeder den folgenden Darstellungssatz[6]:

Satz 2.3 *Es sei $\mathcal{P}$ ein Polyeder und sei E die Menge der Ecken von $\mathcal{P}$, dann gilt*

$$\mathcal{P} = \mathrm{conv}(E) + \mathrm{rec}(\mathcal{P}) \ .$$

Korollar 2.1 *Für ein Polytop $\mathcal{P}$ gilt*

$$\mathcal{P} = \mathrm{conv}(E).$$

Insbesondere ist ein Polytop entweder leer oder spitz. Um eine Aussage über die Existenz von Ecken bei einem beliebigen Polyeder zu treffen, kann man die äußere Beschreibung des Polyeders auf folgende Weise nutzen:

Lemma 2.1 *Es sei $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$ ein nichtleeres Polyeder, dann sind äquivalent:*

(1) $\mathcal{P}$ ist spitz.

(2) $Rg(A) = n$, d. h. A hat maximalen (Spalten-)Rang.

Beweis. Sei $\mathcal{P}$ spitz, x Ecke von $\mathcal{P}$ und angenommen A habe nicht vollen Spaltenrang. Dann gibt es einen nichttrivialen Vektor $y \in \mathbb{R}^n$ mit $Ay = 0$. Damit ist $A(x \pm y) = Ax \geq b$ und daher $x \pm y \in \mathcal{P}$. Also ist $x = \frac{1}{2}(x+y) + \frac{1}{2}(x-y)$ eine echte Konvexkombination für x aus zwei Vektoren aus $\mathcal{P}$, im Widerspruch zur Definition von Ecken.

Hat umgekehrt A vollen Spaltenrang, müssen wir zeigen, daß $\mathcal{P}$ eine Ecke besitzt. Diese Tatsache ist ein Spezialfall eines bekannten Resultats: Besitzt Matrix A genau k ($\leq n$) linear unabhängige Spalten, so hat das zugehörige Polyeder *mindestens eine* Seitenfläche der Dimension $n - k$[7]. ■

Für ein beliebiges Polyeder gilt außerdem:

Lemma 2.2 *Ist ein Polyeder spitz, so enthält jede nichtleere Seitenfläche wenigstens eine Ecke.*

Beweis. Sei $\mathcal{P} = \mathrm{conv}(E) + \mathrm{rec}(\mathcal{P})$ mit der Eckenmenge E von $\mathcal{P}$. Sei F eine nichtleere Seitenfläche von $\mathcal{P}$, also $F = \{x \in \mathcal{P} \mid a^t x = \beta\}$ mit einer gültigen Ungleichung $a^t x \geq \beta$. Angenommen F enthalte keine Ecke, dann ist $a^t e > \beta$ für alle Ecken e von $\mathcal{P}$. Für jedes

[6] Zum Beweis vergleiche etwa [GT 89, Theorem 2.2].
[7] Vergleiche zum Beweis [NW 88, Abschnitt I.4.4, Proposition 4.1].

$x \in conv(E)$ gilt dann ebenfalls $a^t x > \beta$. Daher muß wenigstens ein $y \in rec(\mathcal{P})$ existieren mit $a^t y < 0$, denn sonst wäre $a^t x > \beta$ für alle $x \in \mathcal{P}$ und damit $F = \emptyset$. Zu diesem y existiert per Definition von $rec(\mathcal{P})$ ein $x \in \mathcal{P}$, so daß $x + \lambda y \in \mathcal{P}$ für alle $\lambda \geq 0$. Aber $a^t(x + \lambda y) = a^t x + \lambda a^t y \; < \; \beta$ für ausreichend großes λ. Das steht im Widerspruch zur Gültigkeit der Ungleichung $a^t x \geq \beta$ für $\mathcal{P}$. ■

Wir hatten bereits die Ecken eines Polyeders als Optimallösungen passender linearer Optimierungsprobleme charakterisiert. Im folgenden Abschnitt wollen wir einige Grundlagen der linearen Optimierung vorstellen.

2.2 Lineare Optimierung und Dualität

Für ein allgemeines lineares Optimierungsproblem, das in diesem Zusammenhang stets auch *lineares Programm* oder kurz **LP** genannt wird, gibt es unterschiedliche äquivalente Formulierungen. Unabhängig davon, in welcher dieser Formen ein **LP** vorliegt, geht es stets um die Minimierung oder Maximierung einer *linearen* Zielfunktion $\kappa : \mathbb{R}^n \longrightarrow \mathbb{R}$ unter *linearen* Nebenbedingungen.

Seien dazu $w \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$, $b \in \mathbb{R}^m$ und $\kappa(x) = w^t x = \sum_{i=1}^n w_i \, x_i$. Eine Schreibweise für ein **LP** ist die *kanonische* Form:

$$\min\{w^t x \mid x \in \mathbb{R}^n \;, Ax \geq b\}. \tag{2.2}$$

Unabhängig von der Formulierung besteht die Aufgabe stets darin, für das zugrundeliegende Polyeder $\mathcal{P} = \{x \in \mathbb{R}^n \mid Ax \geq b\}$

- $\mathcal{P} \neq \emptyset$ zu prüfen; für $\mathcal{P} = \emptyset$ heißt (2.2) *unzulässig*;
- $\inf\{w^t x \mid x \in \mathcal{P}\} > -\infty$ zu prüfen, falls nicht, heißt (2.2) *unbeschränkt*;
- und für den Fall, daß (2.2) weder unzulässig noch unbeschränkt ist, eine *Optimallösung* $x^{opt} \in \mathcal{P}$ mit $w^t x^{opt} \leq w^t x$ für alle $x \in \mathcal{P}$ zu finden.

Ein Vektor $x \in \mathcal{P}$ heißt *zulässig* für das lineare Programm (2.2). Die Existenz einer Optimallösung $x^{opt} \in \mathcal{P}$ für ein zulässiges und beschränktes **LP** folgt letztlich aus der Endlichkeit der Beschreibung eines jeden Polyeders:
Ist $\xi := \inf\{w^t x \mid x \in \mathcal{P}\} > -\infty$, kann nicht $w^t x > \xi$ für *alle* $x \in \mathcal{P}$ gelten. Mit der endlichen Darstellung (2.1) eines jeden Vektors $x \in \mathcal{P}$ würde man dann $w^t x \geq \xi + \varepsilon$ für ein $\varepsilon > 0$ erhalten, was ein Widerspruch zur Definition von ξ wäre.
Damit ist – zumindest für den Fall, daß ein **LP** nicht unbeschränkt ist – das Minimum in der Formulierung (2.2) wohldefiniert.

Für Optimallösungen eines linearen Programms über einem Polyeder $\mathcal{P}$ gilt sogar:

Lemma 2.3 *Ist ein* **LP** $\min\{w^t x \mid x \in \mathcal{P}\}$ *für ein Polyeder* $\mathcal{P}$ *weder unzulässig noch unbeschränkt, so ist die Menge der Optimallösungen eine nichtleere Seitenfläche von* $\mathcal{P}$.

Beweis. Sei $\xi := \min\{w^t x \mid x \in \mathcal{P}\}$, dann ist $w^t x \geq \xi$ für alle $x \in \mathcal{P}$. Also ist $w^t x \geq \xi$ eine gültige Ungleichung für $\mathcal{P}$ und damit ist $F = \{x \in \mathcal{P} \mid w^t x = \xi\} \neq \emptyset$ eine Seitenfläche von $\mathcal{P}$. ■

Zusammen mit Lemma 2.2 erhält man:

Korollar 2.2 *Falls ein* **LP** $\min\{w^t x \mid x \in \mathcal{P}\}$ *eine endliche Optimallösung besitzt, und das zugehörige Polyeder ist spitz, so gibt es auch eine Optimallösung, die eine Ecke von* $\mathcal{P}$ *ist.*

Ein sehr wichtiges Hilfsmittel zur Untersuchung von linearen Programmen ist die **Dualitätstheorie**. Wir betrachten ein **LP** der Art

$$\min\{w^t x \mid x \in \mathbb{R}^n,\ Ax \geq b\} \qquad \text{(P)}$$

mit $w \in \mathbb{R}^n$, $A \in \mathbb{R}^{m \times n}$ und $b \in \mathbb{R}^m$ und nennen (P) von nun an auch *primales* Problem. Wir möchten zunächst eine *untere* Schranke für den Zielfunktionswert von (P) finden, falls eine solche existiert. Wir nutzen dazu die Ungleichung $Ax \geq b$ auf folgende Weise: Für einen nichtnegativen Vektor $y \in \mathbb{R}^m_+$ gilt $y^t(Ax) \geq y^t b$ für alle $x \in \mathcal{P}$. Finden wir nun ein y mit $y^t A = w^t$, so folgt

$$w^t x = (y^t A)x = y^t(Ax) \geq y^t b$$

für alle $x \in \mathcal{P}$. Somit ist $y^t b$ untere Schranke für *jeden* Zielfunktionswert von (P). Um die *größte* untere Schranke zu finden, falls eine solche existiert, ist das lineare Programm

$$\max\{y^t b \mid y \in \mathbb{R}^m,\ y^t A = w^t,\ y \geq 0\} \qquad \text{(D)}$$

zu lösen. Dieses **LP** wird das zu (P) *duale* Programm genannt. (D) liegt in der sogenannten *Standardform* vor. Das zu (D) duale Programm ist wieder (P) selbst.

Der Zugang zum dualen Programm über die Suche nach unteren Schranken für das primale Programm liefert sofort einen bekannten Satz:

Satz 2.4 (Schwacher Dualitätssatz) *Ist* x *zulässig für* (P) *und* y *zulässig für* (D), *so gilt stets* $w^t x \geq y^t b$, *also liefert* (D) *eine untere Schranke für* (P), *und umgekehrt auch* (P) *eine obere Schranke für* (D).

Findet man demnach eine primale und eine duale Lösung mit demselben Zielfunktionswert, so folgt aus dem schwachen Dualitätssatz bereits die Optimalität beider Lösungen für die jeweiligen Probleme. Ein solches Paar wird ein *optimales primal-duales Paar* genannt. Eine weitaus stärkere Aussage über zueinander duale Probleme liefert der starke Dualitätssatz, der konstruktiv mit Hilfe der *Simplex-Methode* bewiesen wird; vergleiche etwa [GT 89, Theorem 4.2].

Satz 2.5 (Starker Dualitätssatz) *Haben entweder* (P) *oder* (D) *eine endliche Optimallösung* x^{opt} *bzw.* y^{opt}, *so haben beide eine endliche Optimallösung, und es gilt* $w^t x^{opt} = (y^{opt})^t b$, *d. h.* (P) *und* (D) *haben denselben Optimalwert.*
Ist ferner entweder (P) *oder* (D) *unbeschränkt, so ist das jeweils andere Problem unzulässig.*

Die Lösung allgemeiner linearer Programme soll nicht Thema dieser Arbeit sein, es sei daher an dieser Stelle auf Lehrbücher zur linearen Programmierung verwiesen. Eine Auswahl bieten [NW 88, Abschnitt I.2] sowie auch [GT 89].

Im Vorgriff auf das dritte Kapitel wollen wir noch definieren, was unter einer **LP**-*Relaxierung*[8] zu verstehen ist.

Definition 2.6 (LP-Relaxierung) *Für ein beliebiges Optimierungsproblem*

$$\min\{\kappa(x) \mid x \in X\} \tag{2.3}$$

nennen wir das lineare Programm

$$\min\{w^t x \mid x \in \mathcal{P}\} \tag{2.4}$$

eine **LP**-*Relaxierung von (2.3), falls*

(1) $w^t x \leq \kappa(x)$ *für alle* $x \in X$, *und*

(2) $X \subseteq \mathcal{P}$ *gilt, und* $\mathcal{P}$ *ein Polyeder ist.*

Ist κ selbst bereits linear, ist insbesondere $w^t x = \kappa(x)$ zugelassen. Die Motivation ist die folgende: Wir möchten das Optimierungsproblem (2.3) untersuchen. Verschiedene Umstände können eine „einfache“ Lösung dieses Problems verhindern. Ein vielbeachtetes Beispiel ist etwa die Klasse der *ganzzahligen* linearen Optimierungsprobleme, da diese bereits $\mathcal{NP}$-vollständig ist, vergleiche insbesondere [NW 88].

Ein weiteres Problem kann darin liegen, daß keine vollständige algebraische Beschreibung des Lösungsraums X von (2.3) vorliegt. Dies ist bei den meisten Modellen im Scheduling der Fall, vergleiche insbesondere Kapitel 3 dieser Arbeit.

[8]Vergleiche [NW 88, Abschnitt II.3].

Statt also das eigentliche Problem (2.3) anzugehen, untersuchen wir das „leichtere“, relaxierte Problem (2.4), wobei eine vollständige äußere Beschreibung des Polyeders $\mathcal{P}$ vorliege. Eine Optimallösung für (2.4) liefert uns wegen $X \subseteq \mathcal{P}$ dann zumindest eine *untere Schranke* für (2.3). Weiter liefert uns *jede* zulässige Lösung des zu (2.4) dualen Problems ebenfalls eine untere Schranke. Diese Tatsachen finden im Rahmen von Branch&Bound-Verfahren zur Lösung $\mathcal{NP}$-schwerer Optimierungsprobleme eine häufige Verwendung, vergleiche auch [NW 88, Abschnitt II.4.2].

Falls keine vollständige algebraische Beschreibung von X vorliegt, kann man eine **LP**-Relaxierung durch den Nachweis gültiger Ungleichungen für X gewinnen. Andernfalls kann eine **LP**-Relaxierung auch durch Abschwächen oder Weglassen einiger Restriktionen in der (nicht notwendig linearen) Darstellung von X gewonnen werden.

Bevor wir **LP**-Relaxierungen im letzten Kapitel zur Erzeugung von Approximationslösungen für stochastische Schedulingprobleme aufgreifen, wollen wir uns nun mit speziellen Polyedern befassen.

2.3 Spezielle polyedrische Strukturen

Edmonds [Edm 70] führt 1970 den Begriff *Polymatroid* für spezielle *submodulare Polyeder* ein und gibt einen *greedy*-Algorithmus zur Optimierung linearer Zielfunktionen über Polymatroiden an. Er zeigt, daß dieser Greedy-Ansatz dann und nur dann das Optimum liefert, wenn das zugrundeliegende Polyeder ein Polymatroid ist.

Bhattacharya et al. [BGT 92] definieren *erweiterte Polymatroide* und *erweiterte Contra-Polymatroide* und zeigen, daß für diese Polyeder ein erweiterter *greedy*-Algorithmus zur Optimierung von linearen Zielfunktionen zum Erfolg führt. Erweiterte (Contra-)-Polymatroide stellen eine natürliche Verallgemeinerung der super- bzw. submodularen Polyeder dar, und so ist der Algorithmus von Bhattacharya et al. eine natürliche Verallgemeinerung des Ansatzes von Edmonds.

Das Ziel dieses Abschnittes ist es, die genannten Polyeder gegeneinander abzugrenzen, sowie einen direkten Beweis für die natürliche Überordnung der erweiterten (Contra-)-Polymatroide zu liefern. Weiter soll erläutert werden, welche geometrische Bedeutung die zugeordneten *Basispolytope* der genannten Polyeder haben. Zuvor benötigen wir einige Grundlagen zur Super- und Submodularität.

2.3.1 Super- und Submodularität

Es sei $I = \{1, \ldots, n\}$ eine endliche (Index-)Menge. Super- und Submodularität kann man für Mengenfunktionen f auf beliebigen distributiven Verbänden $\mathcal{D} \subseteq 2^I$ (mit Durchschnitt und Vereinigung als Operationen) definieren. Man nennt das Paar $(\mathcal{D}, f)$ dann *super- oder submodulares System*; vergleiche [Fuj 91, Kapitel II]. Für unsere Zwecke reicht es, sich auf den Fall $\mathcal{D} = 2^I$ zu beschränken.

Sei also $f : 2^I \longrightarrow \mathbb{R}$ eine Mengenfunktion auf der gesamten Potenzmenge von I. Wir betrachten ausschließlich Funktionen f mit

$$f(\emptyset) = 0,$$

und nennen f in diesem Fall auch *normiert*. Es werden nur die in der Arbeit benötigten Eigenschaften super- und submodularer Funktionen angegeben. Für eine ausführliche Behandlung sei daher auf einen Artikel von L. Lovász [Lov 82] hingewiesen.

Definition 2.7 (Super- und Submodularität) *Eine Mengenfunktion $f : 2^I \longrightarrow \mathbb{R}$ heißt supermodular, falls*

$$f(S \cap T) + f(S \cup T) \geq f(S) + f(T) \quad \text{für alle} \quad S, T \subseteq I,$$

und submodular*, falls*

$$f(S \cap T) + f(S \cup T) \leq f(S) + f(T) \quad \text{für alle} \quad S, T \subseteq I.$$

Man nennt eine Mengenfunktion ferner *modular*, wenn sie sowohl super- als auch submodular ist. Es läßt sich relativ leicht zeigen, daß für eine normierte, modulare Funktion $f : 2^I \longrightarrow \mathbb{R}$ stets $f(S) = \sum_{i \in S} f(\{i\})$ für alle $S \subseteq I$ gilt.

Die folgende Charakterisierung von Super- und Submodularität ist ebenso hilfreich wie anschaulich[9]. Wir nennen eine Mengenfunktion $f : 2^I \longrightarrow \mathbb{R}$ *nichtfallend* bzw. *nichtwachsend*, wenn für alle $S \subseteq T$ $f(S) \leq f(T)$ bzw. $f(S) \geq f(T)$ gilt.

Lemma 2.4 *Eine Funktion $f : 2^I \longrightarrow \mathbb{R}$ ist genau dann supermodular (submodular), wenn die Abbildungen*

$$f_k(S) := f(S \cup \{k\}) - f(S) \qquad \textit{für } S \subseteq I \setminus \{k\} \tag{2.5}$$

monoton nichtfallend (nichtsteigend) in S sind für alle $k \in I$.

[9] Vergleiche [Lov 82, Proposition 1.1].

Beweis. Seien f supermodular und $S \subseteq T \subseteq I \setminus \{k\}$ beliebig. Dann gilt

$$\begin{aligned} f_k(S) - f_k(T) &= \Big(f(S \cup \{k\}) - f(S)\Big) - \Big(f(T \cup \{k\}) - f(T)\Big) \\ &= \Big(f(S \cup \{k\}) + f(T)\Big) - \Big(f(S) + f(T \cup \{k\})\Big) \\ &\leq \Big(f((S \cup \{k\}) \cup T) + f((S \cup \{k\}) \cap T)\Big) - \Big(f(S) + f(T \cup \{k\})\Big) \\ &= \Big(f(T \cup \{k\}) + f(S)\Big) - \Big(f(S) + f(T \cup \{k\})\Big) \\ &= 0. \end{aligned}$$

Die Ungleichung gilt dabei aufgrund der Supermodularität von f.

Umgekehrt müssen wir die Supermodularität von f aus (2.5) folgern. Seien dazu $S, T \subseteq I$ beliebig, und um triviale Fälle auszuschließen sei $S \not\subseteq T$ und $T \not\subseteq S$. Ist $S \triangle T$ die symmetrische Differenz von S und T, so sei o. B. d. A. $S \triangle T = \{1, \ldots, k\}$, $T \setminus S = \{1, \ldots, p\}$ und $S \setminus T = \{p+1, \ldots, k\}$. Man kann $f(S \cup T)$ auf folgende Arten darstellen:

$$\begin{aligned} f(S \cup T) &= f(S) + \sum_{i=1}^{p} f_i(S \cup \{1, \ldots, i-1\}) \\ &= f(T) + \sum_{i=p+1}^{k} f_i(T \cup \{p+1, \ldots, i-1\}) \\ &= f(S \cap T) + \sum_{i=1}^{k} f_i((S \cap T) \cup \{1, \ldots, i-1\}), \end{aligned}$$

wobei $\{1, \ldots, 0\} := \emptyset$ sein soll. Die Darstellungen ergeben sich durch sukzessive Anwendung von (2.5), ausgehend von $f_1(S)$, $f_{p+1}(T)$ bzw. $f_1(S \cap T)$. Damit folgt nun die Supermodularität von f, denn:

$$\begin{aligned} f(S) + f(T) &= f(S \cup T) + f(S \cap T) + \sum_{i=1}^{k} f_i((S \cap T) \cup \{1, \ldots, i-1\}) \\ &\quad - \sum_{i=1}^{p} f_i(S \cup \{1, \ldots, i-1\}) - \sum_{i=p+1}^{k} f_i(T \cup \{p+1, \ldots, i-1\}) \\ &\leq f(S \cup T) + f(S \cap T). \end{aligned}$$

Dabei gilt die letzte Ungleichung aufgrund der Isotonie der Funktionen f_i, denn $S \cap T \subseteq S, T$. Für den submodularen Fall verläuft der Beweis analog. ∎

Supermodularität bedeutet also, daß der *„Zuwachs“* der Abbildung f mit wachsendem S steigt. Submodularität bedeutet entsprechend einen mit wachsendem S geringeren *„Zuwachs“* von f – eine Eigenschaft, die mit Konvexität bzw. Konkavität reellwertiger Funktionen verwandt ist. Zu weiteren Parallelen zwischen Super- und Submodularität sowie Konvexität und Konkavität sei mit folgendem Zitat auf [Lov 82] verwiesen:

> „In 'continuous' optimization convex functions play a central role[...]. It is less apparent, but we claim and hope to prove to a certain extent, that a similar role is played in discrete optimization by *submodular set-functions*.“ [Lov 82]

2.3.2 Super- und Submodulare Polyeder, Lineare Transformationen

Jedem super- oder submodularen System $(\mathcal{D}, f)$ über $I = \{1, \dots, n\}$ kann man ein Polyeder des $\mathbb{R}^n$ zuordnen. Wir wollen dies wieder für $\mathcal{D} = 2^I$ tun. Für eine ausführliche Behandlung sei daher nochmals auf [Fuj 91, Kapitel II] hingewiesen.

Definition 2.8 (Super- und Submodulare Polyeder) *Es sei $f : 2^I \longrightarrow \mathbb{R}$ normiert, dann nennt man*

$$\mathcal{P}_{\geq}(f) := \{x \in \mathbb{R}^n \mid x(S) \geq f(S) \quad \text{für alle} \quad S \subseteq I\}$$

ein supermodulares Polyeder, falls f supermodular ist, und

$$\mathcal{P}_{\leq}(f) := \{x \in \mathbb{R}^n \mid x(S) \leq f(S) \quad \text{für alle} \quad S \subseteq I\}$$

ein submodulares Polyeder, falls f submodular ist.

Dabei sei wie üblich $x(S) := \sum_{i \in S} x_i$ für Vektoren $x \in \mathbb{R}^n$.

Gilt für ein submodulares Polyeder $\mathcal{P}_{\leq}(f) \cap \mathbb{R}^n_+ \neq \emptyset$, so entspricht dieser Durchschnitt gerade einem *Polymatroid* im Sinne der ursprünglichen Definition durch Edmonds, die wir der Vollständigkeit halber angeben wollen [10]:

Definition 2.9 (Polymatroid) *Es sei $f : 2^I \longrightarrow \mathbb{R}$ normiert, nichtnegativ, nichtfallend und submodular, dann ist*

$$\mathcal{P}^+_{\leq}(f) = \{x \in \mathbb{R}^n_+ \mid x(S) \leq f(S) \quad \text{für alle} \quad S \subseteq I\}$$

ein Polymatroid.

Da per Definition $x \geq 0$ gilt, ist leicht einzusehen, daß man die Forderungen der Nichtnegativität und Isotonie an f auch durch die Forderung $\mathcal{P}^+_{\leq}(f) \neq \emptyset$ ersetzen kann. Entsprechend kann man ein *Contra-Polymatroid* als nichtleeren Durchschnitt eines supermodularen Polyeders mit dem negativen Orthanten des $\mathbb{R}^n$ definieren. Die Definition der submodularen Polyeder schließt daher auch Polymatroide ein, und wir brauchen uns im folgenden um diese nicht gesondert zu kümmern.

[10] Vergleiche [Edm 70, Definition (3), Theorem (14)].

Man spricht von einer *Linearen Transformation* eines super- oder submodularen Polyeders, im folgenden kurz super- oder submodulares LTP genannt, wenn $\mathcal{P}$ „linear“ durch einen echt positiven Vektor $a \in \mathbb{R}^n$ (d. h. mit $a_i > 0$ für alle $i \in I$) „gestreckt“ bzw. „gestaucht“ wird.

Definition 2.10 (Super- und submodulare LTP) *Sei $a \in \mathbb{R}^n$ ein echt positiver Vektor, und $f : 2^I \longrightarrow \mathbb{R}$ normiert, dann nennt man*

$$\mathcal{P}_{\geq}(a, f) = \{x \in \mathbb{R}^n \mid \sum_{i \in S} a_i x_i \geq f(S) \quad \text{für alle} \quad S \subseteq I\}$$

lineare Transformation eines supermodularen Polyeders (supermodulares LTP), falls f supermodular ist. Analog ist

$$\mathcal{P}_{\leq}(a, f) = \{x \in \mathbb{R}^n \mid \sum_{i \in S} a_i x_i \leq f(S) \quad \text{für alle} \quad S \subseteq I\}$$

die lineare Transformation eines submodularen Polyeders (submodulares LTP), falls f submodular ist.

Offenbar ist $(x_1, \ldots, x_n)^t \in \mathcal{P}_{\geq}(a, f)$ genau dann, wenn $(a_1 x_1, \ldots, a_n x_n)^t \in \mathcal{P}_{\geq}(f)$ gilt. Der Spezialfall $a_i = 1$ für alle $i = 1, \ldots, n$ liefert also gerade die super- und submodularen Polyeder.

2.3.3 Erweiterte Polymatroide und Contra-Polymatroide

Bhattacharya et al. [BGT 92] sowie Bertsimas und Niño-Mora [BN-M 93] schlagen eine weitere Verallgemeinerung der super- und submodularen Polyeder vor. Zur Definition brauchen wir einige Vorbereitungen.

