

Projektplanung

Modelle, Methoden, Management

Bearbeitet von
Jürgen Zimmermann, Christoph Stark, Julia Rieck

2., überarb. u. erw. Auflage 2010. Taschenbuch. 349 S. Paperback

ISBN 978 3 642 11878 4

Format (B x L): 15,5 x 23,5 cm

Gewicht: 556 g

[Wirtschaft > Management > Projektmanagement, Zeitmanagement](#)

Zu [Inhaltsverzeichnis](#)

schnell und portofrei erhältlich bei

beck-shop.de
DIE FACHBUCHHANDLUNG

Die Online-Fachbuchhandlung beck-shop.de ist spezialisiert auf Fachbücher, insbesondere Recht, Steuern und Wirtschaft. Im Sortiment finden Sie alle Medien (Bücher, Zeitschriften, CDs, eBooks, etc.) aller Verlage. Ergänzt wird das Programm durch Services wie Neuerscheinungsdienst oder Zusammenstellungen von Büchern zu Sonderpreisen. Der Shop führt mehr als 8 Millionen Produkte.

Projektplanung unter Zeitrestriktionen

Für die Ausführung eines Vorgangs werden in der Regel Zeit und Ressourcen benötigt. In Abschnitt 1.4.2 haben wir zwei unterschiedliche Arten von Ressourcen kennengelernt, von denen wir in diesem Kapitel nur die erneuerbaren Ressourcen betrachten, wie z.B. Maschinen oder Mitarbeiter.¹ Wir nehmen an, dass die benötigten Ressourcen in „ausreichender“ Menge zur Verfügung stehen, d.h. der kumulierte Ressourcenbedarf aller simultan ausführbaren Vorgänge ist geringer als die vorhandene Ressourcenkapazität. Somit stellen die gegebenen Kapazitäten der benötigten Ressourcen keine Einschränkungen dar, müssen aber gegebenenfalls bei einer zielgerichteten Terminierung der Vorgänge berücksichtigt werden.

Wir betrachten im Folgenden das Problem der *Projektplanung unter Zeitrestriktionen*, wobei die Terminierung der Projektvorgänge unter Beachtung der durch einen MPM-Netzplan gegebenen Zeitbeziehungen erfolgt. Dabei gehen wir, wenn nichts anderes gesagt wird, immer von allgemeinen Zeitbeziehungen in Form von Mindest- und Höchstabständen vom Typ Start-Start aus. Aufgabe der Projektplanung unter Zeitrestriktionen ist die Terminierung der Vorgänge eines gegebenen Projektes, so dass ein vorgegebenes projektbezogenes Ziel (vgl. Abschnitt 1.3.3) bestmöglich erfüllt wird und gleichzeitig die Startzeitpunkte der Vorgänge die vorgegebenen Zeitbeziehungen einhalten.

In Abschnitt 2.1 stellen wir zunächst eine mathematische Modellformulierung des Projektplanungsproblems unter Zeitrestriktionen vor. In Abschnitt 2.1.1 formulieren wir dann für praxisrelevante Ziele eine geeignete mathematische Zielfunktion. Für lineare Zielfunktionen stellt das entsprechende Optimierungsproblem ein lineares Programm (LP) dar. Für nichtlineare Zielfunktion zeigen wir in den Abschnitten 2.1.2 und 2.1.3, wie sich das resultierende nichtlineare Optimierungsproblem in ein LP bzw. mit Hilfe einer zeitindexbasierten Formulierung in ein gemischt-ganzzahliges lineares Programm (MIP) überführen lässt. In Abschnitt 2.1.4 beschreiben wir den

¹ Auf die Betrachtung von nicht-erneuerbaren Ressourcen wird, aus den in Abschnitt 1.4.2 genannten Gründen, verzichtet.

zulässigen Bereich des Projektplanungsproblems unter Zeitrestriktionen. Im Anschluss geben wir für die behandelten Zielfunktionen so genannte ausgezeichnete Punkte des zulässigen Bereichs an, die potentielle Kandidaten für eine optimale Lösung darstellen (vgl. Abschnitt 2.1.5).

In Abschnitt 2.2 stellen wir exakte Lösungsverfahren vor. Diese bestimmen in Abhängigkeit von der betrachteten Zielfunktion unter den jeweiligen ausgezeichneten Punkten eine optimale Lösung. Da die Bestimmung einer optimalen Lösung für einzelne Zielfunktionen sehr aufwändig ist, diskutieren wir in Abschnitt 2.3 ein heuristisches Lösungsverfahren, das eine Näherungslösung für das jeweils zugrunde liegende Problem erzeugt. Abschließend wird in Abschnitt 2.4 ein Anwendungsbeispiel für die Projektplanung unter Zeitrestriktionen behandelt.

2.1 Problemformulierung

Bei der Terminierung der Vorgänge $i \in V$ eines Projektes unter Beachtung vorgegebener Mindest- und Höchstabstände sind den Vorgängen $i = 0, 1, \dots, n+1$ Startzeitpunkte S_i zuzuordnen, so dass für jeden Pfeil $\langle i, j \rangle \in E$ eine Zeitbeziehung der Form

$$S_j - S_i \geq \delta_{ij},$$

eingehalten wird. Gesucht wird ein Vektor von Startzeitpunkten $S = (S_0, S_1, \dots, S_{n+1})$, der eine vorgegebene Zielfunktion $f(S)$, $f : \mathbb{R}_{\geq 0}^{n+2} \rightarrow \mathbb{R}$, optimiert und alle durch den Projektnetzplan vorgegebenen Zeitbeziehungen einhält. Wie in Abschnitt 1.4.3 gehen wir davon aus, dass zur Einhaltung einer vorgegebenen maximalen Projektdauer \bar{d} ein Pfeil $\langle n+1, 0 \rangle$ mit der Bewertung $\delta_{n+1,0} = -\bar{d}$ in der Pfeilmenge E des Netzplans enthalten ist.

Definition 2.1 (Zeitzulässiger Schedule). Ein Vektor von Startzeitpunkten $S = (S_0, S_1, \dots, S_{n+1})$ mit $S_0 = 0$ und $S_i \in \mathbb{R}_{\geq 0}$ für alle Vorgänge $i \in V \setminus \{0\}$ eines Projektes wird *Schedule* genannt. Einen Schedule, der zusätzlich die Zeitbeziehungen $S_j - S_i \geq \delta_{ij}$ für alle $\langle i, j \rangle \in E$ erfüllt, bezeichnen wir als *zeitzulässig*.

Zur Darstellung von Schedules haben sich in der Literatur zur Projektplanung so genannte *Gantt-Charts* etabliert. Jeder reale Vorgang (Vorgang mit positiver Dauer) wird dabei als Rechteck über der Zeitachse dargestellt, wobei die Länge des Rechtecks der Dauer des jeweiligen Vorgangs entspricht. An der Lage des Rechtecks über der Zeitachse kann abgelesen werden, wann der zugehörige Vorgang beginnt bzw. endet. Fiktive Vorgänge wie der Projektstart, das Projektende und Meilensteine werden nicht in einen Gantt-Chart aufgenommen, da sie eine Dauer von 0 besitzen.

Beispiel 2.2. Wir betrachten das in Abbildung 2.1 dargestellte Projekt mit vier realen Vorgängen. Die Einhaltung der maximalen Projektdauer $\bar{d} = 6$

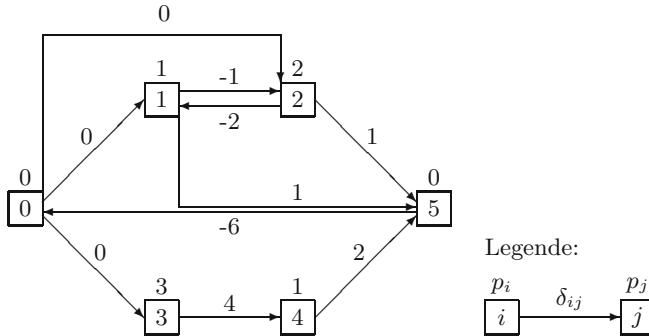


Abb. 2.1. Projektnetzplan mit vier realen Vorgängen

wird durch den Pfeil $\langle 5, 0 \rangle$ mit $\delta_{50} = -6$ im Projektnetzplan gewährleistet. Für den zeitzulässigen Schedule $S = (0, 0, 2, 0, 4, 6)$ erhalten wir den in Abbildung 2.2 dargestellten Gantt-Chart.

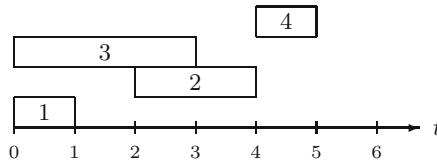


Abb. 2.2. Gantt-Chart für Schedule S

Gantt-Charts geben weder Auskunft über die Ressourceninanspruchnahme der einzelnen Projektvorgänge noch über den gesamten Ressourcenbedarf eines Projektes. Hierfür eignen sich so genannte *Ressourcenprofile*, die angeben, wie viele Einheiten jeder benötigten Ressource zu einem Zeitpunkt $t \in [0, \bar{d}]$ in Anspruch genommen werden.

Seien \mathcal{R} die Menge der erneuerbaren Ressourcen, die für die Projektdurchführung benötigt werden, und $r_{ik} \in \mathbb{Z}_{\geq 0}$ die *Ressourceninanspruchnahme* von Vorgang $i \in V$ an Ressource $k \in \mathcal{R}$, wobei wir davon ausgehen, dass die Ressourceninanspruchnahme r_{ik} während der gesamten Ausführungszeit p_i von Vorgang i konstant ist. Für fiktive Vorgänge $j \in V$ gilt $p_j := 0$ und $r_{jk} := 0$. Ein Vorgang i wird im Ressourcenprofil der Ressource k als Rechteck mit einer Breite von p_i Zeiteinheiten und einer Höhe von r_{ik} Ressourceneinheiten dargestellt. In Abbildung 2.3 skizzieren wir ein Ressourcenprofil für zwei Vorgänge, welches der fett eingezeichneten Treppenfunktion entspricht. Um zu gewährleisten, dass zwei direkt hintereinander ausführbare Vorgänge nicht um eine Ressource konkurrieren, nehmen wir an, dass ein Vorgang die von ihm benötigten Ressourcen zu seinem Startzeitpunkt in Anspruch nimmt und zu seinem Endzeitpunkt wieder freigibt. Man kann sich gewissermaßen vorstellen, dass Vorgang i zum Zeitpunkt S_j die von ihm belegten Ressourcen sofort

in der benötigten Höhe an Vorgang j übergebt und somit nur im Intervall $[S_i, S_i + p_i]$ in Ausführung ist. Dies ist in Abbildung 2.3 durch den leeren und den ausgefüllten Kreis an den Sprungstellen des Ressourcenprofils kenntlich gemacht, wobei der ausgefüllte Kreis den Funktionswert an einer Sprungstelle symbolisiert.

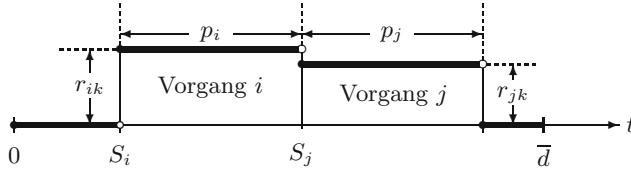


Abb. 2.3. Vorgänge mit Ressourcenbedarf an Ressource $k \in \mathcal{R}$

Für einen gegebenen Schedule S bezeichne

$$\mathcal{A}(S, t) := \{i \in V \mid S_i \leq t < S_i + p_i\}$$

die Menge der realen Vorgänge $i \in V$, die sich zum Zeitpunkt t in Ausführung befinden (auch *aktive Menge* genannt).² Ferner sei

$$r_k(S, t) := \sum_{i \in \mathcal{A}(S, t)} r_{ik}$$

die gesamte Menge an Ressource $k \in \mathcal{R}$, die zum Zeitpunkt t zur Ausführung aller Vorgänge $i \in \mathcal{A}(S, t)$ benötigt wird. Das Ressourcenprofil für einen gegebenen Schedule S und eine Ressource $k \in \mathcal{R}$ stellt somit eine rechtsseitig stetige Treppenfunktion $r_k(S, \cdot) : [0, \bar{d}] \rightarrow \mathbb{R}_{\geq 0}$ dar.

Beispiel 2.3. Wir betrachten das Projekt aus Abbildung 2.1 und nehmen an, dass zur Ausführung der einzelnen Vorgänge zwei erneuerbare Ressourcen benötigt werden. Abbildung 2.4 zeigt den entsprechend erweiterten Projektnetzplan, wobei unterhalb der Vorgänge die Ressourceninanspruchnahme des betreffenden Vorgangs an den Ressourcen 1 und 2 angegeben ist.

Für den zeitzulässigen Schedule $S = (0, 0, 2, 0, 4, 6)$ erhalten wir die in Abbildung 2.5 dargestellten Ressourcenprofile.

Nachfolgend geben wir eine allgemeine mathematische Formulierung des Projektplanungsproblems unter Zeitrestriktionen für eine noch zu spezifizierende Zielfunktion $f(S)$ an:

$$\left. \begin{array}{ll} \text{Minimiere } f(S) \\ \text{u.d.N.} & \begin{array}{ll} S_j - S_i \geq \delta_{ij} & (\langle i, j \rangle \in E) \\ S_0 = 0 & \\ S_i \geq 0 & (i \in V) \\ S_{n+1} \leq \bar{d}. & \end{array} \end{array} \right\} \quad (2.1)$$

² Die Menge $\mathcal{A}(S, t)$ enthält nur Vorgänge $i \in V$ mit positiver Dauer ($p_i > 0$), d.h. es sind keine fiktiven Vorgänge wie Projektstart, Projektende oder Meilensteine in $\mathcal{A}(S, t)$ enthalten.

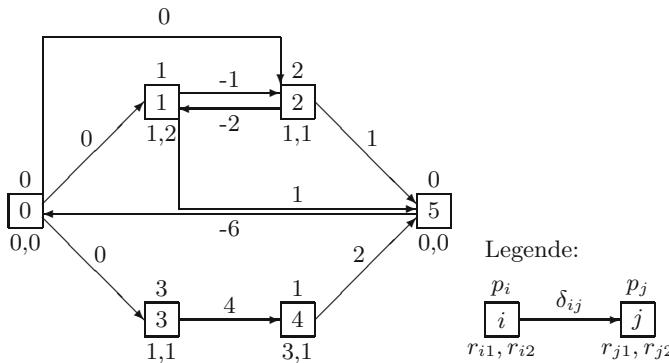
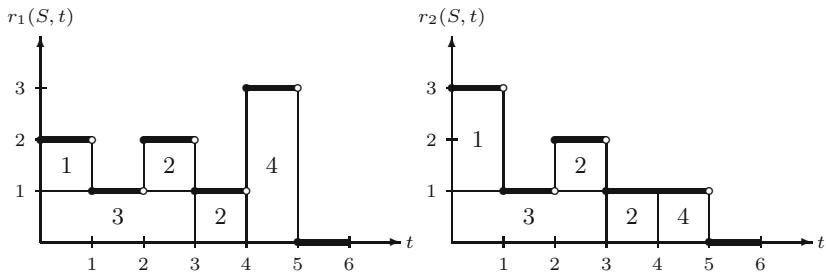


Abb. 2.4. Projektnetzplan mit zwei erneuerbaren Ressourcen

Abb. 2.5. Ressourcenprofile für Schedule S

Dabei nehmen wir o.B.d.A. an, dass die Zielfunktion $f(S)$ zu minimieren ist. Die Menge \mathcal{S}_T aller zeitzulässigen Schedules entspricht dem zulässigen Bereich des Projektplanungsproblems, den wir im Folgenden als *zeitzulässigen Bereich* bezeichnen. Die Nebenbedingungen von (2.1) gewährleisten, dass die vorgegebenen Zeitbeziehungen eingehalten werden, das Projekt zum Zeitpunkt 0 startet, die Startzeitpunkte aller Vorgänge nichtnegativ sind und die maximale Projektdauer \bar{d} nicht überschritten wird. Es sei angemerkt, dass die Bedingungen $S_i \geq 0$ für alle $i \in V$ redundant sind, da wir $S_0 := 0$ setzen und für einen wohldefinierten MPM-Netzplan gilt, dass kein Vorgang vor dem Projektstart begonnen werden darf (vgl. Vereinbarung 1.14). Ebenso ist die Nebenbedingung $S_{n+1} \leq \bar{d}$ redundant, weil $S_0 := 0$ gilt und wir davon ausgehen, dass ein Netzplan stets einen Pfeil $\langle n+1, 0 \rangle$ mit der Bewertung $\delta_{n+1,0} = -\bar{d}$ enthält.

Die Bestimmung eines beliebigen *zeitzulässigen Schedules* entspricht der Lösung des linearen Ungleichungssystems, das den zeitzulässigen Bereich von Problem (2.1) definiert. Es existiert genau dann mindestens ein *zeitzulässiger Schedule* für ein Projekt, wenn der zugehörige Projektnetzplan keinen Zyklus positiver Länge enthält, wovon wir im Folgenden stets ausgehen.

2.1.1 Zielfunktionen

In Abschnitt 1.3.3 haben wir eine Reihe praxisrelevanter Ziele für die Projektplanung vorgestellt. Im Folgenden erläutern wir zugehörige aus der Literatur bekannte Zielfunktionen $f : \mathbb{R}_{\geq 0}^{n+2} \rightarrow \mathbb{R}$. Alle betrachteten Zielfunktionen ordnen einem Schedule $S \in \mathcal{S}_T$ einen Zielfunktionswert $f(S) \in \mathbb{R}$ zu. Wir unterscheiden im Weiteren nach der Dimension (Maßeinheit) des Zielfunktionswertes zwischen zeitbezogenen, monetären und ressourcenabhängigen Zielen.

Zu den *zeitbezogenen Zielen* gehören die Minimierung der Projektdauer, die Minimierung der mittleren Durchlaufzeit sowie die Minimierung der Summe gewichteter Startzeitpunkte.

Ziel der Projektdauerminimierung ist es, die Projektvorgänge so zu terminieren, dass das Projektende S_{n+1} möglichst früh eintritt. Dieses Ziel wird oft bei kleinen Projekten gewählt, da dort im Allgemeinen die Zahlungsvereinbarungen vorsehen, dass ein großer Teil der Projektzahlungen erst zum Projektende geleistet wird (vgl. Abschnitt 1.2.1). Eine kurze Projektdauer führt somit zu einer Reduzierung der Kapitalbindungskosten. Weiterhin wird bei der Minimierung der Projektdauer die Gefahr einer Terminüberschreitung und damit verbundener Konventionalstrafen verringert. Die Zielfunktion der *Projektdauerminimierung* ergibt sich zu

$$f(S) := S_{n+1}. \quad (PD)$$

Bei der Minimierung der mittleren Durchlaufzeit (mean flow time) ist der Zeitpunkt zu minimieren, zu dem im Mittel alle Vorgänge beendet werden. Daher müssen alle Vorgänge $i \in V$ so früh wie möglich eingeplant werden. Die damit verbundene frühzeitige Auslastung der Ressourcenkapazitäten führt im Umkehrschluss dazu, dass die in Anspruch genommenen Ressourcen so früh wie möglich wieder für andere Projekte zur Verfügung stehen. Weiterhin besteht die Möglichkeit, dass Projekte, bei denen aufgrund von Störungen des Projektablaufs einzelne Vorgänge verzögert werden müssen, noch ohne Terminüberschreitungen ausgeführt werden können. Für die *Minimierung der mittleren Durchlaufzeit* ergibt sich die Zielfunktion

$$f(S) := \frac{1}{n+2} \sum_{i \in V} (S_i + p_i). \quad (MFT)$$

In praktischen Anwendungen ist z.B. aufgrund technischer Risiken (vgl. Abschnitt 1.2.4) häufig eine Teilmenge $V' \subseteq V$ von Vorgängen möglichst früh einzuplanen. Andere Vorgänge $V'' \subseteq V \setminus V'$ hingegen sollen möglichst spät durchgeführt werden, beispielsweise weil für Endprodukte, die im Rahmen solcher Vorgänge hergestellt werden, hohe Lagerungskosten anfallen. In einem solchen Fall betrachten wir das Ziel der *Minimierung der Summe gewichteter Startzeitpunkte* (weighted start times) aller Vorgänge. Jeder Vorgang $i \in V'$ erhält dabei ein positives Gewicht $w_i \in \mathbb{R}_{>0}$, jeder Vorgang $j \in V''$ ein

negatives Gewicht $w_j \in \mathbb{R}_{<0}$ und für alle Vorgänge $h \in V \setminus (V' \cup V'')$ setzen wir $w_h := 0$. Somit ergibt sich die Zielfunktion

$$f(S) := \sum_{i \in V} w_i S_i. \quad (WST)$$

Zu den *monetären Zielen* gehören die Minimierung der Kosten für die Unter- bzw. Überschreitungen vorgegebener Fälligkeitstermine, die Maximierung des Kapitalwertes, die Minimierung der Bereitstellungskosten für benötigte Ressourcen sowie die Minimierung der Kosten, die durch Abweichungen der kumulierten Ressourceninanspruchnahmen von einem vorgegebenen Ressourcenniveau entstehen.

Vorgeschrriebene Abschlusszeitpunkte einzelner Vorgänge $i \in V$ und vertraglich vereinbarte Fertigstellungstermine führen zu *Fälligkeitsterminen* d_i . Falls ein Vorgang $i \in V$ zu früh ausgeführt wird, d.h. vor d_i endet, können so genannte *Verfrühungskosten* c_i^E (earliness cost) entstehen, z.B. aufgrund zusätzlicher Kühlprozesse bei verderblichen Gütern. Wird ein Vorgang $i \in V$ zu spät ausgeführt, so entstehen *Verzögerungskosten* c_i^T (tardiness cost), beispielsweise durch Konventionalstrafen oder Ausfallkosten. Sind vorgegebene Fälligkeitstermine d_i für alle $i \in V$ weder zu unter- noch zu überschreiten, so ist die *Earliness-Tardiness-Zielfunktion*

$$f(S) := \sum_{i \in V} (c_i^E(d_i - S_i - p_i)^+ + c_i^T(S_i + p_i - d_i)^+) \quad (E + T)$$

zu betrachten. Dabei setzen wir für diejenigen Vorgänge j , für die keine Fälligkeitstermine d_j vorgegeben sind, $c_j^E := 0$, $c_j^T := 0$ und $d_j := 0$.

Die Maximierung des Kapitalwertes (net present value) entspricht dem vorrangigen unternehmerischen Ziel der Gewinnmaximierung. Der Kapitalwert eines Projektes berechnet sich als Summe der auf den Projektstart diskontierten Zahlungen, die durch den Projektverlauf ausgelöst werden. Nehmen wir an, dass eine Zahlung zum Zeitpunkt $t \in [0, \bar{d}]$ anfällt, dann ist der zugehörige *Diskontrate* $\beta^t = \frac{1}{(1+r)^t}$. Hierbei stellt der Kalkulationszinsfuß r eine exogene Größe dar, die sich am Marktzins oder an der branchenüblichen Rendite orientiert. Zur Bestimmung des Projektkapitalwertes ordnen wir dem Startzeitpunkt S_i jedes Vorgangs $i \in V$ eine Zahlung (Cashflow) $c_i^F \in \mathbb{Z}$ zu. Diese entspricht dem auf den Zeitpunkt S_i abgezinsten Saldo aller Ein- und Auszahlungen, die in Verbindung mit der Ausführung des Vorgangs i stehen. Führt die Durchführung eines Vorgangs i nicht zu einer Ein- oder Auszahlung, so setzen wir $c_i^F := 0$. Für einen gegebenen Schedule S ergibt sich der Projektkapitalwert zu $\sum_{i \in V} c_i^F \beta^{S_i}$. Da wir stets von einer zu minimierenden Zielfunktion $f(S)$ ausgehen, lautet die Zielfunktion für die *Kapitalwertmaximierung*

$$f(S) := - \sum_{i \in V} c_i^F \beta^{S_i}. \quad (NPV)$$

Insbesondere bei Projekten, für die beispielsweise neue Ressourcen beschafft oder vorhandene aufwändig transportiert werden müssen, ist es ein

vorrangiges Ziel, die Kosten für die Bereitstellung von Ressourceneinheiten (z.B. Baukräne, Container, Spezialmaschinen) zu minimieren. Um dies zu erreichen, ist der maximale Bedarf an Ressourceneinheiten (Ressourceninanspruchnahme) im Planungszeitraum $[0, \bar{d}]$ einer jeden Ressource $k \in \mathcal{R}$, d.h. $\max_{t \in [0, \bar{d}]} r_k(S, t)$, zu betrachten. Wir nehmen an, dass für jede benötigte Einheit der jeweiligen Ressourcen die gleichen Kosten anfallen, unabhängig davon, wie lange die Ressource im Planungszeitraum zum Einsatz kommt. Ist für jede Ressource $k \in \mathcal{R}$ ein *Beschaffungskostensatz* $c_k^P \in \mathbb{Z}_{\geq 0}$ (procurement cost) pro Mengeneinheit gegeben, so ergibt sich die Zielfunktion des so genannten *Ressourceninvestmentproblems* zu

$$f(S) := \sum_{k \in \mathcal{R}} c_k^P \max_{t \in [0, \bar{d}]} r_k(S, t). \quad (RI)$$

Ist für eine Ressource $k \in \mathcal{R}$ ein *Ressourceniveau* Y_k vorgegeben, bei dessen Überschreitung Kosten $c_k^D \in \mathbb{R}_{\geq 0}$ (deviation cost) pro Einheit anfallen, so gilt es die so genannten *Überschreitungskosten* zu minimieren. Das Ressourceniveau Y_k für Ressource k entspricht in der Praxis beispielsweise der tariflich vereinbarten Arbeitszeit oder der vorhandenen Maschinenkapazität. Überschreitungskosten entstehen, wenn die Ressourceninanspruchnahme $r_k(S, t)$ einer Ressource $k \in \mathcal{R}$ zu einem Zeitpunkt $t \in [0, \bar{d}]$ vom Ressourceniveau Y_k nach oben abweicht, d.h. es gilt $r_k(S, t) > Y_k$. Betrachten wir dazu das Ressourcenprofil in Abbildung 2.6. Für den gegebenen Schedule $S = (0, 0, 1, 2, 5)$ ist die Ressourceninanspruchnahme $r_k(S, t)$ für alle Zeitpunkte $t \in [1, 2[$ größer als das Ressourceniveau $Y_k = 2$, das als waagerechte Linie in das Ressourcenprofil eingezeichnet ist. Demzufolge fallen Überschreitungskosten an.

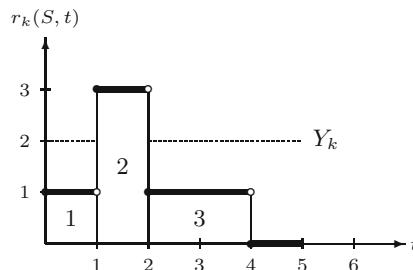


Abb. 2.6. Ressourcenprofil

Für das *Ressourcenabweichungsproblem* (resource deviation problem) erhalten wir die Zielfunktion gemäß

$$f(S) := \sum_{k \in \mathcal{R}} c_k^D \int_{t \in [0, \bar{d}]} [r_k(S, t) - Y_k]^+ dt. \quad (RD)$$

Zu den *ressourcenabhängigen Zielen* gehören Ziele der *Ressourcenivellierung* (resource levelling). Dabei ist die gleichmäßige Auslastung der Ressour-

cen $k \in \mathcal{R}$ im Zeitintervall $[0, \bar{d}]$ zu erreichen. Bei einigen industriellen Anlagen führt eine Änderung der Betriebsintensität der Anlage mitunter zu hohen Anpassungskosten, z.B. bei Hochöfen. Ferner führt ein Wechsel der jeweiligen Betriebsstufe einer Anlage häufig dazu, dass während der nachfolgenden Anlaufphase vermehrt Produkte minderer Qualität produziert werden und der Ausschuss somit zunimmt. Dabei haben die geschilderten Effekte einer solchen Intensitätsanpassung meist umso größere Auswirkungen, je größer der Wechsel in der Betriebsintensität ist. Aus diesen Gründen ist man in der Praxis häufig daran interessiert, ausgewählte Ressourcen gleichmäßig über die Zeit zu belasten und hohe Ressourceninanspruchnahmen stärker zu bestrafen als geringe. Als Zielfunktion für ein solches Problem der Ressourcennivellierung ergibt sich z.B.

$$f(S) := \sum_{k \in \mathcal{R}} \int_{t \in [0, \bar{d}]} r_k^2(S, t) dt. \quad (RL)$$

Beispiel 2.4. Betrachten wir den in Abbildung 2.7 dargestellten Projektnetzplan mit vier realen Vorgängen und einer erneuerbaren Ressource. Die Einhaltung der maximalen Projektdauer $\bar{d} = 6$ wird durch den Pfeil $\langle 5, 0 \rangle$ mit der Bewertung $\delta_{50} = -6$ im Projektnetzplan gewährleistet. Da wir nur eine Ressource betrachten, kann der Index für die Ressource $k \in \mathcal{R}$ in den folgenden Ausführungen entfallen. Für den zeitzulässigen Schedule $S = (0, 3, 5, 0, 4, 6)$ ergibt sich das in Abbildung 2.8 dargestellte Ressourcenprofil.

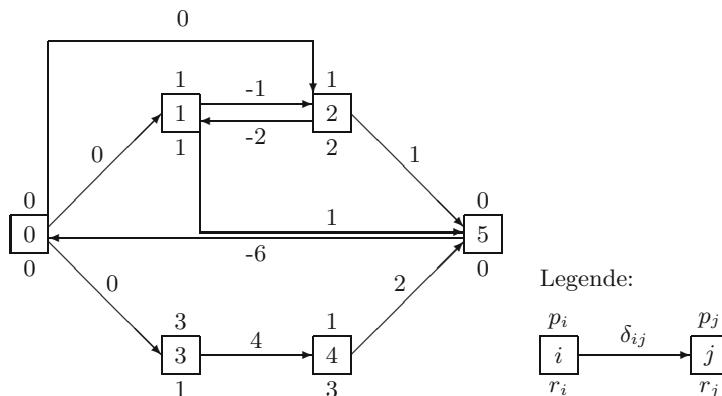
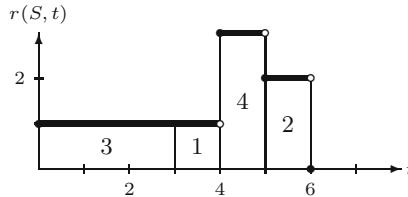


Abb. 2.7. Netzplan mit einer erneuerbaren Ressource

Die Zielfunktionswerte $f(S)$ für Schedule $S = (0, 3, 5, 0, 4, 6)$ und die eingeführten Zielfunktionen ergeben sich nun wie folgt. Für die Zielfunktion der Projektdauerminimierung (PD) gilt

$$f(S) := S_5 = 6$$

und für die Zielfunktion der Minimierung der mittleren Durchlaufzeit (MFT) erhalten wir

Abb. 2.8. Ressourcenprofil für Schedule $S = (0, 3, 5, 0, 4, 6)$

$$f(S) := \frac{1}{6}(0 + 4 + 6 + 3 + 5 + 6) = \frac{24}{6} = 4.$$

Die Zielfunktionswerte der übrigen Zielfunktionen mit den zugehörigen Parametern sind in Tabelle 2.1 angegeben.

Tabelle 2.1. Zielfunktionswerte für Schedule $S = (0, 3, 5, 0, 4, 6)$

Zielfunktion	Parameter	$f(S)$
(WST)	$w_i = 1$ für $i = 0, \dots, 5$	18
$(E + T)$	$d_0 = 0, d_1 = 5, d_2 = 5, d_3 = 3, d_4 = 5, d_5 = 6$ $c_0^E = c_0^T = 0, c_i^E = c_i^T = 1$ für $i = 1, \dots, 5$	2
(NPV)	$c_i^F = 1$ für $i = 0, \dots, 5; \beta = 0, 9$	-4,51
(RI)	$c^P = 5$	15
(RD)	$c^D = 1, Y = 1$	3
(RL)		17

In der Praxis hängt die verwendete Zielfunktion meist vom Planungshorizont des zugrunde liegenden Projektplanungsproblems ab (vgl. auch Abschnitt 1.3.3). Tabelle 2.2 zeigt eine geläufige Klassifizierung der Zielfunktionen in Abhängigkeit vom Planungshorizont (lang-, mittel- und kurzfristig).

Tabelle 2.2. Ziele in Abhängigkeit vom Planungshorizont

	langfristig	mittelfristig	kurzfristig
Planungshorizont	1 – 5 Jahre	1 – 12 Monate	1 – 30 Tage
Periodenlänge	1 Woche – 1 Monat	1 Tag – 1 Woche	1 Stunde – 1 Schicht
Zielfunktionen	(NPV)	$(RI), (RD),$ (RL)	$(PD), (MFT),$ $(WST), (E + T)$

Besitzt ein Projektnetzplan keinen Zyklus positiver Länge, so existiert, wie bereits gesagt, immer eine zeitzulässige Lösung für Problem (2.1). Da es sich bei den Zielfunktionen (PD) , (MFT) , (WST) , $(E + T)$, (NPV) , (RD)

und (RL) um stetige Funktionen handelt und der zeitzulässige Bereich \mathcal{S}_T abgeschlossen und beschränkt ist, existiert nach dem Satz von Weierstrass für diese Zielfunktionen immer auch eine optimale Lösung auf \mathcal{S}_T ; vgl. hierzu bspw. ZEIDLER (2003, S. 251). Des Weiteren ist die Zielfunktion (RI) stetig auf $\mathbb{R}_{\geq 0}^{n+2}$ bis auf alle Schedules S mit $S_j - S_i = p_i$ für mindestens ein Paar $i, j \in V$. An diesen potentiellen Unstetigkeitsstellen gilt aber immer, dass $f(S) \leq f(S')$ für alle Schedules S' , die in einer offenen ε -Umgebung von S liegen, d.h. die Funktion ist *nach unten halbstetig*. Betrachten wir dazu das Ressourcenprofil in Abbildung 2.9. Wird Vorgang 2 ausgehend von seinem $ES_2 = 2$ verzögert, so ist zum Startzeitpunkt $S_2 = 3$ für die Vorgänge 2 und 3 die Bedingung $S_3 - S_2 = p_2$ erfüllt. Des Weiteren gilt für $S_2 = 4$ die Bedingung $S_2 - S_1 = p_1$ und für $S_2 = 6$ die Bedingung $S_2 - S_3 = p_3$. Die resultierende Ressourceninvestition-Zielfunktion (RI) besitzt an den Stellen $S_2 = 3$ und $S_2 = 6$ Unstetigkeiten, es gilt aber $f(S) \leq f(S')$ für alle S' aus einer offenen ε -Umgebung von S (vgl. Abb. 2.9).

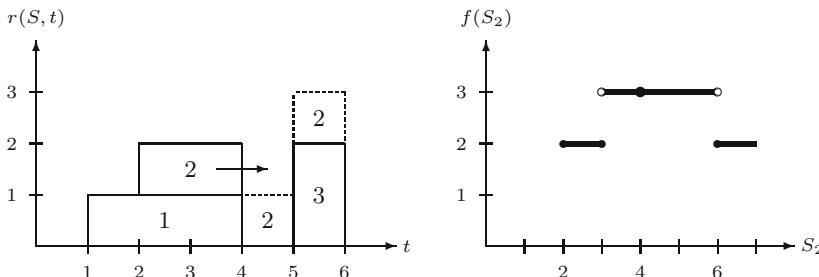


Abb. 2.9. Nach unten halbstetige Funktion (RI)

Allgemein sind nach unten halbstetige Funktionen wie folgt definiert.

Definition 2.5 (Nach unten halbstetige Funktion). Eine Funktion $f : \mathbb{R}_{\geq 0}^{n+2} \rightarrow \mathbb{R}$ wird *nach unten halbstetig* genannt, wenn für alle $S, S' \in \mathbb{R}_{\geq 0}^{n+2}$ gilt, dass $f(S) \leq \liminf_{S' \rightarrow S} f(S')$.

Genau wie für stetige Funktionen gilt auch für nach unten halbstetige Funktionen, dass sie auf einem abgeschlossenen und beschränkten Bereich ihr Minimum annehmen; vgl. hierzu bspw. HEUSER (1998, S. 242).

Korollar 2.6. Für Problem (2.1) existiert für alle stetigen bzw. nach unten halbstetigen Zielfunktionen im Fall $\mathcal{S}_T \neq \emptyset$ eine optimale Lösung.

In den beiden folgenden Abschnitten geben wir an, wie sich Problem (2.1) für die nichtlinearen Zielfunktionen $(E + T)$, (NPV) , (RI) , (RD) und (RL) als lineares Programm (LP) oder gemischt-ganzzahliges lineares Programm (MIP) formulieren und somit prinzipiell mit Hilfe geeigneter Standardsoftware (bspw. CPLEX oder Excel Solver) lösen lässt.³

³ Eine Einführung in die lineare Programmierung findet man z.B. in DOMSCHKE und DREXL (2005) oder NEUMANN und MORLOCK (2002).

2.1.2 Lineare Modelle

Für die Zielfunktionen (PD), (MFT) und (WST) stellt Problem (2.1) ein lineares Programm dar. Ein optimaler Schedule kann für diese Probleme bei ganzzahligen Eingabedaten in polynomialer Zeit bestimmt werden.⁴ In diesem Abschnitt zeigen wir, dass auch das Earliness-Tardiness-Problem und das Kapitalwertmaximierungsproblem in lineare Optimierungsprobleme (LP) transformiert werden können und damit ebenfalls effizient lösbar sind.

Beim Earliness-Tardiness-Problem, d.h. Problem (2.1) mit Zielfunktion ($E + T$), handelt es sich um ein nichtlineares Optimierungsproblem mit linearen Nebenbedingungen und nichtlinearer Zielfunktion. Die *Earliness-Tardiness-Zielfunktion*

$$f(S) := \sum_{i \in V} (c_i^E(d_i - S_i - p_i)^+ + c_i^T(S_i + p_i - d_i)^+)$$

kann durch die Einführung von Hilfsvariablen $e_i \geq 0$ für die Verfrühung und $l_i \geq 0$ für die Verspätung eines Vorgangs $i \in V$ linearisiert werden. Unter Verwendung dieser Hilfsvariablen können wir das folgende lineare Programm formulieren

$$\begin{aligned} \text{Minimiere} \quad & \sum_{i \in V} (c_i^E e_i + c_i^T l_i) \\ \text{u.d.N.} \quad & \begin{aligned} e_i &\geq d_i - S_i - p_i & (i \in V) \\ l_i &\geq S_i + p_i - d_i & (i \in V) \\ e_i &\geq 0 & (i \in V) \\ l_i &\geq 0 & (i \in V) \\ S_j - S_i &\geq \delta_{ij} & (\langle i, j \rangle \in E) \\ S_0 &= 0. \end{aligned} \end{aligned}$$

Die beschriebene Linearisierung führt zu einer Zunahme der Anzahl von Entscheidungsvariablen und Nebenbedingungen. Da für jeden Vorgang $i \in V$ zusätzliche Variablen e_i und l_i eingeführt wurden, erhöht sich die Anzahl der Entscheidungsvariablen von $|V| = n$ auf $3|V| = 3n + 6$. Die Anzahl der Nebenbedingungen erhöht sich um $4|V|$. Das folgende Beispiel veranschaulicht die Linearisierung des Earliness-Tardiness-Problems anhand einer Probleminstanz.

⁴ Ein Optimierungsproblem heißt in der Komplexitätstheorie „in polynomialer Zeit lösbar“, wenn ein Lösungsalgorithmus für das betrachtete Problem existiert, für den der zeitliche Aufwand zur Lösung des Problems durch ein Polynom in Abhängigkeit von der Problemgröße nach oben beschränkt ist. Unter Optimierungsgesichtspunkten werden Probleme, die sich in polynomialer Zeit lösen lassen, als „einfache“ Probleme erachtet. Für eine Einführung in die Komplexitätstheorie verweisen wir beispielsweise auf BACHEM (1980). Eine umfassende Darstellung des Themas findet sich in GAREY und JOHNSON (1979).

Beispiel 2.7. Wir betrachten den in Abbildung 2.10 dargestellten Projektnetzplan mit einer maximalen Projektdauer von $\bar{d} = 6$. Die Knoten des Netzplans sind mit den Dauern p_i und den Fälligkeitsterminen d_i der einzelnen Vorgänge $i \in V$ bewertet.

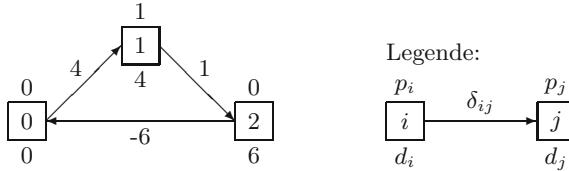


Abb. 2.10. Netzplan

Für die Vorgänge $i = 1, 2$ sollen die vorgegebenen Fälligkeitstermine d_i weder unter- noch überschritten werden. Die Verfrühungs- und Verspätungskosten seien $c_0^E = c_0^T := 0$ und $c_i^E = c_i^T := 1$ für $i = 1, 2$. Mit Hilfe dieser Parameter lässt sich die Probleminstanz aus Abbildung 2.10 wie folgt als Earliness-Tardiness-Problem formulieren

$$\begin{aligned} \text{Minimiere } & f(S) := (3 - S_1)^+ + (S_1 - 3)^+ + (6 - S_2)^+ + (S_2 - 6)^+ \\ \text{u.d.N. } & S_1 - S_0 \geq 4 \\ & S_2 - S_1 \geq 1 \\ & S_0 - S_2 \geq -6 \\ & S_0 = 0. \end{aligned}$$

Durch Einführung der Hilfsvariablen e_i und l_i für $i = 0, 1, 2$ ergibt sich daraus das folgende LP

$$\begin{aligned} \text{Minimiere } & f(S) := e_1 + l_1 + e_2 + l_2 \\ \text{u.d.N. } & e_1 \geq 3 - S_1 \\ & e_2 \geq 6 - S_2 \\ & l_1 \geq S_1 - 3 \\ & l_2 \geq S_2 - 6 \\ & S_1 - S_0 \geq 4 \\ & S_2 - S_1 \geq 1 \\ & S_0 - S_2 \geq -6 \\ & S_0 = 0 \\ & e_i \geq 0 \quad (i = 0, 1, 2) \\ & l_i \geq 0 \quad (i = 0, 1, 2). \end{aligned}$$

Anstelle von 3 Entscheidungsvariablen benötigen wir für die LP-Formulierung 9 Entscheidungsvariablen und anstelle von 4 Nebenbedingungen ergeben sich 14 Nebenbedingungen, wobei wir die beiden redundanten Restriktionen $e_0 \geq 0 - S_0$ und $l_0 \geq S_0 - 0$ vernachlässigt haben. Der optimale Zielfunktionswert wird für $e_0 \geq 0, e_1 = 0, e_2 = 0, l_0 \geq 0, l_1 = 1, l_2 = 0$ bzw. $S = (0, 4, 6)$ angenommen.

Beim Problem der *Kapitalwertmaximierung* mit der Zielfunktion (*NPV*)

$$\left. \begin{array}{l} \text{Maximiere } f(S) := \sum_{i \in V} c_i^F \beta^{S_i} \\ \text{u.d.N.} \quad \begin{array}{l} S_j - S_i \geq \delta_{ij} \\ S_0 = 0 \end{array} \quad \left. \begin{array}{l} (\langle i, j \rangle \in E) \end{array} \right. \end{array} \right\} \quad (2.2)$$

handelt es sich ebenfalls um ein Optimierungsproblem mit nichtlinearer Zielfunktion und linearen Nebenbedingungen. Für den Diskontfaktor $\beta^{S_i} = \frac{1}{(1+r)^{S_i}}$ wird in der Literatur häufig auch die Darstellung $e^{-\alpha S_i}$ verwendet. Bei gegebenem Kalkulationszinsfuß $r \geq 0$ kann α durch $\alpha = \ln(1+r)$ bestimmt werden. Problem (2.2) lässt sich in ein lineares Programm transformieren, indem wir $y_i := e^{-\alpha S_i}$ und $K_{ij} := e^{\alpha \delta_{ij}}$ setzen. Dabei ist $y_i = e^{-\alpha S_i}$ äquivalent zu $S_i = -\frac{\ln(y_i)}{\alpha}$. Aus unserer Vereinbarung, dass das Projekt immer zum Zeitpunkt 0 gestartet wird, folgt

$$S_0 = 0 \Leftrightarrow -\frac{\ln(y_0)}{\alpha} = 0 \Leftrightarrow y_0 = 1.$$

Da die natürliche Logarithmusfunktion monoton wachsend ist, können die Zeitbeziehungen wie folgt transformiert werden:

$$\begin{aligned} S_j - S_i &\geq \delta_{ij} \\ \Leftrightarrow -\frac{\ln(y_j)}{\alpha} + \frac{\ln(y_i)}{\alpha} &\geq \delta_{ij} \\ \Leftrightarrow \ln(y_i) - \ln(y_j) &\geq \alpha \delta_{ij} = \ln(K_{ij}) \\ \Leftrightarrow \ln\left(\frac{y_i}{y_j}\right) &\geq \ln(K_{ij}) \\ \Leftrightarrow \frac{y_i}{y_j} &\geq K_{ij} \\ \Leftrightarrow 0 &\geq y_j K_{ij} - y_i. \end{aligned}$$

Insgesamt erhalten wir das lineare Programm⁵

$$\left. \begin{array}{l} \text{Maximiere } \sum_{i \in V} c_i^F y_i \\ \text{u.d.N.} \quad \begin{array}{l} y_j K_{ij} - y_i \leq 0 \quad (\langle i, j \rangle \in E) \\ y_0 = 1. \end{array} \end{array} \right.$$

⁵ Anders als bei Earliness-Tardiness-Problemen führt die Linearisierung bei der Kapitalwertmaximierung nicht zu zusätzlichen Entscheidungsvariablen und Nebenbedingungen. Aber auch wenn die ursprünglichen Zielfunktionskoeffizienten c_i^F und die Zeitabstände δ_{ij} alle ganzzahlig sind, ergibt sich durch die Substitution eine Koeffizientenmatrix mit Einträgen, die weder ganzzahlig noch rational sind, so dass das entsprechende LP i.d.R. nicht mehr in polynomialer Zeit lösbar ist.

2.1.3 Zeitindexbasierte Modelle

Projektplanungsprobleme mit den Zielfunktionen (RI), (RD) und (RL) lassen sich nicht als lineare Programme formulieren. Mit Hilfe einer geeigneten Diskretisierung der Zeit lassen sie sich aber zumindest als gemischt-ganzzahlige lineare Programme (MIP) beschreiben PRITSKER ET AL. (1969). Dazu nehmen wir an, dass der Planungszeitraum $[0, \bar{d}]$ in \bar{d} Zeitintervalle (Perioden) $[0, 1[$, $[1, 2[$, \dots , $[\bar{d}-2, \bar{d}-1[$, $[\bar{d}-1, \bar{d}]$ zerlegt wird und die Vorgänge $i \in V$ nur zu Beginn eines dieser Zeitintervalle, d.h. zu den (diskreten) Startzeitpunkten $t = 0, 1, \dots, \bar{d}-1$, sowie zum Zeitpunkt $t = \bar{d}$ starten können. Diese Annahme ist i.d.R. nicht einschränkend, da unter der Prämisse, dass die Vorgangsduern und Zeitabstände des zugrunde liegenden Netzplans ganzzahlig sind, für das zugrunde liegende Problem stets eine optimale Lösung mit ganzzahligen Startzeitpunkten $S_i \in \{0, 1, \dots, \bar{d}\}$ für alle $i \in V$ existiert.

Bezeichne $W_i = [ES_i, LS_i]$ das Zeitfenster, in dem Vorgang $i \in V$ unter Beachtung der zugrunde liegenden Zeitbeziehungen starten kann, dann entspricht $\bar{W}_i = \{ES_i, ES_i + 1, \dots, LS_i\} \subseteq \{0, 1, \dots, \bar{d}\}$ der Menge der zugehörigen diskreten Startzeitpunkte. Anstelle der Entscheidungsvariablen S_i können wir dann eine Menge von binären Entscheidungsvariablen x_{it} ($t \in \bar{W}_i$) für jeden Vorgang $i \in V$ verwenden. x_{it} sei genau dann 1, wenn Vorgang $i \in V$ zum Zeitpunkt $t \in \bar{W}_i$ startet, d.h.

$$x_{it} := \begin{cases} 1 & \text{falls } t = S_i \\ 0 & \text{sonst.} \end{cases}$$

Da ein Projekt stets zum Zeitpunkt 0 startet, d.h. $S_0 = 0$, ergibt sich für den Projektstart $x_{00} = 1$. Der Vektor $x = (x_{00}, x_{1ES_1}, \dots, x_{1LS_1}, x_{2ES_2}, \dots, x_{n+1,LS_{n+1}})^T$ fasst alle auftretenden Binärvariablen zusammen.

Mit den Nebenbedingungen

$$\sum_{t \in \bar{W}_i} x_{it} = 1 \quad (i \in V)$$

stellen wir sicher, dass jeder Vorgang eines Projektes genau einmal gestartet wird. Um die Einhaltung der vorgegebenen Zeitbeziehungen im Projekt zu gewährleisten, führen wir zunächst den Ausdruck $\sum_{t \in \bar{W}_i} t x_{it}$ ein. Dieser entspricht für einen Vorgang $i \in V$ genau dann S_i , wenn i zum Zeitpunkt $t = S_i$ startet. Zur Einhaltung der Mindest- und Höchstabstände ergeben sich somit die Nebenbedingungen

$$\sum_{t \in \bar{W}_j} t x_{jt} - \sum_{t \in \bar{W}_i} t x_{it} \geq \delta_{ij} \quad (\langle i, j \rangle \in E).$$

Betrachten wir das *Ressourceninvestmentproblem* mit Zielfunktion

$$f(S) := \sum_{k \in \mathcal{R}} c_k^P \max_{t \in [0, \bar{d}]} r_k(S, t),$$

bei dem die kumulierte mit Bereitstellungskosten c_k^P gewichtete maximale Ressourceninanspruchnahme aller Ressourcen über die Zeit zu minimieren ist. Zunächst gilt es, den Term $r_k(S, t) = \sum_{i \in \mathcal{A}(S, t)} r_{ik}$ mit Hilfe der Binärvariablen x_{it} ($i \in V, t \in \overline{W}_i$) auszudrücken. Ein Vorgang $i \in V$ ist genau dann zum Zeitpunkt t in Ausführung, falls er nicht nach t und höchstens $p_i - 1$ Zeiteinheiten vor t startet, wobei die Subtraktion von 1 der Freigabe der betrachteten Ressource am Ende des Vorgangs Rechnung trägt. Ist ein Vorgang i zum Zeitpunkt t in Ausführung, dann muss also $\sum_{\tau=t-p_i+1}^t x_{i\tau} = 1$ erfüllt sein. Für gegebenes x entspricht

$$\sum_{i \in V} r_{ik} \sum_{\tau=t-p_i+1}^t x_{i\tau}$$

gerade der gesamten Inanspruchnahme der Ressource k zum Zeitpunkt t . Da ein Vorgang $i \in V$ nur zu einem Zeitpunkt $t \in \overline{W}_i = \{ES_i, ES_i + 1, \dots, LS_i\}$ starten kann, ist der obige Ausdruck äquivalent zu

$$\sum_{i \in V} r_{ik} \sum_{\tau=\max\{ES_i, t-p_i+1\}}^{\min\{t, LS_i\}} x_{i\tau}.$$

Wir führen nun für jede Ressource $k \in \mathcal{R}$ eine Hilfsvariable $z_k \geq 0$ ein, die mindestens der maximalen Ressourceninanspruchnahme von k zu einem Zeitpunkt t entspricht, d.h. die Ungleichung

$$z_k \geq \sum_{i \in V} r_{ik} \sum_{\tau=\max\{ES_i, t-p_i+1\}}^{\min\{t, LS_i\}} x_{i\tau}, \quad (2.3)$$

muss für alle $t \in \{0, 1, \dots, \overline{d} - 1\}$ und $k \in \mathcal{R}$ erfüllt sein. Das kleinste z_k , für das (2.3) gerade noch erfüllt ist, entspricht dann der maximalen Ressourceninanspruchnahme von $k \in \mathcal{R}$. Somit kann die folgende zeitindexbasierte Formulierung des Ressourceninvestmentproblems angegeben werden

$$\begin{aligned} \text{Minimiere} \quad & \sum_{k \in \mathcal{R}} c_k^P z_k \\ \text{u.d.N.} \quad & z_k \geq \sum_{i \in V} r_{ik} \sum_{\tau=\max\{ES_i, t-p_i+1\}}^{\min\{t, LS_i\}} x_{i\tau} \quad (t \in \{0, 1, \dots, \overline{d} - 1\}, k \in \mathcal{R}) \\ & \sum_{t \in \overline{W}_i} x_{it} = 1 \quad (i \in V) \\ & \sum_{t \in \overline{W}_j} t x_{jt} - \sum_{t \in \overline{W}_i} t x_{it} \geq \delta_{ij} \quad (\langle i, j \rangle \in E) \\ & x_{00} = 1 \\ & x_{it} \in \{0, 1\} \quad (i \in V, t \in \overline{W}_i). \end{aligned}$$

Es sei angemerkt, dass die Nebenbedingung $x_{00} = 1$ redundant ist, da $ES_0 = LS_0 = 0$ und somit $\overline{W}_0 = \{0\}$ gilt. Das folgende Beispiel veranschaulicht die Formulierung des Ressourceninvestmentproblems mit zeitindexbasierten Entscheidungsvariablen.

Beispiel 2.8. Betrachten wir den in Abbildung 2.11 dargestellten Projektnetzplan N mit einer erneuerbaren Ressource und der maximalen Projektdauer von $\bar{d} = 6$. Wegen $|\mathcal{R}| = 1$ kann der Ressourcenindex k entfallen.

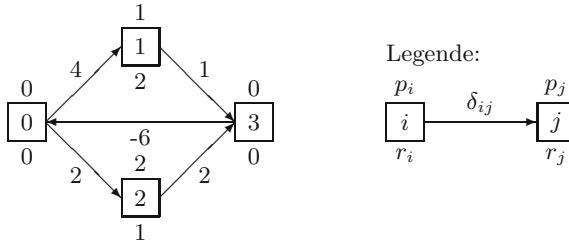


Abb. 2.11. Netzplan N

Wir unterteilen den Planungszeitraum $[0, 6]$ in die diskreten Zeitpunkte $t = 0, \dots, 6$. Mit Hilfe der Metra-Potential-Methode aus Abschnitt 1.4.3 erhalten wir die in Tabelle 2.3 angegebenen frühesten und spätesten Startzeitpunkte der Vorgänge $i \in V$.

Tabelle 2.3. ES_i und LS_i für alle Vorgänge $i \in V$

i	0	1	2	3
ES_i	0	4	2	5
LS_i	0	5	4	6

Aufgrund der durch Netzplan N implizierten Zeitbeziehungen muss Vorgang 1 innerhalb des Intervalls $W_1 = [4, 5]$ starten. Die Menge der diskreten Startzeitpunkte von Vorgang 1 ergibt sich somit zu $\overline{W}_1 = \{4, 5\}$. Für die Vorgänge 2 und 3 gilt entsprechend $\overline{W}_2 = \{2, 3, 4\}$ und $\overline{W}_3 = \{5, 6\}$. Weiterhin definieren wir für jeden Vorgang $i \in V$ eine Menge von Entscheidungsvariablen x_{it} mit $t \in \overline{W}_i$ und eine Hilfsvariable $z \geq 0$, die der maximalen Ressourcenanspruchnahme zu einem Zeitpunkt $t \in \{0, \dots, 6\}$ entspricht. Mit Bereitstellungskosten c^P ergibt sich die folgende MIP-Formulierung für die beschriebene Instanz des Ressourceninvestmentproblems

$$\begin{aligned}
 & \text{Minimiere } c^P z \\
 \text{u.d.N.} \quad & z \geq x_{22} & (t = 2) \\
 & z \geq x_{22} + x_{23} & (t = 3) \\
 & z \geq 2x_{14} + x_{23} + x_{24} & (t = 4)
 \end{aligned}$$

$$\begin{aligned}
z &\geq 2x_{15} + x_{24} & (t = 5) \\
x_{00} &= 1 & (i = 0) \\
x_{14} + x_{15} &= 1 & (i = 1) \\
x_{22} + x_{23} + x_{24} &= 1 & (i = 2) \\
x_{35} + x_{36} &= 1 & (i = 3) \\
4x_{14} + 5x_{15} - 0x_{00} &\geq 4 & (\langle i, j \rangle = \langle 0, 1 \rangle) \\
2x_{22} + 3x_{23} + 4x_{24} - 0x_{00} &\geq 2 & (\langle i, j \rangle = \langle 0, 2 \rangle) \\
5x_{35} + 6x_{36} - 4x_{14} - 5x_{15} &\geq 1 & (\langle i, j \rangle = \langle 1, 3 \rangle) \\
5x_{35} + 6x_{36} - 2x_{22} - 3x_{23} - 4x_{24} &\geq 2 & (\langle i, j \rangle = \langle 2, 3 \rangle) \\
0x_{00} - 5x_{35} - 6x_{36} &\geq -6 & (\langle i, j \rangle = \langle 3, 0 \rangle) \\
z &\geq 0 \\
x_{it} &\in \{0, 1\} & (i = 1, 2, 3; t \in \overline{W}_i) .
\end{aligned}$$

Wenden wir uns nun dem *Ressourcenabweichungsproblem* mit Zielfunktion

$$f(S) := \sum_{k \in \mathcal{R}} c_k^D \int_{t \in [0, \bar{d}]} [r_k(S, t) - Y_k]^+ dt$$

zu, bei dem für jede Ressource $k \in \mathcal{R}$ Kosten c_k^D berücksichtigt werden, die bei Überschreitung des vorgegebenen Ressourcenniveaus Y_k anfallen. Zunächst gilt es, den Term $r_k(S, t) - Y_k = \sum_{i \in \mathcal{A}(S, t)} r_{ik} - Y_k$ mit Hilfe der Binärvariablen x_{it} ($i \in V, t \in \overline{W}_i$) auszudrücken. Wir erhalten

$$\sum_{i \in \mathcal{A}(S, t)} r_{ik} - Y_k = \sum_{i \in V} r_{ik} \sum_{\tau=t-p_i+1}^t x_{i\tau} - Y_k.$$

Nun führen wir Hilfsvariablen $z_{kt} \geq 0$ für alle Ressourcen $k \in \mathcal{R}$ und alle Zeitpunkte $t \in \{0, \dots, \bar{d} - 1\}$ ein, wobei z_{kt} der über das Ressourcenniveau Y_k hinausgehenden Ressourceninanspruchnahme von Ressource k zum Zeitpunkt t entspricht. Berücksichtigen wir, dass ein Vorgang $i \in V$ nur zu einem Zeitpunkt $t \in \overline{W}_i = \{ES_i, ES_i + 1, \dots, LS_i\}$ starten kann, müssen für die Variablen z_{kt} die Ungleichungen

$$z_{kt} \geq \sum_{i \in V} r_{ik} \sum_{\tau=\max\{ES_i, t-p_i+1\}}^{\min\{t, LS_i\}} x_{i\tau} - Y_k$$

erfüllt sein. Für das Ressourcenabweichungsproblem mit Zielfunktion (RD) erhalten wir somit das folgende zeitindexbasierte Modell

$$\begin{aligned}
\text{Min.} \quad & \sum_{k \in \mathcal{R}} \sum_{t \in \{0, 1, \dots, \bar{d} - 1\}} c_k^D z_{kt} \\
\text{u.d.N.} \quad & z_{kt} \geq \sum_{i \in V} r_{ik} \sum_{\tau=\max\{ES_i, t-p_i+1\}}^{\min\{t, LS_i\}} x_{i\tau} - Y_k \quad (t \in \{0, \dots, \bar{d} - 1\}, k \in \mathcal{R})
\end{aligned}$$

$$\begin{aligned}
\sum_{t \in \overline{W}_i} x_{it} &= 1 & (i \in V) \\
\sum_{t \in \overline{W}_j} tx_{jt} - \sum_{t \in \overline{W}_i} tx_{it} &\geq \delta_{ij} & (\langle i, j \rangle \in E) \\
z_{kt} &\geq 0 & (t \in \{0, \dots, \bar{d} - 1\}, k \in \mathcal{R}) \\
x_{it} &\in \{0, 1\} & (i \in V, t \in \overline{W}_i).
\end{aligned}$$

Für Projektplanungsprobleme mit den Zielfunktionen (*RI*) und (*RD*) lässt sich auf Grundlage der vorgestellten MIP-Formulierungen mit Hilfe von Standardsoftware zur Lösung gemischt-ganzzahliger Programme prinzipiell eine optimale Lösung angeben. Praktisch wird man auf diese Weise i.d.R. jedoch nur kleine bis mittelgroße Probleminstanzen erfolgreich lösen können. Für *Ressourcennivellierungsprobleme* lässt sich ebenfalls eine MIP-Formulierung mit zusätzlichen binären Hilfsvariablen y_{ktq} angeben. Die Hilfsvariablen nehmen genau dann den Wert 1 an, falls $r_k^2(S, t) = q$ ist. Dabei kommen für q alle Quadratzahlen $1, 4, 9, \dots$ kleiner als $(\sum_{i \in V} r_{ik})^2$ in Frage. Zu beachten ist, dass die Anzahl an Entscheidungsvariablen schon für kleine Instanzen sehr groß sein kann.