Es sei wieder $f : 2^I \longrightarrow \mathbb{R}$ normiert, und seien $a_{S,i}$ reelle Koeffizienten für alle $S \subseteq I$ und alle $i \in I$ mit folgenden Eigenschaften:

$$\begin{aligned} a_{S,i} &> 0 \quad \text{für alle} \ \ i \in S, \\ a_{S,i} &= 0 \quad \text{für alle} \ \ i \in I \setminus S. \end{aligned}$$

Man beachte, daß im allgemeinen $a_{S,i} \neq a_{T,i}$ für $i \in S \cap T$ und $S \neq T$ gilt. Wir bezeichnen die zugehörige Matrix dieser Koeffizienten mit A, so daß $A = (a_{S,i})_{\substack{S \subseteq I \\ i \in I}}$ eine $\mathbb{R}^{2^n \times n}$-Matrix ist. Wir definieren die Polyeder

$$\mathcal{P}_{\geq}(A, f) = \{x \in \mathbb{R}^n \mid \sum_{i \in S} a_{S,i}\, x_i \geq f(S) \quad \text{für alle} \quad S \subseteq I\}$$

sowie

$$\mathcal{P}_{\leq}(A,f) = \{x \in \mathbb{R}^n \mid \sum_{i \in S} a_{S,i}\, x_i \leq f(S) \quad \text{für alle} \quad S \subseteq I\}.$$

Sei ferner π eine beliebige Permutation auf I, dann hat das lineare Gleichungssystem

$$\begin{pmatrix} a_{\{\pi_1\},\pi_1} & 0 & \cdots & 0 \\ a_{\{\pi_1,\pi_2\},\pi_1} & a_{\{\pi_1,\pi_2\},\pi_2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ a_{\{\pi_1,\ldots,\pi_n\},\pi_1} & a_{\{\pi_1,\ldots,\pi_n\},\pi_2} & \cdots & a_{\{\pi_1,\ldots,\pi_n\},\pi_n} \end{pmatrix} \begin{pmatrix} x_{\pi_1} \\ x_{\pi_2} \\ \vdots \\ x_{\pi_n} \end{pmatrix} = \begin{pmatrix} f(\{\pi_1\}) \\ f(\{\pi_1,\pi_2\}) \\ \vdots \\ f(\{\pi_1,\ldots,\pi_n\}) \end{pmatrix}$$

genau eine Lösung, denn nach Voraussetzung sind alle Diagonalelemente von Null verschieden. Bezeichne $x^\pi \in \mathbb{R}^n$ diese Lösung für eine beliebige Permutation π auf I. Es gilt also für x^π

$$\sum_{i=1}^{k} a_{\{\pi_1,\ldots,\pi_k\},\pi_i}\, x^\pi_{\pi_i} = f(\{\pi_1,\ldots,\pi_k\}) \quad \text{für alle} \quad k = 1,\ldots,n\,. \tag{2.6}$$

Definition 2.11 (Erweitertes Contra-Polymatroid, erweitertes Polymatroid)
Wir nennen $\mathcal{P}_{\geq}(A,f)$ ein erweitertes Contra-Polymatroid, wenn

$$x^\pi \in \mathcal{P}_{\geq}(A,f) \qquad \textit{für jede Permutation } \pi \textit{ auf } I. \tag{2.7}$$

Entsprechend heißt $\mathcal{P}_{\leq}(A,f)$ ein erweitertes Polymatroid, wenn $x^\pi \in \mathcal{P}_{\leq}(A,f)$ für jede Permutation π auf I.

Zunächst einige Bemerkungen zur Definition erweiterter Contra-Polymatroide (analoges gilt für erweiterte Polymatroide):

Der erste Unterschied zur Definition supermodularer LTP ist der, daß die Koeffizienten $a_{S,i}$ zu x_i nun auch davon abhängen, über *welche* Mengen S summiert wird. Weiter entfällt die Forderung der Supermodularität an f, stattdessen wird (2.7) verlangt. Aus (2.6) folgt bereits, daß die Vektoren x^π eine gewisse „Minimalitätseigenschaft“ besitzen, denn *jeder* Vektor aus $\mathcal{P}_{\geq}(A,f)$ erfüllt (2.6) mit „$\geq$“.

Tatsächlich kann man diese Bedingung auch anders auffassen. Bei gegebener Matrix A stellt (2.7) eine Bedingung an die Mengenfunktion f dar: Die Werte von f müssen auf allen Teilmengen von I so beschaffen sein, daß die Vektoren x^π alle in $\mathcal{P}_{\geq}(A,f)$ liegen. Im Spezialfall, daß die Koeffizienten $a_{S,i}$ *nicht* von S abhängen ist die Forderung (2.7) äquivalent zur Supermodularität von f[11].

Wir wollen an einem kleinen Beispiel die Definition erweiterter Contra-Polymatroide illustrieren:

[11] Vergleiche Satz 2.6 dieser Arbeit.

Beispiel 2.1 Wir betrachten ein Contra-Polymatroid im $\mathbb{R}^2$. Es ist also $I = \{1, 2\}$, und seien die (nichttrivialen) Koeffizienten $a_{S,i}$ gegeben durch:

$$a_{\{1\},1} = 2, \quad a_{\{1,2\},1} = 2, \quad a_{\{2\},2} = 4, \quad a_{\{1,2\},2} = 1.$$

Sei ferner

$$f(\{1\}) = 10, \quad f(\{2\}) = 40, \quad f(\{1, 2\} = 40.$$

Die Bedingungen $\sum_{i \in S} a_{S,i}\, x_i \geq f(S)$ für alle $S \subseteq I$ entsprechen dann gerade dem folgenden System:

$$\begin{pmatrix} 2 & 0 \\ 0 & 4 \\ 2 & 1 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} \geq \begin{pmatrix} 10 \\ 40 \\ 40 \end{pmatrix}.$$

Die Vektoren $x^{(1,2)}$ bzw. $x^{(2,1)}$ sind durch die Gleichungssysteme $2x_1 = 10$ und $2x_1 + x_2 = 40$ bzw. $4x_2 = 40$ und $2x_1 + x_2 = 40$ definiert.

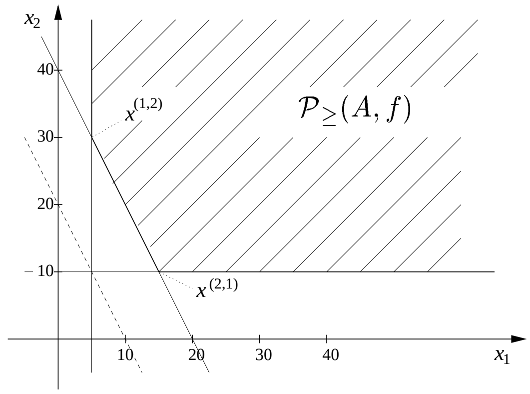


Abbildung 2.1: Ein Contra-Polymatroid im $\mathbb{R}^2$

Wir erhalten:

$$x^{(1,2)} = \begin{pmatrix} 5 \\ 30 \end{pmatrix} \qquad x^{(2,1)} = \begin{pmatrix} 15 \\ 10 \end{pmatrix}$$

Abbildung 2.1 zeigt das Contra-Polymatroid und die Punkte $x^{(1,2)}$ sowie $x^{(2,1)}$.

Bedingung (2.7) garantiert nun, daß die durch eine *beliebige* Ungleichung definierte Seitenfläche *nichtleer* ist. Im Beispiel ist etwa $f(\{1, 2\}) = 15$ unter Beibehaltung aller weiteren Parameter der minimale Wert für f auf der Menge $\{1, 2\}$, da sonst die Vektoren $x^{(1,2)}$ und $x^{(2,1)}$ nicht in $\mathcal{P}_{\geq}(A, f)$ lägen. Für $f(\{1, 2\}) = 15$ ist die zugehörige Gerade gestrichelt in Abbildung 2.1 eingezeichnet. ■

Bei Bertsimas und Niño-Mora [BN-M 93] findet man den Hinweis, daß sich unter Zuhilfenahme des Greedy-Algorithmus von Edmonds [Edm 70], und mit Hilfe der Optimalität des *erweiterten Greedy-Algorithmus*[12] (Contra-)Polymatroide in die erweiterten (Contra-)-Polymatroide einordnen lassen.

Wir geben mit dem folgenden Satz ein Resultat an, das die supermodularen LTP direkt und lediglich unter Benutzung von Lemma 2.4 in die erweiterten Contra-Polymatroide einordnet. Gleichzeitig liefert der Satz eine neue Beschreibung für supermodulare LTP.

Es sei $\mathcal{P}_{\geq}(A,f)$ ein erweitertes Contra-Polymatroid mit zugehöriger Matrix A und normierter Abbildung $f : 2^I \longrightarrow \mathbb{R}$. Falls $a_{S,i} = a_{T,i}$ für alle $i \in I$ und alle $S,T \subseteq I$ gilt, können wir auch a_i statt $a_{S,i}$ schreiben, denn die Koeffizienten hängen dann nicht von S ab. Dann ist $\mathcal{P}_{\geq}(A,f) = \mathcal{P}_{\geq}(a,f)$ ein *supermodulares LTP* mit dem echt positiven Vektor $a = (a_1,\ldots,a_n)^t$ und f. Genauer gilt die folgende Äquivalenz[13]:

Satz 2.6 *Sei*

$$\mathcal{P}_{\geq}(a,f) = \{x \in \mathbb{R}^n \mid \sum_{i\in S} a_i x_i \geq f(S) \quad \textit{für alle} \quad S \subseteq I\}$$

ein Polyeder mit $a_i > 0$ für alle $i \in I$ und normierter Abbildung $f : 2^I \longrightarrow \mathbb{R}$, dann sind äquivalent:

(1) f ist supermodular

(2) Es gilt $x^\pi \in \mathcal{P}_{\geq}(a,f)$ für alle Permutationen π von I.

Beweis. Wenn die Koeffizienten $a_{S,i}$ nicht von S abhängen, so haben die Vektoren x^π die Gestalt

$$x^\pi_{\pi_k} = \frac{f(\{\pi_1,\ldots,\pi_k\}) - f(\{\pi_1,\ldots,\pi_{k-1}\})}{a_{\pi_k}} \quad \text{für alle} \quad k = 1,\ldots,n\ , \tag{2.8}$$

wobei wieder $\{\pi_1,\ldots,\pi_0\} := \emptyset$ sei. Man überzeugt sich leicht anhand von (2.6) durch sukzessive Berechnung von $x^\pi_{\pi_1}, x^\pi_{\pi_2},\ldots$ etc. von der Darstellung.

Sei nun f supermodular und π eine beliebige Permutation von I. Wir haben zu zeigen, daß x^π die Ungleichungen $\sum_{i\in S} a_i\, x^\pi_i \;\geq\; f(S)$ für alle $S \subseteq I$ erfüllt. Aufgrund der Bijektivität von π ist das äquivalent zu: $\sum_{i\in S} a_{\pi_i}\, x^\pi_{\pi_i} \;\geq\; f(\bigcup_{i\in S}\{\pi_i\})$ für alle $S \subseteq I$. Daher können wir o. B. d. A. $\pi = (1,\ldots,n)$ annehmen.
Wir beweisen die Behauptung per vollständiger Induktion über $|S|$.
Wegen Supermodularität von f gilt $f(\{1,\ldots,k\}) \geq f(\{1,\ldots,k-1\}) + f(\{k\})$ und daher $a_k x^\pi_k = f(\{1,\ldots,k\}) - f(\{1,\ldots,k-1\}) \geq f(\{k\})$ für alle $k = 1,\ldots,n$. Also gilt die

[12] Vergleich Abschnitt 2.4 dieser Arbeit.
[13] Der Satz gilt ebenso für erweiterte Polymatroide und submodulare LTP, der Beweis verläuft identisch.

Behauptung für $|S| = 1$. Sei nun $|S| = k+1$ beliebig und gelte die Behauptung für alle S mit $|S| = k$, ferner sei $m := \max\{i \mid i \in S\}$. Dann gilt

$$\begin{aligned}
\sum_{i \in S} a_i x_i^\pi &= \sum_{i \in S \setminus \{m\}} a_i x_i^\pi \;+\; a_m x_m^\pi \\
&\geq f(S \setminus \{m\}) \;+\; a_m x_m^\pi \qquad\qquad \text{(per Induktionsvoraussetzung)} \\
&= f(S \setminus \{m\}) + f(\{1,\dots,m\}) - f(\{1,\dots,m-1\}) \\
&\geq f(S).
\end{aligned}$$

Dabei gilt die letzte Ungleichung nach Definition von m aufgrund der Supermodularität von f, denn

$$f(\underbrace{S \cap \{1,\dots,m-1\}}_{=S\setminus\{m\}}) \;+\; f(\underbrace{S \cup \{1,\dots,m-1\}}_{=\{1,\dots,m\}}) \;\geq\; f(S) \;+\; f(\{1,\dots,m-1\}).$$

Also gilt tatsächlich $x^\pi \in \mathcal{P}_\geq(a,f)$.

Sei umgekehrt $x^\pi \in \mathcal{P}_\geq(a,f)$ für alle Permutationen π von I. Wir wollen Lemma 2.4 zum Nachweis der Supermodularität von f anwenden und zeigen, daß

$$f_k(S) = f(S \cup \{k\}) - f(S) \qquad \text{für } S \subseteq I \setminus \{k\}$$

nichtfallend ist für alle $k \in I$. Wir führen einen Widerspruchsbeweis. Sei also $k \in I$ beliebig und angenommen, es gebe Mengen $S, T \subseteq I \setminus \{k\}$ mit $S \subset T$ und $f_k(S) > f_k(T)$. Damit folgt per Definition von f_k

$$f(S \cup \{k\}) - f(S) \;>\; f(T \cup \{k\}) - f(T). \tag{2.9}$$

Sei π eine Permutation von I mit

$$\{\pi_1,\dots,\pi_{|S|}\} = S, \quad \{\pi_1,\dots,\pi_{|T|}\} = T \quad \textit{und} \quad \pi_{|T|+1} = k.$$

Für den zugehörigen Vektor x^π gilt dann

$$a_k x_k^\pi \;=\; f(T \cup \{k\}) - f(T)$$

nach (2.8) und Wahl von π. Weiter gilt

$$\sum_{i \in S} a_i x_i^\pi \;=\; \sum_{i \in S} a_{\pi_i} x_{\pi_i}^\pi \;=\; f(S)$$

nach (2.6) und Wahl von π. Damit ergibt sich

$$\begin{aligned}
\sum_{i \in S \cup \{k\}} a_i x_i^\pi &= \sum_{i \in S} a_i x_i^\pi \;+\; a_k x_k^\pi \\
&= f(S) + f(T \cup \{k\}) - f(T) \\
&< f(S \cup \{k\})\;. \qquad\qquad \text{(nach (2.9))}
\end{aligned}$$

Da $x^\pi \in \mathcal{P}_\geq(a,f)$ gelten soll, muß in der letzten Zeile aber „ $\geq$ “ gelten. Dies ist der gewünschte Widerspruch und die Äquivalenz ist bewiesen. ∎

Wir gehen mit den folgenden beiden Sätzen kurz auf polyedrische Eigenschaften erweiterter Contra-Polymatroide ein. Für den Rest Kapitels werden wir uns auf erweiterte Contra-Polymatroide beschränken; sämtliche Aussagen gelten für erweiterte Polymatroide analog.

Satz 2.7 *Ein erweitertes Contra-Polymatroid ist spitz.*

Beweis. Wir schreiben $\mathcal{P}_\geq(A,f)$ in der Form $\{x \in \mathbb{R}^n \mid Ax \geq b\}$ mit passender Matrix $A \in \mathbb{R}^{2^n \times n}$ und $b \in \mathbb{R}^{2^n}$; also so, daß jeder der 2^n Zeilen des Systems $Ax \geq b$ eine Ungleichung $\sum_{i\in S} a_{S,i}\, x_i \geq f(S)$ für ein $S \subseteq I$ entspricht. Dann sind die zu den einelementigen Mengen $\{i\}$ gehörigen Zeilen von A linear unabhängig, und es gilt demnach $Rg(A) = n$. Wegen Lemma 2.1 ist $\mathcal{P}_\geq(A,f)$ dann spitz.

Für die innere Beschreibung eines erweiterten Contra-Polymatroids erhält man:

Satz 2.8 *Es sei E die Eckenmenge eines erweiterten Contra-Polymatroids $\mathcal{P}_\geq(A,f)$. Dann gilt*

$$\mathcal{P}_\geq(A,f) = \mathrm{conv}(E) + \mathbb{R}^n_+$$

Beweis. Wir müssen nach Satz 2.3 lediglich nachweisen, daß der Rezessionskegel von $\mathcal{P}_\geq(A,f)$ durch $\mathbb{R}^n_+$ gegeben ist. Ist $\{x \in \mathbb{R}^n \mid Ax \geq b\}$ eine Darstellung von $\mathcal{P}_\geq(A,f)$, so ist der Rezessionskegel gerade $\{x \in \mathbb{R}^n \mid Ax \geq 0\}$. Das bedeutet

$$\mathrm{rec}(\mathcal{P}_\geq(A,f)) = \{x \in \mathbb{R}^n \mid \sum_{i\in S} a_{S,i}\, x_i \geq 0 \quad \text{für alle} \quad S \subseteq I\}.$$

Wegen $a_{S,i} \geq 0$ folgt $\mathbb{R}^n_+ \subseteq \mathrm{rec}(\mathcal{P}_\geq(A,f))$.
Für ein $x \in \mathrm{rec}(\mathcal{P}_\geq(A,f))$ gilt wegen $a_{\{i\},i} > 0$ aber auch $x_i \geq 0$ für alle $i \in I$, also folgt auch $\mathrm{rec}(\mathcal{P}_\geq(A,f)) \subseteq \mathbb{R}^n_+$. ∎

2.3.4 Basispolytope und Optimierung

Wir wollen uns nun mit einer für Optimierungszwecke entscheidenden Seitenfläche von $\mathcal{P}_\geq(A,f)$ befassen. Die Seitenfläche zur gültigen Ungleichung $\sum_{i\in I} a_{I,i} x_i \geq f(I)$ wird das *Basispolytop* eines erweiterten Contra-Polymatroids genannt.

Definition 2.12 (Basispolytop) *Sei $\mathcal{P}_\geq(A,f)$ ein erweitertes Contra-Polymatroid, dann nennt man*

$$\mathcal{B}_\geq(A,f) := \{x \in \mathcal{P}_\geq(A,f) \mid \sum_{i\in I} a_{I,i} x_i = f(I)\ \}$$

das Basispolytop *zu* $\mathcal{P}_\geq(A,f)$.

Die Vektoren des Basispolytops nennt man auch *Basen* des Contra-Polymatroids; vergleiche zur Definition einer Basis auch [Edm 70, Definition (3)]. Eine erste Rechtfertigung für die Namensgebung von $\mathcal{B}_{\geq}(A,f)$ liefert das folgende Lemma.

Lemma 2.5 *$\mathcal{B}_{\geq}(A,f)$ gemäß Definition 2.12 ist nichtleer und beschränkt, also in der Tat ein Polytop.*

Beweis. Wegen $x^{\pi} \in \mathcal{B}_{\geq}(A,f)$ für alle Permutationen π von I ist $\mathcal{B}_{\geq}(A,f)$ trivialerweise nichtleer. Zur Beschränktheit: Wegen $a_{\{i\},i}\, x_i \geq f(\{i\})$ und $a_{\{i\},i} > 0$ für alle $i \in I$ sind alle x_i nach unten beschränkt. Aus $\sum_{i\in I} a_{I,i}\, x_i = f(I)$ und $a_{I,i} > 0$ für alle $i \in I$ folgt damit auch die Beschränktheit aller x_i nach oben. ■

Es gilt für die Vektoren aus $\mathcal{B}_{\geq}(A,f)$:

Satz 2.9 *Das Basispolytop $\mathcal{B}_{\geq}(A,f)$ eines erweiterten Contra-Polymatroids $\mathcal{P}_{\geq}(A,f)$ enthält genau alle bezüglich „ $\leq$ “ minimalen Vektoren von $\mathcal{P}_{\geq}(A,f)$.*

Beweis. Eine Richtung des Beweises ist trivial: Ein Vektor $x \in \mathcal{B}_{\geq}(A,f)$ ist sicherlich minimal bzgl. „$\leq$“, denn wegen $\sum_{i\in I} a_{I,i}x_i = f(I)$ und $a_{I,i} > 0$ für alle $i \in I$ kann keine Komponente von x verkleinert werden, ohne die Ungleichung $\sum_{i\in I} a_{I,i}x_i \geq f(I)$ zu verletzen.

Für die umgekehrte Richtung ist die Gültigkeit von $\sum_{i\in I} a_{I,i}x_i = f(I)$ für alle „$\leq$“-minimalen Vektoren $x \in \mathcal{P}_{\geq}(A,f)$ zu zeigen. Der Beweis beruht auf den folgenden Resultaten für Optimierungsprobleme über erweiterten Contra-Polymatroiden[14] und ist im Beweis zu Lemma 2.8 angegeben. ■

Lemma 2.6 *Sei $\mathcal{P}_{\geq}(A,f)$ ein erweitertes Contra-Polymatroid, $S \subseteq I$ beliebig und $w_i \geq 0$ für alle $i \in S$. Dann hat das lineare Optimierungsproblem*

$$\min\{\sum_{i\in S} w_i x_i \mid x \in \mathcal{P}_{\geq}(A,f)\} \tag{2.10}$$

eine endliche Optimallösung. Eine Lösung ist gegeben durch x^{π} für eine Permutation π von I mit $\bigcup_{i\in S}\{\pi_i\} = S$.

Beweis. Da $\mathcal{P}_{\geq}(A,f)$ nichtleer ist ist (2.10) nicht unzulässig. Da ferner alle x_i vermöge der Ungleichungen $a_{\{i\},i}\, x_i \geq f(\{i\})$ nach unten beschränkt sind, ist (2.10) wegen $w_i \geq 0$ auch nicht unbeschränkt. Zum Beweis der letzten Behauptung wird eine Induktion über $|S|$ geführt, hierzu vergleiche [BGT 92, Lemma 1] und den dort angegebenen Beweis . ■

[14] Vergleiche [BGT 92].

Insbesondere impliziert dieses Lemma die bisher nur vage angedeutete „Minimalitätseigenschaft“ der Punkte x^π:

Korollar 2.3 *Ein* **LP** *der Art*

$$\min\{w^t x \mid x \in \mathcal{P}_{\geq}(A,f)\} \tag{2.11}$$

ist genau dann unbeschränkt, wenn ein $w_i < 0$ existiert. Andernfalls existiert eine Permutation π von I, so daß x^π eine Optimallösung ist.

Die Unbeschränktheit im Falle der Existenz eines $w_i < 0$ folgt unmittelbar aus der Darstellung $\mathcal{P}_{\geq}(A,f) = \text{conv}(E) + \mathbb{R}^n_+$ aus Satz 2.8, denn man kann jede Komponente x_i beliebig groß wählen, ohne $\mathcal{P}_{\geq}(A,f)$ zu verlassen.

Wir zitieren noch das folgende Resultat, dessen Beweis auf Lemma 2.6 beruht[15].

Lemma 2.7 *Es sei $\mathcal{P}_{\geq}(A,f)$ erweitertes Contra-Polymatroid und gelte für ein $x \in \mathcal{P}_{\geq}(A,f)$ sowohl $\sum_{i\in S} a_{S,i}\, x_i = f(S)$ als auch $\sum_{i \in T} a_{T,i}\, x_i = f(T)$ für zwei Mengen $S, T \subseteq I$. Dann gilt auch*

$$\sum_{i \in S\cup T} a_{S\cup T,i}\, x_i = f(S \cup T).$$

Nun sind wir in der Lage, Satz 2.9 zu beweisen, und es gilt:

Lemma 2.8 (Beweis zu Satz 2.9) *Für einen beliebigen „ $\leq$ “-minimalen Vektor $x \in \mathcal{P}_{\geq}(A,f)$ gilt $\sum_{i\in I} a_{I,i}\, x_i = f(I)$, also $x \in \mathcal{B}_{\geq}(A,f)$.*
Weiter existiert für alle $x \in \mathcal{P}_{\geq}(A,f)$ mit $\sum_{i\in I} a_{I,i}\, x_i > f(I)$ ein $x' \in \mathcal{B}_{\geq}(A,f)$ mit $x' \leq x$ und es gilt $x'_j < x_j$ für mindestens ein $j \in I$.

Beweis. Sei x minimal bezüglich „ $\leq$ “ in $\mathcal{P}_{\geq}(A,f)$. Dann ist für jede Komponente x_j von x wenigstens eine der Ungleichungen $\sum_{i\in S} a_{S,i}\, x_i \geq f(S)$ mit $j \in S \subseteq I$ „scharf“, also mit Gleichheit erfüllt. Denn andernfalls könnte man x_j noch verringern, ohne die Ungleichungen von $\mathcal{P}_{\geq}(A,f)$ zu verletzen, und x wäre nicht „$\leq$“-minimal in $\mathcal{P}_{\geq}(A,f)$. Also existiert für alle $j \in I$ ein Menge $S_j \subseteq I$ mit $j \in S_j$ und $\sum_{i\in S_j} a_{S_j,i}\, x_i = f(S_j)$. Zusammen mit Lemma 2.7 folgt dann induktiv

$$\sum_{i\in I} a_{I,i}\, x_i = f(\bigcup_{j \in I} S_j) = f(I),$$

also gilt $x \in \mathcal{B}_{\geq}(A,f)$.

[15] Vergleiche [BGT 92, Lemma 2, Lemma 3] und die dort geführten Beweise.