2.1.4 Zeitzulässiger Bereich

Der *zeitzulässige Bereich* \mathcal{S}_T von Problem (2.1) ist durch Zeitrestriktionen der Form

$$1 \cdot S_j - 1 \cdot S_i \geq \delta_{ij} \quad (\langle i, j \rangle \in E) \quad (2.4)$$

sowie durch $S_0 = 0$ spezifiziert und stellt ein konkaves Polytop dar. Da die Koeffizienten der Entscheidungsvariablen in den Nebenbedingungen (2.4) alle 1 oder -1 sind, verlaufen die Begrenzungslinien von \mathcal{S}_T in binärer Richtung $z \in \{0, 1\}^{n+2}$ parallel zu den Koordinatenachsen oder zur ersten Winkelhalbierenden (vgl. z.B. Abb. 2.13). Ferner besitzt \mathcal{S}_T als eindeutigen *Minimalpunkt* den *ES*-Schedule und als eindeutigen *Maximalpunkt* den *LS*-Schedule. Ein Punkt S^{min} eines Polytops heißt Minimalpunkt, falls er bezüglich jeder Komponente minimal ist, d.h. es gilt $S_i^{min} \leq S_i$ für alle $i \in V$ und alle Punkte S des Polytops. Analog heißt ein Punkt S^{max} eines Polytops Maximalpunkt, falls $S_i^{max} \geq S_i$ für alle $i \in V$ und alle Punkte S des Polytops gilt. Ein *Extremalpunkt* bzw. eine *Ecke* S eines Polytops ist ein Punkt, der sich nicht als Konvexitätskombination zweier von S verschiedener Punkte des Polytops darstellen lässt, d.h. es gibt keine zwei Punkte, so dass S auf der Verbindungsgeraden der beiden Punkte liegt. Der zeitzulässige Bereich $\mathcal{S}_T \subseteq \mathbb{R}_{\geq 0}^{n+2}$ lässt sich für $n \geq 2$ i.d.R. nicht vollständig visualisieren. Besitzt ein Projekt allerdings genau zwei Vorgänge i und j , die nicht zeitlich fixiert sind, d.h. die Vorgänge haben eine positive Gesamtpufferzeit, so können wir den Bereich \mathcal{S}_T als so genannten S_i - S_j -Schnitt darstellen. Dabei projizieren wir \mathcal{S}_T auf die entsprechende S_i - S_j -Ebene.

Beispiel 2.9. Wir betrachten den in Abbildung 2.12 dargestellten Projektnetzplan mit drei realen Vorgängen. Die Einhaltung einer maximalen Projektdauer $\bar{d} = 6$ wird durch den Rückwärtspfeil $\langle 4, 0 \rangle$ mit der Bewertung $\delta_{40} = -6$ sichergestellt.

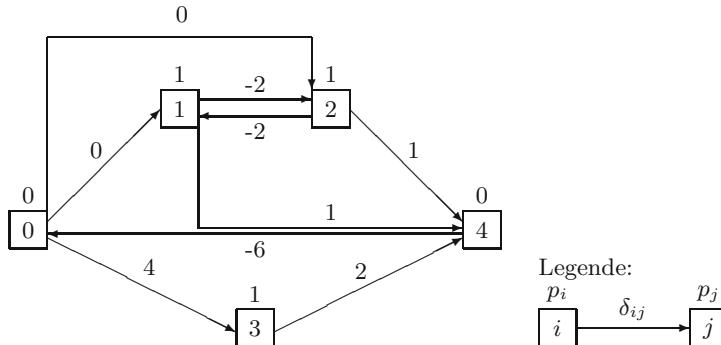


Abb. 2.12. Projektnetzplan N mit drei realen Vorgängen

Für das zugrunde liegende Projekt erhalten wir die in Tabelle 2.4 angegebenen frühesten und spätesten Startzeitpunkte sowie die Gesamtpufferzeiten der Vorgänge $i \in V$. Da die Vorgänge 0, 3 und 4 eine Gesamtpufferzeit von 0 besitzen, sind sie fixiert und müssen zu ihrem frühesten Startzeitpunkt eingeplant werden. Der durch Projektnetzplan N spezifizierte zeitzulässige Bereich \mathcal{S}_T lässt sich folglich als S_1 - S_2 -Schnitt visualisieren.

Tabelle 2.4. ES_i , LS_i und TF_i für alle Vorgänge $i = 0, \dots, 4$

i	0	1	2	3	4
ES_i	0	0	0	4	6
LS_i	0	5	5	4	6
TF_i	0	5	5	0	0

Sobald ein Vorgang eingeplant wird, ändert sich i.d.R. das Zeitfenster $[ES_i, LS_i]$, innerhalb dessen ein noch nicht eingeplanter Vorgang i starten kann, da die Zeitbeziehungen zwischen den Vorgängen berücksichtigt werden müssen (vgl. Abschnitt 1.4.5). Planen wir Vorgang 1 zum Zeitpunkt 0 ein, so kann Vorgang 2 nach wie vor frühestens zum Zeitpunkt 0 starten. Aufgrund der Zeitbeziehung $S_1 - S_2 \geq -2$ ergibt sich aber für Vorgang 2 ein spätester Startzeitpunkt von 2. Wird Vorgang 1 zum Zeitpunkt 1 eingeplant, so ergibt sich für Vorgang 2 das mögliche Einplanungs-Zeitfenster $[0, 3]$. Für $S_1 = 5$ erhalten wir $S_2 \in [3, 5]$. Insgesamt ergibt sich der in Abbildung 2.13 dargestellte S_1 - S_2 -Schnitt des zeitzulässigen Bereichs \mathcal{S}_T . Die Extrempunkte (Ecken) des

S_1 - S_2 -Schnitte sind $S^1 = (0, 0, 0, 4, 6)$, $S^2 = (0, 2, 0, 4, 6)$, $S^3 = (0, 0, 2, 4, 6)$, $S^4 = (0, 5, 5, 4, 6)$, $S^5 = (0, 5, 3, 4, 6)$, $S^6 = (0, 3, 5, 4, 6)$.

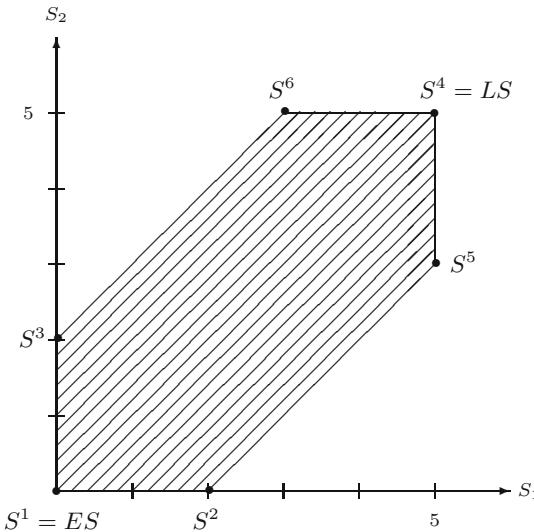


Abb. 2.13. S_1 - S_2 -Schnitt des zeitzulässigen Bereichs

Aus der Theorie der Linearen Optimierung ist bekannt, dass in jedem *Extrempunkt (Ecke)* des zeitzulässigen Bereichs \mathcal{S}_T mindestens $n + 1$ linear unabhängige Nebenbedingungen $S_j - S_i \geq \delta_{ij}$ bindend sind, d.h. sie sind mit Gleichheit erfüllt (vgl. NEUMANN und MORLOCK, 2002, Kapitel 1). Beispielsweise sind im Extrempunkt S^6 in Abbildung 2.13 die Zeitbeziehungen $S_0 - S_4 \geq -6$, $S_1 - S_2 \geq -2$, $S_4 - S_2 \geq 1$ und $S_4 - S_3 \geq 2$ bindend.

Definition 2.10 (Gerüst, Outtree). Sei $D = \langle V, E \rangle$ ein Digraph mit $n + 2$ Knoten. Ein schwach zusammenhängender Teildigraph⁶ des Digraphen D mit $n + 2$ Knoten und $n + 1$ Pfeilen heißt *Gerüst* von D . Ein Gerüst, bei dem jeder Knoten von Knoten 0 aus erreichbar ist, bezeichnen wir als *Outtree* mit Wurzelknoten 0.

Abbildung 2.14 zeigt einen Digraphen sowie ein zugehöriges Gerüst und einen zugehörigen Outtree. Der folgende Satz besagt, dass sich die Extrempunkte des zeitzulässigen Bereichs eines Projektnetzplans als Gerüste des zugrunde liegenden Netzplans darstellen lassen.

Satz 2.11. Jede Ecke des zeitzulässigen Bereichs \mathcal{S}_T kann durch mindestens ein Gerüst des zugehörigen Netzplans N repräsentiert werden, wobei jeder

⁶ In einem schwach zusammenhängenden Teildigraph sind alle Knoten durch eine ungerichtete Pfeilfolge miteinander verbunden.

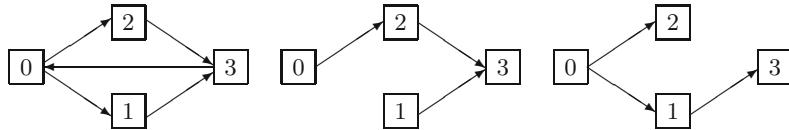


Abb. 2.14. Digraph, Gerüst und Outtree

Pfeil $\langle i, j \rangle$ des Gerüstes eine *bindende Zeitbeziehung* $S_j - S_i = \delta_{ij}$ repräsentiert. Ferner entspricht jedem Gerüst des Netzplans N höchstens ein Extrempunkt von S_T .

Zur Generierung eines Gerüstes für eine gegebene Ecke S des zeitzulässigen Bereichs S_T bestimmen wir sukzessive für jeden Knoten $i \in V \setminus \{0\}$ eine bindende Nebenbedingung $S_j - S_i = \delta_{ij}$ bzw. $S_i - S_j = \delta_{ji}$ zu einem Knoten j , für den bereits eine bindende Nebenbedingung bestimmt wurde. Ausgangspunkt ist hierbei der Knoten 0, für den die Nebenbedingung $S_0 = 0$ immer bindend ist. Algorithmus 2.12 stellt dar, wie für einen gegebenen Extrempunkt S ein zugehöriges Gerüst $G = \langle V, E_G \rangle$ mit der Pfeilmenge E_G erzeugt werden kann. Dabei gibt $m_i = 1$ an, dass Knoten i bereits betrachtet wurde.

Algorithmus 2.12 (Generierung eines Gerüstes für eine Ecke S).

Setze $m_0 := 1$ und $m_i := 0$ für alle $i \in V \setminus \{0\}$.

$E_G := \emptyset$ und $Q := \{0\}$ (Q wird als Schlange verwaltet).

Solange $Q \neq \emptyset$ und mindestens ein $i \in V$ mit $m_i = 0$ existiert:

Entferne i vom Kopf der Schlange Q .

An Knoten i wird nun, falls möglich, ein weiterer Knoten angehängt.

1. **Für alle** $\langle i, j \rangle \in E$:

Falls $m_j = 0$ und $S_j - S_i = \delta_{ij}$:

Setze $E_G := E_G \cup \{\langle i, j \rangle\}$, $m_j := 1$, und füge j am Ende von Q ein.

2. **Für alle** $\langle h, i \rangle \in E$:

Falls $m_h = 0$ und $S_i - S_h = \delta_{hi}$:

Setze $E_G := E_G \cup \{\langle h, i \rangle\}$, $m_h := 1$, und füge h am Ende von Q ein.

Anmerkung 2.13. Bei der Generierung eines Gerüstes für den *ES-Schedule* entfällt die 2. Schleife in Algorithmus 2.12, da der *ES-Schedule* als Outtree in N dargestellt werden kann.

Um die Beziehung zwischen den Ecken des zeitzulässigen Bereichs S_T und den Gerüsten des zugehörigen Projektnetzplans N zu illustrieren, betrachten wir das folgende Beispiel.

Beispiel 2.14. Gegeben seien der in Abbildung 2.15 dargestellte Projektnetzplan mit drei Vorgängen und der zugehörige S_1 - S_2 -Schnitt des zeitzulässigen Bereichs S_T mit den Extrempunkten $S^1 = (0, 1, 4)$, $S^2 = (0, 1, 6)$, $S^3 = (0, 4, 6)$ und $S^4 = (0, 2, 4)$.

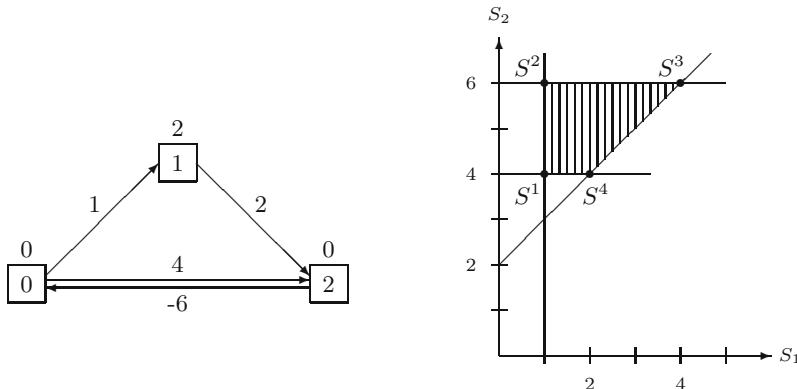


Abb. 2.15. Projektnetzplan und $S_1 - S_2$ -Schnitt des zeitzulässigen Bereichs \mathcal{S}_T

Wir bestimmen mit Hilfe von Algorithmus 2.12 das zu S^1 gehörige Gerüst. In der Initialisierung setzen wir $m_0 := 1$, $m_1 := 0$, $m_2 := 0$, $E_G := \emptyset$ und $Q := \{0\}$. Dann entnehmen wir Vorgang 0 vom Kopf der Schlange Q . Da Pfeil $\langle 0, 1 \rangle$ im Netzplan enthalten ist, $m_1 = 0$ und $S_1 - S_0 = 1$ gilt, hängen wir Knoten 1 an Knoten 0 an, d.h. wir setzen $E_G := \{\langle 0, 1 \rangle\}$, $m_1 := 1$ und $Q := \{1\}$. Weiterhin sind für $\langle 0, 2 \rangle \in E$ die Gleichungen $m_2 = 0$ und $S_2 - S_0 = 4$ erfüllt, deshalb setzen wir $E_G := \{\langle 0, 1 \rangle, \langle 0, 2 \rangle\}$, $m_2 := 1$ und $Q := \{1, 2\}$. Da für alle Vorgänge $i \in V$ nun $m_i = 1$ gilt, terminiert der Algorithmus. Somit sind für Schedule S^1 die beiden linear unabhängigen Nebenbedingungen $S_1^1 - S_0^1 \geq 1$ und $S_2^1 - S_0^1 \geq 4$ bindend, und das korrespondierende Gerüst G^1 enthält die Pfeilmenge $E_{G^1} = \{\langle 0, 1 \rangle, \langle 0, 2 \rangle\}$. Wird Algorithmus 2.12 angewendet, um die zu S^2 , S^3 und S^4 gehörigen Gerüste zu generieren, dann erhalten wir für S^2 das Gerüst G^2 mit der Pfeilmenge $E_{G^2} = \{\langle 0, 1 \rangle, \langle 2, 0 \rangle\}$, für S^3 das Gerüst G^3 mit der Pfeilmenge $E_{G^3} = \{\langle 1, 2 \rangle, \langle 2, 0 \rangle\}$ und für S^4 ergibt sich $E_{G^4} = \{\langle 0, 2 \rangle, \langle 1, 2 \rangle\}$.

Um für ein gegebenes Gerüst G mit Pfeilmenge E_G den zugehörigen Extremalpunkt S zu bestimmen, lösen wir das lineare Gleichungssystem

$$\begin{aligned} S_j - S_i &= \delta_{ij} \quad (\langle i, j \rangle \in E_G) \\ S_0 &= 0 \end{aligned}$$

Für das dem Netzplan aus Abbildung 2.15 zugrunde liegende Gerüst G^3 mit Pfeilmenge $E_{G^3} = \{\langle 1, 2 \rangle, \langle 2, 0 \rangle\}$ erhalten wir beispielsweise das lineare Gleichungssystem

$$\begin{aligned} S_2 - S_1 &= 2 \\ S_0 - S_2 &= -6 \\ S_0 &= 0 \end{aligned}$$

mit der eindeutigen Lösung $S^3 = (0, 4, 6)$.

Es sei angemerkt, dass nicht jedes Gerüst des Netzplans N einen Extremalpunkt des zeitzulässigen Bereichs \mathcal{S}_T repräsentiert. Beispielsweise entspricht das Gerüst G' mit der Pfeilmenge $E_{G'} = \{\langle 0, 1 \rangle, \langle 1, 2 \rangle\}$ dem Schedule $S' = (0, 1, 3)$. S' ist nicht zeitzulässig, da Vorgang 3 frühestens 4 Zeiteinheiten nach dem Projektbeginn starten darf. In Abbildung 2.15 entspricht Schedule S' dem Schnittpunkt der beiden Geraden $S_1 - S_0 = 1$ und $S_2 - S_1 = 2$.

2.1.5 Zielfunktionen und ausgezeichnete Punkte

Wie in Abschnitt 2.1.1 geschildert, existiert für Problem (2.1) mit stetiger bzw. nach unten halbstetiger Zielfunktion $f(S)$ stets eine optimale Lösung. Je nach betrachteter Zielfunktion können wir uns bei der Suche nach einem solchen optimalen Schedule auf eine Menge *ausgezeichneter Punkte* beschränken. Bei den ausgezeichneten Punkten handelt es sich um zeitzulässige Schedules, die potentielle Kandidaten für eine optimale Lösung darstellen. Dabei hängt die Menge der ausgezeichneten Punkte von speziellen Struktureigenschaften der betrachteten Zielfunktionen ab, die wir im Folgenden untersuchen. In Abhängigkeit der Struktureigenschaften lassen sich dann unterschiedliche Lösungsverfahren etablieren.

Bei den Zielfunktionen Projektdauer (PD) und mittlere Durchlaufzeit (MFT) handelt es sich um *reguläre Funktionen*, d.h. f ist monoton wachsend in den Startzeitpunkten der Vorgänge. Für zwei Schedules $S, S' \in \mathbb{R}_{\geq 0}^{n+2}$ mit $S \leq S'$ (d.h. $S_i \leq S'_i$ für alle $i \in V$) gilt somit $f(S) \leq f(S')$. Da der zeitzulässige Bereich \mathcal{S}_T ein konkaves Polytop mit genau einem Minimalpunkt (dem ES -Schedule) darstellt, ist für Problem (2.1) mit regulärer Zielfunktion der ES -Schedule optimal. Die Menge der ausgezeichneten Punkte für Projektplanungsprobleme mit den Zielfunktionen (PD), (MFT) oder anderen regulären Funktionen besteht daher aus dem ES -Schedule.

Für die *lineare* Zielfunktion (WST) der Summe gewichteter Startzeitpunkte stellt Problem (2.1) ein lineares Programm dar, für das immer einer der Extremalpunkte des zeitzulässigen Bereichs optimal ist. Die Menge der ausgezeichneten Punkte für Problem (2.1) mit Zielfunktion (WST) entspricht somit allen Extremalpunkten (Ecken) von \mathcal{S}_T .

Die Zielfunktion des Earliness-Tardiness-Problems ($E + T$) ist *konvex*, d.h. jede Sekante liegt oberhalb des Graphen von f und es gilt $f(\lambda S + (1 - \lambda)S') \leq \lambda f(S) + (1 - \lambda)f(S')$ für alle $S, S' \in \mathbb{R}_{\geq 0}^{n+2}$ und $\lambda \in]0, 1[$ (vgl. Abb. 2.16).

Für konvexe Zielfunktionen ist i.d.R. kein Punkt auf dem Rand des zeitzulässigen Bereichs \mathcal{S}_T optimal. Da aber für Minimierungsprobleme mit konvexer Zielfunktion und konvexem zulässigen Bereich jedes lokale Optimum auch global optimal ist, lässt sich die Suche nach einem optimalen Schedule beschränken auf lokale Minimalstellen S^* der Funktion f , d.h. auf Punkte S^* , für die $f(S^*) \leq f(S')$ für alle S' aus einer offenen ε -Umgebung von S^* erfüllt ist. Ist $c_i^E = 0$ für alle $i \in V$, so handelt es sich bei ($E + T$) um eine reguläre Funktion und der ES -Schedule ist optimal. Gilt $c_i^T = 0$ für alle $i \in V$, so ist Zielfunktion ($E + T$) multipliziert mit -1 regulär, d.h. der LS -Schedule

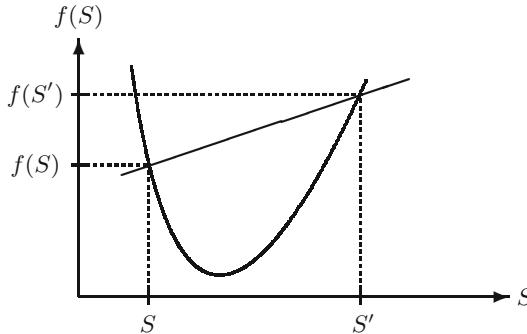


Abb. 2.16. Konvexe Funktion

ist optimal. Eine optimale Lösung für das Earliness-Tardiness-Problem mit $c_i^E, c_i^T > 0$ für mindestens ein $i \in V$ entspricht einer lokalen Minimalstelle S^* , bei der jeder Vorgang i entweder zu $d_i - p_i$ oder zu ES_i bzw. LS_i startet, wobei die ES - und LS -Werte planungsabhängig gemäß Schedule S^* zu verstehen sind (vgl. Abschnitt 1.4.5).

Die Zielfunktion (NPV) des Kapitalwertmaximierungsproblems ist *binär-monoton*, d.h. sie ist monoton auf jeder Halbgeraden $\{S' \in \mathbb{R}_{\geq 0}^{n+2} \mid S' = S + \lambda z, \lambda \in \mathbb{R}\}$ mit binärer Richtung $z \in \{0, 1\}^{n+2}$. Seien z eine binäre Richtung, $V' \subseteq V$ die Menge aller Vorgänge i mit $z_i = 0$ und $V'' = V \setminus V'$ die Menge aller Vorgänge i mit $z_i = 1$. Dann gilt für die Zielfunktion f der Kapitalwertmaximierung

$$\begin{aligned} f(S + \lambda z) &= - \sum_{i \in V} c_i^F \beta^{S_i + \lambda z_i} \\ &= - \sum_{i \in V'} c_i^F \beta^{S_i} - \sum_{i \in V''} c_i^F \beta^{S_i + \lambda} \\ &= - \sum_{i \in V'} c_i^F \beta^{S_i} - \beta^\lambda \sum_{i \in V''} c_i^F \beta^{S_i}. \end{aligned}$$

Für $-\sum_{i \in V''} c_i^F \beta^{S_i} \geq 0$ ist f monoton wachsend und für $-\sum_{i \in V''} c_i^F \beta^{S_i} \leq 0$ monoton fallend in binärer Richtung z . Somit ist f binärmonoton und da ferner die Begrenzungslinien von \mathcal{S}_T in binärer Richtung verlaufen, stellen ebenso wie bei linearen Zielfunktionen die Extrempunkte von \mathcal{S}_T ausgezeichnete Punkte für Problem (2.1) mit Zielfunktion (NPV) dar. Dies ist intuitiv einsichtig, da wir in Abschnitt 2.1.2 bereits gezeigt haben, dass sich das Kapitalwertmaximierungsproblem in ein äquivalentes lineares Programm gleicher Dimension transformieren lässt.

Um ausgezeichnete Punkte für Projektplanungsprobleme mit den Zielfunktionen (RI), (RD) und (RL) angeben zu können, sind einige einleitende Erläuterungen notwendig. Wir führen zunächst den Begriff einer strengen Ordnung auf der Knotenmenge V ein.

Definition 2.15 (Strenge Ordnung). Eine Relation ϱ auf der Menge V ist eine Menge von Paaren $(i, j) \in V \times V$. Anstelle von $(i, j) \in \varrho$ schreiben wir auch $i \prec j$ (wobei „ \prec “ als „vor“ gelesen wird) und $i \not\prec j$ für $(i, j) \notin \varrho$. Eine *strenge Ordnung* auf der Knotenmenge V ist eine asymmetrische, transitive Relation \prec in V , d.h.

- für keine zwei Elemente $i, j \in V$ gilt $i \prec j$ und $j \prec i$ (Asymmetrie),
- wenn für $h, i, j \in V$ die Bedingungen $h \prec i$ und $i \prec j$ erfüllt sind, dann gilt auch $h \prec j$ (Transitivität).

Strenge Ordnung $O \subseteq V \times V$ hat die folgende anschauliche Interpretation. Enthält Ordnung O das Element (i, j) , so wird eine Vorrangbeziehung zwischen Vorgang i und Vorgang j induziert. Eine Vorrangbeziehung zwischen i und j besteht, wenn der Startzeitpunkt von j größer oder gleich dem Endzeitpunkt von i ist, d.h. $S_j - S_i \geq p_i$.

Definition 2.16 (Ordnungspolytop). Sei $O \subseteq V \times V$ eine strenge Ordnung auf der Knotenmenge V und sei $\mathcal{S}_T(O) := \{S \in \mathcal{S}_T \mid S_j - S_i \geq p_i \text{ für alle } (i, j) \in O\}$ die Menge aller zeitzulässigen Schedules, die die durch O induzierten Vorrangbeziehungen einhalten. Dann stellt die Teilmenge $\mathcal{S}_T(O)$ von \mathcal{S}_T ein konvexas Polytop dar, das wir als *Ordnungspolytop* von O bezeichnen. Die Ordnung O wird als *zeitzulässig* bezeichnet, wenn $\mathcal{S}_T(O) \neq \emptyset$ ist.

Beispiel 2.17. Betrachten wir den MPM-Netzplan und den S_1 - S_2 -Schnitt des zugehörigen zeitzulässigen Bereichs \mathcal{S}_T in Abbildung 2.17.

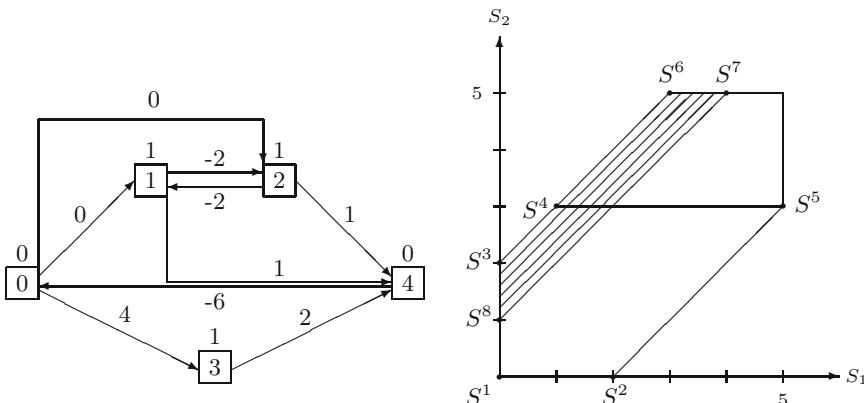


Abb. 2.17. Projektnetzplan und S_1 - S_2 -Schnitt des Bereichs \mathcal{S}_T

Ordnung $O = \{(1, 2)\}$ besagt, dass Vorgang 2 starten kann, sobald Vorgang 1 beendet ist, d.h. $S_2 - S_1 \geq p_1$. Da Vorgang 1 die Ausführungsduer $p_1 = 1$ besitzt, beinhaltet das Ordnungspolytop $\mathcal{S}_T(O)$ alle Schedules $S \in \mathcal{S}_T$, für die gilt, dass Vorgang 2 mindestens eine Zeiteinheit nach Vorgang 1 startet. Startet Vorgang 1 beispielsweise zum Zeitpunkt 0, dann

kann Vorgang 2 zu einem beliebigen Zeitpunkt $t \in [1, 2]$ starten. Somit erhalten wir für die Ordnung $O = \{(1, 2)\}$ das durch die *konvexe Hülle* der Punkte S^3, S^6, S^7, S^8 spezifizierte Ordnungspolytop, d.h. $\mathcal{S}_T(O)$ beinhaltet alle Schedules, die sich als Konvexitätskombination der Punkte S^3, S^6, S^7, S^8 darstellen lassen, und wir schreiben dafür $\mathcal{S}_T(O) = \text{conv}\{S^3, S^6, S^7, S^8\}$ (vgl. Abb. 2.17). Für die Ordnung $O = \{(2, 3)\}$ erhalten wir analog das Ordnungspolytop $\mathcal{S}_T(O) = \text{conv}\{S^1, S^2, S^3, S^4, S^5\}$.

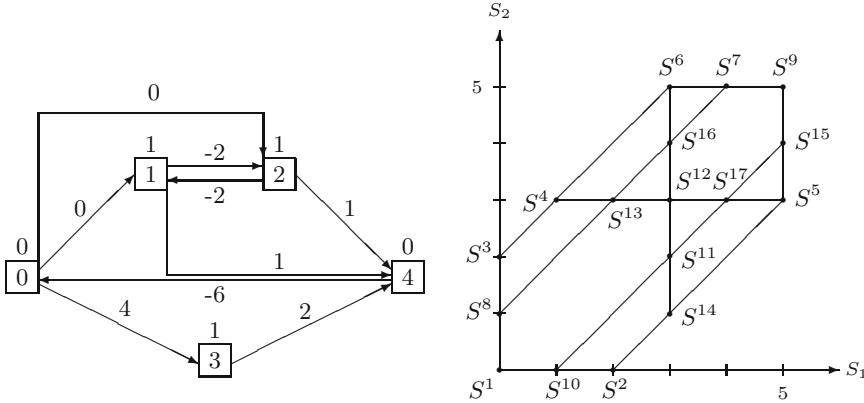
Im Folgenden führen wir das Konzept des so genannten *Ordnungsnetzplans* $N(O)$ einer strengen Ordnung O ein. Dabei ist $N(O)$ der Netzplan, der aus dem Projektnetzplan N resultiert, indem wir für jedes Paar $(i, j) \in O$ einen Pfeil $\langle i, j \rangle$ mit der Bewertung $\delta_{ij} = p_i$ hinzufügen, d.h. Vorgang j kann frühestens nach dem Ende von Vorgang i starten. Besitzt Projektnetzplan N bereits einen Pfeil $\langle i, j \rangle$ mit der Bewertung δ_{ij} , dann ersetzen wir seine Pfeilbewertung durch $\max\{\delta_{ij}, p_i\}$. Die Menge der zeitzulässigen Schedules des Ordnungsnetzplans $N(O)$ entspricht gerade dem Ordnungspolytop $\mathcal{S}_T(O)$ von O . Eine strenge Ordnung O ist demnach genau dann zeitzulässig, wenn der zugehörige Ordnungsnetzplan $N(O)$ keine Zyklen positiver Länge beinhaltet und somit $\mathcal{S}_T(O) \neq \emptyset$ gilt.

Für jeden zeitzulässigen Schedule $S \in \mathcal{S}_T$ existiert eine zugehörige strenge Ordnung $O(S) := \{(i, j) \in V \times V \mid i \neq j, S_j - S_i \geq p_i\}$, die gerade die durch den Schedule S implizierten Vorrangbeziehungen besitzt (wir sagen auch die von Schedule S *induzierte Ordnung*).⁷ Aus Definition 2.16 folgt sofort, dass $S \in \mathcal{S}_T(O(S))$ gilt. Die kanonischen Vorrangbeziehungen $(0, i)$ für alle $i \in V \setminus \{0\}$ sowie $(i, n+1)$ für alle $i \in V \setminus \{n+1\}$ führen wir im Folgenden bei der Darstellung einer Schedule induzierten Ordnung $O(S)$ nicht explizit an, da diese für einen wohldefinierten MPM-Netzplan stets erfüllt sind. Das folgende Beispiel verdeutlicht den Zusammenhang zwischen einem Schedule S , der durch S induzierten Ordnung $O(S)$ und dem Ordnungspolytop $\mathcal{S}_T(O(S))$ von $O(S)$, welches auch *Schedulepolytop* genannt wird.

Beispiel 2.18. Betrachten wir den MPM-Netzplan und den S_1 - S_2 -Schnitt des zugehörigen zeitzulässigen Bereichs \mathcal{S}_T in Abbildung 2.18. Für die eingezeichneten Schedules S^i ($i = 1, \dots, 17$) sollen jeweils die zugehörige Ordnung $O(S^i)$ und das entsprechende Schedulepolytop $\mathcal{S}_T(O(S^i))$ angegeben werden.

Der zeitzulässige Schedule $S^1 = (0, 0, 0, 4, 6)$ impliziert die zugehörige Ordnung $O(S^1) = \{(1, 3), (2, 3)\}$, d.h. Vorgang 3 beginnt nach der Beendigung der Vorgänge 1 und 2. Das zur Ordnung $O(S^1)$ gehörige konvexe Schedulepolytop besitzt die Extrempunkte $S^1, S^2, S^3, S^4, S^{12}, S^{14}$, d.h.

⁷ $O(S)$ repräsentiert für $p_i > 0, i = 1, \dots, n$ eine strenge Ordnung. Im Folgenden gehen wir davon aus, dass nur Vorgänge $i \neq \{0, n+1\}$ mit $p_i > 0$ bei der Generierung von $O(S)$ berücksichtigt werden, Vorgänge mit $p_i = 0$, wie z.B. Meilensteine, sind zwar bei der Bestimmung der planungsabhängigen *ES*- und *LS*-Werte relevant, spielen aber für die Ressourcenprofile eines Schedules keine Rolle.

Abb. 2.18. Projektnetzplan und S_1 - S_2 -Schnitt des Bereichs S_T

$S_T(O(S^1)) = \text{conv}\{S^1, S^2, S^3, S^4, S^{12}, S^{14}\}$. Schedule $S^2 = (0, 2, 0, 4, 6)$ besitzt die zugehörige Ordnung $O(S^2) = \{(1, 3), (2, 1), (2, 3)\}$. Gemäß dieser Ordnung beginnt Vorgang 3 nach Beendigung der Vorgänge 1 und 2 und Vorgang 1 beginnt nach dem Ende von Vorgang 2. Das zu $O(S^2)$ gehörige Schedulepolytop ist $\mathcal{S}_T(O(S^2)) = \text{conv}\{S^2, S^{10}, S^{11}, S^{14}\}$.

Tabelle 2.5 listet zusammenfassend die in Abbildung 2.18 eingezzeichneten Schedules, die zugehörigen Ordnungen $O(S^i)$ und die entsprechenden Schedulepolytope $\mathcal{S}_T(O(S^i))$ auf.

Schedulepolytop $\mathcal{S}_T(O(S))$ enthält alle Schedules S' , die mindestens die durch S implizierten Vorrangbeziehungen enthalten. Ferner kann ein Schedule $S' \in \mathcal{S}_T(O(S))$ aber noch weitere Vorrangbeziehungen implizieren. Betrachten wir z.B. S^{16} in Abbildung 2.18 mit $O(S^{16}) = \{(1, 2), (1, 3)\}$ und $\mathcal{S}_T(O(S^{16})) = \text{conv}\{S^3, S^6, S^8, S^{16}\}$, so enthält $\mathcal{S}_T(O(S^{16}))$ Schedule S^3 mit $O(S^3) = \{(1, 2), (1, 3), (2, 3)\}$.

In einer so genannten Isoordnungsmenge $\mathcal{S}_T^=(O(S))$ hingegen sind nur Schedules enthalten, die dieselbe Ordnung wie S induzieren. Sei S ein zeit-zulässiger Schedule, dann nennen wir die Menge aller Schedules S' aus dem Schedulepolytop von $O(S)$ mit der Eigenschaft, dass $O(S') = O(S)$ ist, die *Isoordnungsmenge* von S . Die Isoordnungsmenge

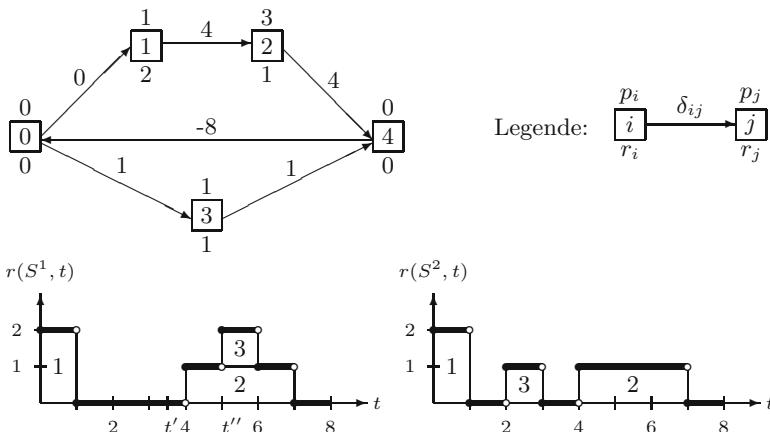
$$\mathcal{S}_T^=(O(S)) := \{S' \in \mathcal{S}_T(O(S)) \mid O(S') = O(S)\}$$

von Schedule S enthält somit alle Schedules $S' \in \mathcal{S}_T$, die die gleiche Ordnung wie S induzieren. Zum Verständnis betrachten wir Abbildung 2.19, in der ein Projektnetzplan und die Ressourcenprofile der beiden Schedules $S^1 = (0, 0, 4, 5, 8)$ und $S^2 = (0, 0, 4, 2, 8)$ dargestellt sind. Für die Vorgänge $i \in \{0, 1, 2, 4\}$ gilt $ES_i = LS_i$; nur Vorgang 3 ist nicht zeitlich fixiert. Da die Schedules $S' = (0, 0, 4, t, 8)$, $t \in]3, 7[$, die Eigenschaft besitzen, dass $O(S') = O(S^1)$ gilt, ist die Isoordnungsmenge von $S^1 = (0, 0, 4, 5, 8)$ gleich

Tabelle 2.5. S^i , $O(S^i)$ und $\mathcal{S}_T(O(S^i))$ für alle $i = 1, \dots, 17$

S	$O(S)$	$\mathcal{S}_T(O(S))$
S^1	$\{(1, 3), (2, 3)\}$	$\text{conv}\{S^1, S^2, S^3, S^4, S^{12}, S^{14}\}$
S^2	$\{(1, 3), (2, 1), (2, 3)\}$	$\text{conv}\{S^2, S^{10}, S^{11}, S^{14}\}$
S^3	$\{(1, 2), (1, 3), (2, 3)\}$	$\text{conv}\{S^3, S^4, S^8, S^{13}\}$
S^4	$\{(1, 2), (1, 3), (2, 3)\}$	$\text{conv}\{S^3, S^4, S^8, S^{13}\}$
S^5	$\{(2, 1), (2, 3), (3, 1)\}$	$\{S^5\}$
S^6	$\{(1, 2), (1, 3), (3, 2)\}$	$\{S^6\}$
S^7	$\{(1, 2), (3, 2)\}$	$\text{conv}\{S^6, S^7\}$
S^8	$\{(1, 2), (1, 3), (2, 3)\}$	$\text{conv}\{S^3, S^4, S^8, S^{13}\}$
S^9	$\{(3, 1), (3, 2)\}$	$\{S^9\}$
S^{10}	$\{(1, 3), (2, 1), (2, 3)\}$	$\text{conv}\{S^2, S^{10}, S^{11}, S^{14}\}$
S^{11}	$\{(1, 3), (2, 1), (2, 3)\}$	$\text{conv}\{S^2, S^{10}, S^{11}, S^{14}\}$
S^{12}	$\{(1, 3), (2, 3)\}$	$\text{conv}\{S^1, S^2, S^3, S^4, S^{12}, S^{14}\}$
S^{13}	$\{(1, 2), (1, 3), (2, 3)\}$	$\text{conv}\{S^3, S^4, S^8, S^{13}\}$
S^{14}	$\{(1, 3), (2, 1), (2, 3)\}$	$\text{conv}\{S^2, S^{10}, S^{11}, S^{14}\}$
S^{15}	$\{(2, 1), (3, 1)\}$	$\text{conv}\{S^5, S^{15}\}$
S^{16}	$\{(1, 2), (1, 3)\}$	$\text{conv}\{S^3, S^6, S^8, S^{16}\}$
S^{17}	$\{(2, 1), (2, 3)\}$	$\text{conv}\{S^2, S^5, S^{10}, S^{17}\}$

$\mathcal{S}_T^=(O(S^1)) = \{(0, 0, 4, t, 8) \mid t \in]3, 7[\}$. Für $S^2 = (0, 0, 4, 2, 8)$ ergibt sich die Isoordnungsmenge $\mathcal{S}_T^=(O(S^2)) = \{(0, 0, 4, t, 8) \mid t \in [1, 3]\}$. Wie wir sehen, sind Isoordnungsmengen i.d.R. nicht abgeschlossen, ihr Abschluss stellt aber ein convexes Polytop dar.

**Abb. 2.19.** Projektnetzplan und Ressourcenprofile von S^1 und S^2

Auf Grundlage der vorhergehenden Ausführungen ist es nun möglich, *ausgezeichnete Punkte* für die Zielfunktionen (RI) , (RD) und (RL) anzugeben.

Wie bereits erläutert, ist die Zielfunktion (RI) nach unten halbstetig. Weiterhin können wir zeigen, dass (RI) konstant auf der Isoordnungsmenge jedes zeitzulässigen Schedules S ist. Seien $S', S'' \in \mathcal{S}_T^=(O(S))$ zwei Schedules aus der Isoordnungsmenge von S , und nehmen wir ferner an, dass $r_k(S', t') = \max_{t \in [0, \bar{d}]} r_k(S', t)$ ist. Da S' und S'' die gleiche Ordnung implizieren, gibt es mindestens einen Zeitpunkt t'' mit $\mathcal{A}(S', t') = \mathcal{A}(S'', t'')$, d.h. bei der Projektausführung gemäß Schedule S' überlappen sich zum Zeitpunkt t' die gleichen Vorgänge wie zum Zeitpunkt t'' bei der Ausführung gemäß S'' (siehe hierzu auch Abb. 2.19, Vorgang 3). Somit gilt, dass der Zielfunktionswert auf der Isoordnungsmenge konstant ist.

Eine stetige oder nach unten halbstetige Funktion $f : \mathbb{R}_{\geq 0}^{n+2} \rightarrow \mathbb{R}$, die auf der Isoordnungsmenge jedes zeitzulässigen Schedules regulär ist, wird *lokal regulär* genannt. Die auf jeder Isoordnungsmenge konstante Funktion (RI) gehört somit zu den lokal regulären Funktionen, für die immer ein Minimalpunkt eines Schedulepolytops $\mathcal{S}_T(O(S))$, $S \in \mathcal{S}_T$ optimal ist. Stellt der Minimalpunkt S' des Abschlusses einer Isoordnungsmenge auch den Minimalpunkt des zugehörigen Schedulepolytops dar und ist damit auch Element der Isoordnungsmenge, so haben wir einen Kandidaten für die optimale Lösung gefunden. Dies ergibt sich direkt aus der Regularität von f auf der Isoordnungsmenge. Gehört der Minimalpunkt S' des Abschlusses selbst nicht zur Isoordnungsmenge, wie beispielsweise S^4 in Abbildung 2.20 nicht zu $\mathcal{S}_T^=(O(S^{18})) = \text{conv}\{S^4, S^6, S^{13}, S^{16}\} \setminus \text{conv}\{S^4, S^{13}\}$ gehört, so gilt wegen der Halbstetigkeit nach unten von f , dass $f(S^4) \leq f(S)$ mit $S \in \mathcal{S}_T^=(O(S^{18}))$. Da f auf der Isoordnungsmenge $\mathcal{S}_T^=(O(S^4)) = \text{conv}\{S^3, S^4, S^8, S^{13}\}$ wieder regulär ist, gilt ferner für den Minimalpunkt S^8 von $\mathcal{S}_T^=(O(S^4))$: $f(S^8) \leq f(S^4) \leq f(S)$ für alle $S \in \mathcal{S}_T^=(O(S^{18}))$. Es sei angemerkt, dass S^8 dem Minimalpunkt der beiden Schedulepolytope $\mathcal{S}_T(O(S^4))$ und $\mathcal{S}_T(O(S^{18}))$ entspricht.

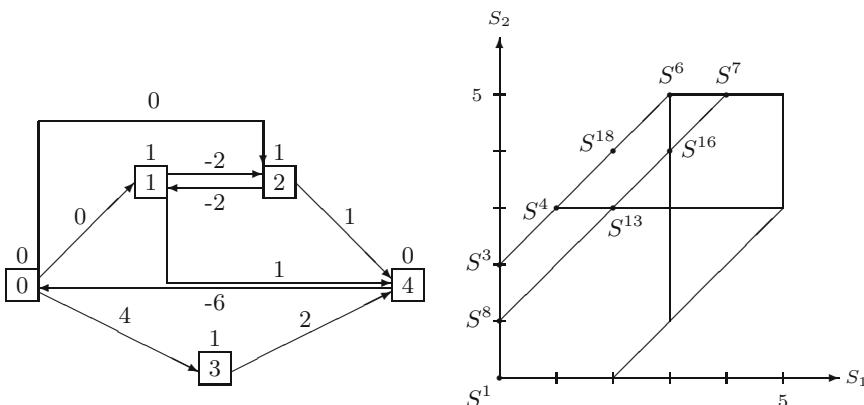


Abb. 2.20. Projektnetzplan und S_1 - S_2 -Schnitt des Bereichs \mathcal{S}_T

Für Zielfunktion (*RI*) kommen also die Minimalpunkte der Schedulepolytope $\mathcal{S}_T(O(S))$ als potentielle Lösungen in Frage und bilden die Elemente der Menge der ausgezeichneten Punkte.

Bei den Zielfunktionen (*RD*) und (*RL*) handelt es sich um stetige Funktionen. Zudem sind (*RD*) und (*RL*) konkav auf der Isoordnungsmenge jedes zeitzulässigen Schedules S . Bei *konkaven Funktionen* liegt jede Sekante unterhalb des Graphen von f , so dass die Funktion f auf einem abgeschlossenen Intervall $[a, b]$ ihr Minimum in einem der beiden Endpunkte a oder b annimmt (vgl. Abb. 2.21), d.h. aufgrund der Stetigkeit entspricht eine Minimalstelle von f immer einer Ecke des Abschlusses der entsprechenden Isoordnungsmenge.

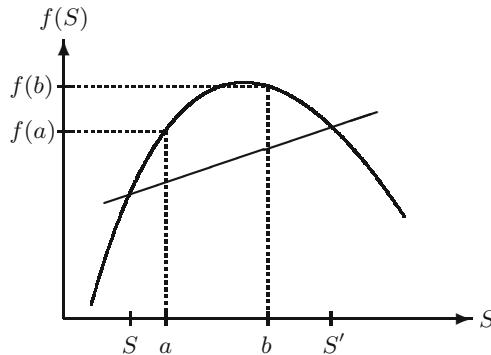


Abb. 2.21. Konkave Funktion

Eine nach unten halbstetige (stetige) Funktion $f : \mathbb{R}_{\geq 0}^{n+2} \rightarrow \mathbb{R}$, die auf der Isoordnungsmenge jedes zeitzulässigen Schedules konkav ist, wird *lokal konkav* genannt. Somit gehören (*RD*) und (*RL*) zu den lokal konkaven Funktionen, für die immer ein Extrempunkt eines Schedulepolytops optimal ist. Da jeder Extrempunkt des Abschlusses einer Isoordnungsmenge auch Extrempunkt eines Schedulepolytops ist, folgt dies sofort aus der Stetigkeit und Konkavität von f . Die Menge der ausgezeichneten Punkte eines Projektplanungsproblems mit Zielfunktion (*RD*) oder (*RL*) entspricht daher den Extrempunkten aller Schedulepolytope $\mathcal{S}_T(O(S))$, $S \in \mathcal{S}_T$.

Die verschiedenen Zielfunktionen und ihre *ausgezeichneten Punkte* sind zusammenfassend in Tabelle 2.6 angegeben.

Tabelle 2.6. Zielfunktionen und ausgezeichnete Punkte

Zielfunktion	ausgezeichnete Punkte von \mathcal{S}_T
(<i>PD</i>), (<i>MFT</i>)	Minimalpunkt von \mathcal{S}_T
(<i>E + T</i>)	lokale Minimalstellen S^* der Funktion f
(<i>WST</i>), (<i>NPV</i>)	Extrempunkte von \mathcal{S}_T
(<i>RI</i>)	Minimalpunkte von $\mathcal{S}_T(O(S))$
(<i>RD</i>), (<i>RL</i>)	Extrempunkte von $\mathcal{S}_T(O(S))$

2.2 Exakte Lösungsverfahren

Wir stellen nun exakte Verfahren zur Lösung von Projektplanungsproblemen unter Zeitrestriktionen mit den in Abschnitt 2.1.1 aufgeführten Zielfunktionen vor. Die exakten Verfahren bestimmen unter den in Abschnitt 2.1.5 eingeführten ausgezeichneten Punkten der jeweiligen Zielfunktion eine optimale Lösung.

2.2.1 Minimierung der Projektdauer

Bei den Zielfunktionen der Minimierung der Projektdauer (PD) und der Minimierung der mittleren Durchlaufzeit (MFT) handelt es sich, wie in Abschnitt 2.1.5 beschrieben, um reguläre Funktionen. Daher muss zur Bestimmung einer optimalen Lösung für das entsprechende Optimierungsproblem der Minimalpunkt des zeitzulässigen Bereichs S_T ermittelt werden. Der Minimalpunkt von S_T entspricht dem Vektor $ES = (ES_i)_{i \in V}$ der frühesten Startzeitpunkte mit $ES_i = d_{0i}$ für alle $i \in V$. Die Bestimmung der Längen längster Wege d_{0i} von 0 nach i kann mit dem *Label-Correcting-Algorithmus* (vgl. Algorithmus 1.15 in Abschnitt 1.4.3) oder dem *Tripel-Algorithmus* (vgl. Algorithmus 1.20 in Abschnitt 1.4.3) erfolgen.

Beispiel 2.19. Betrachten wir den in Abbildung 2.22 dargestellten Projektnetzplan mit drei realen Vorgängen und einer maximalen Projektdauer von $\bar{d} = 7$. Mit Hilfe des Label-Correcting-Algorithmus soll nun eine optimale Lösung für die Projektdauerminimierung und die Minimierung der mittleren Durchlaufzeit bestimmt werden.

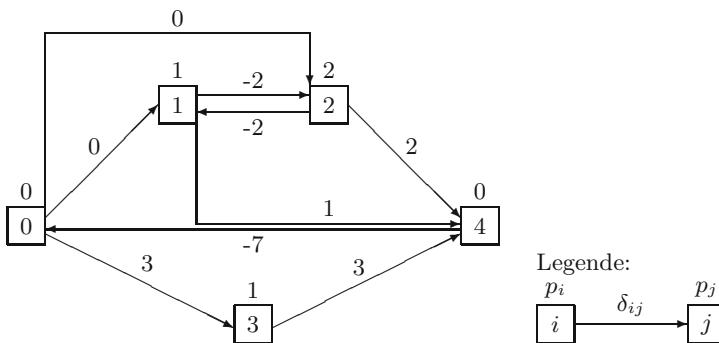


Abb. 2.22. Projektnetzplan mit drei realen Vorgängen

Im Initialisierungsschritt setzen wir $d_{00} := 0$, $p_{00} := 0$ und $Q := \{0\}$ sowie $d_{0i} := -\infty$ und $p_{0i} := -1$ für die Vorgänge $i = 1, 2, 3, 4$. Zu Beginn des ersten Hauptschrittes entfernen wir Vorgang 0 vom Kopf der Schlange Q . Für die Menge der unmittelbaren Nachfolger von Vorgang 0 ergibt

sich $Succ(0) = \{1, 2, 3\}$. Aktualisieren wir die Längen längster Wege dieser Vorgänge, so erhalten wir $d_{01} := 0$, $d_{02} := 0$, $d_{03} := 3$ und setzen $p_{0i} := 0$ für $i = 1, 2, 3$ sowie $Q := \{1, 2, 3\}$. Als Nächstes entfernen wir Vorgang $i = 1$ vom Kopf von Q und fahren auf gleiche Art und Weise fort. Die Ergebnisse der einzelnen Iterationen sind in Tabelle 2.7 angegeben.

Tabelle 2.7. Ermittlung längster Wege von Knoten 0 zu allen anderen Knoten

Iteration	0	1	2	3	4	5
$Q =$	$\{0\}$	$\{1, 2, 3\}$	$\{2, 3, 4\}$	$\{3, 4\}$	$\{4\}$	$\{\}$
Vorgang i	d_{0i}	p_{0i}	d_{0i}	p_{0i}	d_{0i}	p_{0i}
0	0	0	0	0	0	0
1	$-\infty$	-1	0	0	0	0
2	$-\infty$	-1	0	0	0	0
3	$-\infty$	-1	3	0	3	0
4	$-\infty$	-1	$-\infty$	-1	1	1
					2	2
					6	3
					6	3

Der Minimalpunkt von S_T entspricht also dem Vektor $ES = (0, 0, 0, 3, 6)$. Damit beträgt die minimale Projektdauer 6 Zeiteinheiten und die minimale mittlere Durchlaufzeit $\frac{13}{5} = 2,6$ Zeiteinheiten.

2.2.2 Minimierung der Summe gewichteter Startzeitpunkte

Bei der Zielfunktion der Minimierung der Summe gewichteter Startzeitpunkte handelt es sich um eine lineare Funktion. Ein Projektplanungsproblem (2.1) mit linearer Zielfunktion stellt ein lineares Programm dar, für das immer ein Extrempunkt des zeitzulässigen Bereichs optimal ist. Eine optimale Lösung kann daher mit Hilfe eines beliebigen Verfahrens der linearen Programmierung, z.B. dem Simplex-Verfahren, ermittelt werden.