Zur zweiten Behauptung: Für ein $x \in \mathcal{P}_{\geq}(A, f)$ mit $\sum_{i \in I} a_{I,i}\, x_i > f(I)$ existiert wenigstens ein $x_j, j \in I$, für das *keine* der Ungleichungen scharf ist, denn andernfalls wäre x „$\leq$“-minimal, und nach bisher gezeigtem wäre damit $\sum_{i \in I} a_{I,i}\, x_i = f(I)$. Es gilt für alle $J \subseteq I$ mit $j \in J$ dann $\sum_{i \in J} a_{J,i}\, x_i > f(J)$. Daher kann x_j solange verringert werden, bis eine der Ungleichungen scharf wird. Per Konstruktion gilt für den so erhaltenen Punkt x' dann $x' \in \mathcal{P}_{\geq}(A, f)$, $x' \leq x$, $x'_j < x_j$ und x'_j kann nicht mehr verringert werden, ohne eine Ungleichung von $\mathcal{P}_{\geq}(A, f)$ zu verletzen. Entweder liegt x' bereits in $\mathcal{B}_{\geq}(A, f)$, oder dasselbe Verfahren kann für eine weitere Komponente x'_k von x' angewendet werden.
Nach endlich vielen Schritten erhält man so einen Punkt $x' \in \mathcal{B}_{\geq}(A, f)$ mit den geforderten Eigenschaften. ■

Mit Lemma 2.8 folgt für $w > 0$ sogar:

Korollar 2.4 *Sei $w \in \mathbb{R}^n$ echt positiv und $\mathcal{P}_{\geq}(A, f)$ ein erweitertes Contra-Polymatroid. Dann liegt jede Optimallösung x^{opt} des* **LP** *(2.11) bereits im Basispolytop $\mathcal{B}_{\geq}(A, f)$.*

Falls ein $w_i = 0$ existiert, muß das nicht gelten, denn die Zielfunktion bleibt von einer beliebigen Erhöhung von x_i unbeeinflußt. Jedoch existiert nach Lemma 2.8 stets auch eine Optimallösung, die im Basispolytop liegt.

Für Optimierungszwecke können wir uns daher auf das zugehörige Basispolytop beschränken. Das bedeutet, daß wir zur Lösung eines **LP** über einem erweiterten Contra-Polymatroid $\mathcal{P}_{\geq}(A, f)$ o. B. d. A. die zusätzliche Gleichung $\sum_{i \in I} a_{I,i}\, x_i = f(I)$ einführen können.

Ferner wissen wir aus Korollar 2.3, daß stets auch ein x^π existiert, welches eine Optimallösung ist. Erwartungsgemäß stellen die Punkte x^π genau die Ecken des Basispolytops dar:

Satz 2.10 *Es sei $\mathcal{B}_{\geq}(A, f)$ das Basispolytop eines erweiterten Contra-Polymatroids und E die Eckenmenge von $\mathcal{B}_{\geq}(A, f)$. Dann gilt:*

$$E = \{x^\pi \mid \pi \text{ Permutation von } I\}$$

Beweis. Sei e eine Ecke von $\mathcal{B}_{\geq}(A, f)$. Dann gibt es einen Vektor $w \in \mathbb{R}^n$, so daß e *einzige* Optimallösung des linearen Programms $\min\{w^t x \mid x \in \mathcal{B}_{\geq}(A, f)\}$ ist. Das Optimalitätsresultat für Algorithmus G[16] besagt dann, daß $e = x^\pi$ gelten muß mit einer passenden Permutation π auf I. Also gilt $E \subseteq \{x^\pi \mid \pi \textit{ Permutation von } I\}$.

Umgekehrt ist zu zeigen, daß jeder Punkt x^π tatsächlich eine Ecke von $\mathcal{B}_{\geq}(A, f)$ ist. Nehmen wir das Gegenteil an. Also gebe es $y, z \in \mathcal{B}_{\geq}(A, f)$, $y \neq z$ so daß $x^\pi = \lambda y + (1-\lambda) z$

[16] Vergleiche Abschnitt 2.4.

für $0 < \lambda < 1$. Wir können o. B. d. A. $\pi = (1, \ldots, n)$ annehmen. Es gilt per Definition von x^π dann $a_{\{1\},1}\, x_1^\pi = f(\{1\})$, vergleiche (2.6). Wegen $y, z \in \mathcal{B}_\geq(A, f)$ folgt

$$f(\{1\}) = a_{\{1\},1}\, x_1^\pi = \lambda \underbrace{a_{\{1\},1}\, y_1}_{\geq f(\{1\})} + (1-\lambda) \underbrace{a_{\{1\},1}\, z_1}_{\geq f(\{1\})},$$

und daher gilt $x_1^\pi = y_1 = z_1 = \frac{f(\{1\})}{a_{\{1\},1}}$. Fährt man derart sukzessive anhand von (2.6) mit $x_2^\pi,\ x_3^\pi, \ldots$ fort, so erhält man letztendlich den Widerspruch $x^\pi = y = z$. Also ist x^π doch eine Ecke, und der Beweis ist vollständig. ■

Für ein erweitertes Contra-Polymatroid folgern wir mit Satz 2.8:

Korollar 2.5 *Es sei $\mathcal{P}_\geq(A, f)$ ein erweitertes Contra-Polymatroid, dann gilt*

$$\mathcal{P}_\geq(A, f) \ = \mathrm{conv}\{x^\pi \mid \pi \text{ Permutation von } I\} \ + \ \mathbb{R}_+^n.$$

Die folgende Skizze greift auf Beispiel 2.1 zurück, und veranschaulicht das Resultat im zweidimensionalen Fall.

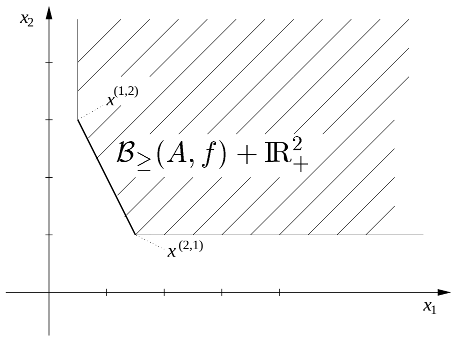


Abbildung 2.2: Das Basispolytop eines Contra-Polymatroids

2.4 Der erweiterte Greedy-Algorithmus

Wir wissen bereits, eine Optimallösung eines linearen Programms über einem erweiterten Contra-Polymatroid im zugehörige Basispolytop zu finden ist, falls eine endliche Optimallösung existiert.

Unsere Aufgabe ist nun, eine passende Permutation π auf I, also eine optimale Ecke x^π zu bestimmen. Dazu werden wir in diesem Abschnitt einen Algorithmus angeben, der sowohl eine passende Permutation π als auch eine Lösung für das *duale* **LP** berechnet. Der Optimalitätsbeweis beruht schließlich auf dem schwachen Dualitätssatz.

Der Algorithmus wird von Bhattacharya et al. [BGT 92] angegeben und von Bertsimas und Niño-Mora *erweiterter greedy-Algorithmus* genannt [BN-M 93]. Dieser Name trägt der

Tatsache Rechnung, daß der Algorithmus im Spezialfall super- und submodularer Polyeder (oder LTP) genau dem durch Edmonds angegebenen *greedy*-Algorithmus für Polymatroide entspricht[17].

Wir betrachten ein lineares Programm über dem Basispolytop $\mathcal{B}_{\geq}(A,f)$ eines erweiterten Contra-Polymatroids $\mathcal{P}_{\geq}(A,f)$, also

$$\min\{w^t x \mid \sum_{i\in S} a_{S,i}\, x_i \geq f(S) \quad \forall \;\; S\subseteq I \;\; \text{und} \;\; \sum_{i\in I} a_{I,i}\, x_i = f(I)\} \tag{P}$$

mit einem beliebigen Vektor $w \in \mathbb{R}^n$. Das zu (P) duale lineare Programm ist

$$\max\{\sum_{S\subseteq I} y_S f(S) \mid \sum_{S\ni i} a_{S,i}\, y_S = w_i \quad \forall \;\; i\in I \;\; \text{und} \;\; y_S \geq 0 \quad \forall \;\; S\subset I\}. \tag{D}$$

Der folgende Algorithmus erzeugt eine Permutation π von I und einen Vektor $\nu \in \mathbb{R}^n$.

Algorithmus G

Input $w = (w_1,\ldots,w_n)^t \in \mathbb{R}^n$, $A = (a_{S,i})_{\substack{S \subseteq I \\ i \in I}} \in \mathbb{R}^{2^n\times n}$ mit

$a_{S,i} > 0$ für alle $i\in S$ und $a_{S,i} = 0$ für alle $i \in I\setminus S$

Output $\pi = (\pi_1,\ldots,\pi_n)$ Permutation auf I

$\nu = (\nu_1,\ldots,\nu_n)^t \in \mathbb{R}^n$

Step 1 $S_n \leftarrow I$

$$\nu_n \leftarrow \min_{i\in S_n} \frac{w_i}{a_{S_n,i}}$$

$\pi_n \leftarrow i \in S_n$, das $\dfrac{w_i}{a_{S_n,i}}$ minimiert

Step k **do for** $k = n-1,\ldots,1$**:**

$S_k \leftarrow S_{k+1}\setminus\{\pi_{k+1}\}$

$$\nu_k \leftarrow \min_{i\in S_k} \frac{w_i - \sum\limits_{j=k+1}^{n} a_{S_j,i}\,\nu_j}{a_{S_k,i}} \tag{$*$}$$

$\pi_k \leftarrow i \in S_k$, das in $(*)$ das Minimum liefert

od

Ausgehend von ν definieren wir eine Lösung $y \in \mathbb{R}^{2^n}$ für das duale **LP**.

[17] Vergleiche zu Edmonds' Algorithmus [Edm 70] sowie auch [Fuj 91] oder [Lov 82].

Für $S \subseteq I$ beliebig sei

$$y_S := \begin{cases} \nu_k & : \quad \text{falls } S = S_k \text{ für ein } k \in I, \\ 0 & : \quad \text{sonst.} \end{cases} \tag{2.12}$$

Zur Korrektheit und Komplexität des Algorithmus:

Korrektheit: Der Algorithmus erzeugt eine korrekte Permutation π von I: Denn ist im $(k+1)$-ten Schritt π_{k+1} gewählt worden, kann im nächsten Schritt (also im k-ten) wegen $\pi_k \in S_k = S_{k+1} \setminus \{\pi_{k+1}\} = I \setminus \{\pi_{k+1}, \dots, \pi_n\}$ niemals derselbe Index ein zweites mal gewählt werden.
Die Definition der ν_k bereitet keine Schwierigkeiten, denn es ist $a_{S_k,i} > 0$ für alle $i \in S_k$ und $k = 1, \dots, n$, also ist der Quotient in Schritt $(*)$ stets wohldefiniert.
Weiter terminiert der Algorithmus trivialerweise nach n Schritten.

Komplexität: Die dominante Operation des Algorithmus ist offenbar Schritt $(*)$. Die Berechnung des Quotienten in $(*)$ für ein festes $i \in S_k$ ist beschränkt durch $\mathcal{O}(n)$ Rechenoperationen. Wir müssen dann das Minimum von maximal n Quotienten bestimmen. Damit ist die Bestimmung von ν_k in $(*)$ beschränkt durch $\mathcal{O}(n^2)$. Folglich ist die Komplexität des Algorithmus gegeben durch $\mathcal{O}(n^3)$.

Falls in $(*)$ sogenannte *ties* auftreten, muß sich der Algorithmus für einen der möglichen Kandidaten zur Setzung von π_k entscheiden. Das Problem kann durch eine beliebige *tie breaking rule* gelöst werden, etwa wähle man stets den lexikographisch größten Index, der in Frage kommt. Dadurch ist die erzeugte Permutation π offenbar von dieser Auswahlregel abhängig. In einem solchen Fall entsprechen die verschiedenen möglichen Lösungen π verschiedenen optimalen Ecken x^π des linearen Programms (dabei können auch Ecken „zusammenfallen“, wie Beispiel 2.1 zeigt).

Wir werden mit Satz 2.11 beweisen, daß die zu π gehörige Ecke x^π von $\mathcal{B}_{\geq}(A, f)$ zusammen mit y gemäß Definition (2.12) ein optimales primal-duales Paar für (P) bzw. (D) ist. Bevor wir dies tun, betrachten wir den Spezialfall supermodularer LTP:

Es sei $\mathcal{P}_{\geq}(a, f)$ ein supermodulares LTP mit zugehörigem Basispolytop $\mathcal{B}_{\geq}(a, f)$. Wir betrachten das lineare Programm $\min\{w^t x \mid x \in \mathcal{B}_{\geq}(a, f)\}$, also

$$\min\{w^t x \mid \sum_{i \in S} a_i\, x_i \geq f(S) \quad \forall \;\; S \subseteq I \;\; \text{und} \;\; \sum_{i \in I} a_i\, x_i = f(I)\}$$

Wir definieren $a_{S,i} := a_i$ für alle S mit $i \in S$ und wenden Algorithmus G mit Input w und der entsprechenden Matrix A an. Sei o. B. d. A.

$$\frac{w_1}{a_1} \geq \frac{w_2}{a_2} \geq \cdots \geq \frac{w_n}{a_n}. \tag{2.13}$$

Als Auswahlregel für eventuell auftretende *ties* wenden wir die Regel „lexikographisch größter Index zuerst“ an, dann erhalten wir:

Lemma 2.9 *Im Falle eines supermodularen LTP liefert die von Algorithmus G erzeugte Permutation π stets eine Ordnung nach absteigenden Quotienten $\frac{w_i}{a_i}$.*

Beweis. Mann kann stets a_i für alle $a_{S,i}$ mit $i \in S$ schreiben - insbesondere ist in Algorithmus G dann $a_{S_j,i} = a_{S_k,i} = a_i$ für alle $j \geq k$ und $i \in S_k$. Unter der Voraussetzung $\pi = (1, \dots, n)$ gilt $\pi_n = n$ und $\nu_n = \dfrac{w_n}{a_n}$ sowie $S_{n-1} = \{1, \dots, n-1\}$. Es folgt nun induktiv $S_k = \{1, \dots, k\}$ und für beliebiges ν_k:

$$\begin{aligned} \nu_k &= \min\{\frac{w_i}{a_i} - \sum_{j=k+1}^{n} \nu_j \mid i \in S_k\} \\ &= \min\{\frac{w_i}{a_i} - \frac{w_{k+1}}{a_{k+1}} \mid i \in S_k\} \quad = \quad \frac{w_k}{a_k} - \frac{w_{k+1}}{a_{k+1}}, \end{aligned}$$

für alle $k = n-1, \dots, 1$. Die letzte Gleichung gilt dabei wegen (2.13) und aufgrund der lexikographischen Auswahlregel. Damit ist $\pi = (1, \dots, n)$. ■

Das bedeutet, daß wir für diesen Spezialfall Algorithmus G durch ein sehr viel einfacheres Verfahren ersetzen können.

Algorithmus G'

Input $w = (w_1, \dots, w_n)^t \in \mathbb{R}^n$

$a = (a_1, \dots, a_n)^t \in \mathbb{R}^n$, $a_i > 0$ für alle $i = 1, \dots, n$

Output $\pi = (\pi_1, \dots, \pi_n)$ Permutation auf I

$\nu = (\nu_1, \dots, \nu_n)^t \in \mathbb{R}^n$

Step 1 Sortiere die Quotienten $\dfrac{w_i}{a_i}$ in absteigender Reihenfolge und definiere π derart, daß

$$\frac{w_{\pi_1}}{a_{\pi_1}} \geq \frac{w_{\pi_2}}{a_{\pi_2}} \geq \cdots \geq \frac{w_{\pi_n}}{a_{\pi_n}} \qquad \text{gilt.}$$

Step 2 Setze

$$\nu_n \leftarrow \frac{w_{\pi_n}}{a_{\pi_n}}, \qquad \nu_k \leftarrow \frac{w_{\pi_k}}{a_{\pi_k}} - \frac{w_{\pi_{k+1}}}{a_{\pi_{k+1}}} \quad \text{für} \quad k = n-1, \dots, 1$$

Die Definition der Permutation π zusammen mit der Definition von x^π (vergleiche (2.6)) entspricht gerade Edmonds' *greedy*-Algorithmus für Polymatroide.

Zur **Korrektheit** von Algorithmus G' ist nichts zu zeigen. Die Laufzeit wird von der Sortierung der Quotienten $\frac{w_i}{a_i}$ dominiert, und die **Komplexität** von Algorithmus G' ist daher $\mathcal{O}(n \log n)$.

Nun zum Beweis der Optimalität der Ecke x^π für (P), wenn π durch Algorithmus G erzeugt wird:

Satz 2.11 *Für das Optimierungsproblem* (P) *sei* π *eine Permutation, die durch Algorithmus G erzeugt wird. Dann ist die durch* x^π *gegebene Ecke von* $\mathcal{B}_\geq(A, f)$ *eine Optimallösung für* (P).

Beweis. Wie bereits angekündigt, werden wir zeigen, daß x^π zusammen mit y gemäß Definition (2.12) ein optimales primal-duales Paar ist. Vorerst müssen wir zeigen, daß y überhaupt eine *zulässige* Lösung für (D) ist.

Sei im folgenden o. B. d. A. $\pi = (1, \dots, n)$, dann ist $S_i = \{1, \dots, i\}$.

Zulässigkeit von y für (D): Wir müssen $\sum_{S \ni k} a_{S,k}\, y_S = w_k$ für alle $k \in I$ zeigen. Sei dazu $k \in I$ beliebig und fest. Per Definition von y sind lediglich die $y_S \neq 0$, für die $S = S_j$ für ein $j \in I$ gilt. Daher vereinfacht sich die Summe zu:

$$
\begin{aligned}
\sum_{S \ni k} a_{S,k}\, y_S &= \sum_{j=k}^{n} a_{S_j,k}\, \nu_j && \text{(per Definition } y \text{ und } S_j\text{)} \\
&= \sum_{j=k+1}^{n} a_{S_j,k}\, \nu_j \quad + \quad a_{S_k,k}\, \nu_k \\
&= \sum_{j=k+1}^{n} a_{S_j,k}\, \nu_j \quad + \quad a_{S_k,k} \left(\frac{w_k - \sum\limits_{j=k+1}^{n} a_{S_j,k}\, \nu_j}{a_{S_k,k}} \right) && \text{(per Definition } \nu_k\text{)} \\
&= w_k
\end{aligned}
$$

Zum vorletzten Schritt sei daran erinnert, daß wir $\pi = (1, \dots, n)$ angenommen hatten und daher ν_k tatsächlich die angegebene Form hat.

Ferner müssen wir $y_S \geq 0$ für alle $S \subset I$ zeigen. Per Definition von y_S ist also $\nu_k \geq 0$ für alle $k < n$ zu zeigen.
Aufgrund der Definition von ν_k (vergleiche Schritt $(*)$ in Algorithmus G), und wegen $a_{S_k,i} > 0$ für alle $i \in S_k$, reicht es

$$
w_i - \sum_{j=k+1}^{n} a_{S_j,i}\, \nu_j \quad \geq \quad 0 \text{ für alle } i \in S_k
$$

zu zeigen. Per Definition von ν_{k+1} als Minimum gilt nun

$$\begin{aligned}
\nu_{k+1} &\leq \frac{w_i - \sum\limits_{j=k+2}^{n} a_{S_j,i}\,\nu_j}{a_{S_{k+1},i}} \quad \text{für alle } i \in S_{k+1}, \\
&\iff \\
a_{S_{k+1},i}\,\nu_{k+1} &\leq w_i - \sum_{j=k+2}^{n} a_{S_j,i}\,\nu_j \quad \text{für alle } i \in S_{k+1}, \\
&\iff \\
w_i - \sum_{j=k+1}^{n} a_{S_j,i}\,\nu_j &\geq 0 \text{ für alle } i \in S_{k+1}.
\end{aligned}$$

Wegen $S_k \subset S_{k+1}$ folgt die Behauptung.

Optimalität des Lösungspaares x^π und y: Wir nutzen die Definitionen von y und x^π und die soeben bewiesene Zulässigkeit von y[18]. Mit $\pi = (1, \ldots, n)$ und $S_i = \{1, \ldots, i\}$ folgt dann

$$\begin{aligned}
\sum_{S \subseteq I} y_S f(S) &= \sum_{i=1}^{n} \nu_i f(S_i) && \text{(per Def. } y) \\
&= \sum_{i=1}^{n} \left(\sum_{k=1}^{i} a_{S_i,k}\, x_k^\pi \right) \nu_i && \text{(per Def. } x^\pi) \\
&= \sum_{k=1}^{n} x_k^\pi \left(\sum_{i=k}^{n} a_{S_i,k}\, \nu_i \right) && \text{(nach Umordnung)} \\
&= \sum_{k=1}^{n} x_k^\pi \left(\sum_{S \ni k} a_{S,k}\, y_S \right) && \text{(per Def. } y) \\
&= \sum_{k=1}^{n} x_k^\pi w_k. && \text{(per Zulässigkeit von } y)
\end{aligned}$$

Damit sind x^π und y ein optimales primal-duales Paar für (P) bzw. (D), und die Behauptung folgt aus dem schwachen Dualitätssatz. ∎

Im Falle von *ties* können je nach Auswahlregel zwar unterschiedliche Lösungen x^π bzw. y erzeugt werden,deren Zielfunktionswert ist jedoch stets derselbe. Diese Tatsache kann man auch direkt beweisen, denn es ist $\sum_{i=1}^{n} \nu_i f(S_i)$ unabhängig von der Auswahlregel. Wir verzichten auf den Beweis und verweisen auf die Diskussion dieser Tatsache in [BN-M 93]. Für das primale Problem ist die geometrische Interpretation, daß der Zielfunktionsvektor w in einem solchen Fall senkrecht auf einer Seitenfläche von $\mathcal{B}_{\geq}(A, f)$ steht, welche die entsprechenden Lösungen x^π enthält.

Wir geben noch einen für unsere Zwecke wichtigen Spezialfall an, der direkt aus Lemma 2.9 und Satz 2.6 folgt:

[18] Vergleiche zu den Definitionen (2.12) und (2.6).

Korollar 2.6 *Ein lineares Optimierungsproblem der Art*

$$\min\{w^t x \mid \sum_{i \in S} a_i x_i \ \geq f(S) \quad \forall\, S \subseteq I\}$$

mit $w \geq 0$*, einer supermodularen Funktion* $f : 2^I \longrightarrow \mathbb{R}$ *und echt positiven Koeffizienten* a_i *kann in* $\mathcal{O}(n \log n)$ *gelöst werden: Eine Optimallösung* x^{opt} *ist nach (2.8) durch*

$$x^{opt}_{\pi_k} = \frac{f(\{\pi_1, \ldots, \pi_k\}) - f(\{\pi_1, \ldots, \pi_{k-1}\})}{a_{\pi_k}} \qquad \textit{für alle} \quad k = 1, \ldots, n,$$

gegeben, wenn π *derart gewählt ist, daß*

$$\frac{w_{\pi_1}}{a_{\pi_1}} \geq \frac{w_{\pi_2}}{a_{\pi_2}} \geq \cdots \geq \frac{w_{\pi_n}}{a_{\pi_n}}$$

gilt.

Dabei sei wieder $\{\pi_1, \ldots, \pi_0\} := \emptyset$. Diese Tatsache entspricht der Optimalität des *greedy*-Algorithmus von Edmonds.

Wieso nennen wir Algorithmus G *greedy* ?

Die primale Lösung: Die Definition des Vektors x^π zu einer Permutation π verfährt auf folgende (gierige) Weise:
Sei o. B. d. A. $\pi = (1, \ldots, n)$. Wir wissen, daß *jeder* zulässige Punkt x von $\mathcal{B}_{\geq}(A, f)$ die Ungleichung

$$a_{\{1\},1}\, x_1 \geq f(\{1\})$$

erfüllt. Wählen wir also x^π_1 (gierig) minimal:

$$x^\pi_1 := \frac{f(\{1\})}{a_{\{1\},1}}.$$

Seien nun $x^\pi_1, x^\pi_2, \ldots, x^\pi_k$ gewählt. Wir wissen wieder, daß jeder zulässige Punkt x die Ungleichung

$$\sum_{i=1}^{k+1} a_{\{1,\ldots,k+1\},i}\, x_i \geq f(\{1, \ldots, k+1\}) \tag{2.14}$$

erfüllt. Unter der Bedingung der bereits gewählten Komponenten wählen wir wieder x^π_{k+1} minimal, so daß (2.14) mit Gleichheit erfüllt wird. Diese Vorgehensweise entspricht gerade der Berechnung von x^π anhand des Gleichungssystems (2.6). Die Tatsache, daß uns ein erweitertes Contra-Polymatroid vorliegt, sichert dabei, daß diese (gierig bestimmten) Punkte auch tatsächlich zulässige Punkte sind.

Dieses Verfahren entspricht für $a_{S,i} = 1$ für alle $i \in S$ und $S \subseteq I$ genau dem bekannten *greedy*-Verfahren für Polymatroide bzw. super- oder submodulare Polyeder[19]: Eine Optimallösung erhält man, indem man sukzessive (unter Beachtung der Zulässigkeit) diejenigen Komponenten x_i mit dem größten Gewicht w_i in der Zielfunktion minimal wählt.