Wesentlich effizienter ist es aber in diesem speziellen Fall, das zu (2.1) duale Problem zu lösen, welches einem kostenminimalen Flussproblem entspricht.⁸ Dualisieren wir Problem (2.1) mit Zielfunktion (WST), d.h.

$$\begin{aligned}
 & \text{Minimiere } f(S) := \sum_{i \in V} w_i S_i \\
 \text{u.d.N. } & \begin{aligned} S_j - S_i & \geq \delta_{ij} & ((i, j) \in E) \\ S_0 & = 0 \\ S_i & \geq 0 & (i \in V), \end{aligned}
 \end{aligned}$$

so erhalten wir das lineare Programm

⁸ Eine Einführung in die Dualitätstheorie findet man z.B. in DOMSCHKE und DREXL (2005) oder NEUMANN und MORLOCK (2002).

$$\left. \begin{array}{l} \text{Maximiere } \sum_{\langle i,j \rangle \in E} \delta_{ij} \cdot x_{ij} + 0 \cdot z \\ \text{u.d.N. } \begin{aligned} z + \sum_{\langle h,0 \rangle \in E} x_{h0} - \sum_{\langle 0,j \rangle \in E} x_{0j} &= w_0 \\ \sum_{\langle h,i \rangle \in E} x_{hi} - \sum_{\langle i,j \rangle \in E} x_{ij} &= w_i \quad (i \in V \setminus \{0\}) \\ x_{ij} &\geq 0 \quad (\langle i,j \rangle \in E) \\ z &\in \mathbb{R}, \end{aligned} \end{array} \right\} \quad (2.5)$$

wobei $Pred(i)$ wieder die Menge der Vorgänger und $Succ(i)$ die Menge der Nachfolger von Knoten i im Projektnetzplan bezeichnet. Die Dualvariable z ist nicht vorzeichenbeschränkt und der zugehörige Zielfunktionskoeffizient ist 0, daher können wir z beliebig wählen, z.B. $z := \sum_{i \in V} w_i$. Multiplizieren wir weiter die Zielfunktion mit -1 , ergibt sich das zu (2.5) äquivalente Minimierungsproblem

$$\left. \begin{array}{l} \text{Minimiere } \sum_{\langle i,j \rangle \in E} -\delta_{ij} \cdot x_{ij} \\ \text{u.d.N. } \begin{aligned} \sum_{\langle h,i \rangle \in E} x_{hi} - \sum_{\langle i,j \rangle \in E} x_{ij} &= w_i \quad (i \in V) \\ x_{ij} &\geq 0 \quad (\langle i,j \rangle \in E), \end{aligned} \end{array} \right\} \quad (2.6)$$

mit $w_0 = -\sum_{i \in V \setminus \{0\}} w_i$. Problem (2.6) entspricht einem *kostenminimalen Flussproblem* mit einer unteren Schranke (Minimalkapazität) $\lambda_{ij} = 0$ für alle $x_{ij}, \langle i,j \rangle \in E$. Außerdem gilt aufgrund der geschickten Wahl von w_0 , dass $\sum_{i \in V} w_i = 0$. Die Variable x_{ij} gibt an, wie viele Flusseinheiten von Knoten i zu Knoten j transportiert werden. In der Literatur wird gemeinhin angenommen, dass für jede Flussvariable $x_{ij}, \langle i,j \rangle \in E$, neben einer unteren Schranke λ_{ij} auch eine obere Schranke κ_{ij} vorgegeben ist. Im vorliegenden Fall gilt dann $\kappa_{ij} = \infty$ für alle $\langle i,j \rangle \in E$. Das Gewicht w_i bezeichnet für jeden Knoten $i \in V$ den Bedarf an Flusseinheiten. Entsprechend bezeichnen wir einen Knoten i mit $w_i > 0$ als Nachfrageknoten, einen Knoten i mit $w_i < 0$ als Angebotsknoten und einen Knoten mit $w_i = 0$ als Umladeknoten. Gesucht ist ein Fluss mit minimalen Kosten, der den Flussbedingungen genügt, d.h. dass für alle Knoten $i \in V$ die Summe der in Knoten i einmündenden Flusseinheiten abzüglich der aus Knoten i ausmündenden Flusseinheiten gerade w_i entspricht.

Zunächst konstruieren wir das dem Flussproblem (2.6) zugrunde liegende Flussnetzwerk $N^F = \langle V^F, E^F; \delta^F, \lambda, \kappa \rangle$. Knoten- und Pfeilmenge des Flussnetzwerks N^F entsprechen der Knoten- und Pfeilmenge des ursprünglichen Projektnetzplans $N = \langle V, E; \delta \rangle$. Für die Pfeile $\langle i,j \rangle \in E^F$ ergeben sich die Bewertungen $\delta_{ij}^F := -\delta_{ij}$ sowie die Kapazitäten $\lambda_{ij} := 0$ und $\kappa_{ij} := \infty$. Knoten 0 erhält das Gewicht $w_0 := -\sum_{i \in V \setminus \{0\}} w_i$, während alle anderen Knoten ihr Gewicht beibehalten.

Beispiel 2.20. Betrachten wir den Projektnetzplan N in Abbildung 2.23 mit drei realen Vorgängen und einer maximalen Projektdauer von $\bar{d} = 4$. Die Knoten i des Netzplans sind mit Gewichten w_i bewertet.

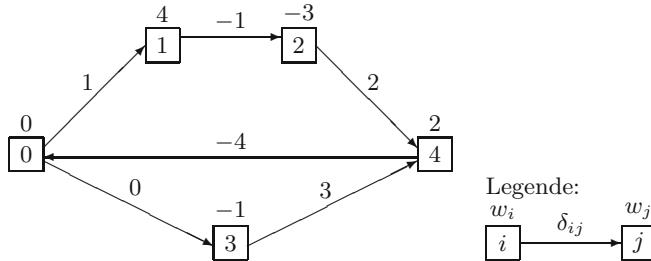


Abb. 2.23. Projektnetzplan N mit drei realen Vorgängen

Das *Flussnetzwerk* N^F ergibt sich aus dem Projektnetzplan N , indem wir den Pfeilen $\langle i, j \rangle \in E^F$ die Bewertungen $\delta_{ij}^F := -\delta_{ij}$ zuordnen und das Gewicht des Knotens 0 auf $w_0 := -\sum_{i \in V \setminus \{0\}} w_i = -2$ setzen. Das resultierende Flussnetzwerk N^F ist in Abbildung 2.24 dargestellt.

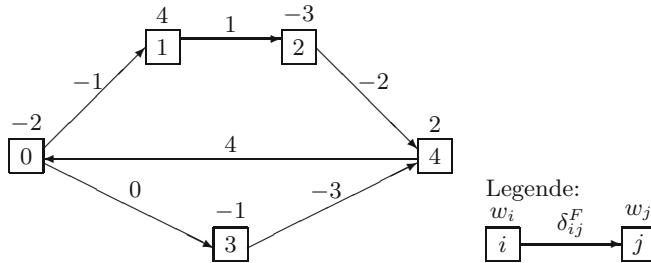


Abb. 2.24. Flussnetzwerk $N^F = \langle V^F, E^F; \delta^F, \lambda, \kappa \rangle$

Aus dem folgenden *Satz vom komplementären Schlupf* folgt, dass aus einer optimalen Lösung $(x_{ij})_{(i,j) \in E}$ des Flussproblems (2.6) eine optimale Lösung für Problem (2.1) mit Zielfunktion (WST) konstruiert werden kann, indem wir das lineare Gleichungssystem $S_j - S_i = \delta_{ij}$ für alle $\langle i, j \rangle \in E$ mit $x_{ij} > 0$ lösen; vgl. DOMSCHKE und DREXL (2005, S. 34).

Satz 2.21 (Satz vom komplementären Schlupf). Gegeben seien zwei zueinander duale lineare Programme. Ist für eine optimale Lösung des einen Problems eine der Nebenbedingungen nicht bindend, d.h. sie hat Schlupf, dann ist die mit dieser Nebenbedingung korrespondierende Variable des anderen Problems in einer optimalen Lösung gleich 0. Ist andererseits in einer optimalen Lösung des einen Problems eine der Entscheidungsvariablen positiv, so ist die mit dieser Variablen korrespondierende Nebenbedingung des anderen Problems in einer optimalen Lösung bindend, d.h. sie hat keinen Schlupf.

Wie wir noch sehen werden, stellt für das in Beispiel 2.20 vorgestellte Problem der Fluss $x_{01} = 4$, $x_{24} = 3$, $x_{34} = 1$, $x_{40} = 2$ und $x_{ij} = 0$ für alle

übrigen Variablen einen optimalen Fluss dar. Aus Satz 2.21 können wir nun folgern, dass in einer optimalen Lösung von Problem (2.1) die Nebenbedingungen $S_0 - S_4 \geq -4$, $S_1 - S_0 \geq 1$, $S_4 - S_2 \geq 2$ und $S_4 - S_3 \geq 3$ bindend, d.h. mit Gleichheit erfüllt sind. Die Lösung $S = (0, 1, 2, 1, 4)$ des entsprechenden linearen Gleichungssystems

$$\begin{aligned} S_0 - S_4 &= -4 \\ S_1 - S_0 &= 1 \\ S_4 - S_2 &= 2 \\ S_4 - S_3 &= 3 \\ S_0 &= 0 \end{aligned}$$

entspricht daher einer optimalen Lösung für die Minimierung der Summe gewichteter Startzeitpunkte.

Eine optimale Lösung für Flussproblem (2.6) kann ermittelt werden, indem wir einen zulässigen Ausgangsfluss sukzessive verbessern. Zur Bestimmung eines zulässigen Ausgangsflusses x für das Flussnetzwerk N^F ergänzen wir die Knotenmenge V^F um einen Umladeknoten $n + 2$ mit dem Gewicht $w_{n+2} := 0$. Dann fügen wir für jeden Nachfrageknoten l einen Pfeil $\langle n + 2, l \rangle$ und für jeden Angebotsknoten h einen Pfeil $\langle h, n + 2 \rangle$ mit der Bewertung $\delta_{n+2,l}^F := \sum_{\langle i,j \rangle \in E} |\delta_{ij}|$ bzw. $\delta_{h,n+2}^F := \sum_{\langle i,j \rangle \in E} |\delta_{ij}|$ ein. Der Transport einer Flusseinheit auf diesen Pfeilen ist also teurer als auf einem beliebigen Weg im Flussnetzwerk, der Knoten $n + 2$ nicht enthält. Alle eingeführten Pfeile erhalten eine Minimalkapazität von Null und eine Maximalkapazität von ∞ . Das entsprechend erweiterte Flussnetzwerk bezeichnen wir mit $\hat{N}^F = \langle \hat{V}^F, \hat{E}^F, \delta^F, \lambda, \kappa \rangle$. Ein zulässiger Ausgangsfluss x im erweiterten Flussnetzwerk ergibt sich nun ganz kanonisch, indem wir $x_{i,n+2} := -w_i$ für alle Pfeile $\langle i, n + 2 \rangle$ und $x_{n+2,i} := w_i$ für alle Pfeile $\langle n + 2, i \rangle$ setzen. Die übrigen Pfeile erhalten einen Fluss von 0. Zur sukzessiven Verbesserung des Ausgangsflusses benötigen wir das zum gegebenen Fluss x gehörige *Inkrementnetzwerk* $N^I = \langle V^I, E^I, \delta^I, \lambda, \kappa \rangle$, das wie folgt definiert ist. Für die Knotenmenge des Inkrementnetzwerks gilt $V^I := \hat{V}^F$. Die Pfeilmenge des Inkrementnetzwerks entspricht zunächst der Pfeilmenge \hat{E}^F des erweiterten Flussnetzwerks mit den jeweiligen Bewertungen. Gilt $x_{ij} > 0$ für einen Pfeil $\langle i, j \rangle \in \hat{E}^F$, dann ergänzen wir das Inkrementnetzwerk N^I um einen Rückwärtspfeil $\langle j, i \rangle$ mit der Bewertung $-\delta_{ij}^F$ und den unteren bzw. oberen Kapazitäten $\lambda_{ji} = 0$ bzw. $\kappa_{ji} = x_{ij}$. Die Optimalität des aktuellen Flusses x kann nun mit Hilfe des folgenden aus der Graphentheorie bekannten Satzes überprüft werden; vgl. z.B. NEUMANN und MORLOCK (2002, Abschnitt 2.6.6).

Satz 2.22. Eine zulässige Lösung (Fluss) für das Flussproblem (2.6) ist genau dann optimal, falls das entsprechende Inkrementnetzwerk keinen Zyklus negativer Länge besitzt.

Beinhaltet das Inkrementnetzwerk N^I einen Zyklus negativer Länge, so kann der aktuelle Fluss verbessert werden. Dazu bestimmen wir einen solchen

Zyklus, sagen wir Zyklus C , und ermitteln die minimale Maximalkapazität aller in C enthaltenen Pfeile, d.h. $\kappa^* = \min_{\langle i,j \rangle \in C} \{\kappa_{ij}\}$.⁹ Seien $E^+ := \{\langle i,j \rangle \in C \mid \langle i,j \rangle \in \hat{N}^F\}$ und $E^- := \{\langle i,j \rangle \in C \mid \langle i,j \rangle \notin \hat{N}^F \wedge \langle j,i \rangle \in \hat{N}^F\}$. Dann wird nun im erweiterten Flussnetzwerk \hat{N}^F auf allen Pfeilen $\langle i,j \rangle \in E^+$ der Fluss x_{ij} um κ^* Einheiten erhöht und auf allen Pfeilen $\langle j,i \rangle$, für die $\langle i,j \rangle \in E^-$ gilt, um κ^* Einheiten verringert. Ausgehend von der so erhaltenen neuen Lösung x wird abermals das zugehörige Inkrementnetzwerk konstruiert und ein Zyklus negativer Länge gesucht. Diese Schritte werden solange wiederholt, bis das Inkrementnetzwerk für den aktuellen Fluss schließlich keinen Zyklus negativer Länge mehr enthält. Der aktuelle Fluss ist dann optimal. Ein zugehöriger optimaler Schedule für das zugrunde liegende Projektplanungsproblem kann dann, wie bereits gesagt, mit dem Satz vom komplementären Schlupf bestimmt werden.

Beispiel 2.23. Betrachten wir abermals den Projektnetzplan N in Abbildung 2.23 und das zugehörige Flussnetzwerk N^F in Abbildung 2.24. Um für das Flussnetzwerk einen zulässigen Ausgangsfluss zu bestimmen, fügen wir den Umladeknoten 5 mit dem Gewicht $w_5 = 0$ ein. Weiter ist für jeden Nachfrageknoten $l = 1, 4$ ein Pfeil $\langle 5, l \rangle$ und für jeden Angebotsknoten $h = 0, 2, 3$ ein Pfeil $\langle h, 5 \rangle$ mit der Bewertung $\delta_{5l}^F = 11$ bzw. $\delta_{h5}^F = 11$ in das Flussnetzwerk aufzunehmen. Das resultierende erweiterte Flussnetzwerk \hat{N}^F ist mit einem zulässigen Ausgangsfluss $x_{05} := 2$, $x_{25} := 3$, $x_{35} := 1$, $x_{51} := 4$, $x_{54} := 2$ und $x_{ij} = 0$ für alle übrigen Variablen in Abbildung 2.25 dargestellt.

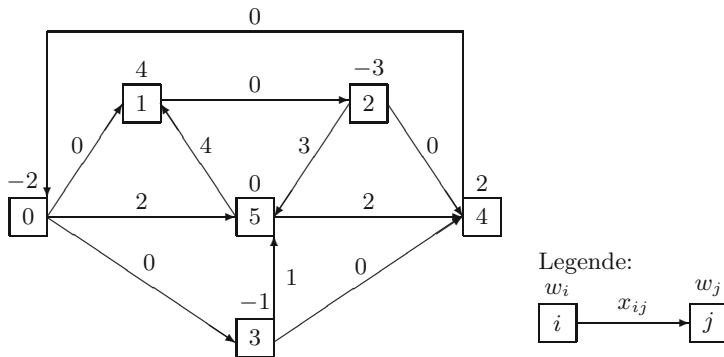
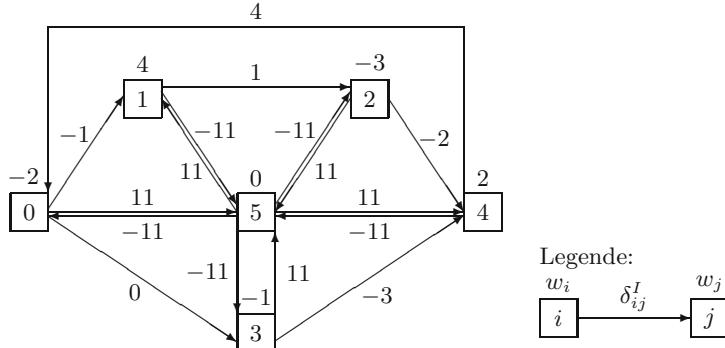


Abb. 2.25. Ausgangsfluss im erweiterten Flussnetzwerk \hat{N}^F

Das zum Ausgangsfluss in \hat{N}^F gehörende Inkrementennetzwerk N^I erhalten wir, indem wir für alle $\langle i, j \rangle \in \hat{E}^F$ mit $x_{ij} > 0$ das erweiterte Flussnetzwerk \hat{N}^F um einen Rückwärtspfeil $\langle j, i \rangle$ mit den Bewertungen $-\delta_{ij}^F$ sowie $\lambda_{ji} = 0$ und $\kappa_{ji} = x_{ij}$ ergänzen (vgl. Abb. 2.26).

⁹ Um die Notation zu vereinfachen, stellen wir einen Zyklus $\langle 1, 2, 3, 1 \rangle$ in diesem Abschnitt als Pfeilfolge $\{\langle 1, 2 \rangle, \langle 2, 3 \rangle, \langle 3, 1 \rangle\}$ dar.

Abb. 2.26. Inkrementnetzwerk N^I

Das Inkrementnetzwerk N^I besitzt mehrere Zyklen negativer Länge. Die Länge von Zyklus $C^1 = \{\langle 5, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 5 \rangle\}$ beträgt beispielsweise $\delta_{50} + \delta_{01} + \delta_{15} = -23$. Wir konstruieren die zugehörigen Mengen $E^+ := \{\langle 0, 1 \rangle\}$ und $E^- := \{\langle 5, 0 \rangle, \langle 1, 5 \rangle\}$. Die minimale Maximalkapazität im Zyklus C^1 ist $\kappa^* := \min\{2, \infty, 4\} = 2$. Daher erhöhen wir im Netzwerk \hat{N}^F den Fluss auf dem Pfeil $\langle 0, 1 \rangle$ um 2 Einheiten, auf den Pfeilen $\{\langle 0, 5 \rangle, \langle 5, 1 \rangle\}$ wird der Fluss um 2 Einheiten verringert. Es ergeben sich veränderte Flusswerte $x_{05} = 0, x_{01} = 2$ und $x_{51} = 2$. Das zu dem resultierenden Fluss konstruierte Inkrementnetzwerk besitzt den Zyklus $C^2 = \{\langle 5, 2 \rangle, \langle 2, 4 \rangle, \langle 4, 5 \rangle\}$, für den die kumulierten Kosten -24 betragen. Es ergeben sich die Mengen $E^+ := \{\langle 2, 4 \rangle\}$ und $E^- := \{\langle 5, 2 \rangle, \langle 4, 5 \rangle\}$. Die minimale Maximalkapazität im Zyklus C^2 ist $\min\{3, \infty, 2\} = 2$; somit wird der Fluss in \hat{N}^F auf dem Pfeil $\langle 2, 4 \rangle$ um 2 Einheiten erhöht und auf den Pfeilen $\{\langle 2, 5 \rangle, \langle 5, 4 \rangle\}$ um 2 Einheiten verringert. Damit ergeben sich die neuen Flusswerte $x_{25} = 1, x_{24} = 2$ und $x_{54} = 0$. Das zu dem erhaltenen Fluss gehörige Inkrementnetzwerk beinhaltet den Zyklus $C^3 = \{\langle 5, 3 \rangle, \langle 3, 4 \rangle, \langle 4, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 5 \rangle\}$ mit den Gesamtkosten -22 und einer minimalen Maximalkapazität von $\min\{1, \infty, \infty, \infty, 2\} = 1$. Wir konstruieren die Mengen $E^+ := \{\langle 3, 4 \rangle, \langle 4, 0 \rangle, \langle 0, 1 \rangle\}$ und $E^- := \{\langle 5, 3 \rangle, \langle 1, 5 \rangle\}$. Erhöhen wir den Fluss in \hat{N}^F auf den Pfeilen $\{\langle 3, 4 \rangle, \langle 4, 0 \rangle, \langle 0, 1 \rangle\}$ um eine Einheit und verringern wir den Fluss auf den Pfeilen $\{\langle 3, 5 \rangle, \langle 5, 1 \rangle\}$ um eine Einheit, dann erhalten wir die Flusswerte $x_{35} = 0, x_{34} = 1, x_{40} = 1, x_{01} = 3$ und $x_{51} = 1$. Im neu konstruierten Inkrementnetzwerk existiert nur noch der Zyklus $C^4 = \{\langle 5, 2 \rangle, \langle 2, 4 \rangle, \langle 4, 0 \rangle, \langle 0, 1 \rangle, \langle 1, 5 \rangle\}$ mit negativen kumulierten Kosten in Höhe von -21 und einer minimalen Maximalkapazität von $\min\{1, \infty, \infty, \infty, 1\} = 1$. Wird der Fluss in \hat{N}^F auf den Pfeilen $\{\langle 2, 4 \rangle, \langle 4, 0 \rangle, \langle 0, 1 \rangle\}$ um eine Einheit erhöht und auf den Pfeilen $\{\langle 2, 5 \rangle, \langle 5, 1 \rangle\}$ um eine Einheit verringert, so ergibt sich der in Abbildung 2.27 dargestellte Fluss: $x_{01} = 4, x_{24} = 3, x_{34} = 1$ und $x_{40} = 2$ sowie $x_{ij} = 0$ für alle übrigen Variablen. Das entsprechende Inkrementnetzwerk enthält keinen Zyklus negativer Länge. Somit ist der gefundene Fluss optimal. Der zugehörige Schedule kann nun mit dem Satz vom komplementären Schlupf bestimmt werden. Wir

erhalten mit $S = (0, 1, 2, 1, 4)$ die optimale Lösung für das Problem der Minimierung der Summe gewichteter Startzeitpunkte.

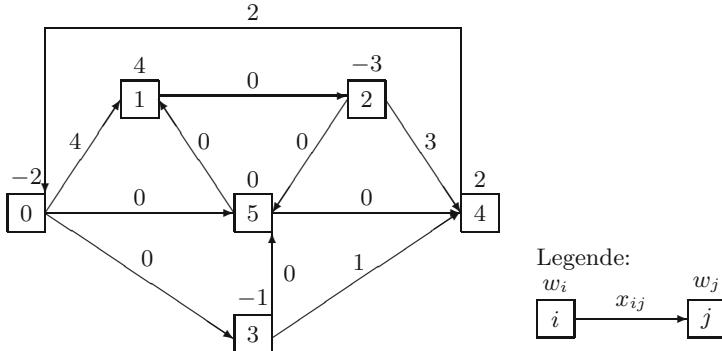


Abb. 2.27. Optimaler Fluss im erweiterten Flussnetzwerk \hat{N}^F

2.2.3 Maximierung des Kapitalwertes

Wie bereits in Abschnitt 2.1.4 beschrieben, lässt sich das Problem der Kapitalwertmaximierung in ein lineares Programm transformieren. Zur Bestimmung einer optimalen Lösung kann folglich das Simplex-Verfahren verwendet werden.

Eine weitaus effizientere Methode zur Bestimmung eines Schedules mit maximalem Kapitalwert ist jedoch das im Folgenden beschriebene *Anstiegsverfahren*. Aus didaktischen Gründen betrachten wir dabei anstelle der zu minimierenden Zielfunktion (NPV) die äquivalente zu maximierende Zielfunktion $f(S) := \sum_{i \in V} c_i^F e^{-\alpha S_i}$. Ausgehend von einem Extrempunkt $S \in \mathcal{S}_T$ bestimmen wir in jeder Iteration des Anstiegsverfahrens eine möglichst steile zulässige Anstiegsrichtung z . Sei $\phi_i := \partial f(S) / \partial S_i$ die partielle Ableitung der Zielfunktion f nach S_i im Punkt S , die die Veränderung des Kapitalwertes angibt, den eine marginale Erhöhung des Startzeitpunktes S_i von Vorgang i bewirkt. Dann wählen wir eine zulässige Richtung z , für die die so genannte Richtungsableitung $z^T \nabla f(S) := \sum_{i \in V} z_i \phi_i$ maximal ist (steilste Anstiegsrichtung). Weiterhin bestimmen wir eine Schrittweite λ , die angibt, wie weit wir uns in Richtung z bewegen können, bevor eine der vorgegebenen Zeitbeziehungen verletzt oder die Richtungsableitung negativ wird (letzteres passiert nicht, wenn wir in eine binäre Richtung gehen). Ausgehend von einem Extrempunkt S bewegen wir uns so zu einem zeitzulässigen Schedule $S' := S + \lambda z$ mit höherem Kapitalwert.

Betrachten wir zunächst die Bestimmung einer zulässigen *steilsten Anstiegsrichtung* z für einen beliebigen Extrempunkt $S \in \mathcal{S}_T$. In jedem Extrempunkt $S \in \mathcal{S}_T$ sind mindestens $n + 1$ linear unabhängige Nebenbedingungen bindend. Daher lässt sich S als Gerüst $G = \langle V, E_G \rangle$ des Projektnetzplans

N darstellen, wobei jeder Pfeil $\langle i, j \rangle \in E_G$ zu einer bindenden Nebenbedingung $S_j - S_i = \delta_{ij}$ gehört (vgl. Abschnitt 2.1.4). Wählen wir zur Normierung des Richtungsvektors z die Maximumsnorm, d.h. $\|z\|_\infty = \max_{i \in V} |z_i|$, dann kann die Bestimmung einer zulässigen steilsten Anstiegsrichtung in S mit dem zugehörigen Gerüst G als lineares Programm in den Veränderlichen z_i wie folgt formuliert werden

$$\left. \begin{array}{l} \text{Maximiere } z^T \nabla f(S) := \sum_{i \in V} -\alpha c_i^F e^{-\alpha S_i} z_i \\ \text{u.d.N.} \quad \begin{array}{l} z_j - z_i \geq 0 \quad (\langle i, j \rangle \in E_G) \\ z_0 = 0 \\ -1 \leq z_i \leq 1 \quad (i \in V) \end{array} \end{array} \right\} \quad (2.7)$$

Die Nebenbedingungen $z_j - z_i \geq 0$, $\langle i, j \rangle \in E_G$, stellen sicher, dass bei einer Verzögerung des Startzeitpunktes von Vorgang i um $z_i > 0$ Zeiteinheiten auch alle Nachfolger j von i im Gerüst G um $z_j \geq z_i$ Zeiteinheiten verzögert werden, da alle Pfeile im zugrunde liegenden Gerüst bindende Zeitbeziehungen repräsentieren. Bei einer Verzögerung des Startzeitpunktes von Vorgang j um $z_j < 0$ Zeiteinheiten, d.h. Vorgang j wird z_j Zeiteinheiten früher gestartet, müssen aufgrund der bindenden Zeitbeziehungen alle Vorgänger i von j im Gerüst G um $z_i \leq z_j$ Zeiteinheiten früher starten. Die Bedingung $z_0 = 0$ besagt, dass der Projektstart weder vorgezogen noch verzögert werden darf. Die beiden Nebenbedingungen sorgen somit dafür, dass es sich bei z um eine zulässige Anstiegsrichtung handelt. Durch die Zielfunktion wird gewährleistet, dass wir eine steilste Anstiegsrichtung erhalten.

Es lässt sich zeigen, dass die Koeffizientenmatrix von Problem (2.7) total unimodular ist, d.h. die Determinante jeder quadratischen Teilmatrix nimmt nur die Werte 0, 1 bzw. -1 an. Aufgrund dieser Eigenschaft der Koeffizientenmatrix und der gewählten Normierung existiert für Problem (2.7) immer eine optimale Lösung $z \in \{-1, 0, 1\}^{n+2}$. SCHWINDT und ZIMMERMANN (1998) zeigen ferner, dass ausgehend vom ES -Schedule immer eine zulässige steilste Anstiegsrichtung $z \in \{0, 1\}^{n+2}$ für unser Anstiegsverfahren existiert. Sei S^1, \dots, S^r eine endliche Folge von Schedules mit $S^1 = ES$ und $S^{q+1} = S^q + \lambda^q z^q$ für $q = 1, \dots, r-1$, wobei $\lambda^q \in \mathbb{R}_{\geq 0}$ ist, und $z^q \neq \mathbf{0}$ für alle $q = 1, \dots, r$ eine steilste Anstiegsrichtung für Schedule S^q repräsentiert. Dann gilt für jede optimale Lösung z^r des Problems (2.7) für Schedule S^r , dass $z^r \geq \mathbf{0}$, d.h. in jedem Schritt kann $z \in \{0, 1\}^{n+2}$ gewählt werden. Die Folge der mit Hilfe des steilsten Anstiegsverfahrens generierten Schedules S^1, \dots, S^r ist somit komponentenweise monoton wachsend, d.h. $S^1 \leq S^2 \leq \dots \leq S^{r-1} \leq S^r$.

Die Bestimmung einer steilsten Anstiegsrichtung $z \in \{0, 1\}^{n+2}$ für Schedule S mit zugehörigem Gerüst G basiert auf dem folgenden Lemma.

Lemma 2.24. Sei ϱ_i die Anzahl der Pfeile auf einem Semiweg (ungerichtete Pfeilfolge) von Knoten 0 zu Knoten i im Gerüst G des Projektnetzplans N . Die Vorgänge i des Gerüstes mit maximalem ϱ_i stellen dann entweder Quellen mit genau einem Nachfolger oder Senken mit genau einem Vorgänger dar.

In Abbildung 2.28 ist ein Gerüst mit den zugehörigen Zahlungen c_i^F für alle Vorgänge $i \in V$ angegeben. Für $i = 3, 4$ ist $\varrho_i = 2$ maximal, wobei Vorgang 3 eine Senke mit genau einem Vorgänger und Vorgang 4 eine Quelle mit genau einem Nachfolger darstellt. Wir skizzieren nun anhand dieses Beispiels, wie man eine steilste Anstiegsrichtung bestimmt, bevor wir im Anschluss eine allgemeine Beschreibung des Vorgehens zur Bestimmung einer steilsten Anstiegsrichtung geben.