Die duale Lösung: Auch die Definition der Duallösung y ist in der Tat *greedy*. Wir wollen $\sum_{S\subseteq I} y_S f(S)$ maximieren und beginnen mit der Auswahl von y_I (dies ist immerhin für nichtfallende Funktionen f plausibel). Wir müssen die Bedingungen

$$\sum_{S\ni i} a_{S,i}\, y_S = w_i \quad \text{für alle} \quad i \in I$$

einhalten. Wir wählen y_I (gierig) so groß wie möglich, das bedeutet:

$$y_I := \min\{\frac{w_i}{a_{I,i}} \mid i \in I\},$$

und werde dieses Minimum o. B. d. A. für $i = n$ angenommen. Wieso das *Minimum*, wo wir doch y_I so groß wie möglich wählen wollten?
Würden wir etwa $y_I = \frac{w_k}{a_{I,k}}$ wählen mit $\frac{w_k}{a_{I,k}} > \frac{w_n}{a_{I,n}}$, so folgte $a_{I,n}\, y_I > w_n$. Wegen $a_{S,n} > 0$ für alle $S \subseteq I$ mit $n \in S$ erhielte man damit $\sum_{S\ni n} a_{S,n}\, y_S > w_n$, womit y nicht zulässig wäre.
Gelte also $a_{I,n}\, y_I = w_n$. Wir müssen dann $y_S = 0$ setzen für alle weiteren $S \subset I$ mit $n \in S$: Denn wieder wegen $a_{S,n} > 0$ für alle $S \subseteq I$ mit $n \in S$ läßt sich nur so die Zulässigkeitsbedingung $\sum_{S\ni n} a_{S,n}\, y_S = w_n$ einhalten.
Als nächstes fahren wir mit der größten Menge $S \subset I$ mit $n \notin S$, also $S \setminus \{n\}$ fort. Wir wählen wieder $y_{S\setminus\{n\}}$ so groß wie möglich und müssen nun $a_{I,n}\, y_I = w_n$ berücksichtigen. Die maximale Wahl ist:

$$y_{S\setminus\{n\}} = \min\{\frac{w_i - a_{I,n}\, y_I}{a_{S\setminus\{n\},i}} \mid i \in S \setminus \{n\}\}$$

mit denselben Überlegungen wie oben. Wird dieses Minimum etwa für $i = n-1$ angenommen, müssen wir wieder $y_S = 0$ setzen für alle weiteren $S \subset S \setminus \{n\}$ mit $n-1 \in S$. Auf diese Weise fahren wir bis zum Ende fort.
Diese (gierige) Vorgehensweise entspricht genau der Definition der ν_k in Algorithmus G zusammen mit der Definition der y_S in (2.12).

2.5 Anwendungen im Stochastischen Scheduling

In diesem Abschnitt wollen wir den Nutzen der vorgestellten Algorithmen für die *exakte* Lösung stochastischer Schedulingprobleme herausstellen. Prinzipiell hatten wir die Vorgehensweise bereits in der Einleitung zu diesem Kapitel vorgestellt.

[19] Vergleiche [Edm 70], [Lov 82] oder [Fuj 91].

Sei also ein stochastisches Schedulingproblem $(\Theta, \mathcal{N}, P, \kappa)$ gegeben. Sei ferner eine beliebige Klasse σ von Strategien für dieses Problem gegeben. Wir setzen eine *lineare* Kostenfunktion $\kappa : \mathbb{R}_+^n \longrightarrow \mathbb{R}_+$ voraus, und wollen eine Strategie minimaler erwarteter Kosten $E(\kappa^\Pi)$ angeben.

Unter jeder Strategie Π erhalten wir bekanntlich einen Vektor $E(\mathcal{C}^\Pi) \in \mathbb{R}_+^n$ für die unter einer Strategie Π zu erwartenden Fertigstellungszeiten.

Mit der Linearität der Kostenfunktion läßt sich das Problem unter Beachtung von Lemma 1.1 in der Form

$$\min\{\kappa(E(\mathcal{C}^\Pi)) \mid \Pi \in \sigma\} \tag{2.15}$$

schreiben. Unser Ziel ist es, eine äußere Beschreibung für den zugehörigen Lösungsraum

$$X := \operatorname{conv}\{E(\mathcal{C}^\Pi) \mid \Pi \in \sigma\}$$

zu gewinnen. Damit ist das *lineare Programm*

$$\min\ \{\kappa(x) \quad | \quad x \in X\} \tag{2.16}$$

zu lösen. Können wir den Lösungsraum $\{E(\mathcal{C}^\Pi) \mid \Pi \in \sigma\}$ als *Polyeder* identifizieren, bestimmen wir eine Optimallösung x^{opt} für (2.16) mit **LP**-Methoden. Für den Fall, daß eine solche polyedrische Beschreibung nicht gelingt, sei auf das dritte Kapitel verwiesen.

Falls wir nun noch anhand einer Optimallösung x^{opt} von (2.16) auf eine Strategie Π mit $E(\mathcal{C}^\Pi) = x^{opt}$ schließen können, ist das Schedulingproblem (2.15) gelöst. Denn damit ist $E(\kappa^\Pi) = \kappa(\mathcal{C}^\Pi)$ minimal.

Wir geben in diesem Abschnitt Bedingungen an, die garantieren, daß die konvexe Hülle der unter allen Strategien zu erwartenden Fertigstellungszeiten das Basispolytop eines erweiterten Contra-Polymatroids (oder supermodularen LTP) ist. Mit Algorithmus G (oder G') können wir dann eine optimale Ecke x^π für das lineare Programm (2.16) berechnen.

Schließlich müssen wir zu diesem Vektor noch die zugehörige Strategie Π mit $E(\mathcal{C}^\Pi) = x^\pi$ gewinnen. Um dazu in der Lage zu sein, werden wir zusätzlich fordern, daß die Ecken des Lösungsraums X gerade speziellen Strategien, nämlich den statischen Prioritätsstrategien entsprechen.

Das Bündel dieser Forderungen wird in der Literatur *generalized conservation laws* oder kurz GCL genannt[20]. Wir wollen diese Bedingungen hier die *Erhaltungssätze GCL* nennen.

[20] Vergleiche [BN-M 93] oder auch [GG 95]. Es sei darauf hingewiesen, daß in diesen Artikeln die *generalized conservation laws* im allgemeinen Rahmen von *performance measures* für Modelle der stochastischen dynamischen Programmierung definiert werden, und wir in dieser Arbeit nur die Übertragung auf das im ersten Kapitel vorgestellte Modell für stochastisches Scheduling untersuchen.

2.5.1 „Erhaltungssätze“ für stochastisches Scheduling

Es seien ein stochastisches Schedulingproblem mit einer *linearen* Kostenfunktion κ, und eine beliebige Klasse σ von Strategien gegeben, die die statischen Prioritätsstrategien enthält.

Definition 2.13 (Erhaltungssätze GCL) *Ein Schedulingproblem* $(\Theta, \mathcal{N}, P, \kappa)$ *erfüllt die Erhaltungssätze GCL, wenn es eine normierte Abbildung* $f : 2^I \longrightarrow \mathbb{R}$ *und Koeffizienten* $a_{S,i} > 0$ *für alle* $S \subseteq I$ *und* $i \in S$ *gibt, so daß folgendes gilt:*

$$\sum_{i \in S} a_{S,i}\, E(\mathcal{C}_i^{\Pi}) \;\geq\; f(S) \qquad \text{für alle } S \subseteq I \text{ und alle } \Pi \in \sigma, \tag{2.17}$$

$$\sum_{i \in I} a_{I,i}\, E(\mathcal{C}_i^{\Pi}) \;=\; f(I) \qquad \text{für alle } \Pi \in \sigma, \qquad \text{sowie} \tag{2.18}$$

$$\sum_{i \in S} a_{S,i}\, E(\mathcal{C}_i^{L_\pi}) \;=\; f(S) \qquad \text{für alle } S \subseteq I \text{ und alle } \pi \text{ mit} \{\pi_1, \ldots, \pi_{|S|}\} = S. \tag{2.19}$$

Was wird mit den Erhaltungssätzen gefordert?
Wir können die Erhaltungsätze folgendermaßen interpretieren: (2.18) besagt, daß es Koeffizienten $a_{I,i} > 0$ für alle $i \in I$ gibt, so daß *alle* Strategien denselben Wert liefern, wenn wir als Zielfunktion den Vektor $a_I = (a_{I,1}, \ldots, a_{I,n})^t$ ansetzen.

Weiter besagen (2.17) und (2.19), daß es darüber hinaus für *jede* Teilmenge S der Jobs Koeffizienten $a_{S,i} > 0$ für alle $i \in S$ gibt, so daß *jede beliebige* Prioritätsstrategie L_π, die die Jobs aus S denen aus $I \setminus S$ vorzieht, bezüglich dieser Koeffizienten als Zielfunktion optimal in der Klasse aller Strategien ist.

Es gilt nun der folgende Satz:

Satz 2.12 *Sei ein stochastischen Schedulingproblem* $(\Theta, \mathcal{N}, P, \kappa)$ *mit linearer Kostenfunktion gegeben. Sind dann die Erhaltungssätze GCL mit einer passenden Abbildung* $f : 2^I \longrightarrow \mathbb{R}$ *und passenden Koeffizienten* $a_{S,i}$ *erfüllt, so sei* $A = (a_{S,i})_{\substack{S \subseteq I \\ i \in I}}$ *die zugehörige Matrix dieser Koeffizienten. Dann gilt:*

(1) $\operatorname{conv}\{E(\mathcal{C}^{\Pi}) \mid \Pi \in \sigma\} = \mathcal{B}_{\geq}(A, f)$, *und* $\mathcal{B}_{\geq}(A, f)$ *ist das Basispolytop des zu* A *und* f *gehörigen Contra-Polymatroids.*

(2) Die Ecken von $\mathcal{B}_{\geq}(A, f)$ *sind genau die Vektoren* $E(\mathcal{C}^{L_\pi})$ *der unter den statischen Prioritätsstrategien* L_π *zu erwartenden Fertigstellungszeiten.*

Beweis. Wir erinnern an die Bezeichnung $X = \operatorname{conv}\{E(\mathcal{C}^\Pi) \mid \Pi \in \sigma\}$. Zur ersten Behauptung: Aus (2.17) zusammen mit (2.18) folgt

$$X \subseteq \mathcal{B}_{\geq}(A, f), \tag{2.20}$$

wobei $\mathcal{B}_{\geq}(A,f) = \{x \in \mathbb{R}^n \mid \sum_{i \in S} a_{S,i}\, x_i \geq f(S) \quad \forall S \subseteq I \text{ und } \sum_{i \in I} a_{I,i}\, x_i = f(I)\}$ ist. Es sei angemerkt, daß der Beweis, daß $\mathcal{B}_{\geq}(A,f)$ tatsächlich das Basispolytop eines Contra-Polymatroids ist, erst erbracht werden muß. Sei nun π eine beliebige Permutation auf I. Es folgt aus (2.19) dann mit $S_k := \{\pi_1, \ldots, \pi_k\}$

$$\sum_{i \in S_k} a_{S_k,i}\, \underbrace{E(\mathcal{C}_i^{L_\pi})}_{=:x_i} = f(S_k) \qquad \text{für alle } k = 1, \ldots, n.$$

Substituieren wir $E(\mathcal{C}_i^{L_\pi})$ durch x_i und schreiben die Gleichungen in der Form

$$\sum_{i=1}^{k} a_{\{\pi_1, \ldots \pi_k\}, \pi_i}\, x_{\pi_i} = f(\{\pi_1, \ldots, \pi_k\}) \qquad \text{für alle } k = 1, \ldots, n,$$

so sehen wir, daß x genau der zu A und f gehörige Vektor x^π ist[21]. Damit gilt also $x^\pi \in X$ für alle Permutationen π auf I. Wegen (2.20) gilt damit auch $x^\pi \in \mathcal{B}_{\geq}(A,f)$. Daher ist $\mathcal{B}_{\geq}(A,f)$ tatsächlich das Basispolytop des Contra-Polymatroids $\mathcal{P}_{\geq}(A,f)$.
Da alle Ecken x^π von $\mathcal{B}_{\geq}(A,f)$ auch in X liegen, und X per Definition konvex ist, folgt sofort $\mathcal{B}_{\geq}(A,f) \subseteq X$ und die erste Behauptung ist bewiesen.

Die zweite Behauptung folgt nun direkt aus dem bisher gezeigten, denn es gilt $E(\mathcal{C}_i^{L_\pi}) = x^\pi$ für jede beliebige Permutation π auf I. ■

Eine wichtige Folgerung ergibt sich aus der Optimalität des im vorigen Abschnitt vorgestellten Algorithmus für Basispolytope von erweiterten Contra-Polymatroiden:

Satz 2.13 *Es sei ein stochastisches Schedulingproblem $(\Theta, \mathcal{N}, P, \kappa)$ mit linearer Kostenfunktion $\kappa(x) = w^t x$ und einer Klasse σ von Strategien gegeben. Sind dann die Erhaltungssätze GCL mit passender Koeffizientenmatrix A und Abbildung f erfüllt, so wird der Optimalwert von einer statischen Prioritätsstrategie L_π erreicht, und wir können L_π und $\rho(\kappa)$ in $\mathcal{O}(n^3)$ bestimmen.*

Beweis. Wir wissen bereits nach Satz 2.12 (1), daß für den Lösungsraum des Schedulingproblems

$$\min\{\kappa(E(\mathcal{C}^\Pi)) \mid \Pi \in \sigma\} \tag{2.21}$$

$\{E(\mathcal{C}^\Pi) \mid \Pi \in \sigma\} = \mathcal{B}(A,f)$ gilt. Wir können also (2.21) auch in der Form

$$\min\{\kappa(x) \mid x \in \mathcal{B}_{\geq}(A,f)\} \tag{2.22}$$

[21] Vergleiche (2.6).

schreiben. Mit A und w wenden wir Algorithmus G an. Wir erhalten eine Permutation π auf I und wissen, daß x^π eine optimale Lösung für (2.22) ist. Nach Satz 2.12 (2) ist x^π genau der Vektor $E(\mathcal{C}^{L_\pi})$ der erwarteten Fertigstellungszeiten der Prioritätsstrategie L_π. Demnach ist $\kappa(E(\mathcal{C}^{L_\pi})) = \kappa(x^\pi)$ sowohl für (2.22) als auch für (2.21) minimal und L_π eine optimale Strategie für das Schedulingproblem. Ferner ist $\kappa(x^\pi) = \rho(\kappa)$. ■

Aus Satz 2.12 und Korollar 2.6 folgt:

Korollar 2.7 *Sind die Erhaltungssätze GCL für ein stochastisches Schedulingproblem derart erfüllt, daß die Koeffizienten $a_{S,i}$ nicht von S abhängen für alle $i \in I$, so können die optimale statische Prioritätsstrategie L_π und $\rho(\kappa)$ in $\mathcal{O}(n \log n)$ bestimmt werden.*

Im folgenden Abschnitt wollen wir ein Beispiel für die polyedrische Charakterisierung des Lösungsraums stochastischer Schedulingprobleme geben und uns zu diesem Zweck mit einem „klassischen" Resultat beschäftigen.

2.5.2 Die Optimalität der Ratio Rule im stochastischen Fall

Wir betrachten das Problem

$$((I, \emptyset), \{S \subseteq I \mid |S| = 2\}, P, \sum_{i \in I} w_i \, C_i)$$

mit $P = \bigotimes_{i \in I} P_i$, also beliebigen, stochastisch unabhängigen Dauerverteilungen P_i.

Wir hatten bereits in Satz 1.5 erwähnt, daß die *ratio rule* auch im stochastischen Scheduling auf einer Maschine eine optimale Strategie ist. Dieses Ergebnis wollen wir nun in den polyedrischen Zusammenhang supermodularer LTP einorden und nachträglich den Beweis für Satz 1.5 liefern. Es gilt:

Satz 2.14 *Das stochastische Schedulingproblem $((I, \emptyset), \{S \subseteq I \mid |S| = 2\}, P, \sum_{i \in I} w_i \, C_i)$ mit stochastisch unabhängigen Dauerverteilungen P_i erfüllt die Erhaltungssätze GCL, und die Menge* $\mathrm{conv}\{E(\mathcal{C}^\Pi) \mid \Pi \,\text{beliebige Strategie}\ \}$ *ist das Basispolytop eines supermodularen LTP.*
Die optimale Strategie ist die ratio rule – also diejenige statische Prioritätsstrategie, die die Vorgänge in einer Reihenfolge absteigender Quotienten $\frac{w_i}{E(P_i)}$ startet.

Erstaunlich an dieser Tatsache ist, daß die *ratio rule* bei *beliebigen* Dauerverteilungen der Vorgänge noch optimal ist, denn die verwendeten Informationen sind lediglich die erwarteten Dauern $E(P_i)$ und die Kosten w_i für jeden Vorgang i.
Intuitiv hätte man schließlich erwarten können, daß bei gleichen Kosten ein Vorgang mit

geringer Varianz einem weiteren Vorgang mit hoher Varianz und geringfügig kürzerer erwarteter Dauer vorgezogen werden sollte. Es sei allerdings daran erinnert, daß wir lediglich eine Optimierung dem Erwartungswert nach anstreben.

Beweis zu Satz 2.14. Wir definieren

$$a_i := E(P_i) \qquad \text{für alle } i \in I,$$
$$f(S) := \frac{1}{2}\Big(\Big(\sum_{i\in S} E(P_i)\Big)^2 + \sum_{i\in S}(E(P_i))^2\Big) \qquad \text{für alle } S \subseteq I,$$

und zeigen, daß

$$\operatorname{conv}\{E(\mathcal{C}^\Pi) \mid \Pi \text{ beliebige Strategie }\} = \mathcal{B}_{\geq}(a,f) \tag{2.23}$$

für das Basispolytop $\mathcal{B}_{\geq}(a,f)$ des supermodularen LTP $\mathcal{P}_{\geq}(a,f)$ gilt. Dazu weisen wir die Gültigkeit der Erhaltungssätze GCL (2.17) - (2.19) nach.
Wir erinnern an den Zusammenhang

$$P \xrightarrow{\Pi} \mathcal{S}^\Pi \xrightarrow{\mathrm{id}+P} \mathcal{C}^\Pi.$$

Es gilt ferner $E(\mathcal{C}_i^\Pi) = E(\mathcal{S}_i^\Pi) + E(P_i)$ für alle $i \in I$. Wir beginnen mit (2.17):
Sei $p = (p_1, \ldots, p_n)$ eine beliebige Realisierung der Vorgangsdauern und Π eine beliebige Strategie. Bezeichne dann s_i die Startzeit und C_i die Fertigstellungszeit von Vorgang i unter Strategie Π bei Realisierung p (es ist $s_i = (\Pi(p))_i$ und $C_i = (\Pi(p)_i + p_i)$. Weiter bezeichne $S_i(\Pi, p) \subseteq I$ die Menge derjenigen Vorgänge, die unter Π und p *vor* Vorgang $i \in I$ bearbeitet werden. Aufgrund der Eigenschaft (STO 3) einer Strategie ist $S_i(\Pi, p)$ wohldefiniert.

Es gilt nun für beliebiges $S \subseteq I$:

$$\begin{aligned}
\sum_{i\in S} p_i\, s_i &= \sum_{i\in S} p_i\Big(\sum_{j\in S_i(\Pi,p)} p_j\Big) \\
&\geq \sum_{i\in S} p_i\Big(\sum_{j\in S\cap S_i(\Pi,p)} p_j\Big) \qquad \text{(da alle } p_i, p_j \geq 0) && (2.24)\\
&= \sum_{i,j\in S,\, i<j} p_i\, p_j. && (2.25)
\end{aligned}$$

Dabei gilt die erste Gleichung aufgrund der Tatsache, daß eine Strategie niemals totale Leerzeiten läßt, und daher die Vorgänge im Einmaschinenfall stets konsekutiv ohne Leerzeit gestartet werden. Weiter gilt die letzte Gleichung, da in (2.24) jedes Produkt $p_i\, p_j$ für $i \neq j$ *genau einmal* auftaucht; schließlich wird *entweder* Vorgang i vor j *oder* umgekehrt Vorgang j vor i geplant.

Für $S = I$ gilt die obige Ungleichungskette offenbar bereits mit Gleichheit, und wir erhalten

$$\sum_{i\in I} p_i\, s_i = \sum_{i,j\in I,\, i<j} p_i\, p_j. \tag{2.26}$$

Dasselbe gilt für beliebiges $S \subset I$ und eine Prioritätsstrategie L_π mit $\{\pi_1, \ldots, \pi_{|S|}\} = S$, denn für ein solches L_π gilt offenbar $S_i(L_\pi, p) \subseteq S$ für alle $i \in S$ (schließlich wird kein Vorgang aus $I \setminus S$ einem Vorgang aus S vorgezogen). Es gilt daher für eine Prioritätsstrategie L_π

$$\sum_{i \in S} p_i \, s_i = \sum_{i,j \in S,\, i<j} p_i \, p_j, \tag{2.27}$$

falls $\{\pi_1, \ldots, \pi_{|S|}\} = S$ erfüllt ist.

Entscheidend ist nun, daß die Zufallsvariablen $\mathcal{S}_i^\Pi$ und P_i stochastisch unabhängig (und daher insbesondere unkorreliert) sind für alle $i \in I$:
Diese Tatsache macht man sich folgendermaßen klar. Aufgrund der Eigenschaft (STO 3) ist die Startzeit für Vorgang i abhängig von den Dauern derjenigen Vorgänge j, die *vor* i gestartet werden; letztlich ist $\mathcal{S}_i^\Pi$ also eine Funktion in den Zufallsvariablen P_j, $j \neq i$. Die stochastische Unabhängigkeit der P_i, $i \in I$ hat demnach auch die Unabhängigkeit von $\mathcal{S}_i^\Pi$ und P_i zur Folge. Aufgrund der Unkorreliertheit von $\mathcal{S}_i^\Pi$ und P_i gilt daher:

$$E(P_i \, \mathcal{S}_i^\Pi) = E(P_i) E(\mathcal{S}_i^\Pi) \qquad \text{für alle } i \in I \ \text{ und alle } \Pi.$$

Aus (2.25) folgt damit unter Ausnutzung der Linearität des Erwartungswertes:

$$\begin{aligned}
\sum_{i \in S} E(P_i) E(\mathcal{S}_i^\Pi) \;&\geq\; \sum_{i,j \in S,\, i<j} E(P_i \, P_j) \\
&=\; \sum_{i,j \in S,\, i<j} E(P_i) E(P_j) \\
&=\; \frac{1}{2} \Big(\Big(\sum_{i \in S} E(P_i) \Big)^2 - \sum_{i \in S} (E(P_i))^2 \Big),
\end{aligned}$$

wobei der zweite Schritt aus der stochastischen Unabhängigkeit (bzw. der paarweisen Unkorreliertheit) der Bearbeitungsdauern folgt. Zusammen mit $E(\mathcal{C}_i^\Pi) = E(\mathcal{S}_i^\Pi) + E(P_i)$ folgt letztlich

$$\sum_{i \in S} E(P_i) E(\mathcal{C}_i^\Pi) \;\geq\; \frac{1}{2} \Big(\Big(\sum_{i \in S} E(P_i) \Big)^2 + \sum_{i \in S} (E(P_i))^2 \Big) \;=\; f(S).$$

Damit ist

$$\sum_{i \in S} a_i \, E(\mathcal{C}_i^\Pi) \;\geq\; f(S) \qquad \text{für alle } S \subseteq I \text{ und } \Pi,$$

also ist (2.17) erfüllt.

Die Gültigkeit von (2.18) und (2.19) folgt nun aus (2.26) und (2.27), denn mit denselben Überlegungen wie oben ergibt sich sowohl

$$\sum_{i \in I} E(P_i) E(\mathcal{C}_i^\Pi) \;=\; \frac{1}{2} \Big(\Big(\sum_{i \in I} E(P_i) \Big)^2 + \sum_{i \in I} (E(P_i))^2 \Big) \;=\; f(I)$$

für alle Strategien Π als auch

$$\sum_{i \in S} E(P_i)E(\mathcal{C}_i^{L_\pi}) = \frac{1}{2}\Big(\Big(\sum_{i \in S} E(P_i)\Big)^2 + \sum_{i \in S}(E(P_i))^2\Big) = f(S)$$

für alle $S \subseteq I$ und alle Prioritätsstrategien L_π mit $\{\pi_1, \ldots, \pi_{|S|}\} = S$.

Damit sind die Erhaltungsätze GCL vollständig erfüllt, und nach Satz 2.12 (1) ist die konvexe Hülle der Inzidenzvektoren der erwarteten Abschlußtermine tatsächlich das Basispolytop eines supermodularen LTP.

Weiter folgt aus Satz 2.12 (2) zusammen mit Korollar 2.6 und der Definition der Koeffizienten $a_i = E(P_i)$ die Optimalität jeder Prioritätstrategie L_π, die gemäß einer Reihenfolge

$$\frac{w_{\pi_1}}{E(P_{\pi_1})} \geq \frac{w_{\pi_2}}{E(P_{\pi_2})} \geq \cdots \geq \frac{w_{\pi_n}}{E(P_{\pi_n})}$$

verfährt. ■

Damit können wir eine optimale Strategie für ein stochastisches Schedulingproblem auf einer Maschine – zumindest im Fall stochastisch unabhängiger Vorgangsdauern und für $\prec = \emptyset$ – in $\mathcal{O}(n \log n)$ angeben, denn wir müssen lediglich die Quotienten $\frac{w_i}{E(P_i)}$ sortieren.

Im dritten Kapitel dieser Arbeit untersuchen wir den Einmaschinenfall unter Berücksichtigung von *release dates* und einer Präzedenzrelation.

2.5.3 Mehrmaschinenmodelle und weitere Anwendungen

Die in diesem Abschnitt vorgestellte Anwendung polyedrischer Methoden ordnet die *ratio rule* von Smith in einen allgemeinen Ansatz für erweiterte Contra-Polymatroide, oder genauer, supermodulare LTP ein. Von diesem Standpunkt aus weist das hier untersuchte Einmaschinenproblem Parallelen zu *branching bandit problems* und *multi-armed bandit problems* auf, mit deren Untersuchung sich D. Bertsimas und J. Niño-Mora beschäftigen[22]. Es wird dort gezeigt, daß die entsprechenden Lösungsräume ebenfalls Basispolytope erweiterter (Contra-)Polymatroide sind.

Für eine eingehende Untersuchung dieser Modelle und im Hinblick auf die Definition von *generalized conservation laws* für ein beliebiges *performance measure* bei stochastischen Schedulingproblemen der stochastischen dynamischen Programmierung sei auf [BN-M 93] sowie [GG 95] verwiesen.