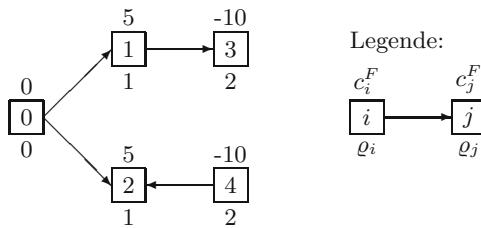


Abb. 2.28. Gerüst mit $\varrho_3 = \varrho_4 = 2$

Da die partielle Ableitung $\phi_3 = -\alpha c_3^F e^{-\alpha S_3}$ der Zielfunktion f nach S_3 im Punkt S für beliebige $\alpha > 0$ positiv ist, führt eine Erhöhung von S_3 auch zu einer Erhöhung des Projektkapitalwertes. Wir setzen daher $z_3 := 1$. Für den Vorgänger von Vorgang 3, Vorgang 1, ist $\phi_1 < 0$, d.h. eine Verzögerung des Vorgangs führt zu einer Verminderung des Kapitalwertes. Aus diesem Grund setzen wir $z_1 := 0$. Für Vorgang 4 ist die partielle Ableitung ϕ_4 positiv und eine Verzögerung von Vorgang 4 führt zu einer Erhöhung des Kapitalwertes. Vorgang 4 hat im Gerüst G jedoch Knoten 2 als Nachfolger, d.h. es existiert eine bindende Zeitbeziehung zwischen den Vorgängen 4 und 2. Wollen wir Vorgang 4 verzögern, so müssen wir Vorgang 2 gemeinsam mit Vorgang 4 verzögern, um die im zugrunde liegenden Netzplan gegebenen Zeitbeziehungen einzuhalten. Wir betrachten daher die Summe der partiellen Ableitungen der beiden Vorgänge. Ist $\phi_2 + \phi_4 > 0$, d.h. die gemeinsame Verzögerung von Vorgang 2 und 4 führt zu einer Erhöhung des Kapitalwertes, so setzen wir $z_2 := z_4 := 1$. Andernfalls gilt $z_2 := z_4 := 0$, d.h. die Vorgänge 2 und 4 werden nicht verzögert. Da wir annehmen, dass Vorgang 0 zum Zeitpunkt $S_0 = 0$ startet, gilt naturgemäß $z_0 := 0$.

Zur Bestimmung einer steilsten zulässigen Anstiegsrichtung gehen wir nun im Einzelnen wie folgt vor. Seien S ein Extrempunkt des zeitzulässigen Bereichs S_T und G ein zugehöriges Gerüst. Ausgehend von den Vorgängen i mit maximaler Pfeilanzahl ϱ_i bestimmen wir dann Teilgerüste, deren Verzögerung zu einer Erhöhung des Projektkapitalwertes führen. Für die Vorgänge i eines solchen Teilgerüstes setzen wir $z_i := 1$.

Wir bestimmen dazu zunächst die Werte ϱ_i für alle $i \in V \setminus \{0\}$ und sortieren sie nach nichtwachsenden Werten. Anschließend ermitteln wir die partiellen Ableitungen $\phi_i = \partial f(S) / \partial S_i$ von f nach S_i im Punkt S . Wir entnehmen

jeweils den Knoten i mit maximalem ϱ_i und untersuchen, ob es sich um eine Quelle oder eine Senke im zugrunde liegenden Gerüst G handelt. Ist Vorgang i eine Senke, dann ist zu prüfen, ob die Verzögerung von Vorgang i zu einer Erhöhung des Zielfunktionswertes führt. Dies ist für $\phi_i > 0$ der Fall und wir setzen $z_i := 1$. Führt allerdings die Verzögerung des Vorgangs i nicht zu einer Erhöhung des Kapitalwertes, d.h. $\phi_i \leq 0$, dann „verschmelzen“ wir Vorgang i mit seinem eindeutigen Vorgänger j , indem wir $\phi_j := \phi_j + \phi_i$ setzen.¹⁰ ϕ_j gibt dann die Veränderung des Kapitalwertes an, wenn wir das Teilgerüst bestehend aus den Knoten i und j verzögern. Da Knoten i mit seinem Vorgänger j verschmolzen wurde, braucht Knoten i im Folgenden nicht weiter betrachtet zu werden.

Handelt es sich bei Vorgang i um eine Quelle in G , so kann Vorgang i nur gemeinsam mit seinem eindeutig direkten Nachfolger j in G in einer binären Richtung $z_i \in \{0, 1\}$ verzögert werden, da zwischen i und j eine bindende Zeitbeziehung existiert. Gehen wir vom ES -Schedule aus, so ist ϕ_i für eine Quelle i außerdem immer positiv. Wir verschmelzen daher Quelle i stets mit ihrem Nachfolger j , setzen $\phi_j := \phi_j + \phi_i$ und brauchen Knoten i im Weiteren nicht mehr zu berücksichtigen.

Diese Schritte werden solange ausgeführt, bis alle Vorgänge $i \in V \setminus \{0\}$ betrachtet wurden. Ist die Verzögerung eines Teilgerüstes vorteilhaft, d.h. die (kumulierte) partielle Ableitung ϕ_j des Knotens j , der ein Teilgerüst repräsentiert, ist positiv, so werden alle zu diesem Teilgerüst gehörenden Vorgänge verzögert. Bezeichne $C(j)$ die Menge der mit Vorgang j verschmolzenen Vorgänge, dann setzen wir also $z_i := 1$ für alle $i \in C(j)$. Für die mit Knoten 0 verschmolzenen Vorgänge $i \in C(0)$ gilt $z_i := 0$. Algorithmus 2.25 fasst die beschriebene Vorgehensweise zusammen.

Algorithmus 2.25 (Bestimmung einer steilsten Anstiegsrichtung z für Schedule S).

Setze $z_i := 0$, $\phi_i := \partial f(S)/\partial S_i = -\alpha c_i^F e^{-\alpha S_i}$ und $C(i) := \{i\}$ für alle $i \in V$.

Bestimme ein zu S gehöriges Gerüst G und setze $V' := V \setminus \{0\}$.

Bestimme ϱ_i für alle $i \in V'$.

Solange $V' \neq \emptyset$:

Entnimm einen Vorgang i mit größtem ϱ_i aus V' .

Falls i eine Quelle im aktuellen Gerüst G darstellt, d.h. i besitzt keinen Vorgänger, aber genau einen Nachfolger j :

Verschmelze Knoten i mit Knoten j und setze $\phi_j := \phi_j + \phi_i$ und $C(j) := C(j) \cup C(i)$.

Falls i eine Senke im aktuellen Gerüst G darstellt, d.h. i besitzt keinen Nachfolger, aber genau einen Vorgänger j :

¹⁰ Eine Senke i mit maximalem ϱ_i besitzt nach Lemma 2.24 immer genau einen Vorgänger.

Falls $\phi_i > 0$: Setze $z_h := 1$ für alle $h \in C(i)$.

Andernfalls verschmelze Knoten i mit Knoten j und setze $\phi_j := \phi_j + \phi_i$ und $C(j) := C(j) \cup C(i)$.

Mit Hilfe von Algorithmus 2.25 bestimmen wir zum einen eine binäre steilste Anstiegsrichtung $z \in \{0, 1\}^{n+2}$ und zum anderen Teilgerüste mit der Knotenmenge $C(j)$, $j \in V$, die gemeinsam verschoben werden können. Hierbei ist zu beachten, dass die Verzögerung jedes einzelnen Teilgerüstes (d.h. die Verzögerung jedes Vorgangs dieses Teilgerüstes um denselben Betrag) eine Verschiebung in eine binäre Richtung darstellt und somit für sich eine Verbesserung des Zielfunktionswertes bewirkt.

Nachdem eine steilste Anstiegsrichtung $z \in \{0, 1\}^{n+2}$ ermittelt wurde, müssen wir eine *Schrittweite* λ berechnen, die angibt, wie weit wir uns in Richtung z bewegen. Da die Zielfunktion (NPV) binärmonoton und die bestimmte steilste Anstiegsrichtung $z \in \{0, 1\}^{n+2}$ ist, können wir die Schrittweite so groß wie möglich wählen, ohne dass eine der Nebenbedingungen $S_j - S_i \geq \delta_{ij}$, $\langle i, j \rangle \in E$, verletzt wird. Der Durchstoßpunkt $S' = S + \lambda z$ liegt also immer auf dem Rand des zeitzulässigen Bereichs (vgl. Abb. 2.29). Die *optimale Schrittweite* λ ergibt sich dabei gemäß

$$\lambda := \min_{i \in V: z_i=1} \min_{j \in V: z_j=0} \{S_j - S_i - \delta_{ij}\}, \quad (2.8)$$

mit $f(S + \lambda z) = \max_{\lambda \in \mathbb{R}_{\geq 0}} \{f(S + \lambda z) \mid S + \lambda z \in \mathcal{S}_T\}$. Im Allgemeinen stellt der Durchstoßpunkt $S' = S + \lambda z$ keinen Extrempunkt des zeitzulässigen Bereichs \mathcal{S}_T dar (vgl. Abb. 2.29).

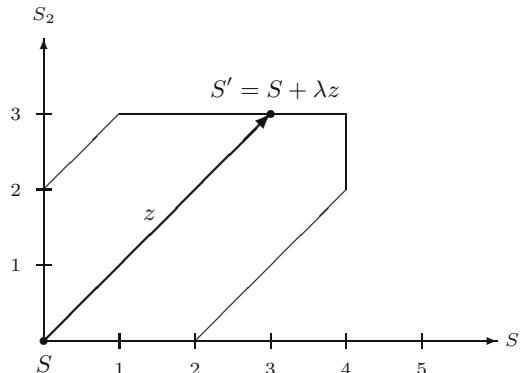


Abb. 2.29. Vorgehensweise des Anstiegsverfahrens

Verzögern wir alle Vorgänge $i \in V$ mit $z_i = 1$ um λ dann wird i.d.R. zunächst eine Zeitbeziehung $S_j - S_i \geq \delta_{ij}$ mit $z_i = 1$ und $z_j = 0$ bindend, d.h. eines der verzögerten Teilgerüste „dickt“ an das Teilgerüst der stehengebliebenen Vorgänge j mit $z_j = 0$ an. Weiteres Verzögern der verbleibenden

Teilgerüste bewirkt, aufgrund der binärmonotonen Eigenschaft der Zielfunktion (NPV) und da die steilste Anstiegsrichtung z binär ist, eine weitere Erhöhung des Kapitalwertes. Deshalb führen wir ausgehend von Schedule S' zur Beschleunigung des Verfahrens einen so genannten *Eckenanstieg* durch. Dabei wird nacheinander für alle Teilgerüste eine bindende Zeitbeziehung eingefügt. Auf diese Weise erhalten wir ein neues Gerüst und somit eine neue Ecke des zeitzulässigen Bereichs (vgl. Abb. 2.30).

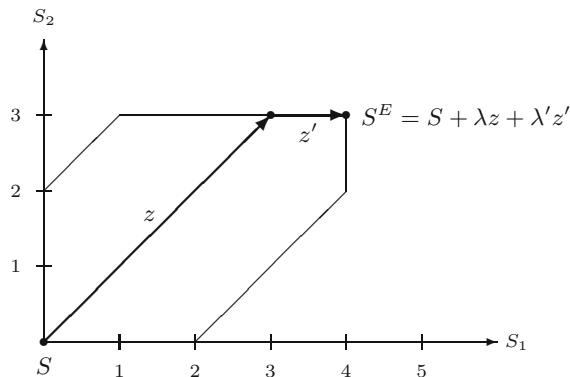


Abb. 2.30. Eckenanstieg

Zur Bestimmung der maximalen Schrittweite in den einzelnen Schritten des Eckenanstiegs gehen wir wie folgt vor. Seien G das zu Schedule S gehörende Gerüst und V^s die Menge aller Knoten, die verzögert werden, d.h. $V^s := \{i \in V \mid z_i = 1\}$. Für alle $i \in V \setminus V^s$ initialisieren wir $S_i^E := S_i$. Wir entfernen zunächst alle Pfeile $\langle i, j \rangle \in E_G$ mit $i \notin C(j)$ und $j \notin C(i)$ aus G , d.h. wir entfernen die Pfeile aus G , an denen die zu verschiebenden Teilgerüste „abgerissen“ werden. Danach bestimmen wir eine Schrittweite λ gemäß (2.8) und den zugehörigen Pfeil $\langle h, l \rangle$ mit $\lambda = S_l - S_h - \delta_{hl}$. Für das entsprechende Teilgerüst bzw. die Knotenmenge $C(j)$, die den Knoten h enthält, erhöhen wir die Startzeitpunkte um λ Zeiteinheiten, d.h. wir setzen $S_i^E := S_i + \lambda$. Danach werden die Knoten des Teilgerüstes $C(j)$ aus der Menge V^s eliminiert und der Pfeil $\langle h, l \rangle$ dem Gerüst G hinzugefügt. Diese Schritte werden solange wiederholt, bis die Menge V^s leer ist. Algorithmus 2.26 fasst die einzelnen Schritte zur Durchführung eines Eckenanstiegs zusammen.

Algorithmus 2.26 (Eckenanstieg).

Setze $V^s := \{i \in V \mid z_i = 1\}$.

Für alle $i \in V \setminus V^s$: Setze $S_i^E := S_i$.

Für alle $\langle i, j \rangle \in E_G$ mit $j \notin C(i)$ und $i \notin C(j)$: Setze $E_G := E_G \setminus \{\langle i, j \rangle\}$.

Solange $V^s \neq \emptyset$.

Ermittle $\langle h, l \rangle \in E$ mit $h \in V^s$ und $l \in V \setminus V^s$, so dass $\lambda := S_l^E - S_h - \delta_{hl} = \min_{i \in V: z_i=1} \min_{j \in V: z_j=0} (S_j^E - S_i - \delta_{ij})$.

Ermittle die Knotenmenge $C(j)$ mit $h \in C(j)$.

Für alle $i \in C(j)$: Setze $S_i^E := S_i + \lambda$ und $z_i := 0$.

$V^s := V^s \setminus C(j)$.

$E_G := E_G \cup \{\langle h, l \rangle\}$.

Rückgabe S^E .

In Algorithmus 2.26 werden die Teilgerüste einzeln nacheinander verschoben, indem die jeweiligen Startzeitpunkte erhöht werden. Formal ergibt sich die zweite Anstiegsrichtig z' aus der ersten Anstiegsrichtig z durch Nullsetzen der Komponenten aus der Menge $C(j)$. Die zweite Schrittweite $\lambda' \geq \lambda$ umfasst die Schrittweite von S nach S' sowie von S' nach S'' (vgl. Abb. 2.30).

Das Verfahren des steilsten Anstiegs zur Lösung von Kapitalwertmaximierungsproblemen unter Zeitrestriktionen ist zusammenfassend in Algorithmus 2.27 beschrieben und wird anhand von Beispiel 2.28 veranschaulicht. Die Zeitkomplexität von Algorithmus 2.25 zur Bestimmung einer steilsten Anstiegsrichtung für einen Schedule S lässt sich mit $\mathcal{O}(|V|)$ angeben. Für die Zeitkomplexität des Algorithmus 2.26 zur Durchführung eines Eckenanstiegs ergibt sich $\mathcal{O}(|E| \log |E|)$. Obwohl die Anzahl an benötigten Iterationen in Algorithmus 2.27 i.d.R. gering ist, konnte bislang nicht gezeigt werden, dass die Anzahl an Wiederholungen für beliebige Probleminstanzen polynomial beschränkt ist. Nichts desto trotz zeigen experimentelle Tests im Vergleich zur Lösung des auf S. 134 dargestellten LP-Ansatzes einen entscheidenden Laufzeitvorteil.

Algorithmus 2.27 (Verfahren des steilsten Anstiegs).

Ermittle den Schedule der frühesten Startzeitpunkte ES und zugehöriges Gerüst $G = \langle V, E_G \rangle$ des Netzwerkes N .

Setze $S := ES$.

Wiederhole:

Ermittle eine steilste Anstiegsrichtung z in S mit Algorithmus 2.25.

Falls $z \neq 0$:

Bestimme eine Ecke S^E mit Algorithmus 2.26.

Setze $S := S^E$.

Solange bis $z = 0$.

Rückgabe S

Beispiel 2.28. Wir betrachten den Projektnetzplan in Abbildung 2.31 mit vier realen Vorgängen und einer maximalen Projektdauer von $\bar{d} = 10$. Zur Maximierung des Projektkapitalwertes wenden wir Algorithmus 2.27 an, wobei wir $\alpha = 0,01$ wählen.

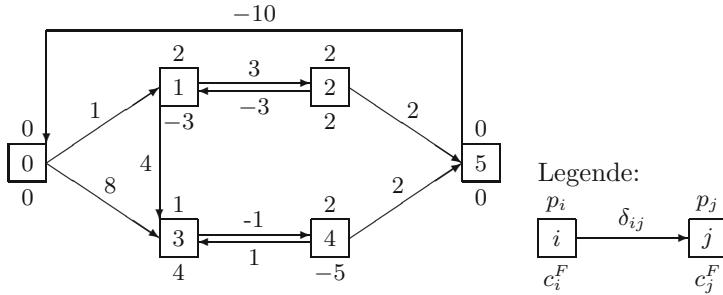
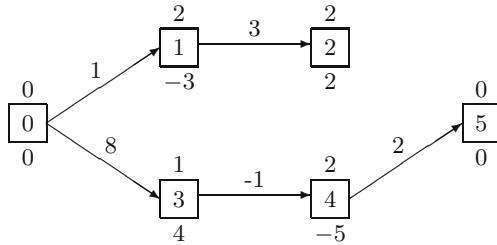


Abb. 2.31. Projektnetzplan mit vier realen Vorgängen

Wir erhalten den Vektor der frühesten Startzeitpunkte gemäß $ES = (0, 1, 4, 8, 7, 9)$. Ein zugehöriges Gerüst (Outtree) ist in Abbildung 2.32 angegeben.

Abb. 2.32. Gerüst zum ES -Schedule

Zu Beginn des Verfahrens zur Bestimmung einer steilsten Anstiegsrichtung setzen wir $z_i := 0$ und $C(i) := \{i\}$ für alle $i \in V$ sowie $V' := V \setminus \{0\}$. Weiter berechnen wir für alle $i \in V$ die partiellen Ableitungen $\phi_i = -0,01 \cdot c_i^F \cdot e^{-0,01 \cdot S_i}$ von f nach S_i im Punkt S (vgl. Tab. 2.8). Im ersten Hauptschritt entnehmen

Tabelle 2.8. Partielle Ableitungen von f nach S_i im Punkt S

$i \in V$	0	1	2	3	4	5
ϕ_i	0	0,030	-0,019	-0,037	0,047	0

wir Knoten 5 mit $\varrho_5 = 3$ aus der Menge V' . Knoten 5 besitzt keinen Nachfolger, aber genau einen Vorgänger (Knoten 4) und stellt damit eine Senke in dem zum ES -Schedule gehörigen Gerüst dar. Da $\phi_5 = 0$ gilt, wird Knoten 5 nicht verzögert, sondern wir verschmelzen Vorgang 5 mit Vorgang 4 und setzen $\phi_4 := \phi_4 + \phi_5 = 0,047$ sowie $C(4) := \{4, 5\}$. Für die Vorgänge 2 und 4 gilt $\varrho_2 = \varrho_4 = 2$. Zunächst wählen wir Knoten 2 (Senke) aus V' . Da für Vorgang 2 die Bedingung $\phi_2 < 0$ erfüllt ist, verschmelzen wir Knoten 2 mit

Knoten 1. Es ergibt sich $\phi_1 := \phi_1 + \phi_2 = 0,011$ und $C(1) := \{1, 2\}$. Als Nächste entfernen wir Vorgang 4 aus V' , der ebenfalls eine Senke darstellt. Für Knoten 4 gilt $\phi_4 > 0$, d.h. die Vorgänge $C(4) = \{4, 5\}$ bilden ein Teilgerüst, das zu verzögern ist, und wir setzen $z_4 := z_5 := 1$. Für die Vorgänge 1 und 3 gilt $\varphi_1 = \varphi_3 = 1$. Vorgang 1 stellt eine Senke dar und wird aus V' eliminiert. Wegen $\phi_1 = 0,011 > 0$ setzen wir $z_1 := 1$ und $z_2 := 1$. Knoten 3 ist wiederum eine Senke und wird aus V' entfernt. Da ϕ_3 negativ ist, verschmelzen wir Vorgang 3 mit Vorgang 0. Es ergibt sich $\phi_0 := \phi_0 + \phi_3 = -0,037$ und $C(0) := \{0, 3\}$. Da nun die Menge V' leer ist, terminiert der Algorithmus. Als zulässige steilste Anstiegsrichtung erhalten wir $z = (0, 1, 1, 0, 1, 1)^T$.

Wir führen nun einen Eckenanstieg durch. Die Menge der Vorgänge, die verzögert werden, ist $V^s = \{1, 2, 4, 5\}$. Wir setzen $S_0^E := 0$ und $S_3^E := 8$. Zunächst sind die Pfeile $\langle 0, 1 \rangle$ und $\langle 3, 4 \rangle$ aus dem Gerüst G zu entfernen, da $0 \notin C(1)$ und $1 \notin C(0)$ sowie $3 \notin C(4)$ und $4 \notin C(3)$. Dann ist Schrittweite λ gemäß (2.8) zu bestimmen. In der ersten Iteration ergibt sich für $h \in \{1, 2, 4, 5\}$ und $l \in \{0, 3\}$ eine Schrittweite von $\lambda := \min\{S_0^E - S_5 - \delta_{50} = 0 - 9 + 10 = 1, S_3^E - S_1 - \delta_{13} = 8 - 1 - 4 = 3, S_3^E - S_4 - \delta_{43} = 8 - 7 - 1 = 0\} = 0$. Die Vorgänge aus $C(4)$ werden also um 0 Zeiteinheiten verzögert, d.h. wir setzen $S_4^E := 7, S_5^E := 9$ und wir fügen den Pfeil $\langle 4, 3 \rangle$ zu E_G hinzu. Danach setzen wir $z_4 = z_5 := 0$ und eliminieren die Vorgänge 4 und 5 aus V^s . In der zweiten Iteration ergibt sich mit $h \in \{1, 2\}$ und $l \in \{0, 3, 4, 5\}$ eine Schrittweite von $\lambda := \min\{S_3^E - S_1 - \delta_{13} = 8 - 1 - 4 = 3, S_5^E - S_2 - \delta_{25} = 9 - 4 - 2 = 3\} = 3$. Die Vorgänge aus $C(1)$ sind um 3 Zeiteinheiten zu verzögern. Wir setzen $S_1^E := 4, S_2^E := 7, z_1 = z_3 := 0$ und fügen den Pfeil $\langle 2, 5 \rangle$ zu E_G hinzu und entnehmen die Vorgänge 1 und 2 der Menge V^s . Nun ist $V^s = \emptyset$ und wir haben die Ecke $S^E = (0, 4, 7, 8, 7, 9)$ erreicht. Das zu S^E gehörige Gerüst ist in Abbildung 2.33 dargestellt.

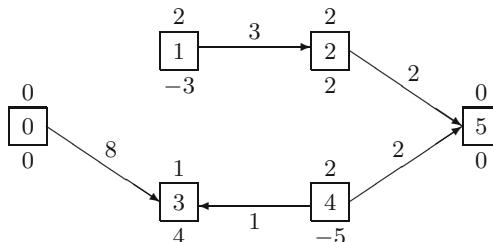


Abb. 2.33. Gerüst zu $S^E = (0, 4, 7, 8, 7, 9)$

In der zweiten Iteration wird in analoger Weise für $S := S^E$ fortgefahrene. Zunächst setzen wir $z_i := 0$ sowie $C(i) := \{i\}$ für alle $i \in V$ und bestimmen die partiellen Ableitungen von f nach S_i im Punkt S (vgl. Tab. 2.9).

Die einzelnen Schritte zur Bestimmung einer steilsten Anstiegsrichtung sind in Tabelle 2.10 zusammengefasst.

Tabelle 2.9. Partielle Ableitungen von f nach S_i im Punkt S

$i \in V$	0	1	2	3	4	5
ϕ_i	0	0,029	-0,019	-0,037	0,047	0

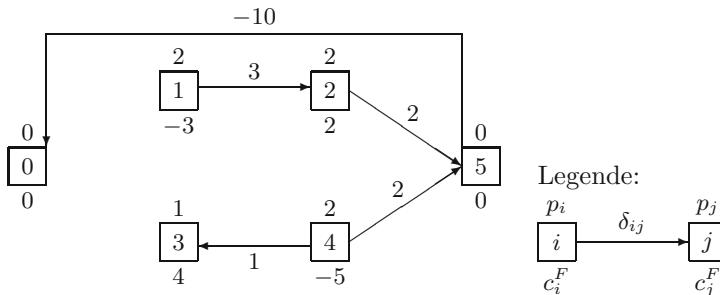
Tabelle 2.10. Bestimmung einer steilsten Anstiegsrichtung

$i \in V$	ϱ_i	Quelle/Senke	ϕ_i	Aktion
1	5	Quelle	$\phi_1 = 0,029$	Verschmelzung mit Knoten 2, $\phi_2 := 0,010, C(2) := \{1, 2\}$
2	4	Quelle	$\phi_2 = 0,010$	Verschmelzung mit Knoten 5, $\phi_5 := 0,010, C(5) := \{1, 2, 5\}$
5	3	Senke	$\phi_5 = 0,010$	$z_1 := 1, z_2 := 1, z_5 := 1$
4	2	Quelle	$\phi_4 = 0,047$	Verschmelzung mit Knoten 3, $\phi_3 := 0,010, C(3) := \{3, 4\}$
3	1	Senke	$\phi_3 = 0,010$	$z_3 := 1, z_4 := 1$

Danach führen wir einen Eckenanstieg durch (vgl. Tab. 2.11). Es ergibt sich die Anstiegsrichtung $z = (0, 1, 1, 1, 1, 1)^T$ und wir erreichen die Ecke $S^E = (0, 5, 8, 9, 8, 10)$. Das zu S^E gehörige Gerüst ist in Abbildung 2.34 angegeben.

Tabelle 2.11. Bestimmung der maximalen Schrittweite

V^s	λ	Aktion
$\{1, 2, 3, 4, 5\}$	1	Teilgerüst $\{1, 2, 5\} \in C(5)$ um 1 Zeiteinheit verzögern
$\{3, 4\}$	1	Teilgerüst $\{3, 4\} \in C(3)$ um 1 Zeiteinheit verzögern

**Abb. 2.34.** Gerüst zu $S^E = (0, 5, 8, 9, 8, 10)$

In der dritten Iteration terminiert das Verfahren, da wir als steilste Anstiegsrichtung $z = \mathbf{0}$ erhalten. Die einzelnen Schritte zur Bestimmung von z sind in Tabelle 2.12 angegeben. Die optimale Lösung für das Kapitalwertmaximierungsproblem lautet $S = (0, 5, 8, 9, 8, 10)$.

Tabelle 2.12. Bestimmung einer steilsten Anstiegsrichtung

$i \in V$	ϱ_i	Quelle/Senke	ϕ_i	Aktion
1	3	Quelle	$\phi_1 = 0,029$	Verschmelzung mit Knoten 2, $\phi_2 := 0,011, C(2) := \{1, 2\}$
3	3	Senke	$\phi_3 = -0,037$	Verschmelzung mit Knoten 4, $\phi_4 := 0,009, C(4) := \{3, 4\}$
2	2	Quelle	$\phi_2 = 0,011$	Verschmelzung mit Knoten 5, $\phi_5 := 0,011, C(5) := \{1, 2, 5\}$
4	2	Quelle	$\phi_4 = 0,009$	Verschmelzung mit Knoten 5, $\phi_5 := 0,020, C(5) := \{1, 2, 3, 4, 5\}$
5	1	Quelle	$\phi_5 = 0,020$	Verschmelzung mit Knoten 0, $\phi_0 := 0,020, C(0) := \{0, 1, 2, 3, 4, 5\}$

Das in diesem Abschnitt vorgestellte steilste Anstiegsverfahren kann ebenfalls zur Lösung von Earliness-Tardiness-Problemen angewendet werden. Im Gegensatz zu einem Problem der Kapitalwertmaximierung wird bei einem Earliness-Tardiness-Problem ein Schedule i.d.R. nicht durch ein Gerüst, sondern durch einen *Wald* (Menge von Teilgerüsten) bindender Zeitbeziehungen repräsentiert. Dies resultiert daraus, dass für Earliness-Tardiness-Probleme bei der Terminierung eines Vorgangs i nicht nur der planungsabhängige ES_i bzw. LS_i in Frage kommt, sondern zusätzlich Zeitpunkte $d_i - p_i$, da die partielle Ableitung ϕ_i in diesem Punkt das Vorzeichen wechselt. Vorgänge $i \in V$, die zu $d_i - p_i$ eingeplant werden, oder Teilgerüste, die einen solchen Vorgang enthalten, können „im Raum stehen“, d.h. für diese Vorgänge bzw. Teilgerüste müssen keine Zeitbeziehungen zu Knoten außerhalb des Teilgerüstes bindend sein (vgl. Abschnitt 2.1.5).