Eine Motivation dieser Arbeit war die Überlegung, auch den Lösungsraum von Mehrmaschinenmodellen im stochastischen Scheduling auf die vorgestellte Weise polyedrisch zu charakterisieren, um so auf eine Aussage über das *deliberate idleness*-Problem zu stoßen.

[22] Vergleiche [BN-M 93], [GG 95] und zu *branching bandits* auch [Wei 88].

Es muß jedoch festgehalten werden, daß ein Schedulingproblem, welches die Erhaltungssätze GCL erfüllt, stets durch eine *statische Prioritätsstrategie* gelöst wird. Beispiel 1.2 zeigt jedoch, daß bereits für exponentielle Schedulingprobleme mit $\prec = \emptyset$ die Klasse der statischen Prioritätsstrategien nicht hinreichend ist (in dem Sinne, daß der Optimalwert ausschließlich durch eine *echt* dynamische Strategie erreicht wird).

Damit wird klar, daß der Lösungsraum bereits für exponentielle Schedulingprobleme auf mehreren Maschinen und ohne Präzedenzrelation im allgemeinen *keine* polymatroidiale Struktur aufweisen kann.

Im dritten Kapitel dieser Arbeit werden wir dennoch einen Bogen zwischen Mehrmaschinenmodellen und den vorgestellten polyedrischen Methoden schlagen, indem wir Approximationsheuristiken angeben, deren Basis die Lösung linearer Programme über supermodularen LTP ist. Korollar 2.6 wird sich dabei als wichtiges Hilfsmittel zur Lösung der entsprechenden linearen Programme herausstellen.

Kapitel 3

Approximationsheuristiken für Stochastisches Scheduling

Insbesondere in jüngerer Vergangenheit wurde Approximationsheuristiken zur Behandlung $\mathcal{NP}$-schwerer Schedulingprobleme viel Aufmerksamkeit gewidmet. Eine Auswahl von Veröffentlichungen, die sich mit der Minimierung der gewichteten Summe der Fertigstellungszeiten bei *deterministischen* Schedulingproblemen beschäftigen, ist [Sch 96], [CPS+ 96] und [HSSW 96].

In diesem Kapitel geben wir Approximationsheuristiken für *stochastische* Schedulingprobleme zur Minimierung der gewichteten Summe der Fertigstellungszeiten an. Wir übernehmen dabei einen Ansatz von A. S. Schulz [Sch 96] für deterministisches Scheduling, und zeigen, daß sich einige Resultate auf den allgemeineren, stochastischen Fall übertragen lassen.

Für Schedulingprobleme auf einer Maschine ist dabei lediglich die stochastische Unabhängigkeit der einzelnen Verteilungen für die Vorgangsdauern notwendig. Im Falle mehrerer Maschinen konnte im Rahmen des gewählten Ansatzes auf eine zusätzliche Bedingung an die Varianz der einzelnen Verteilungen nicht verzichtet werden.

Der Begriff Approximationsheuristik beinhaltet im stochastischen Fall die Angabe einer Strategie Π^{appr} – möglichst in polynomialer Zeit – derart, daß die erwarteten Kosten $E(\kappa^{\Pi^{appr}})$ beweisbar nah am Optimalwert $\rho(\kappa)$ des Schedulingproblems liegen. Wir sprechen von einer δ-approximativen Heuristik, wenn $E(\kappa^{\Pi^{appr}}) \leq \delta\, \rho(\kappa)$ für ein $\delta \geq 1$ gilt.

Eine Rechtfertigung für die Beschäftigung mit approximativen Methoden im stochastischen Scheduling liegt natürlich in der Unkenntnis optimaler Strategien begründet. Wie wir beispielsweise in Abschnitt 1.5.5 sehen konnten, ist sogar unter sehr restriktiven Modellannahmen bislang noch nicht bekannt, ob optimale Prioritätsstrategien existieren.

Es stellt sich insbesondere die Frage, wie gut etwa statische Prioritätsstrategien im Vergleich zum Optimalwert sind. Hierzu sind bislang keine Ergebnisse in der Literatur angegeben, und in diesem Zusammenhang sind die folgenden Resultate von besonderem Interesse.

Darüberhinaus beinhaltet das Modell des stochastischen Scheduling stets auch den deterministischen Fall – sofern keine speziellen Dauerverteilungen vorausgesetzt werden. Aus diesem Grund ist ein stochastisches Problem ebenfalls $\mathcal{NP}$-schwer, wenn die entsprechende deterministische Version es ist. Diese Tatsache liefert unter der Annahme $\mathcal{P} \neq \mathcal{NP}$ eine weitere Rechtfertigung für die Untersuchung von Approximationsheuristiken für stochastische Schedulingprobleme.

An geeigneter Stelle werden wir die Komplexitätsresultate der entsprechenden deterministischen Probleme zitieren.

3.1 Gütegarantien für Heuristiken auf LP-Basis – Genereller Ansatz

Wir untersuchen stochastische Schedulingprobleme $(\Theta, \mathcal{N}, P, \kappa)$ mit $I = \{1, \ldots, n\}$, der Kostenfunktion $\kappa = \sum_{i \in I} w_i C_i$ und $P = \bigotimes_{i \in I} P_i$, also stochastisch unabhängigen Vorgangsdauern P_i.

Das Schedulingproblem ist durch

$$\min\{E(\kappa^{\Pi}) \mid \Pi \text{ beliebige Strategie } \} \tag{3.1}$$

gegeben. Aufgrund der Linearität von κ können wir statt $E(\kappa^{\Pi})$ auch $\kappa(E(\mathcal{C}^{\Pi}))$ schreiben[1], vergleiche hierzu Lemma 1.1. Diese Tatsache macht das Problem zunächst prinzipiell zugänglich für Methoden der Linearen Programmierung. Der Lösungsraum, den wir untersuchen ist dann die konvexe Hülle aller Vektoren erwarteter Fertigstellungszeiten, also

$$X := \operatorname{conv}\{E(\mathcal{C}^{\Pi}) \mid \Pi \text{ beliebige Strategie } \} \subseteq \mathbb{R}^n_+ .$$

Das Problem ist nun, daß eine algebraische Beschreibung für diesen Lösungsraum im allgemeinen nicht vorliegt. Wir geben stattdessen gültige, lineare Ungleichungen für X an und erhalten auf diese Weise eine **LP**-Relaxierung für (3.1). D. h. wir geben ein *Polyeder* $\mathcal{P} \subseteq \mathbb{R}^n_+$ mit

$$X \subseteq \mathcal{P}$$

an, und lösen das „relaxierte“ lineare Programm

$$\min\{\kappa(x) \mid x \in \mathcal{P}\}. \tag{3.2}$$

[1] Zur Erinnerung: $E(\mathcal{C}^{\Pi})$ ist der Vektor der unter Π erwarteten Fertigstellungszeiten.

Bezeichnet dann x^{LP} eine Optimallösung für (3.2), so gilt im allgemeinen *nicht* $x^{LP} \in X$, und daher ist x^{LP} *keine* Lösung für das Schedulingproblem.

Wir können die Optimallösung x^{LP} von (3.2) dennoch zur Erzeugung einer Heuristik für das Schedulingproblem (3.1) nutzen. Dabei gehen wir auf folgende, intuitive Weise vor: Unter der Annahme

$$x^{LP} \approx E(\mathcal{C}^{\Pi^{opt}})$$

liefert die Reihenfolge aufsteigender x_i^{LP} zumindest eine plausible Planungsreihenfolge für das Schedulingproblem.

Sei daher π eine Permutation auf I mit

$$x_{\pi_1}^{LP} \leq x_{\pi_2}^{LP} \leq \cdots \leq x_{\pi_n}^{LP}.$$

Dann definieren wir durch die Prioritätsliste $\pi_1 < \cdots < \pi_n$ eine statische Prioritätsstrategie L_π für das Schedulingproblem (3.1).

Ist Π^{opt} eine optimale Strategie für (3.1), so gilt

$$\kappa(x^{LP}) \quad \leq \quad \underbrace{\kappa(E(\mathcal{C}^{\Pi^{opt}}))}_{=\rho(\kappa)} \quad \leq \quad \underbrace{\kappa(E(\mathcal{C}^{L_\pi}))}_{=E(\kappa^{L_\pi})}, \tag{3.3}$$

denn es ist $X \subseteq \mathcal{P}$ und x^{LP} ist Optimallösung für (3.2).

Unser Ziel ist nun eine Abschätzung in umgekehrter Richtung, also

$$E(\kappa^{L_\pi}) \quad \leq \quad \delta\, \kappa(x^{LP})$$

für ein $\delta \geq 1$. Damit folgt wegen (3.3) zweierlei:

$$\begin{aligned} E(\kappa^{L_\pi}) &\leq \delta\, \rho(\kappa), \qquad \text{sowie} \\ \kappa(x^{LP}) &\geq \frac{1}{\delta}\, \rho(\kappa). \end{aligned}$$

Das bedeutet, daß wir nicht nur eine *worst case*-Gütegarantie für die Strategie L_π im Vergleich zum Optimalwert nachweisen, sondern gleichzeitig eine Garantie für die Qualität der unteren Schranke $\kappa(x^{LP})$ erhalten, die das lineare Programm (3.2) für den Optimalwert des Schedulingproblems (3.1) liefert.

Alle folgenden Heuristiken basieren auf der Lösung gewisser linearer Programme. Auf die polynomiale Lösbarkeit dieser linearen Programme werden wir im letzten Abschnitt des Kapitels genauer eingehen.

3.2 Stochastisches Scheduling auf einer Maschine

Für stochastisches Scheduling auf einer Maschine ohne Präzedenzrelation oder *release dates* ist die *ratio rule* bereits eine optimale Strategie, vergleiche Abschnitt 2.5.2. Wir geben in diesem Abschnitt Heuristiken unter Berücksichtigung nichttrivialer *release dates* und/oder einer Präzedenzrelation an. Dabei setzten wir lediglich die stochastische Unabhängigkeit der Verteilungen P_i voraus.

3.2.1 Gültige Ungleichungen

Wir hatten bereits in Abschnitt 2.5.2 bewiesen, daß für stochastische Schedulingprobleme auf einer Maschine der Vektor der erwarteten Fertigstellungszeiten $E(\mathcal{C}^\Pi)$ unter einer beliebigen Strategie Π die Ungleichungen

$$\sum_{i \in S} E(P_i)\, E(\mathcal{C}_i^\Pi) \quad \geq \quad f(S) \qquad \text{für alle } S \subseteq I \tag{3.4}$$

erfüllt. Dabei ist

$$f(S) \;=\; \frac{1}{2}\Big(\Big(\sum_{i \in S} E(P_i)\Big)^2 + \sum_{i \in S}(E(P_i))^2\Big).$$

Genauer gilt für $\prec = \emptyset$ und ohne *release dates* sogar:

$$\operatorname{conv}\{E(\mathcal{C}^\Pi) \mid \Pi \text{ Strategie }\} + \mathbb{R}^n_+ \quad = \quad \{x \in \mathbb{R}^n \mid \sum_{i \in S} E(P_i)\, x_i \geq f(S) \quad \forall\, S \subseteq I\},$$

und $\operatorname{conv}\{E(\mathcal{C}^\Pi) \mid \Pi \text{ Strategie }\}$ ist das Basispolytop des supermodularen LTP $\mathcal{P}_\geq(E(P), f)$, vergleiche Sätze 2.8 und 2.14.

In diesem Fall wird also der Lösungsraum der zu erwartenden Fertigstellungszeiten vollständig durch die Ungleichungen (3.4) zusammen mit der Gleichung $\sum_{i \in I} E(P_i)\, x_i = f(I)$ beschrieben[2], und nach Satz 2.14 ist Smith's *ratio rule* eine optimale Strategie.

Durch die Einführung einer Präzedenzrelation und/oder *release dates* geht diese vollständige Beschreibung des Lösungsraums zwar verloren, aber die Ungleichungen (3.4) bleiben dennoch gültig. Diese Tatsache macht man sich dadurch klar, daß durch die zusätzliche Einführung einer Präzedenzrelation und/oder *release dates* lediglich die Menge der *zulässigen* Strategien verringert wird[3].

Die Ungleichungen (3.4) und die folgende Abschätzung bilden die Grundlage aller Heuristiken dieses Abschnitts. Die Abschätzung findet man in [Sch 96, Lemma 1], und sie überträgt sich analog auf den stochastischen Fall.

[2] Dieses Resultat beweist M. Queyranne für den deterministischen Fall, vergleiche [Que 93, Lemma 2.1].

[3] Wir erinnern daran, daß Strategien, die die Präzedenzrelation (und/oder *release dates*) respektieren, auch zulässig genannt werden.

Lemma 3.1 *Sei $x \in \mathbb{R}^n$ ein beliebiger Punkt, der den Ungleichungen*

$$\sum_{i \in S} E(P_i)\, x_i \geq f(S) \qquad \text{für alle} \quad S \subseteq I \tag{3.5}$$

genügt. Sei o. B. d. A. $x_1 \leq \cdots \leq x_n$, dann gilt

$$\sum_{i=1}^{k} E(P_i) \ \leq \ 2x_k \qquad \text{für alle} \quad k \in I.$$

Beweis. Mit $S = \{1, \ldots, k\}$ folgt aus (3.5)

$$\sum_{i=1}^{k} E(P_i)\, x_i \ \geq \ \frac{1}{2}\Big(\Big(\sum_{i=1}^{k} E(P_i)\Big)^2 + \sum_{i=1}^{k} (E(P_i))^2\Big).$$

Wegen $x_k \geq x_{k-1} \geq \cdots \geq x_1$ folgt damit

$$x_k \sum_{i=1}^{k} E(P_i) \ \geq \ \frac{1}{2}\Big(\Big(\sum_{i=1}^{k} E(P_i)\Big)^2 + \sum_{i=1}^{k} (E(P_i))^2\Big).$$

Wir teilen durch $\frac{1}{2} \sum_{i=1}^{k} E(P_i)$, und erhalten:

$$2x_k \ \geq \ \sum_{i=1}^{k} E(P_i) \ + \ \frac{\sum_{i=1}^{k} (E(P_i))^2}{\sum_{i=1}^{k} E(P_i)},$$

und damit folgt die Behauptung. ■

3.2.2 Präzedenzrelationen

Wir wollen in diesem Abschnitt $\prec \neq \emptyset$ zulassen und erinnern an die Schreibweise $(k, l) \in \prec$, wenn Vorgang k beendet sein muß, bevor l gestartet werden darf.

Es sei erwähnt, daß die deterministische Version dieses Problems, also $1 \mid \text{prec} \mid \sum_{i \in I} w_i\, C_i$ bereits $\mathcal{NP}$-schwer ist, vergleiche [LRKB 77]. Damit ist auch das stochastische Problem

$$(\Theta, \{S \subseteq I \mid |S| = 2\}, P, \sum_{i \in I} w_i\, C_i) \tag{3.6}$$

für stochastisch unabhängige Dauerverteilungen P_i $\mathcal{NP}$-schwer.

Zunächst halten wir fest:

Lemma 3.2 *Es gilt für alle $(k, l) \in \prec$*

$$E(\mathcal{C}_l^{\Pi}) \geq E(\mathcal{C}_k^{\Pi}) + E(P_k) \tag{3.7}$$

unter jeder zulässigen Strategie Π.

Beweis. Seien Π eine zulässige Strategie und $p \in \mathbb{R}^n_+$ eine beliebige Realisierung der Vorgangsdauern, dann gilt

$$C_l \geq C_k + p_k \qquad \text{für alle } (k,l) \in \prec,$$

wenn $C_i = (\Pi(p))_i + p_i$ die Fertigstellungszeit von Vorgang $i \in I$ ist. Also gilt die Behauptung punktweise, und daher folgt (3.7) für alle zulässigen Strategien Π. ■

Damit erfüllt jeder Vektor erwarteter Fertigstellungszeiten $E(\mathcal{C}^\Pi)$ unter einer zulässigen Strategie Π sowohl (3.4) als auch (3.7). Es liegt auf der Hand, das durch diese Ungleichungen gegebene lineare Programm zu untersuchen:

$$\begin{aligned} \min\{\kappa(x) \quad | \quad & \sum_{i \in S} E(P_i)\, x_i \geq f(S) \quad \forall\, S \subseteq I, \\ & x_l \geq x_k + E(P_k) \quad \forall\, (k,l) \in \prec \qquad \}. \end{aligned} \tag{3.8}$$

Bezeichne x^{LP} eine Optimallösung für (3.8), und sei π eine Permutation auf I mit

$$x^{LP}_{\pi_1} \leq \cdots \leq x^{LP}_{\pi_n}.$$

Dann untersuchen wir die Prioritätsstrategie L_π. Wegen $x^{LP}_l > x^{LP}_k$ für alle $(k,l) \in \prec$ ist die Planungsreihenfolge $\pi_1 < \pi_2 < \cdots < \pi_n$ mit der Präzedenzrelation $\prec$ verträglich, und die Vorgänge dürfen tatsächlich in dieser Reihenfolge gestartet werden. Es gilt:

Satz 3.1 *L_π ist eine 2-approximative Heuristik für das Schedulingproblem (3.6), es gilt also*

$$E(\kappa^{L_\pi}) \leq 2\,\rho(\kappa).$$

Beweis. Sei o. B. d. A. $\pi_i = i$ für alle $i \in I$. Aufgrund der *greedyness* von L_π gilt

$$E(\mathcal{C}^{L_\pi}_k) \;=\; \sum_{i=1}^{k} E(P_i) \qquad \text{für alle } k \in I.$$

Diese Behauptung folgt aus der punktweisen Gültigkeit der Gleichung zusammen mit der Linearität des Erwartungswertes. Wir wenden Lemma 3.1 auf x^{LP} an und erhalten die Abschätzung

$$E(\mathcal{C}^{L_\pi}_k) \;\leq\; 2x^{LP}_k \qquad \text{für alle } k \in I.$$

Damit folgt:

$$\begin{aligned} E(\kappa^{L_\pi}) \;&=\; \sum_{i \in I} w_i\, E(\mathcal{C}^{L_\pi}_i) \\ &\leq\; 2 \sum_{i \in I} w_i\, x^{LP}_i \;=\; 2\,\kappa(x^{LP}). \end{aligned}$$

Wegen $\kappa(x^{LP}) \leq \rho(\kappa)$ folgt die Behauptung. ■

Damit folgt unmittelbar dieselbe Gütegarantie für die untere Schranke, die x^{LP} für den Optimalwert des Schedulingproblems liefert:

Korollar 3.1 *Ist x^{LP} eine Optimallösung für (3.8) und $\rho(\kappa)$ der Optimalwert für das Schedulingproblem (3.6), so gilt $\kappa(x^{LP}) \ \leq \ \rho(\kappa) \ \leq \ 2\kappa(x^{LP})$.*

Bemerkung 3.1 *Im Beweis zu Lemma 3.1 haben wir gesehen, daß die Abschätzung $\sum_{i=1}^{k} E(P_i) \leq 2x_k$ sich noch um den Term $\sum_{i=1}^{k}(E(P_i))^2 \ / \ \sum_{i=1}^{k} E(P_i)$ verfeinern läßt. Man kann damit zeigen, daß L_π in der Tat eine $(2-\frac{2}{n+1})$-approximative Heuristik ist.*

Details findet man im Beweis zu [Sch 96, Theorem 2].

3.2.3 Release Dates

Wir betrachten das stochastische Schedulingproblem (3.6) unter zusätzlicher Berücksichtigung nichttrivialer *release dates* r_i für jeden Vorgang i, d. h. r_i ist der frühste Zeitpunkt, zu dem Vorgang i gestartet werden darf. Dann gelten die folgenden Ungleichungen für $E(\mathcal{C}^\Pi)$:

Lemma 3.3 *Für jede zulässige Strategie Π gilt*

$$E(\mathcal{C}_i^\Pi) \ \geq \ r_i + E(P_i) \qquad \text{für alle} \ \ i \in I \tag{3.9}$$

Beweis. Wieder folgt die Behauptung aus der punktweisen Gültigkeit von $C_i \geq r_i + p_i$ für jede Realisierung $p \in \mathbb{R}_+^n$ der Vorgangsdauern unter jeder zulässigen Strategie Π. ■

Damit erfüllt jeder Vektor erwarteter Fertigstellungszeiten $E(\mathcal{C}^\Pi)$ unter einer zulässigen Strategie Π die Ungleichungen (3.4), (3.7) und (3.9). Wir untersuchen das durch diese Ungleichungen gegebene lineare Programm:

$$\begin{aligned} \min\{\kappa(x) \quad | \quad & \sum_{i \in S} E(P_i)\, x_i \geq f(S) \quad \forall\, S \subseteq I, \\ & x_l \geq x_k + E(P_k) \quad \forall\, (k,l) \in \prec, \\ & x_i \geq E(P_i) + r_i \quad \forall\, i \in I \qquad\qquad \}. \end{aligned} \tag{3.10}$$

Bezeichne wieder x^{LP} eine Optimallösung für (3.10), und sei π eine Permutation auf I mit

$$x_{\pi_1}^{LP} \ \leq \ \cdots \ \leq x_{\pi_n}^{LP}.$$

Wir definieren die folgende Strategie $\widetilde{L}_\pi$:
$\widetilde{L}_\pi$ startet die Vorgänge in der Reihenfolge $\pi_1 < \pi_2 < \cdots < \pi_n$, und berücksichtigt dabei

die *release dates*. Das bedeutet, daß Vorgang π_1 zum Zeitpunkt r_{π_1} gestartet wird. Die restlichen Startzeiten definieren wir rekursiv durch

$$(\widetilde{L}_\pi(p))_{\pi_k} := \max\{r_{\pi_k}, (\widetilde{L}_\pi(p))_{\pi_{k-1}} + p_{\pi_{k-1}}\}.$$

Es sei erwähnt, daß $\widetilde{L}_\pi$ keine Prioritätsstrategie ist, denn $\widetilde{L}_\pi$ weicht auch dann nicht von der Planungsreihenfolge $\pi_1 < \pi_2 < \cdots < \pi_n$ ab, wenn dies zur Vermeidung von Leerzeiten bezüglich $\prec$ und der *release dates* möglich wäre.

In diesem Sinne verletzt $\widetilde{L}_\pi$ sogar die Bedingung (STO 4) für Strategien (da totale Leerzeiten nicht immer – wenn möglich – vermieden werden), wir lassen solches Verhalten in diesem Fall jedoch zu, und es gilt:

Satz 3.2 *$\widetilde{L}_\pi$ ist eine 3-approximative Heuristik für das Schedulingproblem (3.6) unter Berücksichtigung nichttrivialer release dates, es gilt also*

$$E(\kappa^{\widetilde{L}_\pi}) \leq 3\,\rho(\kappa).$$

Beweis. Sei o. B. d. A. $\pi_i = i$ für alle $i \in I$. Die Behauptung ist gezeigt, wenn wir

$$E(\mathcal{C}_k^{\widetilde{L}_\pi}) \;\leq\; 3\,x_k^{LP} \qquad \text{für alle } k \in I \tag{3.11}$$

gezeigt haben, denn die restlichen Schritte entsprechen dem Beweis zu Satz 3.1.

Wir benötigen dazu die folgende Abschätzung:

$$E(\mathcal{C}_k^{\widetilde{L}_\pi}) \;\leq\; \max_{i=1,\ldots,k} r_i \;+\; \sum_{i=1}^{k} E(P_i) \qquad \text{für alle } k \in I. \tag{3.12}$$

Seien dazu $p \in \mathbb{R}_+^n$ eine beliebige Realisierung der Vorgangsdauern und $k \in I$ beliebig aber fest. Jeder der Vorgänge $i = 1, \ldots, k$ darf bezüglich seines *release dates* zum Zeitpunkt $\max_{i=1,\ldots,k} r_i$ gestartet werden. Unter einer Strategie, die mit dem Start des ersten Vorganges bis zu diesem Zeitpunkt wartet und dann gemäß $1 < 2 < \cdots < k < \cdots < n$ fortfährt, werden die Vorgänge $1, \ldots, k$ konsekutiv und ohne Leerzeit in dieser Reihenfolge gestartet. Die Fertigstellungszeit für k ist dann $\max_{i=1,\ldots,k} r_i \;+\; \sum_{i=1}^{k} p_i$.

Offenbar wird k unter $\widetilde{L}_\pi$ höchstens früher gestartet. Dieses Argument können wir für alle $p \in \mathbb{R}_+^n$ und alle $k \in I$ anwenden, und es folgt

$$C_k \leq \max_{i=1,\ldots,k} r_i \;+\; \sum_{i=1}^{k} p_i$$

für jede Realisierung $p \in \mathbb{R}_+^n$ und alle $k \in I$, wenn $C_k = (\widetilde{L}_\pi(p))_k + p_k$ ist. Also gilt die Abschätzung (3.12) punktweise und daher auch im Erwartungswert.

Weiter gilt

$$\max_{i=1,\dots,k} r_i \ \le\ x_k^{LP} \qquad \text{für alle} \ \ k \in I, \tag{3.13}$$

denn wir hatten für *jeden* zulässigen Punkt des linearen Programms (3.10) $x_i \ge E(P_i)+r_i$ für alle $i \in I$ gefordert. Damit folgt insbesondere $x_i \ge r_i$ für alle $i \in I$.
Wegen $x_k^{LP} \ge x_{k-1}^{LP} \ge \cdots \ge x_1^{LP}$ gilt dann (3.13).

Nun sind wir in der Lage, (3.11) zu beweisen:

$$\begin{aligned} E(\mathcal{C}_k^{\widetilde{L}_\pi}) &\le \max_{i=1,\dots,k} r_i + \sum_{i=1}^k E(P_i) && \text{(nach (3.12))} \\ &\le x_k^{LP} + \sum_{i=1}^k E(P_i) && \text{(nach (3.13))} \\ &\le x_k^{LP} + 2x_k^{LP} = 3x_k^{LP}. && \text{(nach Lemma 3.1)} \end{aligned}$$

■

Korollar 3.2 *Ist x^{LP} eine Optimallösung für (3.10), und $\rho(\kappa)$ der Optimalwert für (3.6) unter Berücksichtigung nichttrivialer* release dates, *so gilt* $\kappa(x^{LP}) \ \le\ \rho(\kappa) \ \le\ 3\kappa(x^{LP})$.

Bemerkung 3.2 *Es läßt sich sogar zeigen, daß Strategie $\widetilde{L}_\pi$ tatsächlich eine $(3-\frac{2}{n+1})$-approximative Heuristik ist.*

Da diese Verfeinerung weder in [Sch 96] noch in [HSSW 96] angegeben ist, soll sie hier bewiesen werden:

Beweis. Es gilt unter Verwendung der Cauchy-Schwarz'schen Ungleichung

$$\Big(\sum_{i=1}^k E(P_i)\Big)^2 \ \le\ k \sum_{i=1}^k (E(P_i))^2 \qquad \text{für alle } k \in I. \tag{3.14}$$

Aus dem Beweis zu Lemma 3.1 wissen wir:

$$\sum_{i=1}^k E(P_i) \ \le\ 2x_k \ - \ \frac{\sum_{i=1}^k (E(P_i))^2}{\sum_{i=1}^k E(P_i)}$$

für alle $k \in I$ und jeden zulässigen Vektor des linearen Programms (3.10). Es folgt

$$\begin{aligned} E(\mathcal{C}_k^{\widetilde{L}_\pi}) &\le \max_{i=1,\dots,k} r_i + \sum_{i=1}^k E(P_i) && \text{(nach (3.12))} \\ &\le \max_{i=1,\dots,k} r_i + 2x_k^{LP} - \frac{\sum_{i=1}^k (E(P_i))^2}{\sum_{i=1}^k E(P_i)} \end{aligned}$$

$$
\begin{aligned}
&= \max_{i=1,\dots,k} r_i \;+\; 2x_k^{LP} \;-\; \sum_{i=1}^{k} E(P_i)\, \frac{\sum_{i=1}^{k}(E(P_i))^2}{\left(\sum_{i=1}^{k} E(P_i)\right)^2} \\
&\le \max_{i=1,\dots,k} r_i \;+\; 2x_k^{LP} \;-\; \sum_{i=1}^{k} E(P_i)\, \frac{1}{k} && \text{(nach (3.14))} \\
&\le \max_{i=1,\dots,k} r_i \;+\; 2x_k^{LP} \;-\; \frac{1}{n} \sum_{i=1}^{k} E(P_i) \\
&\le \max_{i=1,\dots,k} r_i \;+\; 2x_k^{LP} \;-\; \frac{1}{n} \left(E(\mathcal{C}_k^{\widetilde{L}_\pi}) \;-\; \max_{i=1,\dots,k} r_i \right) && \text{(nach (3.12))} \\
&= (1+\frac{1}{n}) \max_{i=1,\dots,k} r_i \;+\; 2x_k^{LP} \;-\; \frac{1}{n} E(\mathcal{C}_k^{\widetilde{L}_\pi}).
\end{aligned}
$$

Damit erhalten wir

$$
\begin{aligned}
(1+\frac{1}{n}) E(\mathcal{C}_k^{\widetilde{L}_\pi}) &\le (1+\frac{1}{n}) \max_{i=1,\dots,k} r_i \;+\; 2x_k^{LP} \\
&\le (1+\frac{1}{n})\, x_k^{LP} \;+\; 2x_k^{LP}, && \text{(nach (3.13))}
\end{aligned}
$$

und letztlich gilt

$$
E(\mathcal{C}_k^{\widetilde{L}_\pi}) \;\le\; (3 - \frac{2}{n+1})\, x_k^{LP}.
$$

Damit ist die Behauptung bewiesen, denn die restlichen Schritte entsprechen denen im Beweis zu Satz 3.1. ∎

3.3 Stochastisches Scheduling auf mehreren Maschinen

In diesem Abschnitt geben wir verschiedene **LP**-Relaxierungen mit zugehörigen Heuristiken für stochastisches Scheduling auf $m \in \mathbb{N}$ identischen Maschinen ohne Präzedenzrelation an.

Das Modell, mit dem wir uns beschäftigen, ist durch

$$
((I,\emptyset), \{S \subseteq I \mid |S| = m+1\}, P, \sum_{i \in I} w_i\, C_i) \tag{3.15}
$$

gegeben. Dabei ist $m \in \mathbb{N}$ beliebig und seien wieder stochastisch unabhängige Dauerverteilungen P_i vorausgesetzt, also $P = \bigotimes_{i \in I} P_i$. Es ist bekannt, daß bereits das deterministische Problem $\mathbf{P} \mid\mid \sum_{i \in I} w_i\, C_i$ $\mathcal{NP}$-schwer ist, vergleiche [GJ 79]. Demnach ist (3.15) ebenfalls $\mathcal{NP}$-schwer, wenn wir bei den Dauerverteilungen auch deterministische Dauern zulassen.

Die erzielten Resultate beschränken sich auf Dauerverteilungen P_i, deren Varianz sich nach oben durch das Quadrat des Erwartungswertes abschätzen läßt (vergleiche Bedingung

(3.20)). Insbesondere sind exponentialverteilte und deterministische Vorgangsdauern als Spezialfälle enthalten.

Das Problem (3.15) bleibt daher auch unter Bedingung (3.20) an die Dauerverteilungen der Vorgänge $\mathcal{NP}$-schwer.

3.3.1 Gültige Ungleichungen

In [Sch 96, Lemma 7] zeigt A. S. Schulz, daß im *deterministischen* Scheduling der Vektor $C = (C_1, \ldots, C_n)$ der Fertigstellungszeiten für jeden zulässigen Terminplan den folgenden Ungleichungen genügt:

$$\sum_{i \in S} p_i\, C_i \quad \geq \quad \frac{1}{2m} \left(\sum_{i \in S} p_i \right)^2 \; + \; \frac{1}{2} \sum_{i \in S} p_i^2 \qquad \text{für alle } S \subseteq I.$$

Dabei bezeichnet p_i in diesem Fall die (deterministische) Dauer von Vorgang i.

Dieselbe Klasse von Ungleichungen konnten wir für *beliebige Dauerverteilungen* im stochastischen Fall nicht nachweisen.

Stattdessen führen wir die folgenden Ungleichungen ein:

$$\begin{aligned} \sum_{i \in S} E(P_i)\, E(\mathcal{C}_i^{\Pi}) \quad \geq \quad & \frac{1}{2m} \left(\sum_{i \in S} E(P_i) \right)^2 \; + \; \frac{1}{2} \sum_{i \in S} (E(P_i))^2 \\ & - \left(\frac{1}{2} - \frac{1}{2m} \right) \sum_{i \in S} \mathrm{Var}(P_i) \qquad \textit{für alle } S \subseteq I \end{aligned} \tag{3.16}$$

Wir definieren $h : 2^I \longrightarrow \mathbb{R}$ entsprechend (3.16) durch die rechte Seite der Ungleichungen.

Es gilt:

Lemma 3.4 *Sei Π eine beliebige Strategie für das stochastische Schedulingproblem (3.15) unter stochastisch unabhängig verteilten Vorgangsdauern. Dann genügt der Vektor $E(\mathcal{C}^{\Pi})$ den Ungleichungen (3.16).*

Beweis. Wir gehen von einer beliebigen Realisierung $p \in \mathbb{R}_+^n$ der Vorgangsdauern und einer beliebigen Strategie Π aus. Der Beweis wird ähnlich geführt wie schon der Beweis zu Satz 2.14.

Wieder sei $s = (s_1, \ldots, s_n)$ der Vektor der Startzeiten unter Π, und bezeichne $S_j \subseteq I$ für alle $j \in \{1, \ldots, m\}$ die Menge derjenigen Vorgänge, die unter Π und p auf Maschine j bearbeitet werden ($S_j = S_j(\Pi, p)$). Aufgrund der Eigenschaft (STO 3) von Strategien ist S_j wohldefiniert, denn Π erzeugt – abgesehen von einer möglichen Permutation der

Maschinen – stets denselben Terminplan für eine feste Realisierung p. Es gilt wie schon im Einmaschinenfall

$$\sum_{i \in S_j} p_i\, s_i \;\geq\; \sum_{k,l \in S_j,\; k<l} p_k\, p_l \qquad \text{für alle } j \in \{1, \ldots, m\}. \tag{3.17}$$

Denn werden alle Vorgänge auf Maschine j konsekutiv *ohne* Leerzeit geplant, gilt (3.17) bereits mit „ $=$ “; vergleiche (2.25) im Beweis zu Satz 2.14. Aufgrund etwaiger Leerzeiten, die unter einer beliebigen Strategie auftreten können, gilt im Mehrmaschinenfall auf jeder Maschine „ $\geq$ “.

Weiter gilt

$$\sum_{j=1}^{m} \Big(\sum_{i \in S_j} p_i \Big)^2 \;\geq\; \frac{1}{m} \Big(\sum_{i \in S} p_i \Big)^2 \tag{3.18}$$

aufgrund der Cauchy-Schwarz'schen Ungleichung.

Damit erhalten wir für alle $S \subseteq I$ die Abschätzung

$$\begin{aligned}
\sum_{i \in S} p_i\, s_i &= \sum_{j=1}^{m} \sum_{i \in S_j} p_i\, s_i \\
&\geq \sum_{j=1}^{m} \sum_{k,l \in S_j,\; k<l} p_k\, p_l && \text{(nach (3.17))} \\
&= \sum_{j=1}^{m} \frac{1}{2} \left(\Big(\sum_{i \in S_j} p_i \Big)^2 - \sum_{i \in S_j} p_i^2 \right) \\
&= \frac{1}{2} \sum_{j=1}^{m} \Big(\sum_{i \in S_j} p_i \Big)^2 - \frac{1}{2} \sum_{i \in S} p_i^2 \\
&\geq \frac{1}{2m} \Big(\sum_{i \in S} p_i \Big)^2 - \frac{1}{2} \sum_{i \in S} p_i^2 && \text{(nach (3.18))} \\
&= \frac{1}{m} \Big(\sum_{k,l \in S,\; k<l} p_k\, p_l \Big) - \Big(\frac{1}{2} - \frac{1}{2m} \Big) \sum_{i \in S} p_i^2
\end{aligned} \tag{3.19}$$

für eine beliebige Strategie Π und jede Realisierung $p \in \mathbb{R}^n_+$.

Wir erinnern an den Zusammenhang der Verteilungen für Dauern, Startzeiten und Fertigstellungszeiten der Vorgänge:

$$P \xrightarrow{\Pi} \mathcal{S}^{\Pi} \xrightarrow{\text{id}+P} \mathcal{C}^{\Pi}.$$

Wie wir schon im Beweis zu Satz 2.14 festgestellt hatten, sind die Zufallsgrößen $\mathcal{S}_i^{\Pi}$ und P_i stochastisch unabhängig für alle $i \in I$, und es gilt daher insbesondere:

$$E(P_i\, \mathcal{S}_i^{\Pi}) = E(P_i) E(\mathcal{S}_i^{\Pi}) \qquad \text{für alle } i \in I \text{ und alle } \Pi.$$

Ferner sei an

$$E(P_i^2) \;=\; (E(P_i))^2 \;+\; \mathrm{Var}(P_i)$$

erinnert. Wegen der stochastischen Unabhängigkeit der P_i folgt insbesondere die paarweise Unkorreliertheit, also $E(P_i\,P_j) = E(P_i)E(P_j)$ für alle $i \neq j$. Damit folgt aus (3.19) unter Ausnutzung der Linearität des Erwartungswertes:

$$\begin{aligned}
\sum_{i\in S} E(P_i)\,E(\mathcal{S}_i^{\Pi}) \;&\geq\; \frac{1}{m}\Big(\sum_{k,l\in S,\;k<l} E(P_k\,P_l)\Big) \;-\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\sum_{i\in S} E(P_i^2) \\
&=\; \frac{1}{m}\Big(\sum_{k,l\in S,\;k<l} E(P_k)\,E(P_l)\Big) \;-\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\sum_{i\in S}\Big((E(P_i))^2 \;+\mathrm{Var}(P_i)\Big) \\
&=\; \frac{1}{2m}\left(\sum_{i\in S} E(P_i)\right)^2 \;-\frac{1}{2m}\sum_{i\in S}(E(P_i))^2 \\
&\quad -\Big(\frac{1}{2}-\frac{1}{2m}\Big)\sum_{i\in S}(E(P_i))^2 \;-\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\sum_{i\in S}\mathrm{Var}(P_i) \\
&=\; \frac{1}{2m}\left(\sum_{i\in S} E(P_i)\right)^2 \;-\; \frac{1}{2}\sum_{i\in S}(E(P_i))^2 \\
&\quad -\;\left(\frac{1}{2}-\frac{1}{2m}\right)\sum_{i\in S}\mathrm{Var}(P_i) \qquad \text{für alle } S\subseteq I.
\end{aligned}$$

Wegen $E(\mathcal{C}_i^{\Pi}) = E(\mathcal{S}_i^{\Pi}) + E(P_i)$ folgt die Behauptung. ∎

Unter einer weiteren Voraussetzung an die Varianz der einzelnen Dauerverteilungen können wir eine zu Lemma 3.1 vergleichbare Abschätzung beweisen, auf der die Analyse der nachfolgenden Heuristiken beruht.

Die Bedingung ist:

$$\mathrm{Var}(P_i) \;\leq\; (E(P_i))^2 \qquad \text{für alle } i\in I. \tag{3.20}$$

Diese Forderung ist – abgesehen von der kaum hilfreichen Bemerkung, daß die Varianz nicht „zu groß“ sein darf – nicht sehr anschaulich interpretierbar. Daher werden wir einige Verteilungen (ohne Anspruch auf Vollständigkeit) in dieser Hinsicht untersuchen. Von Interesse sind letztlich nur Verteilungen auf $\mathbb{R}_+$, da negative Dauern keinen Sinn ergäben[4].

Stetige Verteilungen :

- Die **Exponentialverteilung** $X \sim \exp(\lambda)$ erfüllt (3.20) wegen $\mathrm{Var}(X) = (E(X))^2 = \frac{1}{\lambda^2}$ mit Gleichheit.
- Für die **Normalverteilung** $N(\mu,\sigma^2)$ (entsprechend der Nichtnegativitätsbedingung passend modifiziert) gilt (3.20) offenbar nur für $\sigma^2 \leq \mu^2$.

[4] Eine ausführliche Untersuchung der angeführten Verteilungen findet man etwa in [Ros 85, Kapitel 2].

- Die **Gleichverteilung** auf einem Intervall $[a,b] \subset \mathbb{R}_+$ erfüllt (3.20) wegen $E(X) = \frac{a+b}{2}$ und $\mathrm{Var}(X) = \frac{(b-a)^2}{12}$ mit „ < “.

Diskrete Verteilungen :

- Für eine **bernoulliverteilte** Zufallsvariable X mit Parameter p und Werten x_1 und x_2 gilt $E(X) = px_1 + (1-p)x_2$ und $\mathrm{Var}(X) = p(1-p)(x_1 - x_2)^2$. Etwa für $p = \frac{1}{2}$ ist die Bedingung dann für alle $x_1, x_2 \geq 0$ erfüllt.
- Die **geometrische Verteilung** mit Parameter p erfüllt (3.20) wegen $E(X) = \frac{1}{p}$ und $\mathrm{Var}(X) = \frac{1-p}{p^2}$ mit „<“.
- Für die **Poissonverteilung** mit Parameter λ gilt $E(X) = \mathrm{Var}(X) = \lambda$, also gilt (3.20) nur für $\lambda \geq 1$.
- Für **Einpunktverteilungen**, also den deterministischen Fall gilt die Bedingung trivialerweise.

Der Versuch, Bedingung (3.20) mit dem Begriff des dritten zentralen Moments (der *Schiefe*) $E((X - E(X))^3)$ in Verbindung zu bringen, ergab keine hilfreichen Ergebnisse. Grob gesagt sind Verteilungen mit positiver Schiefe (wie die Exponentialverteilung) „eher ungünstig“, da sie einen „kleinen“ Erwartungswert besitzen. Die Exponentialverteilung bildet hier gewissermaßen den Grenzfall, und erfüllt (3.20) mit Gleichheit. Wie die obige Aufzählung zeigt, läßt sich auch für symmetrische Verteilungen keine allgemeine Aussage treffen.

Entscheidend ist, daß wir unter dieser Voraussetzung an die Varianz der einzelnen Dauerverteilungen das folgende Lemma zeigen können.

Lemma 3.5 *Sei $x \in \mathbb{R}^n$ ein beliebiger Punkt, der den Ungleichungen*

$$\sum_{i \in S} E(P_i)\, x_i \geq h(S) \qquad \forall\ S \subseteq I \tag{3.21}$$

genügt, und sei o. B. d. A. $x_1 \leq x_2 \leq \cdots \leq x_n$. Sei ferner für die Dauerverteilungen P_i Bedingung $\mathrm{Var}(P_i) \leq (E(P_i))^2$ für alle $i \in I$ erfüllt. Dann gilt

$$\frac{1}{m} \sum_{i=1}^{k} E(P_i) \ \leq\ 2\, x_k \qquad \text{für alle } k = 1, \ldots, n.$$

Beweis. Sei $k \in I$ und $S = \{1, \ldots, k\}$, dann besagt $\sum_{i \in S} E(P_i)\, x_i \geq h(S)$ gerade

$$\sum_{i=1}^{k} E(P_i)\, x_i \ \geq\ \frac{1}{2m} \left(\sum_{i=1}^{k} E(P_i) \right)^2 \ +\ \frac{1}{2} \sum_{i=1}^{k} (E(P_i))^2 \ -\ \left(\frac{1}{2} - \frac{1}{2m} \right) \sum_{i=1}^{k} \mathrm{Var}(P_i).$$

Wegen $x_k \geq x_{k-1} \geq \cdots \geq x_1$ erhalten wir

$$x_k \sum_{i=1}^{k} E(P_i) \;\geq\; \frac{1}{2m}\left(\sum_{i=1}^{k} E(P_i)\right)^2 \;+\; \frac{1}{2}\sum_{i=1}^{k}(E(P_i))^2 \;-\; \left(\frac{1}{2}-\frac{1}{2m}\right)\sum_{i=1}^{k}\mathrm{Var}(P_i),$$

und nach Teilen durch $\frac{1}{2}\sum_{i=1}^{k} E(P_i)$ ergibt sich

$$\begin{aligned} 2\,x_k \;&\geq\; \frac{1}{m}\sum_{i=1}^{k} E(P_i) \;+\; \frac{\sum_{i=1}^{k}(E(P_i))^2}{\sum_{i=1}^{k} E(P_i)} - \left(1-\frac{1}{m}\right)\frac{\sum_{i=1}^{k}\mathrm{Var}(P_i)}{\sum_{i=1}^{k} E(P_i)} \\ &=\; \frac{1}{m}\sum_{i=1}^{k} E(P_i) \;+\left(1-\frac{1}{m}\right)\frac{\sum_{i=1}^{k}\left(\frac{m}{m-1}(E(P_i))^2-\mathrm{Var}(P_i)\right)}{\sum_{i=1}^{k} E(P_i)} \\ &\geq\; \frac{1}{m}\sum_{i=1}^{k} E(P_i). \end{aligned}$$

■

Ein wichtiges Hilfsmittel bildet ferner die folgende Abschätzung, die für eine beliebige statische Prioritätsstrategie L_π gilt.

Lemma 3.6 *Es sei L_π eine beliebige Prioritätsstrategie für ein stochastisches Scheduling-problem auf m Maschinen, dann gilt*

$$E(C_{\pi_k}^{L_\pi}) \;\leq\; \frac{1}{m}\sum_{i=1}^{k-1} E(P_{\pi_i}) \;+\; E(P_{\pi_k}) \qquad \textit{für alle } k \in I.$$

Beweis. Wir zeigen die punktweise Gültigkeit der Ungleichung. Die Behauptung folgt dann aus der Linearität des Erwartungswertes.

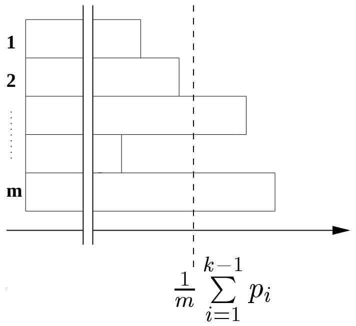


Abbildung 3.1: Terminplan für die Vorgänge $1,\ldots,k-1$ unter L_π

Sei o. B. d. A. $\pi_i = i$ für alle $i \in I$, und sei $p \in \mathbb{R}_+^n$ beliebig. Abbildung 3.1 zeigt schematisch den von L_π erzeugten Terminplan für die Vorgänge $1,\ldots,k-1$.

Aufgrund der *greedyness* von L_π gilt stets

$$s_k \leq \frac{1}{m}\sum_{i=1}^{k-1} p_i$$

wenn wieder $s_k = (L_\pi(p))_k$ die Startzeit von Vorgang k bezeichnet. Damit folgt

$$C_k \leq \frac{1}{m} \sum_{i=1}^{k-1} p_i \; + \; p_k,$$

was zu zeigen war. ■

3.3.2 Heuristiken ohne Berücksichtigung von Release Dates

Wir wissen, daß der Vektor der Fertigstellungszeiten $E(\mathcal{C}^\Pi)$ unter einer beliebigen Strategie Π den Ungleichungen (3.16) genügt. Ferner erfüllt $E(\mathcal{C}^\Pi)$ trivialerweise die Bedingung

$$E(\mathcal{C}_i^\Pi) \;\geq\; E(P_i) \qquad \text{für alle } i \in I \tag{3.22}$$

und für jede Strategie Π. Wir betrachten das durch (3.16) und (3.22) gegebene lineare Programm:

$$\begin{aligned} \min\{\kappa(x) \quad | \quad & \sum_{i \in S} E(P_i)\, x_i \geq h(S) \quad \forall\, S \subseteq I, \\ & x_i \;\geq\; E(P_i) \quad \forall\, i \in I \qquad\qquad \}. \end{aligned} \tag{3.23}$$

Bezeichne x^{LP} eine Optimallösung für (3.23), und sei π eine Permutation auf I mit

$$x_{\pi_1}^{LP} \leq \cdots \leq x_{\pi_n}^{LP}.$$

Wir definieren wieder durch π die statische Prioritätsstrategie L_π, und es gilt:

Satz 3.3 *Erfüllen die Dauerverteilungen P_i die Bedingung $Var(P_i) \leq (E(P_i))^2$ für alle $i \in I$, so ist die statische Prioritätsstrategie L_π eine $(3 - \frac{1}{m})$-approximative Heuristik für das Schedulingproblem (3.15), es gilt also*

$$E(\kappa^{L_\pi}) \leq (3 - \frac{1}{m})\, \rho(\kappa).$$

Beweis. Sei o. B. d. A. $\pi_i = i$ für alle $i \in I$. Wir zeigen wieder

$$E(\mathcal{C}_k^{L_\pi}) \leq (3 - \frac{1}{m})\, x_k^{LP} \qquad \text{für alle } k \in I,$$

und die Behauptung folgt wie im Beweis zu Satz 3.1.

Es gilt für alle $k \in I$:

$$E(\mathcal{C}_k^{L_\pi}) \;\; \leq \;\; \frac{1}{m} \sum_{i=1}^{k-1} E(P_i) \; + \; E(P_k) \qquad\qquad \text{(nach Lemma 3.6)}$$

$$
\begin{aligned}
&= \frac{1}{m} \sum_{i=1}^{k} E(P_i) \; + \; (1 - \frac{1}{m}) \, E(P_k) \\
&\leq 2 \, x_k^{LP} \; + \; (1 - \frac{1}{m}) \, E(P_k) && \text{(nach Lemma 3.5)} \\
&\leq 2 \, x_k^{LP} \; + \; (1 - \frac{1}{m}) \, x_k^{LP} && \text{(wegen } x_k^{LP} \geq E(P_k)\text{)} \\
&= (3 - \frac{1}{m}) \, x_k^{LP}.
\end{aligned}
$$

■

Wieder erhalten wir eine entsprechende Gütegarantie der unteren Schranke x^{LP}, wenn die Dauerverteilungen der Bedingung (3.20) genügen.

Korollar 3.3 *Ist x^{LP} eine Optimallösung für (3.23) und $\rho(\kappa)$ der Optimalwert für das Schedulingproblem (3.15), so gilt $\kappa(x^{LP}) \;\leq\; \rho(\kappa) \;\leq\; (3 - \frac{1}{m})\kappa(x^{LP})$.*

In Korollar 3.4 werden wir zeigen, daß sogar $\kappa(x^{LP}) \;\leq\; \rho(\kappa) \;\leq\; (2 - \frac{1}{m})\kappa(x^{LP})$ für die **LP**-Relaxierung (3.23) gilt. Zum Beweis benötigen wir die folgende **LP**-Relaxierung, die zusätzlich eine weitere Heuristik für das Schedulingproblem (3.15) liefert.

Eine $(2 - \frac{1}{m})$-approximative Heuristik

Wir geben nun eine weitere **LP**-Relaxierung an, die zu einer $(2 - \frac{1}{m})$-approximativen Heuristik führt.

Die Heuristik beruht auf der vermeintlich „schlechteren“ linearen Relaxierung, die man erhält, wenn man die Ungleichungen (3.22) aus dem oben untersuchten linearen Programm entfernt.

Wie sich herausstellen wird, entspricht die Heuristik im Fall exponentialverteilter Dauern gerade der *ratio rule* von Smith. Damit liefern wir im Hinblick auf das erste Kapitel zusätzlich ein Ergebnis über die Güte der *ratio rule* bei Nichtverträglichkeit absoluter und relativer Kosten.