2.2.4 Ressourceninvestment-, Ressourcenabweichungs- und Ressourcennivellierungsprobleme

Bei der Zielfunktion des Ressourceninvestmentproblems (RI) handelt es sich um eine lokal reguläre Funktion. Für ein Projektplanungsproblem (2.1) mit lokal regulärer Zielfunktion kommen, wie in Abschnitt 2.1.5 erläutert, die Minimalpunkte aller Schedulepolytope $S_T(O(S))$, $S \in \mathcal{S}_T$, als Lösungskandidaten in Frage. Die Zielfunktionen des Ressourcenabweichungsproblems (RD) und des Ressourcennivellierungsproblems (RL) sind lokal konkav (vgl. Abschnitt 2.1.5). Daher stellen die Extrempunkte aller Schedulepolytope $S_T(O(S))$ Kandidaten für eine optimale Lösung des entsprechenden Optimierungsproblems dar.

Um ein Ressourceninvestment-, Ressourcenabweichungs- bzw. Ressourcennivellierungsproblem zu lösen, machen wir uns das folgende Lemma zu Nutze, das sich unmittelbar aus Satz 2.11 ergibt.

Lemma 2.29. Jeder Minimalpunkt eines Schedulepolytops $S_T(O(S))$ kann durch mindestens einen Outtree des Netzplans $N(O(S))$ repräsentiert wer-

den. Analog dazu kann jede Ecke eines Schedulepolytops $S_T(O(S))$ durch mindestens ein Gerüst des Netzplans $N(O(S))$ repräsentiert werden. Jeder Pfeil des entsprechenden Outtrees bzw. Gerüstes stellt dabei eine bindende Zeitbeziehung $S_j - S_i = \delta_{ij}$ mit $\langle i, j \rangle \in E$ oder eine bindende Vorrangbeziehung $S_j - S_i = p_i$ mit $(i, j) \in O(S)$ dar.

Die Extrempunkte (und damit auch die Minimalpunkte) aller Schedulepolytope $S_T(O(S))$, $S \in \mathcal{S}_T$, können mit Hilfe eines so genannten *gerüstbasierten Enumerationsschemas* bestimmt werden. Im Verlauf dieses Verfahrens werden die Gerüste der Ordnungsnetzpläne $N(O(S))$ schrittweise aufgebaut. Da Vorgang 0 aufgrund der Nebenbedingung $S_0 = 0$ zeitlich fixiert ist, beginnen wir das gerüstbasierte Enumerationsschema mit dem Teilgerüst, das nur aus dem Knoten $i = 0$ besteht. Anschließend wird in jedem Schritt ein weiterer Knoten j über einen Pfeil $\langle i, j \rangle$ oder $\langle j, i \rangle$ an das aktuelle Teilgerüst angefügt. Dazu prüfen wir für jeden Knoten i des aktuellen Teilgerüstes, ob eine der folgenden Beziehungen

- (i) $S_i + \delta_{ij} = ES_j$
- (ii) $S_i - \delta_{ji} = LS_j$
- (iii) $ES_j \leq S_i + p_i \leq LS_j$
- (iv) $ES_j \leq S_i - p_j \leq LS_j$

zu einem Knoten j besteht, der nicht Knoten des Teilgerüstes ist. Ist Bedingung (i) erfüllt, so kann eine bindende Zeitbeziehung der Form $S_j - S_i = \delta_{ij}$ zu Knoten j etabliert werden und wir fügen dem Gerüst den Pfeil $\langle i, j \rangle$ mit Bewertung δ_{ij} hinzu. Der Startzeitpunkt von Vorgang j ergibt sich zu $S_j = ES_j = S_i + \delta_{ij}$; vgl. Abb. 2.35 (i). Gilt Bedingung (ii), dann kann eine bindende Zeitbeziehung der Form $S_i - S_j = \delta_{ji}$ zu Knoten j eingefügt werden. Wir erhalten den Startzeitpunkt von Vorgang j gemäß $S_j = LS_j = S_i - \delta_{ji}$; vgl. Abb. 2.35 (ii). Ist eine der Bedingungen (iii) oder (iv) erfüllt, so kann eine Vorrangbeziehung der Form $S_j - S_i = p_i$ bzw. $S_i - S_j = p_j$ zu Knoten j etabliert werden. Der Startzeitpunkt von Vorgang j ergibt sich dann zu $S_j = S_i + p_i$ bzw. $S_j = S_i - p_j$; vgl. Abb. 2.35 (iii) und (iv).

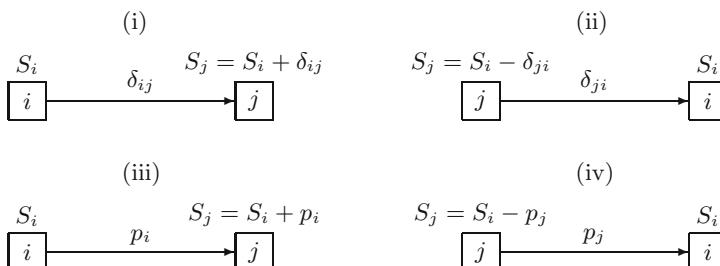


Abb. 2.35. Bindende Zeit- und Vorrangbeziehungen

Nachdem auf diese Weise ein Knoten an das aktuelle Teilgerüst angefügt wurde, ändern sich i.d.R. die planungsabhängigen frühesten und spätesten Startzeitpunkte der noch nicht im Teilgerüst enthaltenen Knoten. Für diese Vorgänge müssen wir daher zunächst ein so genanntes *ES-LS-Update* durchführen, d.h. wir aktualisieren für alle nicht im aktuellen Teilgerüst enthaltenen Vorgänge j die *ES*- und *LS*-Werte gemäß der Vorschriften

$$ES_j := \max\{d_{0j}, \max_{i \in V_G}(S_i + d_{ij})\} \text{ bzw.}$$

$$LS_j := \min\{-d_{j0}, \min_{i \in V_G}(S_i - d_{ji})\}.$$

Hierbei bezeichnet $V_G \subseteq V$ die Menge aller im aktuellen Teilgerüst G enthaltenen Knoten. Im Folgenden behandeln wir ein kleines Beispiel zum *ES-LS-Update*.

Beispiel 2.30. Betrachten wir den MPM-Netzplan in Abbildung 2.36 mit vier realen Vorgängen. Der zugehörige Vektor der frühesten Startzeitpunkte lautet $ES = (0, 0, 1, 0, 1, 5)$ und für die spätesten Startzeitpunkte gilt $LS = (0, 3, 4, 4, 5, 8)$. Die Startzeitpunkte der Vorgänge 0, 2, 3 und 5 seien nun zu $S_0 := 0$, $S_2 := 4$, $S_3 := 0$ und $S_5 := 8$ festgelegt. Das zugehörige Teilgerüst ist in Abbildung 2.36 angegeben. Für die noch nicht im Teilgerüst enthaltenen Knoten 1 und 4 muss nun ein *ES-LS-Update* durchgeführt werden.

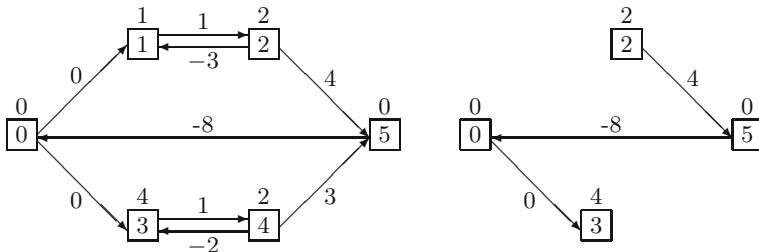


Abb. 2.36. Projektnetzplan und Teilgerüst mit Knotenmenge $V_G = \{0, 2, 3, 5\}$

Für Vorgang 1 ändert sich der früheste Startzeitpunkt zu

$$ES_1 := \max\{d_{01}, \max(S_0 + d_{01}, S_2 + d_{21})\}$$

$$= \max\{0, \max(0 + 0, 4 - 3)\} = 1,$$

während der späteste Startzeitpunkt von Vorgang 1 unverändert $LS_1 = 3$ bleibt. Für den Vorgang 4 ändert sich der späteste Startzeitpunkt zu

$$LS_4 := \min\{-d_{40}, \min(S_3 - d_{43}, S_5 - d_{45})\}$$

$$= \min\{5, \min(0 + 2, 8 - 3)\} = 2.$$

und der früheste Startzeitpunkt von Vorgang 4 ist unverändert $ES_4 = 1$.

Bevor wir den Algorithmus zur Bestimmung aller Extrempunkte der Schedulepolytope $S_T(O(S))$, $S \in \mathcal{S}_T$ vorstellen, gehen wir auf Möglichkeiten ein, die Bildung redundanter Gerüste im Algorithmus zu reduzieren. Eine Möglichkeit ist, die Wahl des nächsten an das aktuelle Teilgerüst anzubindenden Knotens zu beschränken. Sind beispielsweise zwei Teilgerüste mit Pfeilmengen $\{\langle 0, 2 \rangle\}$ und $\{\langle 0, 1 \rangle\}$ vorhanden, so darf die erste Pfeilmenge nur dann um den Pfeil $\langle 0, 1 \rangle$ erweitert werden, wenn die zweite Pfeilmenge nicht um den Pfeil $\langle 0, 2 \rangle$ erweitert wird. Wir vereinbaren daher, dass an einen Knoten i , an den bereits Knoten angebunden wurden, nur solche Knoten neu angebunden werden dürfen, deren Vorgangsnummer größer ist als die Vorgangsnummern aller bislang an i angebundenen Knoten. Dabei ist zu beachten, dass der Knoten, an den i angebunden wurde und der deshalb auf dem eindeutigen Semiweg von 0 nach i liegt, nicht berücksichtigt wird. Dies gewährleistet, dass wir trotz beliebiger Nummerierung der Knoten ein vollständiges Gerüst generieren können. Betrachten wir dazu das Gerüst in Abbildung 2.37. Knoten 5 besitzt zwar eine größere Vorgangsnummer als die Knoten 2 und 3, da 5 aber auf einem Semiweg von Knoten 0 zu Knoten 1 liegt, muss er beim Anbinden der Knoten 2 und 3 an 1 vernachlässigt werden.

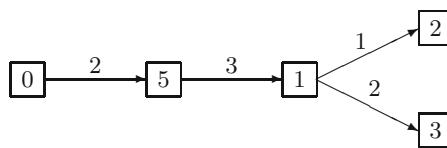


Abb. 2.37. Teilgerüst

Im Einzelnen lässt sich das *gerüstbasierte Enumerationsschema* wie folgt beschreiben. Seien Γ die Menge bereits generierter Gerüste und Ω die Menge generierter und noch zu erweiternder Teilgerüste $\langle V_G, E_G \rangle$, wobei V_G die Knotenmenge und E_G die Pfeilmenge eines Teilgerüstes darstellt. Im Initialisierungsschritt setzen wir $V_G := \{0\}$, $E_G := \emptyset$ sowie $\Omega := \{\langle V_G, E_G \rangle\}$, $\Gamma := \emptyset$ und bestimmen die Längen längster Wege d_{ij} . In jeder Iteration wählen wir dann ein Teilgerüst $\langle V_G, E_G \rangle \in \Omega$ und prüfen, ob bereits ein vollständiges Gerüst bestimmt wurde, d.h. ob V_G alle Knoten des zugrunde liegenden Projektnetzplans beinhaltet. Ist dies nicht der Fall, wird versucht, das Teilgerüst zu erweitern. Dazu bestimmen wir zunächst, wie in Abschnitt 2.1.4 beschrieben, die Startzeitpunkte aller Vorgänge in V_G sowie die *ES*- und *LS*-Werte aller Vorgänge aus $V \setminus V_G$. Für jeden Knoten $i \in V_G$ ermitteln wir alle Knoten $j \in V \setminus V_G$, die an das aktuelle Teilgerüst V_G angebunden werden können. Um, wie oben beschrieben, die Bildung redundanter Gerüste zu vermeiden, vereinbaren wir, dass nur solche Knoten j an einen im Teilgerüst enthaltenen Knoten i angebunden werden, die größer sind als der größte über einen Pfeil $\langle h, i \rangle \in E_G$ bzw. $\langle i, h \rangle \in E_G$ an i bereits angebundene Knoten h , wobei $\langle h, i \rangle$ bzw. $\langle i, h \rangle$ nicht auf einem Semiweg von 0 nach i liegt. Danach prüfen wir für

alle diese j mit Hilfe der Bedingungen (i)–(iv), ob eine bindende Zeit- bzw. Vorrangbeziehung zwischen Knoten $i \in V_G$ und Knoten $j \in V \setminus V_G$ eingeführt werden kann. Ist eine der Bedingungen (i)–(iv) erfüllt, dann wird das aktuelle Teilgerüst $\langle V_G, E_G \rangle$ um den Knoten j erweitert und das so entstandene Teilgerüst $\langle V_{G'}, E_{G'} \rangle$ in die Menge Ω aufgenommen. Die Knotenmenge $V_{G'}$ ergibt sich dabei aus $V_G \cup \{j\}$ und die Pfeilmenge $E_{G'}$ aus E_G vereinigt mit dem zur jeweiligen bindenden Zeit- bzw. Vorrangbeziehung gehörigen Pfeil. Das Verfahren terminiert, wenn die Menge Ω leer ist.

Algorithmus 2.31 (Gerüstbasiertes Enumerationsschema für Problem (2.1) mit Zielfunktion (RD) bzw. (RL)).

Initialisierung:

Setze $V_G := \{0\}$, $E_G := \emptyset$, $\Omega := \{\langle V_G, E_G \rangle\}$, $\Gamma := \emptyset$.

Bestimme die Längen längster Wege d_{ij} für alle $i, j \in V$.

Hauptschritt:

Solange $\Omega \neq \emptyset$:

Entferne ein Paar $\langle V_G, E_G \rangle$ aus Ω .

Falls $V_G = V$: Setze $\Gamma := \Gamma \cup \{\langle V_G, E_G \rangle\}$.

Andernfalls :

Bestimme die Startzeitpunkte S_i für alle $i \in V_G$.

Bestimme $ES_j := \max\{d_{0j}, \max_{i \in V_G} (S_i + d_{ij})\}$ und $LS_j := \min\{-d_{j0}, \min_{i \in V_G} (S_i - d_{ji})\}$ für alle $j \in V \setminus V_G$.

Für alle $i \in V_G$:

Für alle $j \in V \setminus V_G$ für die gilt: es existiert kein Vorgang h mit $\langle i, h \rangle \in E_G$ oder $\langle h, i \rangle \in E_G$ mit $h > j$, der nicht auf einem Semiweg von 0 nach i liegt:

(i) **Falls** $\langle i, j \rangle \in E$ und $S_i + \delta_{ij} = ES_j$:

Füge $\langle i, j \rangle$ mit der Bewertung δ_{ij} dem Gerüst hinzu, d.h. $V_{G'} := V_G \cup \{j\}$, $E_{G'} := E_G \cup \{\langle i, j \rangle\}$, $\Omega := \Omega \cup \{\langle V_{G'}, E_{G'} \rangle\}$.

(ii) **Falls** $\langle j, i \rangle \in E$ und $S_i - \delta_{ji} = LS_j$:

Füge $\langle j, i \rangle$ mit der Bewertung δ_{ji} dem Gerüst hinzu, d.h. $V_{G'} := V_G \cup \{j\}$, $E_{G'} := E_G \cup \{\langle j, i \rangle\}$, $\Omega := \Omega \cup \{\langle V_{G'}, E_{G'} \rangle\}$.

(iii) **Falls** $ES_j \leq S_i + p_i \leq LS_j$:

Füge $\langle i, j \rangle$ mit der Bewertung p_i dem Gerüst hinzu, d.h. $V_{G'} := V_G \cup \{j\}$, $E_{G'} := E_G \cup \{\langle i, j \rangle\}$, $\Omega := \Omega \cup \{\langle V_{G'}, E_{G'} \rangle\}$.

(iv) **Falls** $ES_j \leq S_i - p_j \leq LS_j$:

Füge $\langle j, i \rangle$ mit der Bewertung p_j dem Gerüst hinzu, d.h. $V_{G'} := V_G \cup \{j\}$, $E_{G'} := E_G \cup \{\langle j, i \rangle\}$, $\Omega := \Omega \cup \{\langle V_{G'}, E_{G'} \rangle\}$.

Rückgabe Γ .

Algorithmus 2.31 generiert alle Gerüste der Netzpläne $N(O(S))$, $S \in \mathcal{S}_T$, und somit mindestens ein Gerüst für jeden Extrempunkt eines Schedulepolytops $\mathcal{S}_T(O(S))$. Für ein Projektplanungsproblem (2.1) mit Zielfunktion (RI) ist aber nur die Bestimmung von Outtrees der Ordnungsnetzpläne $N(O(S))$ erforderlich, d.h. von Minimalpunkten der Schedulepolytope $\mathcal{S}_T(O(S))$. Um dies zu erreichen, sind die Bedingungen (ii) und (iv) in Algorithmus 2.31 zu streichen.

Im Folgenden demonstrieren wir anhand eines Beispiels die Generierung aller Gerüste für Problem (2.1) mit Zielfunktion (RD) bzw. (RL) mittels des gerüstbasierten Enumerationsschemas.

Beispiel 2.32. Wir betrachten den Projektnetzplan in Abbildung 2.38 mit zwei realen Vorgängen. Der zugehörige S_1 - S_2 -Schnitt des zeitzulässigen Bereichs \mathcal{S}_T ist ebenfalls in Abbildung 2.38 angegeben.

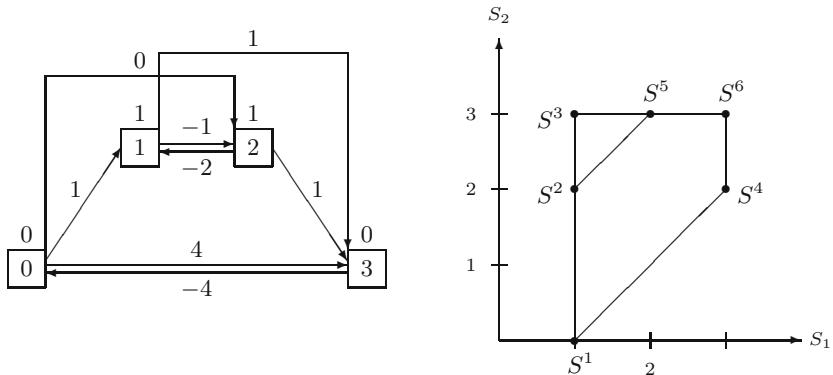


Abb. 2.38. Projektnetzplan und S_1 - S_2 -Schnitt des zeitzulässigen Bereichs \mathcal{S}_T

Im Initialisierungsschritt des gerüstbasierten Enumerationsschemas setzen wir $V_G := \{0\}$, $E_G := \emptyset$, $\Omega := \{\{\{0\}, \emptyset\}\}$ und $\Gamma := \emptyset$. Danach bestimmen wir die Längen längster Wege d_{ij} für alle $i, j \in V$ (vgl. Tab. 2.13).

Tabelle 2.13. Längen längster Wege d_{ij} für alle $i, j \in V$

d_{ij}	0	1	2	3
0	0	1	0	4
1	-3	0	-1	1
2	-3	-2	0	1
3	-4	-3	-4	0

Im ersten Hauptschritt entnehmen wir Ω das Teilgerüst $\{\{0\}, \emptyset\}$. Für den Startzeitpunkt von Vorgang 0 gilt $S_0 := 0$ und die frühesten und spätesten

Startzeitpunkte der Vorgänge $j \in V \setminus V_G$ ergeben sich zu $ES_1 := 1$, $LS_1 := 3$, $ES_2 := 0$, $LS_2 := 3$, $ES_3 := 4$ und $LS_3 := 4$. Nun ist anhand der Bedingungen (i)–(iv) zu prüfen, welche bindenden Zeit- bzw. Vorrangbeziehungen zwischen dem Knoten 0 und einem Knoten $j \in \{1, 2, 3\}$ etabliert werden können. Da für den Pfeil $\langle 0, 1 \rangle \in E$ die Bedingung $S_0 + 1 = 1 = ES_1$ erfüllt ist (Bedingung (i)), kann Knoten 1 durch eine bindende Zeitbeziehung der Form $S_1 - S_0 = \delta_{01}$ an das Teilgerüst $\langle \{0\}, \emptyset \rangle$ angehängt werden. Es ergibt sich das Teilgerüst $\langle \{0, 1\}, \{\langle 0, 1 \rangle\} \rangle$, das Ω hinzugefügt wird. Für die Knoten 0 und 2 sind die Bedingungen (i) und (iii) erfüllt, d.h. das Teilgerüst $\langle \{0\}, \emptyset \rangle$ kann zum einen durch eine bindende Zeitbeziehung und zum anderen durch eine bindende Vorrangbeziehung zu Knoten 2 erweitert werden. Da aber $p_0 = \delta_{02}$ gilt, sind die beiden resultierenden Teilgerüste gleich, und wir nehmen das Teilgerüst $\langle \{0, 2\}, \{\langle 0, 2 \rangle\} \rangle$ in die Menge Ω auf. Für die Knoten 0 und 3 ist die Bedingung (i) und die Bedingung (ii) erfüllt. Somit können bindende Zeitbeziehungen der Form $S_3 - S_0 = \delta_{03}$ und $S_0 - S_3 = \delta_{30}$ etabliert werden. Die Menge Ω ist um die beiden Teilgerüste $\langle \{0, 3\}, \{\langle 0, 3 \rangle\} \rangle$ und $\langle \{0, 3\}, \{\langle 3, 0 \rangle\} \rangle$ zu erweitern. Am Ende des ersten Hauptschrittes ergibt sich die Menge Ω zu $\Omega = \{\langle \{0, 1\}, \{\langle 0, 1 \rangle\} \rangle, \langle \{0, 2\}, \{\langle 0, 2 \rangle\} \rangle, \langle \{0, 3\}, \{\langle 0, 3 \rangle\} \rangle, \langle \{0, 3\}, \{\langle 3, 0 \rangle\} \rangle\}$.

Im zweiten Hauptschritt entnehmen wir Ω das Teilgerüst $\langle \{0, 1\}, \{\langle 0, 1 \rangle\} \rangle$. Die Startzeitpunkte der Vorgänge 0 und 1 ergeben sich zu $S_0 := 0$, $S_1 := 1$. Wir bestimmen $ES_2 := 0$, $LS_2 := 3$, $ES_3 := 4$ und $LS_3 := 4$. Ausgehend von Knoten 0 sind wie im ersten Hauptschritt für Knoten 2 die Bedingungen (i) und (iii) erfüllt. Des Weiteren gelten für die Knoten 0 und 3 die Bedingungen (i) und (ii). Somit fügen wir die Teilgerüste $\langle \{0, 1, 2\}, \{\langle 0, 1 \rangle, \langle 0, 2 \rangle\} \rangle$, $\langle \{0, 1, 3\}, \{\langle 0, 1 \rangle, \langle 0, 3 \rangle\} \rangle$ und $\langle \{0, 1, 3\}, \{\langle 0, 1 \rangle, \langle 3, 0 \rangle\} \rangle$ der Menge Ω hinzu. Weiterhin sind für die Knoten 1 und 2 alle Bedingungen (i)–(iv) erfüllt. Da $p_1 \neq \delta_{12}$ und $p_2 \neq \delta_{21}$ gilt, werden alle vier bindenden Zeit- und Vorrangbeziehungen zu Knoten 2 eingefügt. Es resultieren die Teilgerüste $\langle \{0, 1, 2\}, \{\langle 0, 1 \rangle, \langle 1, 2 \rangle\} \rangle$, $\langle \{0, 1, 2\}, \{\langle 0, 1 \rangle, \langle 1, 2 \rangle\} \rangle$, $\langle \{0, 1, 2\}, \{\langle 0, 1 \rangle, \langle 2, 1 \rangle\} \rangle$, $\langle \{0, 1, 2\}, \{\langle 0, 1 \rangle, \langle 2, 1 \rangle\} \rangle$, die in der Menge Ω gespeichert werden.

Im dritten Hauptschritt entnehmen wir das Teilgerüst $\langle \{0, 1, 2\}, \{\langle 0, 1 \rangle, \langle 0, 2 \rangle\} \rangle$ aus der Menge Ω und bestimmen die Startzeitpunkte der Vorgänge 0, 1 und 2. Es ergeben sich $S_0 := 0$, $S_1 := 1$ sowie $S_2 := 0$. Dann aktualisieren wir $ES_3 := 4$ und $LS_3 := 4$. Für die Knoten 0 und 3 sind die Bedingungen (i) und (ii) erfüllt. Daher fügen wir die vollständigen Gerüste $\langle \{0, 1, 2, 3\}, \{\langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 0, 3 \rangle\} \rangle$ und $\langle \{0, 1, 2, 3\}, \{\langle 0, 1 \rangle, \langle 0, 2 \rangle, \langle 3, 0 \rangle\} \rangle$ zuerst der Menge Ω und dann der Menge Γ hinzu.

In den nächsten Schritten wird auf gleiche Art und Weise fortgefahrene, bis $\Omega = \emptyset$ gilt. Insgesamt werden 20 unterschiedliche Gerüste generiert. Die 10 Gerüste, die den Pfeil $\langle 0, 3 \rangle$ enthalten, sind in Abbildung 2.39 veranschaulicht, sie entsprechen jeweils einem Extrempunkt S^1, \dots, S^6 eines Schedulepolytops $\mathcal{S}_T(O(S))$. Die anderen 10 Gerüste unterscheiden sich von denen in Abbildung 2.39 lediglich darin, dass sie anstelle des Pfeils $\langle 0, 3 \rangle$ den Pfeil $\langle 3, 0 \rangle$ besitzen.

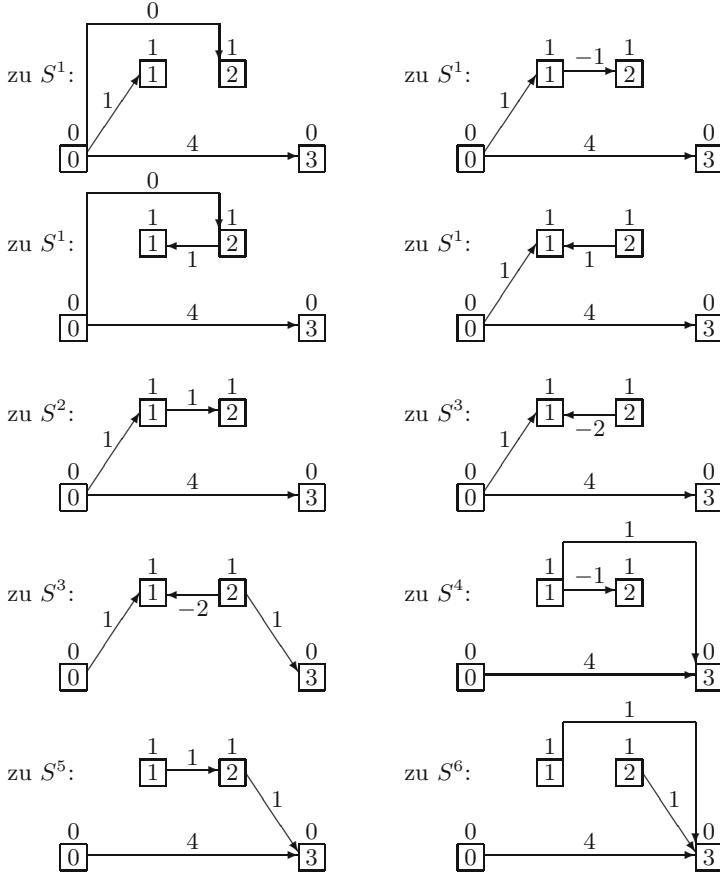


Abb. 2.39. Gerüste für den Projektnetzplan aus Abbildung 2.38

Im Prinzip können wir mit Algorithmus 2.31 eine optimale Lösung für ein Ressourceninvestment-, Ressourcenabweichungs- oder Ressourcennivellierungsproblem bestimmen, indem wir für jedes generierte Gerüst den Zielfunktionswert berechnen und dann das gemäß der Zielfunktion beste Gerüst auswählen. Da die Anzahl möglicher Gerüste aber schon für kleine Probleminstanzen sehr groß sein kann, ist dies i.d.R. nicht sinnvoll.

Normalerweise existieren für einen Extrempunkt eines Ordnungspolytops mehrere zugehörige Gerüste (vgl. z.B. die vier zu S^1 gehörenden Gerüste in Abb. 2.39). Daher lässt sich eine deutliche Beschleunigung von Algorithmus 2.31 erzielen, wenn wir anstelle der Teilgerüste die entsprechenden Teilschedules betrachten. Diese ergeben sich aus den Startzeitpunkten der im jeweiligen Teilgerüst enthaltenen Vorgänge. Im Folgenden bezeichnen wir mit $\mathcal{C} := V_G$ die Menge der bereits an ein Teilgerüst angegebundenen Knoten, d.h. die Menge der bereits eingeplanten Knoten. Einen Vektor von Startzeit-

punkten S^C mit $S_i^C \in \mathbb{R}_{\geq 0}$ für Vorgänge $i \in \mathcal{C} \subseteq V$ und $S_0^C = 0$ nennen wir *Teilschedule*. Die Spezifikation eines Teilschedules erfolgt immer durch Angabe der Menge \mathcal{C} und der zugehörigen Startzeitpunkte $(S_i^C)_{i \in \mathcal{C}}$.