Wir lösen das lineare Programm

$$
\min\{\kappa(x) \mid \sum_{i \in S} E(P_i) \, x_i \geq h(S) \quad \forall \, S \subseteq I\}, \tag{3.24}
$$

und sei wieder x^{LP} eine Optimallösung von (3.24). Sei π eine Permutation auf I mit

$$
x_{\pi_1}^{LP} \leq \cdots \leq x_{\pi_n}^{LP},
$$

und sei L_π die zugehörige Prioritätsstrategie. Dann gilt:

Satz 3.4 *Erfüllen die Dauerverteilungen P_i die Bedingung $Var(P_i) \leq (E(P_i))^2$ für alle $i \in I$, so ist die statische Prioritätsstrategie L_π eine $(2 - \frac{1}{m})$-approximative Heuristik für (3.15), es gilt also*

$$E(\kappa^{L_\pi}) \leq (2 - \frac{1}{m})\, \rho(\kappa).$$

Beweis. Zunächst stellen wir fest, daß die Funktion h *supermodular* ist [5]. Damit können wir Korollar 2.6 zur genauen Bestimmung einer Optimallösung x^{LP} für das lineare Programm (3.24) anwenden. Wir erinnern an $\kappa = w^t\, C$, und sei o. B. d. A.

$$\frac{w_1}{E(P_1)} \geq \frac{w_2}{E(P_2)} \geq \cdots \geq \frac{w_n}{E(P_n)}.$$

Dann gilt unter Verwendung von Korollar 2.6:

$$\begin{aligned} x_k^{LP} &= \frac{h(\{1,\ldots,k\}) - h(\{1,\ldots,k-1\})}{E(P_k)} \\ &= \frac{1}{2m}\, 2 \sum_{i=1}^{k-1} E(P_i) \;+\; \frac{1}{2m} E(P_k) \;+\; \frac{1}{2} E(P_k) \;-\; \left(\frac{1}{2} - \frac{1}{2m}\right) \frac{\mathrm{Var}(P_k)}{E(P_k)} \\ &= \frac{1}{m} \sum_{i=1}^{k} E(P_i) \;+\; \left(\frac{1}{2} - \frac{1}{2m}\right) \frac{(E(P_k))^2 - \mathrm{Var}(P_k)}{E(P_k)} \end{aligned} \tag{3.25}$$

für alle $k = 1, \ldots, n$. Daraus folgt unter Beachtung von $\mathrm{Var}(P_k) \leq (E(P_k))^2$:

$$\frac{1}{m} \sum_{i=1}^{k} E(P_i) \;\leq\; x_k^{LP} \qquad \text{für alle } k \in I. \tag{3.26}$$

Bezeichnet Π^{opt} eine optimale Strategie für das Schedulingproblem (3.15), so gilt trivialerweise

$$E(\mathcal{C}_k^{\Pi^{opt}}) \;\geq E(P_k) \qquad \text{für alle } k \in I. \tag{3.27}$$

Nun können wir folgern:

$$\begin{aligned} E(\mathcal{C}_k^{L_\pi}) &\leq \frac{1}{m} \sum_{i=1}^{k-1} E(P_i) \;+\; E(P_k) && \text{(nach Lemma 3.6)} \\ &= \frac{1}{m} \sum_{i=1}^{k} E(P_i) \;+\; (1 - \frac{1}{m})\, E(P_k) \\ &\leq x_k^{LP} + (1 - \frac{1}{m})\, E(P_k) && \text{(nach (3.26))} \\ &\leq x_k^{LP} + (1 - \frac{1}{m})\, E(\mathcal{C}_k^{\Pi^{opt}}). && \text{(nach (3.27))} \end{aligned}$$

[5] Diese Tatsache macht man sich leicht anhand von Lemma 2.4 klar, denn es gilt

$$h(S \cup \{k\}) - h(S) = \frac{1}{m} E(P_k) \sum_{i \in S} E(P_i) + \frac{1}{2}(E(P_k))^2 - (\frac{1}{2} - \frac{1}{2m})\mathrm{Var}(P_k)$$

für $S \subseteq I - \{k\}$. Dieser Term ist streng monoton steigend in S für festes k. Mit Lemma 2.4 folgt damit die Supermodularität von h.

Damit gilt

$$\begin{aligned}
E(\kappa^{L_\pi}) &= \sum_{i\in I} w_i\, E(\mathcal{C}_k^{L_\pi}) \\
&\leq \sum_{i\in I} w_i\, x_i^{LP} \ + \ (1-\frac{1}{m}) \sum_{i\in I} w_i\, E(\mathcal{C}_k^{\Pi^{opt}}) \\
&= \kappa(x^{LP}) \ + \ (1-\frac{1}{m}) E(\kappa^{\Pi^{opt}}) \\
&\leq (2-\frac{1}{m}) E(\kappa^{\Pi^{opt}}).
\end{aligned}$$

■

Die Optimallösung x^{LP} des linearen Programms (3.24) liefert ebenfalls eine untere Schranke für den Optimalwert des Schedulingproblems (3.15). Über deren Güte können wir allerdings in diesem Fall *keine* entsprechende Aussage machen.

Wir sind nun in der Lage, die in Korollar 3.3 angegebene Gütegarantie für die **LP**-Relaxierung (3.23) zu verbessern.

Korollar 3.4 *Ist x^{LP} eine Optimallösung für (3.23) und $\rho(\kappa)$ der Optimalwert für das Schedulingproblem (3.15), so gilt $\kappa(x^{LP}) \ \leq \ \rho(\kappa) \ \leq \ (2-\frac{1}{m})\kappa(x^{LP})$.*

Beweis. Wir bezeichnen in diesem Beweis die linearen Programme (3.23) durch **LP1** und (3.24) durch **LP2**.

Offenbar ist dann **LP2** eine **LP**-Relaxierung für **LP1**, und es gilt $\kappa(x^{\mathbf{LP2}}) \leq \kappa(x^{\mathbf{LP1}})$, wenn $x^{\mathbf{LP1}}$ und $x^{\mathbf{LP2}}$ die jeweiligen Optimallösungen sind. Aus dem Beweis zu Satz 3.4 wissen wir, daß

$$E(\mathcal{C}_k^{L_\pi^2}) \leq x_k^{\mathbf{LP2}} + \ (1-\frac{1}{m})\, E(P_k) \qquad \text{für alle } k \in I$$

gilt. Dabei bezeichnet L_π^2 die Heuristik zum linearen Programm (3.24).

Wegen $E(P_k) \leq x_k^{LP1}$ für alle $k \in I$ können wir folgern:

$$\begin{aligned}
E(\kappa^{L_\pi^2}) &= \sum_{i\in I} w_i\, E(\mathcal{C}_k^{L_\pi^2}) \\
&\leq \sum_{i\in I} w_i \left(x_i^{\mathbf{LP2}} + \ (1-\frac{1}{m})\, E(P_i) \right) \\
&\leq \sum_{i\in I} w_i\, x_i^{\mathbf{LP2}} \ + \ (1-\frac{1}{m}) \sum_{i\in I} w_i\, x_i^{\mathbf{LP1}} \\
&= \kappa(x^{\mathbf{LP2}}) \ + \ (1-\frac{1}{m})\, \kappa(x^{\mathbf{LP1}}) \\
&\leq (2-\frac{1}{m})\, \kappa(x^{\mathbf{LP1}})
\end{aligned}$$

Damit gilt:

$$\rho(\kappa) \;\leq\; E(\kappa^{L^2_\pi}) \;\leq\; (2 - \frac{1}{m})\,\kappa(x^{\mathbf{LP1}}).$$

■

Die Tatsache, daß die vermeintlich „schlechtere" Relaxierung hier zu einer zumindest theoretisch besseren Gütegarantie führt, liegt daran, daß im zugehörigen Beweis der Term $\frac{1}{m}\sum_{i=1}^{k} E(P_i)$ gegen die *Optimallösung* x_k^{LP} abgeschätzt werden konnte. Im vorigen Beweis hatten wir die Abschätzung gegen $2x_k^{LP}$ verwendet, die bereits für *jeden* zulässigen Vektor der **LP**-Relaxierung gilt.

Dennoch kann diese Tatsache als Motivation für weitere Anstrengungen in Bezug auf die folgenden Fragen dienen:

- Können durch genauere Analysis – zumindest für Spezialfälle – bessere Schranken als die angegebenen bewiesen werden?
- Was ist eine „gute" Relaxierung für ein Schedulingproblem?
  Liefert eine Heuristik zu „genaueren" Relaxierungen auch bessere Ergebnisse?

Im Hinblick auf das erste Kapitel wollen wir den Fall exponentialverteilter Dauern genauer untersuchen.

Die Ratio Rule bei exponentialverteilten Dauern

Die Heuristik zu dem linearen Programm (3.24) entspricht genau dann der *ratio rule* von Smith, wenn aus

$$\frac{w_1}{E(P_1)} \geq \frac{w_2}{E(P_2)} \geq \cdots \geq \frac{w_n}{E(P_n)}$$

bereits

$$x_1^{LP} \leq x_2^{LP} \leq \cdots \leq x_n^{LP}$$

folgt. Dies ist für exponentialverteilte Dauern der Fall, denn es gilt unter Beachtung von (3.25) dann:

$$x_k^{LP} \;=\; \frac{1}{m}\sum_{i=1}^{k} E(P_i) \qquad \text{für alle } k \in I,$$

und demnach $x_1^{LP} < x_2^{LP} < \cdots < x_n^{LP}$. Damit können wir folgern:

Korollar 3.5 *Für das Schedulingproblem (3.15) unter stochastisch unabhängigen, exponentialverteilten Vorgangsdauern ist die ratio rule eine* $(2 - \frac{1}{m})$*-approximative Heuristik.*

Bekanntlich ist die *ratio rule* im Falle der Verträglichkeit absoluter und relativer Kosten sogar optimal[6], also ist die Heuristik zu (3.24) dann bereits eine optimale Strategie.

Es sind bisher keine Beispiele für *Nichtverträglichkeit* absoluter und relativer Kosten bekannt, so daß die in Korollar 3.5 angegebene Schranke tatsächlich scharf oder zumindest asymptotisch scharf wird.

Im Falle der Nichtverträglichkeit hatte sich herausgestellt, daß die Klasse der statischen Prioritätsstrategien nicht hinreichend für das Modell exponentieller m-Maschinen Probleme ohne Präzedenzrelation war[7]. Mit Korollar 3.5 können wir in diesem Zusammenhang zumindest folgern:

Korollar 3.6 *Für exponentielle Schedulingprobleme auf m Maschinen mit $\prec = \emptyset$ gilt für $\kappa = \sum_{i \in I} w_i C_i$*

$$\rho_{PRIOR}(\kappa) \;\leq\; \rho_{STAT.PRIOR}(\kappa) \;\leq\; (2 - \frac{1}{m})\rho(\kappa).$$

Die Ratio Rule bei deterministischen Dauern

Wir untersuchen eine weitere **LP**-Relaxierung für den *deterministischen* Fall. Das Problem ist in der üblichen Klassifikation deterministischer Schedulingprobleme durch

$$\mathbf{P} \mid \mid \sum_{i \in I} w_i C_i$$

gegeben. Es bezeichne im folgenden p_i die Dauer und C_i die Fertigstellungszeit von Vorgang i. Dann können wir aus (3.16) die Gültigkeit der Ungleichungen:

$$\sum_{i \in S} p_i C_i \quad \geq \quad \frac{1}{2m}\left(\sum_{i \in S} p_i\right)^2 \;+\; \frac{1}{2}\sum_{i \in S} p_i^2 \qquad \text{für alle } S \subseteq I$$

für jeden zulässigen Terminplan folgern; vergleiche [Sch 96, Lemma 7]. Insbesondere gelten die schwächeren Ungleichungen

$$\sum_{i \in S} p_i C_i \quad \geq \quad \underbrace{\frac{1}{2m}\left(\left(\sum_{i \in S} p_i\right)^2 \;+\; \sum_{i \in S} p_i^2\right)}_{=:j(S)} \qquad \text{für alle } S \subseteq I$$

für jeden zulässigen Terminplan; vergleiche [HSSW 96, Lemma 3.1].

Wir untersuchen das lineare Programm

$$\min\{\kappa(x) \mid \sum_{i \in S} p_i x_i \geq j(S) \qquad \text{für alle } S \subseteq I\} \tag{3.28}$$

[6]Vergleiche Satz 1.6.
[7]Vergleiche Beispiel 1.2.

Aufgrund der Supermodularität von j können wir Korollar 2.6 zur Lösung von (3.28) heranziehen. Wir erinnern an $\kappa = w^t C$, und sei o. B. d. A.

$$\frac{w_1}{p_1} \geq \frac{w_2}{p_2} \geq \cdots \geq \frac{w_n}{p_n}.$$

Dann gilt mit Korollar 2.6

$$\begin{aligned} x_k^{LP} &= \frac{j(\{1,\ldots,k\}) - j(\{1,\ldots,k-1\})}{p_k} \\ &= \frac{1}{m}\sum_{i=1}^{k} p_i \end{aligned}$$

für alle $k = 1,\ldots,n$. Demnach folgt $x_1^{LP} < x_2^{LP} < \cdots < x_n^{LP}$.

Die einer Optimallösung des linearen Programms (3.28) zugeordnete Heuristik entspricht also wieder der *ratio rule*. Analog zum Beweis von Satz 3.4 läßt sich zeigen[8]:

Korollar 3.7 *Für das deterministische Schedulingproblem* $\mathbf{P} \mid\mid \sum_{i\in I} w_i C_i$ *ist die ratio rule eine* $(2 - \frac{1}{m})$*-approximative Heuristik.*

T. Kawaguchi und S. Kyan [KK 86] konnten sogar zeigen, daß die Kosten der *ratio rule* niemals höher als das $\frac{\sqrt{2}+1}{2}$-fache der Kosten eines optimalen Terminplans sind, also die *ratio rule* tatsächlich eine $\frac{\sqrt{2}+1}{2}$-approximative Heuristik ist. Dieses Resultat konnten wir auf den allgemeinen stochastischen Fall nicht übertragen.

3.3.3 Eine Heuristik unter Berücksichtigung von Release Dates

Wir untersuchen das Schedulingproblem (3.15) unter zusätzlicher Berücksichtigung nichttrivialer *release dates* r_i für jeden Vorgang i. Unter jeder zulässigen Strategie Π erfüllt der Vektor $E(\mathcal{C}^\Pi)$ dann die Ungleichungen (3.9):

$$E(\mathcal{C}_k^\Pi) \geq r_k + E(P_k) \qquad \text{für alle } k \in I,$$

und wir untersuchen das durch diese Ungleichungen erweiterte lineare Programm:

$$\begin{aligned} \min\{\kappa(x) \quad \mid \quad & \sum_{i\in S} E(P_i)\, x_i \geq h(S) \quad \forall\, S \subseteq I, \\ & x_i \;\geq\; r_i + E(P_i) \quad \forall\, i \in I \qquad \}. \end{aligned} \tag{3.29}$$

Bezeichne wieder x^{LP} eine Optimallösung für (3.29), und sei π eine Permutation auf I mit

$$x_{\pi_1}^{LP} \;\leq\; \cdots \;\leq x_{\pi_n}^{LP}.$$

[8] Vergleiche [HSSW 96, Corollary 3.5]

Wir definieren eine Strategie $\widetilde{L}_\pi$ auf der Basis dieser Reihenfolge:
$\widetilde{L}_\pi$ startet die Vorgänge genau in der Reihenfolge $\pi_1 < \pi_2 < \cdots < \pi_n$, und berücksichtigt dabei die *release dates* r_i für jeden Vorgang i.

Das bedeutet, daß wir Vorgang π_1 zum Zeitpunkt r_{π_1} starten. Für alle weiteren Vorgänge π_k mit $k > 1$ verfahren wir nach der Regel: π_k wird gestartet, sobald eine Maschine frei ist, jedoch erst dann, wenn

- sein *release date* es zuläßt, und
- π_{k-1} bereits gestartet wurde oder zumindest gleichzeitig gestartet werden kann.

Wieder ist $\widetilde{L}_\pi$ keine Prioritätsstrategie, denn $\widetilde{L}_\pi$ plant auch dann Leerzeiten ein, wenn ein Vorgang π_k zwar bezüglich seines *release dates* auf einer freien Maschine gestartet werden könnte, aber Vorgang π_{k-1} noch nicht gestartet werden darf.

Es gilt für $\widetilde{L}_\pi$ der Satz:

Satz 3.5 *Erfüllen die Dauerverteilungen P_i die Bedingung $\mathrm{Var}(P_i) \le (E(P_i))^2$ für alle $i \in I$, so ist Strategie $\widetilde{L}_\pi$ eine $(4 - \frac{1}{m})$-approximative Heuristik für das Schedulingproblem (3.15) unter Berücksichtigung nichttrivialer release dates. Es gilt also*

$$E(\kappa^{\widetilde{L}_\pi}) \le (4 - \frac{1}{m})\,\rho(\kappa).$$

Beweis. Sei o. B. d. A. $\pi_i = i$ für alle $i \in I$. Der Beweis beruht auf der Abschätzung

$$E(\mathcal{C}_k^{\widetilde{L}_\pi}) \;\le\; \max_{i=1,\dots,k} r_i \;+\; \frac{1}{m}\sum_{i=1}^{k-1} E(P_i) \;+\; E(P_k) \tag{3.30}$$

für alle $k \in I$. Diese Abschätzung folgt sofort, wenn man die Überlegungen zum Beweis der entsprechenden Abschätzung (3.12) im Einmaschinenfall mit Lemma 3.6 kombiniert.

Damit gilt:

$$\begin{aligned}
E(\mathcal{C}_k^{\widetilde{L}_\pi}) \;&\le\; \max_{i=1,\dots,k} r_i \;+\; \frac{1}{m}\sum_{i=1}^{k} E(P_i) \;+\; (1-\frac{1}{m})E(P_k) \\
&\le\; x_k^{LP} \;+\; 2x_k^{LP} \;+\; (1-\frac{1}{m})E(P_k) \qquad \text{(nach Lemma 3.5, bzw. (3.13))} \\
&\le\; (4-\frac{1}{m})x_k^{LP}. \qquad \text{(wegen } x_k^{LP} \ge E(P_k))
\end{aligned}$$

Die Behauptung folgt wie schon im Beweis zu Satz 3.1. ■

Wir erhalten die entsprechende Gütegarantie wieder für die untere Schranke $\kappa(x^{LP})$:

Korollar 3.8 *Ist x^{LP} eine Optimallösung für (3.29), und $\rho(\kappa)$ der Optimalwert für das Schedulingproblem (3.15) unter Berücksichtigung nichttrivialer release dates, so gilt $\kappa(x^{LP}) \;\le\; \rho(\kappa) \;\le\; (4 - \frac{1}{m})\kappa(x^{LP})$.*

3.4 Zur polynomialen Lösbarkeit der LP-Relaxierungen

In diesem Abschnitt geben wir einen $\mathcal{O}(n^2)$-Algorithmus zur Lösung der linearen Programme (3.10) für $\prec = \emptyset$, (3.23) sowie (3.29) an.

Die entsprechenden Schedulingprobleme zeichnen sich durch die *Abwesenheit einer Präzedenzrelation* aus, und es stellt sich heraus, daß die Polyeder der zugeordneten linearen Programme sogenannte *Reduktionen* supermodularer LTP sind; vergleiche hierzu [Fuj 91, Abschnitt II.3.1].

Solche Polyeder lassen sich über die Einführung einer (supermodularen) Hilfsabbildung als „normale" supermodulare LTP identifizieren, und zu deren Lösung kann wieder Korollar 2.6 verwendet werden. Das einzige Problem besteht dann noch in der Berechnung dieser Hilfsfunktion.

Die Grundlage für den hier angegebenen Algorithmus zu deren Berechnung bildet ein von M. Queyranne [Que 93, Abschnitt 5] vorgestellter $\mathcal{O}(n \log n)$-Separationsalgorithmus, der auf der Maximierung einer bestimmten supermodularen Mengenfunktion beruht. Wie sich herausstellen wird, läßt sich dieser Algorithmus für die Berechnung der Werte der oben erwähnten Hilfsfunktion modifizieren, so daß letztlich zusammen mit Korollar 2.6 die Lösung der genannten linearen Programme mit einem Gesamtaufwand von $\mathcal{O}(n^2)$ möglich ist.

Zunächst jedoch eine Bemerkung zu den denjenigen linearen Programmen, die auch die einer Präzedenzrelation $\prec$ zugeordneten Ungleichungen berücksichtigen.

Zur Lösung der linearen Programme (3.8) und (3.10)

Beide linearen Programme haben gemeinsam, daß zusätzlich zu den (exponentiell vielen) Ungleichungen

$$\sum_{i \in S} E(P_i)\, E(\mathcal{C}_i^{\Pi}) \quad \geq \quad f(S) \qquad \text{für alle } S \subseteq I$$

eine (polynomiale) Anzahl weiterer Ungleichungen, nämlich

$$x_l \geq x_k + E(P_k) \quad \forall\ (k,l) \in \prec,$$

und bei (3.10) desweiteren

$$x_i \geq r_i + E(P_i) \quad \forall\ i \in I$$

berücksichtigt werden muß.

Für dieses Problem skizziert M. Queyranne einen Algorithmus[9], der jedoch auf die Lösung allgemeiner linearer Programme (etwa via Ellipsoid-Methode) nicht verzichten kann. Wir

[9] Den *'Rudimentary cutting plane algorithm'*, vergleiche [Que 93, Abschnitt 5].

halten also fest, daß zumindest ein (pseudopolynomialer) Algorithmus für diese Probleme existiert und verweisen zu Einzelheiten bezüglich dieses Verfahrens auf [Que 93].

Eine Möglichkeit zur „schnelleren“ Lösung der genannten linearen Programme könnte in dem von S. Fujishige [Fuj 91, Abschnitt II.3.5] vorgestellten *greedy*-Algorithmus für *Polypseudomatroide* liegen.

Eine Formulierung sämtlicher der Präzedenzrelation zugeordneten Ungleichungen zusammen mit den weiteren Ungleichungen derart, daß man die entsprechenden Polyeder als Polypseudomatroide identifizieren könnte, ist jedoch bislang nicht gelungen.

Zur Lösung der linearen Programme (3.10) für $\prec = \emptyset$, (3.23) und (3.29)

Wir beschränken uns auf die Behandlung des Mehrmaschinenfalles (also (3.23) und (3.29)), da eine Modifikation des Verfahrens für den Einmaschinenfall lediglich die Ersetzung der Abbildung h durch Abbildung f beinhaltet und keinerlei Probleme bereitet.

Wir betrachten ein lineares Programm der Art

$$\begin{aligned} \min\{\kappa(x) \quad | \quad & \sum_{i \in S} E(P_i)\, x_i \geq h(S) \quad \forall\, S \subseteq I, \\ & x_i \;\geq\; \omega_i \quad \forall\, i \in I \qquad\qquad \}. \end{aligned} \tag{3.31}$$

Dabei steht ω_i im Fall (3.23) für $E(P_i)$ und im Fall (3.29) entsprechend für $E(P_i) + r_i$.

Bezeichne $\mathcal{P}$ das durch sämtliche Ungleichungen in (3.31) gegebene Polyeder. Dann ist $\mathcal{P}$ die *Reduktion* eines supermodularen LTP durch den Vektor $\omega = (\omega_1, \ldots, \omega_n)^t$; vergleiche hierzu [Fuj 91, Abschnitt II.3.1].

Wir definieren eine Hilfsabbildung h^ω auf 2^I, mit deren Hilfe wir $\mathcal{P}$ alternativ beschreiben werden. Es sei

$$h^\omega(S) \;:=\; \max_{T \subseteq S}\{h(T) \;+ \sum_{i \in S-T} E(P_i)\, \omega_i\} \qquad \text{für alle } S \subseteq I. \tag{3.32}$$

Dann gilt das folgende Lemma[10]:

Lemma 3.7

$$\mathcal{P} \;=\; \{x \in \mathbb{R}^n \mid \sum_{i \in S} E(P_i)\, x_i \geq h^\omega(S) \qquad \forall\, S \subseteq I\}.$$

Beweis. Wir haben die Äquivalenz von

[10] Das entsprechende Resultat für supermodulare Polyeder liefert [Fuj 91, Abschnitt II.3.1, Theorem 3.3].

(1) $\sum\limits_{i \in S} E(P_i)\, x_i \geq h^{\omega}(S) \qquad \forall\ S \subseteq I \qquad$ zu

(2) $\sum\limits_{i \in S} E(P_i)\, x_i \geq h(S) \qquad \forall\ S \subseteq I \qquad$ und $\qquad x_i \geq \omega_i \qquad \forall\ i \in I$

zu zeigen.

Gelte (1), dann ist wegen $h^{\omega} \geq h$ auch $\sum_{i \in S} E(P_i)\, x_i \geq h(S)$ für alle $S \subseteq I$. Weiter gilt $h^{\omega}(\{i\}) = \max\{h(\{i\}), E(P_i)\, \omega_i\}$ für alle $i \in I$. Daher ist auch $E(P_i)\, x_i \geq E(P_i)\, \omega_i$ für alle $i \in I$, also $x_i \geq \omega_i$ für alle $i \in I$.

Gelte umgekehrt (2). Dann folgt für beliebige $S \subseteq I$ und $T \subseteq S$

$$\begin{aligned} \sum_{i \in S} E(P_i)\, x_i &= \sum_{i \in T} E(P_i)\, x_i + \sum_{i \in S-T} E(P_i)\, x_i \\ &\geq h(T) + \sum_{i \in S-T} E(P_i)\, \omega_i . \end{aligned}$$

Also gilt auch

$$\sum_{i \in S} E(P_i)\, x_i \geq \max_{T \subseteq S} \{h(T) + \sum_{i \in S-T} E(P_i)\, \omega_i\} = h^{\omega}(S).$$

■

Weiter gilt[11]:

Lemma 3.8 *Die Funktion $h^{\omega} : 2^I \longrightarrow \mathbb{R}$ ist supermodular.*

Beweis. Es ist

$$h^{\omega}(A) + h^{\omega}(B) \leq h^{\omega}(A \cup B) + h^{\omega}(A \cap B) \qquad \forall\ A, B \subseteq I$$

zu zeigen. Seien dazu $A, B \subseteq I$ beliebig. Per Definition von h^{ω} gibt es Mengen $X \subseteq A$ und $Y \subseteq B$ mit $h^{\omega}(A) = h(X) + \sum_{i \in A-X} E(P_i)\, \omega_i$ und $h^{\omega}(B) = h(Y) + \sum_{i \in B-Y} E(P_i)\, \omega_i$.

Damit folgt wegen Supermodularität von h und per Definition von h^{ω}

$$\begin{aligned} h^{\omega}(A) + h^{\omega}(B) &= h(X) + h(Y) + \sum_{i \in A-X} E(P_i)\, \omega_i + \sum_{i \in B-Y} E(P_i)\, \omega_i \\ &\leq h(X \cup Y) + h(X \cap Y) + \sum_{i \in A-X} E(P_i)\, \omega_i + \sum_{i \in B-Y} E(P_i)\, \omega_i \\ &= h(X \cup Y) + \sum_{i \in (A \cup B)-(X \cup Y)} E(P_i)\, \omega_i \\ &\quad + h(X \cap Y) + \sum_{i \in (A \cap B)-(X \cap Y)} E(P_i)\, \omega_i \end{aligned}$$

[11] In [Lov 82, Theorem 2.5] ist ein allgemeines Resultat für die Submodularität der sogenannten *Konvolution* einer submodularen mit einer modularen Funktion angegeben. In diesem Sinne handelt es sich bei h^{ω} um die Konvolution der supermodularen Funktion h mit der modularen Funktion $\sum_{i \in S} E(P_i)\, \omega_i$.

$$\leq \; h^\omega(A \cup B) \; + \; h^\omega(A \cap B).$$

■

Mit Lemma 3.7 kann man das lineare Programm (3.31) in der Form

$$\min\{\kappa(x) \mid \sum_{i \in S} E(P_i)\, x_i \geq h^\omega(S) \qquad \forall\, S \subseteq I\}$$

schreiben. Aufgrund der Supermodularität von h^ω folgt dann unter Verwendung von Korollar 2.6:

Satz 3.6 *Es gelte o. B. d. A.*

$$\frac{w_1}{E(P_1)} \geq \frac{w_2}{E(P_2)} \geq \cdots \geq \frac{w_n}{E(P_n)},$$

dann ist eine Optimallösung für das lineare Programm (3.31) durch

$$x_k^{LP} \;=\; \frac{h^\omega(\{1,\ldots,k\}) - h^\omega(\{1,\ldots,k-1\})}{E(P_k)} \qquad \text{für alle } k \in I$$

gegeben.

Damit können wir das lineare Programm (3.31) mit einem Aufwand von $\mathcal{O}(n \log n)$ lösen, sofern die entsprechenden Werte für h^ω bekannt sind.

Ein Algorithmus zur Berechnung von h^ω

Es sei zunächst eine beliebige Menge $S \subseteq I$ gegeben. Wir möchten $h^\omega(S)$ berechnen. Wir werden zeigen, daß diese Berechnung in $\mathcal{O}(|S| \log |S|)$ möglich ist.

Es gilt

$$\begin{aligned} h^\omega(S) &= \max_{T \subseteq S}\{h(T) + \sum_{i \in S-T} E(P_i)\,\omega_i\} \\ &= \sum_{i \in S} E(P_i)\,\omega_i \;+\; \max_{T \subseteq S}\underbrace{\{h(T) - \sum_{i \in T} E(P_i)\,\omega_i\}}_{=:\,\widetilde{h}(T)}, \end{aligned}$$

so daß für festes S das Maximum der Funktion $\widetilde{h}$ auf allen Teilmengen von S gefunden werden muß. Wir orientieren uns zur Berechnung von $\max_{T \subseteq S} \widetilde{h}(T)$ an einem Algorithmus von M. Queyranne, der auf der Maximierung einer vergleichbaren Mengenfunktion beruht; vergleiche hierzu [Que 93, Abschnitt 5, 'Separation Algorithm'].

Bezeichne im folgenden S^* eine Teilmenge von S mit $\widetilde{h}(S^*) = \max_{T \subseteq S} \widetilde{h}(T)$, und sei o. B. d. A. S^* inklusionsminimal gewählt. Dann gilt [12]:

[12] Ein ähnliches Resultat liefert [Que 93, Lemma 5.1].

Lemma 3.9

$$k \in S^* \quad \Longleftrightarrow \quad \frac{1}{m}\sum_{i\in S^*} E(P_i) \;>\; \underbrace{\omega_k \;+\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\frac{\mathrm{Var}(P_k)-(E(P_k))^2}{E(P_k)}}_{=:\,\eta_k}.$$

Beweis. Gelte $k \in S^*$, dann muß aufgrund der Definition von S^* $\widetilde{h}(S^*) > \widetilde{h}(S^* - \{k\})$ gelten. Es ergibt sich nun

$$\begin{aligned}
\widetilde{h}(S^*) - \widetilde{h}(S^* - \{k\}) \;&=\; \frac{E(P_k)}{m}\sum_{i\in S^*-\{k\}} E(P_i) \;+\; \frac{1}{2m}(E(P_k))^2 \\
&\quad +\frac{1}{2}(E(P_k))^2 \;-\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\mathrm{Var}(P_k) \;-\; E(P_k)\,\omega_k \\
&=\; \frac{E(P_k)}{m}\sum_{i\in S^*} E(P_i) \\
&\quad -\bigg(E(P_k)\,\omega_k \;+\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\Big(\mathrm{Var}(P_k)-(E(P_k))^2\Big)\bigg) \\
&=\; E(P_k)\left(\frac{1}{m}\sum_{i\in S^*} E(P_i) \;-\; \eta_k\right) \\
&>\; 0.
\end{aligned}$$

Wegen $E(P_k) > 0$ folgt sofort $\frac{1}{m}\sum_{i\in S^*} E(P_i) \;>\; \eta_k$.

Gelte umgekehrt $\frac{1}{m}\sum_{i\in S^*} E(P_i) \;>\; \eta_k$ und angenommen, es sei $k \notin S^*$. Wegen

$$\begin{aligned}
\widetilde{h}(S^* \cup \{k\}) - \widetilde{h}(S^*) \;&=\; \frac{1}{2m}(E(P_k))^2 \;+\; \frac{E(P_k)}{m}\sum_{i\in S^*} E(P_i) \\
&\quad +\frac{1}{2}(E(P_k))^2 \;-\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\mathrm{Var}(P_k) \;-\; E(P_k)\,\omega_k \\
&=\; \frac{1}{m}(E(P_k))^2 \;+\; \frac{E(P_k)}{m}\sum_{i\in S^*} E(P_i) \\
&\quad -\bigg(E(P_k)\,\omega_k \;+\; \Big(\frac{1}{2}-\frac{1}{2m}\Big)\Big(\mathrm{Var}(P_k)-(E(P_k))^2\Big)\bigg) \\
&=\; \frac{1}{m}(E(P_k))^2 \;+ E(P_k)\left(\frac{1}{m}\sum_{i\in S^*} E(P_i) \;-\; \eta_k\right) \\
&>\; \frac{1}{m}(E(P_k))^2 \quad > 0
\end{aligned}$$

folgt $\widetilde{h}(S^* \cup \{k\}) > \widetilde{h}(S^*)$, was jedoch ein Widerspruch zur Definition von S^* ist. Damit folgt die Behauptung. ∎

Mit Lemma 3.9 können wir folgern:

Korollar 3.9 *Gilt $k \in S^*$ für ein $k \in S$, so folgt $l \in S^*$ für alle $l \in S$ mit $\eta_l \leq \eta_k$.*

Wir brauchen also alle $i \in S$ lediglich in aufsteigender Reihenfolge

$$\eta_{i_1} \leq \eta_{i_2} \leq \cdots \leq \eta_{i_{|S|}}$$

zu sortieren, und $\max_{T \subseteq S} \widetilde{h}(T)$ wird auf einer der Mengen $S_k \subseteq S$ mit $S_k := \{i_1, \ldots, i_k\}$ für $k = 0, \ldots, |S|$ angenommen. Dabei setzen wir $S_0 := \emptyset$.

Der folgende Algorithmus berechnet $\max_{T \subseteq S} \widetilde{h}(T)$ für beliebiges $S \subseteq I$.

Algorithmus $\widetilde{h}$

Input $S = \{1, \ldots, s\} \subseteq I$

$E(P_i)$, $\mathrm{Var}(P_i)$, ω_i und η_i für alle $i \in S$

Output $S^* \subseteq S$, sowie $\widetilde{h} = \widetilde{h}(S^*) = \max_{T \subseteq S} \widetilde{h}(T)$

Step 1 Sortiere die Elemente aus S gemäß

$\eta_{i_1} \leq \eta_{i_2} \leq \cdots \leq \eta_{i_s}$

$S^* \leftarrow \emptyset,\ S' \leftarrow \emptyset,\ \widetilde{h} \leftarrow 0,\ \widetilde{h}' \leftarrow 0,\ \sum \leftarrow 0$

Step k **do for** $k = 1, \ldots, s$**:**

$$\widetilde{h}' \leftarrow \widetilde{h}' + \frac{E(P_{i_k})}{m} \sum + \frac{1}{m}(E(P_{i_k}))^2 - \Big(E(P_{i_k})\,\omega_{i_k} + \Big(\frac{1}{2} - \frac{1}{2m}\Big)\Big(\mathrm{Var}(P_{i_k}) - (E(P_{i_k}))^2\Big)\Big)$$

$\sum \leftarrow \sum + E(P_{i_k}),\ S' \leftarrow S' \cup \{i_k\}$

if $\widetilde{h}' > \widetilde{h}$ **do** $\widetilde{h} \leftarrow \widetilde{h}'$, $S^* \leftarrow S'$ **od**

fi

od

Zur Korrektheit und Komplexität des Algorithmus:

Korrektheit: Wegen $\widetilde{h}(\emptyset) = 0$ ist die Initialisierung trivialerweise korrekt. In jedem der Schritte $k = 1, \ldots, s$ wird ausgehend von $\widetilde{h}' = \widetilde{h}(S_{i_{k-1}})$ der Wert für $\widetilde{h}'$ auf $\widetilde{h}' = \widetilde{h}(S_{i_k}) = \widetilde{h}(S_{i_{k-1}} \cup \{i_k\})$ gesetzt. Vergleiche zur rekursiven Setzung von $\widetilde{h}'$ die Differenz $\widetilde{h}(S^* \cup \{k\}) - \widetilde{h}(S^*)$ im Beweis zu Lemma 3.9.

Genau dann, wenn $\widetilde{h}' = \widetilde{h}(S_{i_k})$ den bisher größten Wert von $\widetilde{h}(\emptyset), \widetilde{h}(S_{i_1}), \ldots, \widetilde{h}(S_{i_{k-1}})$ übertrifft, werden die Werte $\widetilde{h}$ bzw. S^* entsprechend aktualisiert.

Die Ausgabe ist also $\widetilde{h} = \max_{k=0,\ldots,s} \widetilde{h}(S_{i_k})$ zusammen mit der zugehörigen Menge $S^* = S_{i_l}$ mit $\widetilde{h}(S_{i_l}) = \max_{k=0,\ldots,s} \widetilde{h}(S_{i_k})$. Zusammen mit Korollar 3.9 folgt sofort, daß damit bereits $\widetilde{h} = \widetilde{h}(S^*) = \max_{T \subseteq S} \widetilde{h}(T)$ gilt.

Komplexität: Da jeder der Schritte $k = 1, \dots, s$ lediglich lineare Zeit erfordert, wird die Gesamtkomplexität durch die Sortierung der Parameter η_i dominiert und beträgt folglich $\mathcal{O}(s \log s)$.

Wir können also $\max_{T \subseteq S} \widetilde{h}(T)$ und damit auch

$$h^\omega(S) = \sum_{i \in S} E(P_i)\,\omega_i \;+\; \max_{T \subseteq S} \widetilde{h}(T)$$

mit einem Aufwand von $\mathcal{O}(|S| \log |S|)$ für beliebiges $S \subseteq I$ berechnen.

Da wir zur Berechnung der Optimallösung für das lineare Programm (3.31) letztlich die Werte

$$h^\omega(\{1, \dots, k\}) \qquad \text{für alle } k = 1, \dots, n$$

benötigen (vergleiche Satz 3.6), bietet sich das folgende Vorgehen zur Berechnung sämtlicher $h^\omega(\{1, \dots, k\})$ an:

Algorithmus h^ω

Input $I = \{1, \dots, n\}$, $E(P_i)$, $\mathrm{Var}(P_i)$, ω_i und η_i für alle $i \in I$

Output $h^\omega(S^k)$ für $S^k = \{1, \dots, k\}$ und alle $k = 1, \dots, n$

Step 1 $\sum \leftarrow 0$, $S \leftarrow \emptyset$, Liste $L \leftarrow \emptyset$

Step k **do for** $k = 1, \dots, n$**:**

(1) $S \leftarrow S \cup \{k\}$, $\sum \leftarrow \sum + E(P_k)\,\omega_k$,

Ordne k gemäß aufsteigender η_k in die Liste L ein

(2) Berechne mit Algorithmus $\widetilde{h}$ $\max_{T \subseteq S} \widetilde{h}(T)$,

Dabei wird Liste L an Algorithmus $\widetilde{h}$ übergeben

(3) $h^\omega(S^k) \leftarrow \sum + \max_{T \subseteq S} \widetilde{h}(T)$

od

Der Aufwand der Schritte (1) und (2) ist nun durch $\mathcal{O}(n)$ gegeben, denn die Sortierung der η_i in Algorithmus $\widetilde{h}$ entfällt. Schritt (3) erfordert lediglich einen Aufwand von $\mathcal{O}(1)$. Damit ist die Berechnung aller $h^\omega(\{1, \dots, k\})$ in insgesamt $\mathcal{O}(n^2)$ möglich.

Man kann durch „günstigere“ Initialisierung der Eingabe für Algorithmus $\widetilde{h}$ in Schritt (2) die Laufzeit günstig beeinflussen, wir wollen hier jedoch nicht weiter darauf eingehen.

Wir halten schließlich unter Beachtung von Satz 3.6 fest:

Satz 3.7 *Die Lösung des linearen Programms (3.31) ist unter Benutzung von Algorithmus h^ω und wegen Satz 3.6 mit einem Aufwand von insgesamt $\mathcal{O}(n^2)$ möglich.*

3.5 Zusammenfassung der Resultate

In der folgenden Tabelle fassen wir die Resultate der untersuchten Heuristiken zusammen. Das zugrundeliegende Modell ist

$$((I, \prec), \mathcal{N} = \{S \subseteq I \mid |S| = m+1\}, P, \sum_{i \in I} w_i C_i)$$

mit $P = \bigotimes_{i \in I} P_i$, also stochastisch unabhängigen Verteilungen der Dauern und $m \in \mathbb{N}$. In der Tabelle schreiben wir r für die Berücksichtigung nichttrivialer *release dates* und $\prec$ für die Berücksichtigung einer Präzedenzrelation.

Es bezeichnet n die Anzahl der Vorgänge und m die Anzahl der Maschinen. Die aufgeführten Werte sind sowohl *worst case*-Garantien für die Heuristiken, als auch Gütegarantien der unteren Schranken, die die Optimallösungen der jeweiligen **LP**-Relaxierungen für die Schedulingprobleme liefern. Sofern ein polynomialer Algorithmus zur Lösung der **LP**-Relaxierungen vorliegt, ist dessen Aufwand angegeben.

	$m = 1$		$m \in \mathbb{N}$	
	1	$\mathcal{O}(n \log n)$ (*ratio rule*)	$2 - \frac{1}{m}$ *† $3 - \frac{1}{m}$ *	$\mathcal{O}(n \log n)$ $\mathcal{O}(n^2)$
$\prec$	$2 - \frac{2}{n+1}$	-	-	-
r	$3 - \frac{2}{n+1}$	$\mathcal{O}(n^2)$	$4 - \frac{1}{m}$ *	$\mathcal{O}(n^2)$
$r, \prec$	$3 - \frac{2}{n+1}$	-	-	-

*Falls $\mathrm{Var}(P_i) \leq (E(P_i))^2$ für alle $i \in I$ gilt.

†Dieser Wert ist keine Gütegarantie für die untere Schranke, die das zugrundeliegende lineare Programm für das Schedulingproblem liefert.

Tabelle 3.1: Zusammenfassung der Resultate des dritten Kapitels.

Symbolverzeichnis

$\mathbb{N}$	Menge der natürlichen Zahlen (ausschließlich 0)
$\mathbb{R}^n_+$	Menge aller Vektoren $(x_1, \ldots, x_n)^t \in \mathbb{R}^n$ mit $x_i \geq 0 \quad \forall \quad i = 1, \ldots, n$
x^t	Transposition eines Vektors $x \in \mathbb{R}^n$
$\mathrm{conv}(E)$	Konvexe Hülle einer Menge $E \subseteq \mathbb{R}^n$
$\mathrm{cone}(E)$	Konische Hülle einer Menge $E \subseteq \mathbb{R}^n$
$\mathcal{P}$	Polyeder; $\mathcal{P} \subseteq \mathbb{R}^n$
$\mathcal{P}_{\geq}(A, f)$	Erweitertes Contra-Polymatroid zu A und f; $\mathcal{P}_{\geq}(A, f) \subseteq \mathbb{R}^n$
$\mathcal{B}_{\geq}(A, f)$	Basispolytop eines erweiterten Contra-Polymatroids; $\mathcal{B}_{\geq}(A, f) \subseteq \mathcal{P}_{\geq}(A, f)$
I	Endliche Menge von Vorgängen eines Schedulingproblems; $I = \{1, \ldots, n\}$
2^I	Potenzmenge von I
π	Permutation auf I
$\prec$	Präzedenzrelation auf der Menge der Vorgänge; $\prec \subset I \times I$
$\mathcal{N}$	Familie der verbotenen Mengen; $\mathcal{N} \subset 2^I$
P	Verteilung für die Vorgangsdauern; $P = (P_1, \ldots, P_n)$
p	Realisierung für die Vorgangsdauern; $p \in \mathbb{R}^n_+$
κ	Kostenfunktion; $\kappa : \mathbb{R}^n_+ \longrightarrow \mathbb{R}_+$
g	Kostenrate einer additiven Kostenfunktion; $g : 2^I \longrightarrow \mathbb{R}_+$
Π	Strategie für ein stochastisches Schedulingproblem; $\Pi : \mathbb{R}^n_+ \longrightarrow \mathbb{R}^n_+$
L_π	Statische Prioritätsstrategie gemäß Prioritätsliste $\pi_1 < \pi_2 < \cdots < \pi_n$
s_i	Startzeit von Vorgang i; $s_i \in \mathbb{R}_+$
C_i	Fertigstellungszeit von Vorgang i; $C_i \in \mathbb{R}_+$
$\mathcal{S}^\Pi$	Verteilung für die Startzeiten der Vorgänge unter Strategie Π
$\mathcal{C}^\Pi$	Verteilung für die Fertigstellungszeiten der Vorgänge unter Strategie Π
κ^Π	Verteilung für die Projektkosten unter Strategie Π
ρ	Optimalwert eines stochastischen Schedulingproblems; $\rho = \inf_\Pi E(\kappa^\Pi)$
$E(X)$	Erwartungswert einer Zufallsvariablen X
$\mathrm{Var}(X)$	Varianz einer Zufallsvariablen X

Literaturverzeichnis

[BGT 92] P. P. BHATTACHARYA, L. GEORGIADIS AND P. TSOUCAS, *Extended polymatroids: Properties and optimization*, Integer Programming and Combinatorial Optimization (E. Balas, G. Cornuéjols and R. Kannan, eds.), **1992**, Proc. of the 2nd International IPCO Conference, pp. 298-315

[BN-M 93] D. BERTSIMAS AND J. NIÑO-MORA, *Conservation laws, extended polymatroids and multi-armed bandit problems: A unified approach to indexable systems*, Integer Programming and Combinatorial Optimization (G. Rinaldi and L. A. Wolsey, eds.), **1993**, Proc. of the 3rd International IPCO Conference, pp. 355-384

[CPS+ 96] S. CHAKRABARTI, C. PHILLIPS, A. S. SCHULZ, D. B. SHMOYS, C. STEIN AND J. WEIN *Improved scheduling algorithms for minsum criteria*, in: Automata, Languages and Programming (F. Meyer auf der Heide and B. Monien, eds.), Lecture Notes in Computer Science, vol. 1099, Springer, Berlin, **1996**, Proc. of ICALP '96

[Edm 70] J. EDMONDS, *Submodular functions, matroids and certain polyhedra*, in: Proc. of the International Conference on Combinatorics, Calgary, **1970**, pp. 69-87

[Fuj 91] S. FUJISHIGE, *Submodular functions and optimization*, Annals of Discrete Mathematics, vol. 47, North-Holland, Amsterdam, **1991**

[GG 95] R. GARBE AND K. D. GLAZEBROOK, *Submodular returns and greedy heuristics for queueing scheduling problems*, Preprint, Department of Mathematics and Statistics, Newcastle University, Newcastle upon Tyne, U. K., **1995**

[GJ 79] M. R. GAREY AND D. S. JOHNSON, *Computers and intractability – A guide to the theory of $\mathcal{NP}$-completeness*, W. H. Freeman & Co., New York, **1979**

[GT 89] D. GOLDBERG AND M. J. TODD, *Linear programming*, in: Optimization (G. L. Nemhauser, A. H. G. Rinnooy Kan and M. J. Todd, eds.), Handbooks in Operations Research and Management Science, vol. 1, North-Holland, Amsterdam, **1989**

[HSSW 96] L. A. HALL, A. S. SCHULZ, D. B. SHMOYS AND J. WEIN, *Scheduling to minimize average completion time: Off-line and on-line approximation algorithms*, Preprint No. 516/1996, Department of Mathematics, Technical University of Berlin, Berlin, Germany, **1996**

[Jac 55] J. R. JACKSON, *Scheduling a production line to minimize maximum tardiness*, Management Science Research Project, University of California - Los Angeles, Research Report, vol. 43, **1955**

[Joh 54] S. M. JOHNSON, *Optimal two- and three-stage production schedules with setup times included*, Naval Research and Logistics Quarterly, vol. 1, **1954**, pp. 61-68

[Käm 85] TH. KÄMPKE, *Optimalitätsaussagen für spezielle stochastische Schedulingprobleme*, Dissertation, RWTH Aachen, **1985**

[KK 86] T. KAWAGUCHI AND S. KYAN, *Worst case bound on an LRF schedule for the mean weighted flow-time problem*, SIAM Journal on Computing, vol. 15, **1986**, pp. 1119-1129

[Law 82] E. L. LAWLER, *Recent results in the theory of machine scheduling*, in: Mathematical Programming: The State of the Art - Bonn 1982 (A. Bachem, M. Grötschel and B. Korte, eds.), Springer, **1983**, pp. 202-234

[Lov 82] L. LOVÁSZ, *Submodular functions and convexity*, in: Mathematical Programming: The State of the Art - Bonn 1982 (A. Bachem, M. Grötschel and B. Korte, eds.), Springer, **1983**, pp. 235-257

[LRKB 77] J. K. LENSTRA, A. H. G. RINNOOY KAN AND P. BRUCKER, *Complexity of machine scheduling problems*, Annals of Discrete Mathematics, vol. 1, **1977**, pp. 343-362

[MRW 84] R. H. MÖHRING, F. J. RADERMACHER AND G. WEISS, *Stochastic scheduling problems I – General strategies*, ZOR - Zeitschrift für Operations Research, vol. 28, **1984**, pp. 193-260

[MRW 85] R. H. MÖHRING, F. J. RADERMACHER AND G. WEISS, *Stochastic scheduling problems II – Set strategies*, ZOR - Zeitschrift für Operations Research, vol. 29, **1985**, pp. 65-104

[NW 88] G. L. NEMHAUSER AND A. L. WOLSEY, *Linear and integer programming*, John Wiley and Sons, New York, **1988**

[Que 93] M. QUEYRANNE, *Structure of a simple scheduling polyhedron*, Mathematical Programming, vol. 58, **1993**, pp. 263-285

[QS 94] M. QUEYRANNE AND A. S. SCHULZ, *Polyhedral approaches to machine scheduling*, Preprint No. 408/1994, Department of Mathematics, Technical University of Berlin, Berlin, Germany, **1994**

[RSZ 79] B. RAUHUT, N. SCHMITZ UND E.-W. ZACHOW, *Spieltheorie*, B. G. Teubner, Stuttgart, **1979**

[Ros 83] SH. M. ROSS, *Introduction to stochastic dynamic programming*, Series: Probability and Mathematical Statistics (Z. W. Birnbaum and E. Lukacs, eds.), Academic Press, New York, **1983**

[Ros 85] SH. M. ROSS, *Introduction to probability models*, 3rd Edition, Academic Press, New York, **1985**

[Sch 96] A. S. SCHULZ, *Scheduling to minimize total weighted completion time – Performance guarantees of LP-based heuristics and lower bounds*, in: Integer and Combinatorial Optimization (W. H. Cunningham, S. T. Mc Cormick and M. Queyranne, eds.), Lecture Notes in Computer Science, vol. 1084, Springer, Berlin, **1996**, Proc. of the 5th International IPCO Conference, pp. 430-440

[Smi 56] W. E. SMITH, *Various optimizers for single-stage production*, Naval Research and Logistics Quarterly, vol. 3, **1956**, pp. 59-66

[Wei 82] G. WEISS, *Multiserver stochastic scheduling*, In: Deterministic and Stochastic Scheduling (M. A. H. Dempster, J. K. Lenstra and A. H. G. Rinnooy Kan, eds.), D. Reidel Publishing Company, Dordrecht, **1982**, pp. 157-179

[Wei 88] G. WEISS, *Branching bandit processes*, Probability in the Engineering and Informational Sciences, vol. 2, **1988**, pp. 269-278