Um zu gewährleisten, dass wir nicht alle, sondern nur „erfolgversprechende“ Teilschedules zu vollständigen Schedules ergänzen, indem wir die Startzeitpunkte von Vorgängen $i \in V \setminus \mathcal{C}$ festsetzen, betten wir das gerüstbasierte Enumerationsschema in ein *Branch-and-Bound-Verfahren* (im Folgenden auch als *gerüstbasierter Enumerationsansatz* bezeichnet) ein.

Branch-and-Bound-Verfahren gehören zu den Enumerationsbaum-Verfahren und besitzen als Hauptkomponenten eine *Branching-Strategie*, um den Lösungsbereich in immer kleinere Teilbereiche aufzuspalten, und eine *Schrankenfunktion*, die es erlaubt, verschiedene Teilbereiche von der weiteren Suche auszuschließen. Die Branching-Strategie unseres Branch-and-Bound-Verfahrens beruht auf dem gerüstbasierten Enumerationsschema, wobei wir, wie bereits gesagt, anstelle von Teilgerüsten Teilschedules betrachten, die sukzessive zu Schedules ergänzt werden. Wir legen ausgehend von der Wurzel des Enumerationsbaums, bei der Vorgang 0 zu $S_0 := 0$ eingeplant ist, bzw. dem Teilschedule $(\mathcal{C}, S^C) = (\{0\}, (0))$ schrittweise Startzeitpunkte von Vorgängen fest, indem wir auf jeder Ebene des Enumerationsbaumes mit Hilfe der Bedingungen (i)–(iv) prüfen, ob eine bindende Zeit- oder Vorrangbeziehung zu einem Vorgang j etabliert werden kann, der noch nicht eingeplant wurde. Ist eine der Bedingungen (i)–(iv) erfüllt, dann wird der aktuelle Teilschedule durch den Startzeitpunkt S_j von Vorgang j erweitert. Durch das Festlegen von Startzeitpunkten für die einzelnen Vorgänge wird der zeitzulässige Bereich des zugrunde liegenden Projektplanungsproblems immer weiter eingeschränkt. Legen wir etwa den Startzeitpunkt von Vorgang 1 fest, so ist der entsprechende Teilbereich, d.h. der zeitzulässige Bereich des aktuellen Enumerationsknotens, durch die Menge $\{S' \in \mathcal{S}_T \mid S'_1 = S_1\}$ bestimmt. Sind alle Startzeitpunkte fixiert, so besteht der aktuelle Teilbereich nur noch aus dem generierten Schedule selbst.

Für jeden Enumerationsknoten (\mathcal{C}, S^C) berechnen wir eine untere Schranke (lower bound) für den optimalen Zielfunktionswert des entsprechenden Teilbereichs $\{S' \in \mathcal{S}_T \mid S'_i = S_i \text{ für alle } i \in \mathcal{C}\}$. Da die Zielfunktionen (RI) , (RD) und (RL) monoton wachsend bzgl. der Ressourcenprofile sind, d.h. der Zielfunktionswert ist größer oder bleibt gleich, falls sich $r_k(S, t)$ für ein $t \in [0, \bar{d}]$ vergrößert,¹¹ stellt

$$LB0 := f(S^C)$$

die am einfachsten zu berechnende untere Schranke für den Teilschedule S^C dar. $LB0$ repräsentiert den Zielfunktionswert aller bereits eingeplanten Vorgänge. Beispielsweise erhalten wir für einen Teilschedule $S^C = (0, 1)$ mit $\mathcal{C} = \{0, 1\}$ und den Ressourceninanspruchnahmen $r_{01} = r_{02} = 0, r_{11} = 3, r_{12} = 2$ an den Ressourcen 1 und 2 sowie den Dauern $p_0 = 0, p_1 = 1$ für die

¹¹ Entsprechende Funktionen werden in der Literatur auch r-monoton genannt (vgl. ZIMMERMANN, 2001).

Zielfunktion eines Ressourceninvestmentproblems (*RI*) mit $c_1^P = c_2^P = 1$ die untere Schranke $LB0 = 3 + 2 = 5$.

Um im Enumerationsbaum möglichst schnell eine zeitzulässige Lösung zu finden, verwenden wir als Suchstrategie eine *Tiefensuche*. Dazu wählen wir unter allen Söhnen des aktuell betrachteten Knotens im Enumerationsbaum denjenigen Knoten zur weiteren Bearbeitung aus, der die kleinste untere Schranke $LB0$ aufweist. Sobald eine erste zeitzulässige Lösung S^* gefunden wurde, stellt deren Zielfunktionswert $f(S^*)$ eine *obere Schranke UB* (upper bound) für eine optimale Lösung des zugrunde liegenden Problems dar. Mit ihrer Hilfe können alle Knoten zusammen mit ihren Nachfolgern ausgelotet (d.h. von den weiteren Betrachtungen ausgeschlossen) werden, die eine untere Schranke besitzen, welche größer oder gleich UB ist. Jedes Mal, wenn eine neue beste Lösung gefunden wurde, wird UB entsprechend aktualisiert. Existiert unter den Söhnen eines aktuell betrachteten Knotens keine zeitzulässige Lösung oder kann dieser aufgrund seiner unteren Schranke ausgelotet werden, so führen wir einen Backtracking-Schritt aus. Das bedeutet, dass wir im Enumerationsbaum wieder zum Vaterknoten des aktuell betrachteten Knotens zurückspringen und im Folgenden dessen noch nicht untersuchte Söhne betrachten. Eine Tiefensuche kann implementiert werden, indem die Menge der noch zu betrachteten Knoten als Stapel, d.h. als LIFO-Liste (last-in first-out), verwaltet wird, und die generierten Enumerationsknoten gemäß nicht anwachsender unterer Schrankenwerte $LB0$ auf den Stapel gelegt werden.

Für das Ausloten von Enumerationsknoten verwenden wir eine *workloadbasierte* (auf dem Arbeitseinsatz basierte) untere Schranke LBA . Diese erhalten wir aus der unteren Schranke $LB0$, indem der Workload (Arbeitseinsatz) der noch nicht eingeplanten Vorgänge in zwei Schritten zusätzlich berücksichtigt wird.

Im ersten Schritt betrachten wir eine Menge $\tilde{\mathcal{C}} \subseteq V \setminus \mathcal{C}$ bestehend aus einem oder mehreren noch nicht eingeplanten Vorgängen, deren planungsabhängige Zeitfenster $[ES, LC]$ sich nicht überlappen.¹² Da diese Vorgänge $i \in \tilde{\mathcal{C}}$ nicht simultan in Ausführung sein können, planen wir jeden von ihnen unabhängig von den anderen Vorgängen aus $\tilde{\mathcal{C}}$ bestmöglich ein. Hierbei ist zu beachten, dass Vorgang $n + 1$ immer zu seinem LS eingeplant wird, weil dadurch die zur Verfügung stehenden Ressourcen am längsten genutzt werden können. Der Zielfunktionswert des entstandenen Teilschedules $S^{\mathcal{C} \cup \tilde{\mathcal{C}}}$ stellt wieder eine untere Schranke für den optimalen Zielfunktionswert dar.

Sei im zweiten Schritt $\hat{\mathcal{C}} = V \setminus \{\mathcal{C} \cup \tilde{\mathcal{C}}\}$ die Menge der noch nicht betrachteten Vorgänge. Dann entspricht

$$w_k(\hat{\mathcal{C}}) := \sum_{i \in \hat{\mathcal{C}}} p_i r_{ik}$$

¹² Planungsabhängig bedeutet, wie bereits in Abschnitt 1.4.5 erläutert, dass wir die ES - und LC -Werte unter Berücksichtigung der bereits eingeplanten Vorgänge bestimmen.

dem Workload der noch nicht betrachteten Vorgänge an Ressource $k \in \mathcal{R}$. Den Workload $w_k(\hat{\mathcal{C}})$ berücksichtigen wir, indem wir die Vorgänge $j \in \hat{\mathcal{C}}$ in unterbrechbare Teilvergänge mit einer Dauer und einer Ressourceninanspruchnahme von 1 zerlegen. Diese Teilvergänge werden dann für jede Ressource $k \in \mathcal{R}$ zu Zeitpunkten eingeplant, an denen die Ressourceninanspruchnahme der bereits eingeplanten Vorgänge minimal ist. Wir vernachlässigen also die Zeitbeziehungen, nehmen an, die Vorgänge wären unterbrechbar und planen die unterbrechbaren Vorgänge so gut wie möglich im Planungszeitraum $[0, \bar{d}]$ zu Zeitpunkten $\{0, \dots, \bar{d} - 1\}$ ein. Das beschriebene Vorgehen setzt voraus, dass die Pfeilbewertungen des Netzplans, die Vorgangsdauern und die Ressourceninanspruchnahmen der Vorgänge ganzzahlig sind.

Betrachten wir beispielsweise das Ressourcenprofil $r_k(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, t)$ in Abbildung 2.40 mit $\bar{d} = 14$. Der Workload der Vorgänge aus der Menge $\hat{\mathcal{C}}$ sei $w_k(\hat{\mathcal{C}}) = 12$. Dann ergibt sich die durch die gepunkteten Quadrate angedeutete „Auffüllung“ des Ressourcenprofils.

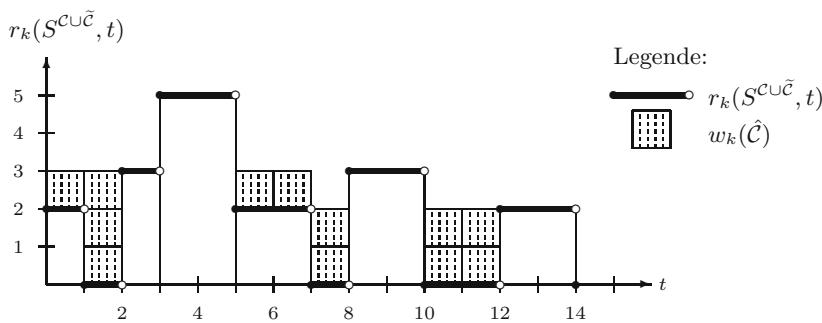


Abb. 2.40. Ressourcenprofil $r_k(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, t)$

Sei $r_k^a(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, \cdot)$, $k \in \mathcal{R}$ das Ressourcenprofil, das aus dem Ressourcenprofil $r_k(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, \cdot)$ durch Hinzufügen des Workloads (Arbeitseinsatzes) $w_k(\hat{\mathcal{C}})$ entsteht. Da, wie bereits gesagt, für die Zielfunktionen (RI), (RD) und (RL) der Zielfunktionswert mit der Ressourceninanspruchnahme in den einzelnen Perioden wächst, gilt $r_k^a(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, t) \leq r_k(S, t)$ für alle $S \in \mathcal{S}_T$, $k \in \mathcal{R}$ und $t = [0, \bar{d}]$. Folglich stellt

$$LBA := \sum_{k \in \mathcal{R}} c_k^P \max_{t \in [0, \bar{d}]} r_k^a(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, t)$$

eine untere Schranke für das Ressourceninvestmentproblem,

$$LBA := \sum_{k \in \mathcal{R}} c_k^D \int_{t \in [0, \bar{d}]} [r_k^a(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, t) - Y_k]^+ dt$$

eine untere Schranke für das Ressourcenabweichungsproblem und

$$LBA := \sum_{k \in \mathcal{R}} \int_{t \in [0, \bar{d}]} (r_k^a)^2 (S^{\mathcal{C} \cup \tilde{\mathcal{C}}}, t) dt$$

eine untere Schranke für das Ressourcennivellierungsproblem dar.

Algorithmus 2.33 beschreibt das Branch-and-Bound-Verfahren zur Lösung von Ressourcenabweichungs- und Ressourcennivellierungsproblemen. Seien Ψ die Menge aller bislang generierten Teilschedules und Ω ein Stapel, der die noch zu erweiternden Teilschedules $(\mathcal{C}, S^{\mathcal{C}})$ enthält. Die Menge Ψ dient dazu bereits betrachtete oder in Ω aufgenommene Gerüste zu erkennen, so dass sie von der weiteren Betrachtung ausgeschlossen werden können. Im Initialisierungsschritt setzen wir $\mathcal{C} := \{0\}$ und $S^{\mathcal{C}} := (0)$. Weiterhin werden die frühesten und spätesten Startzeitpunkte bestimmt. Gilt $ES_i = LS_i$ zu Beginn des Verfahrens oder im weiteren Verfahrensverlauf für einen Vorgang i , so wird Vorgang i zum Zeitpunkt $S_i := ES_i$ eingeplant und \mathcal{C} und $S^{\mathcal{C}}$ werden entsprechend aktualisiert. Als erste „beste“ Lösung wählen wir $S^* := ES$ und initialisieren die obere Schranke $UB := f(ES)$. Danach setzen wir $\Omega := \{(\mathcal{C}, S^{\mathcal{C}})\}$ und $\Psi := \{(\mathcal{C}, S^{\mathcal{C}})\}$. In jeder Iteration entnehmen wir den jeweils obersten Teilschedule $(\mathcal{C}, S^{\mathcal{C}})$ des Stapels Ω , d.h. als Suchstrategie wird eine Tiefensuche durchgeführt. Gilt für den aktuellen Teilschedule $\mathcal{C} = V$, dann ist zu prüfen, ob die gefundene Lösung S besser als die beste bisher gefundene Lösung S^* ist. Ist $f(S^{\mathcal{C}}) < UB$ erfüllt, so setzen wir $S^* := S$ und verringern die obere Schranke auf $UB := f(S^*)$. Gilt $\mathcal{C} \neq V$, dann überprüfen wir, ob $LBA(S^{\mathcal{C}})$ kleiner ist als die obere Schranke UB . In diesem Fall, erweitern wir den aktuellen Teilschedule gemäß der vier Fälle (i)–(iv). Dabei speichern wir für alle Vorgänge $j \in V \setminus \mathcal{C}$ in einer Menge \mathcal{T}_j jeweils alle möglichen Startzeitpunkte, zu denen Vorgang j an das betrachtete Gerüst angebunden werden kann. Die neu entstandenen Teilschedules $(\mathcal{C}', S^{\mathcal{C}'})$ werden, wenn sie noch nicht in der Menge Ψ enthalten sind und ihre untere Schranke $LB0(S^{\mathcal{C}'})$ kleiner als UB ist, den Mengen Λ und Ψ hinzugefügt. Danach werden die Teilschedules $(\mathcal{C}, S^{\mathcal{C}})$ der Menge Λ entnommen und dem Stapel Ω gemäß nicht wachsender Werte von $LB0(S^{\mathcal{C}})$ hinzugefügt. Das oberste Element mit kleinstem Schrankenwert wird dann als Nächstes gewählt. Es werden solange Teilschedules aus Ω entnommen, bis $\Omega = \emptyset$ gilt.

Algorithmus 2.33 (Branch-and-Bound-Verfahren für Problem (2.1) mit Zielfunktion (RD) bzw. (RL)).

Setze $\mathcal{C} := \{0\}$, $S^{\mathcal{C}} := (0)$.

Bestimme die Längen längster Wege d_{ij} für alle $i, j \in V$.

Setze $ES_i := d_{0i}$ und $LS_i := -d_{i0}$ für alle $i \in V \setminus \{0\}$.

Falls $ES_i = LS_i$ für ein $i \in V$: Setze $S_i := ES_i$, $\mathcal{C} := \mathcal{C} \cup \{i\}$.

Setze $S^* := ES$ und $UB := f(S^*)$.

Initialisiere den Stapel $\Omega := \{(\mathcal{C}, S^{\mathcal{C}})\}$ und die Menge $\Psi := \{(\mathcal{C}, S^{\mathcal{C}})\}$.

Solange $\Omega \neq \emptyset$:

Entnimm das oberste Paar $(\mathcal{C}, S^{\mathcal{C}})$ vom Stapel Ω .

Falls $\mathcal{C} = V$:

Falls $f(S^{\mathcal{C}}) < UB$: Setze $S^* := S^{\mathcal{C}}$ und $UB := f(S^*)$.

Andernfalls :

Falls $LBA(S^{\mathcal{C}}) < UB$:

Initialisiere die Menge $\Lambda := \emptyset$.

Für alle $j \in V \setminus \mathcal{C}$:

Setze $\mathcal{T}_j := \emptyset$.

Setze $ES_j := \max\{ES_j, \max_{i \in \mathcal{C}}(S_i + d_{ij})\}$ und

$LS_j := \min\{LS_j, \min_{i \in \mathcal{C}}(S_i - d_{ji})\}$.

Für alle $i \in \mathcal{C}$:

(i) **Falls** $\langle i, j \rangle \in E$ und $S_i + \delta_{ij} = ES_j$: Setze $\mathcal{T}_j := \mathcal{T}_j \cup \{S_i + \delta_{ij}\}$.

(ii) **Falls** $\langle j, i \rangle \in E$ und $S_i - \delta_{ji} = LS_j$: Setze $\mathcal{T}_j := \mathcal{T}_j \cup \{S_i - \delta_{ji}\}$.

(iii) **Falls** $ES_j \leq S_i + p_i \leq LS_j$: Setze $\mathcal{T}_j := \mathcal{T}_j \cup \{S_i + p_i\}$.

(iv) **Falls** $ES_j \leq S_i - p_j \leq LS_j$: Setze $\mathcal{T}_j := \mathcal{T}_j \cup \{S_i - p_j\}$.

Für alle $t \in \mathcal{T}_j$: Setze $S_j := t$, $\mathcal{C}' := \mathcal{C} \cup \{j\}$.

Für alle $h \in V \setminus \mathcal{C}'$:

Falls $\max(ES_h, S_j + d_{jh}) = \min(LS_h, S_j - d_{jh})$:

Setze $S_h := \max(ES_h, S_j + d_{jh})$ und $\mathcal{C}' := \mathcal{C}' \cup \{h\}$.

Falls $(\mathcal{C}', S^{\mathcal{C}'}) \notin \Psi$ und $LB0(S^{\mathcal{C}'}) < UB$:

Setze $\Lambda := \Lambda \cup \{(\mathcal{C}', S^{\mathcal{C}'})\}$ und $\Psi := \Psi \cup \{(\mathcal{C}', S^{\mathcal{C}'})\}$.

Entnimm die Teilschedules $(\mathcal{C}, S^{\mathcal{C}})$ aus Λ und füge sie dem Stapel Ω gemäß nicht wachsender Werte von $LB0(S^{\mathcal{C}'})$ hinzu.

Rückgabe S^* .

Die Verwaltung der Menge Ψ aller bereits generierten Teilschedules ist i.d.R. sehr aufwändig. Um die Bildung von redundanten Teilschedules schon bei der Konstruktion der Teilschedules zu verhindern, existieren verschiedene Techniken, die z.B. in NÜBEL (1999) beschrieben sind. Zur Lösung von Ressourceninvestmentproblemen können wieder die Bedingungen (ii) und (iv) in Algorithmus 2.33 eliminiert werden.

Beispiel 2.34. Wir betrachten den Projektnetzplan in Abbildung 2.41 mit sechs realen Vorgängen, zwei erneuerbaren Ressourcen und einer maximalen Projektdauer von $\bar{d} = 16$. Im Folgenden geben wir die ersten beiden Iterationen von Algorithmus 2.33 zur Lösung eines Ressourceninvestmentproblems mit den Bereitstellungskosten $c_k^P = 1$ für $k = 1, 2$ an.

Für die Vorgänge $i \in V$ erhalten wir die in Tabelle 2.14 aufgeführten frühesten und spätesten Startzeitpunkte.

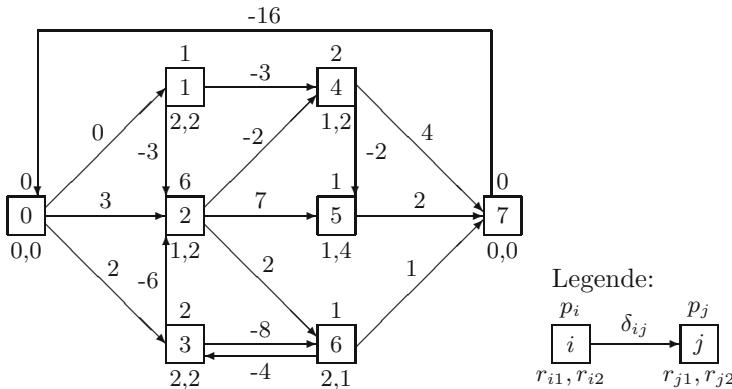


Abb. 2.41. Projektnetzplan mit zwei erneuerbaren Ressourcen

Tabelle 2.14. ES_i und LS_i für alle Vorgänge $i \in V$

i	0	1	2	3	4	5	6	7
ES_i	0	0	3	2	1	10	5	12
LS_i	0	10	7	13	12	14	15	16

Wir setzen $S^* := (0, 0, 3, 2, 1, 10, 5, 12)$. Für S^* ergeben sich die in Abbildung 2.42 dargestellten Ressourcenprofile und eine obere Schranke gemäß $UB := f(S^*) = \sum_{k \in \mathcal{R}} \max_{t \in [0, \bar{d}]} r_{kt} = 3 + 4 = 7$.

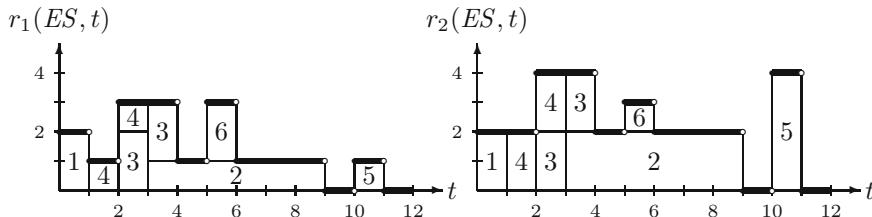


Abb. 2.42. Ressourcenprofile zum *ES*-Schedule

Weiter initialisieren wir $\mathcal{C} := \{0\}$, $S^{\mathcal{C}} := (0)$, den Stapel $\Omega := \{(\mathcal{C}, S^{\mathcal{C}})\}$ und die Menge $\Psi := \{(\mathcal{C}, S^{\mathcal{C}})\}$. In der ersten Iteration entnehmen wir den Teilschedule $(\{0\}, (0))$ dem Stapel Ω . Dann berechnen wir die workloadbasierte untere Schranke $LBA(S^{\mathcal{C}})$ für $S^{\mathcal{C}} = (0)$ mit $\mathcal{C} = \{0\}$. Da sich für die Vorgänge 1 und 7 die planungsabhängigen Zeitfenster $[ES, LC]$ nicht überlappen, planen wir Vorgang 1 und Vorgang 7 bestmöglich ein, d.h. beispielsweise zu $S_1 = 0$ und $S_7 = 16$. Die Menge der noch nicht betrachteten Vorgänge ist nun $\hat{\mathcal{C}} = \{2, \dots, 6\}$. Für den Workload der Vorgänge aus $\hat{\mathcal{C}}$ ergibt sich $w_1(\hat{\mathcal{C}}) := 15$ und $w_2(\hat{\mathcal{C}}) := 25$. Dieser Workload wird berücksichtigt, indem

wir die Vorgänge $j \in \hat{\mathcal{C}}$ in unterbrechbare Televorgänge mit einer Dauer und einer Ressourceninanspruchnahme von 1 zerlegen und diese zu Zeitpunkten einplanen, an denen die Ressourceninanspruchnahme der bereits eingeplanten Vorgänge minimal ist. Somit ergibt sich für die workloadbasierte untere Schranke

$$LBA(S^{\mathcal{C}}) := \sum_{k \in \mathcal{R}} \max_{t \in [0, d]} r_k^a(S^{\mathcal{C} \cup \tilde{\mathcal{C}}}, t) = 2 + 2 = 4.$$

Da $LBA(S^{\mathcal{C}}) < UB$ ist, fahren wir fort und initialisieren die Menge $\Lambda := \emptyset$. Die Aktualisierung der ES - und LS -Werte führt zu keinen Veränderungen. Anhand der Bedingungen (i) und (iii) ist nun zu prüfen (Zielfunktion (RI)), ob zwischen einem Knoten $j \in \{1, \dots, 7\}$ und Knoten 0 eine bindende Zeit- bzw. Vorrangbeziehung etabliert werden kann. Für die Knoten 0 und 1 gelten die Bedingungen $\langle 0, 1 \rangle \in E$ und $S_0 + 0 = 0 = ES_1$ sowie $ES_1 \leq S_0 + p_0 \leq LS_1$; Bedingungen (i) und (iii). Daher ergibt sich die Menge aller möglichen Einplanungszeitpunkte von 1 zu $\mathcal{T}_1 := \{0\}$. Wir erweitern den aktuellen Teilschedule um den Knoten 1 und erhalten $S^{\mathcal{C}} = (0, 0)$ mit $\mathcal{C} = \{0, 1\}$. Für den Teilschedule $S^{\mathcal{C}} = (0, 0)$ ergibt sich die untere Schranke

$$LB0(S^{\mathcal{C}}) := f(S^{\mathcal{C}}) = 2 + 2 = 4.$$

Da $LB0(S^{\mathcal{C}}) < UB$ erfüllt ist und wir Teilschedule $S^{\mathcal{C}}$ nicht bereits schon einmal generiert haben, fügen wir den Teilschedule $(\{0, 1\}, (0, 0))$ den Mengen Λ und Ψ hinzu.

Für die Knoten 0 und 2 ist die Bedingung (i) erfüllt, denn es gilt $\langle 0, 2 \rangle \in E$ und $S_0 + 3 = ES_2$. Wir speichern den Zeitpunkt $t = 3$ in der Menge \mathcal{T}_2 und erweitern dann den aktuellen Teilschedule um Knoten 2. Dabei ergibt sich $S^{\mathcal{C}} = (0, 3)$ mit $\mathcal{C} = \{0, 2\}$. Für Teilschedule $S^{\mathcal{C}} = (0, 3)$ erhalten wir die untere Schranke

$$LB0(S^{\mathcal{C}}) = 1 + 2 = 3.$$

Da $LB0(S^{\mathcal{C}}) < UB$ gilt, fügen wir Teilschedule $(\{0, 2\}, (0, 3))$ den Mengen Λ und Ψ hinzu.

Für die Knoten 0 und 3 ist ebenfalls die Bedingung (i) erfüllt und wir setzen $\mathcal{T}_3 := \{2\}$. Es ergibt sich der Teilschedule $S^{\mathcal{C}} = (0, 2)$ mit $\mathcal{C} = \{0, 3\}$. Wir erhalten die untere Schranke $LB0$ für Teilschedule $S^{\mathcal{C}} = (0, 2)$ zu

$$LB0(S^{\mathcal{C}}) = 2 + 2 = 4.$$

Es gilt $LB0(S^{\mathcal{C}}) < UB$ und daher speichern wir Teilschedule $(\{0, 3\}, (0, 2))$ in den Mengen Λ und Ψ . Für die anderen Knoten 4–7 ist weder Bedingung (i) noch (iii) erfüllt. Nun werden alle Teilschedules aus der Menge Λ entfernt und dem Stapel Ω gemäß wachsender Werte von $LB0$ hinzugefügt. Somit erhalten wir $\Omega := \{(\{0, 2\}, (0, 3)), (\{0, 1\}, (0, 0)), (\{0, 3\}, (0, 2))\}$.

In der zweiten Iteration entnehmen wir Teilschedule $(\{0, 2\}, (0, 3))$ dem Stapel Ω . Zunächst ist die workloadbasierte untere Schranke $LBA(S^{\mathcal{C}})$ des Teilschedules $S^{\mathcal{C}} = (0, 3)$ mit $\mathcal{C} = \{0, 2\}$ zu berechnen. Da sich für die

Vorgänge 1 und 7 die planungsabhängigen Zeitfenster $[ES, LC]$ nicht überlappen, planen wir sie zu $S_1 = 0$ und $S_7 = 16$ ein. Somit besteht die Menge der noch nicht betrachteten Vorgänge aus $\hat{\mathcal{C}} = \{3, \dots, 6\}$. Für den Workload dieser Vorgänge ergibt sich $w_1(\hat{\mathcal{C}}) := 9$ und $w_2(\hat{\mathcal{C}}) := 13$ und wir erhalten die workloadbasierte untere Schranke

$$LBA(S^{\mathcal{C}}) := \sum_{k \in \mathcal{R}} \max_{t \in [0, \bar{d}]} r_k^a(S^{\mathcal{C} \cup \hat{\mathcal{C}}}, t) = 2 + 2 = 4.$$

Da $LBA(S^{\mathcal{C}}) < UB$ gilt, wird versucht, den Teilschedule $(\{0, 2\}, (0, 3))$ zu erweitern. Wir initialisieren die Menge $\Lambda = \emptyset$. Die Aktualisierung der ES - und LS -Werte führt zu den folgenden Änderungen: $LS_1 = 6$, $LS_3 = 9$ und $LS_6 = 13$. Es ist nun anhand der Bedingungen (i) und (iii) zu prüfen, ob zwischen einem Knoten $j \in \{1, 3, \dots, 7\}$ und Knoten 0 oder 2 eine bindende Zeit- oder Vorrangbeziehung etabliert werden kann. Für die Vorgänge 0 und 1 treffen die Bedingungen (i) und (iii) zu und wir setzen $\mathcal{T}_1 := \{0\}$. Die untere Schranke für den entstandenen Teilschedule $S^{\mathcal{C}} = (0, 0, 3)$ mit $\mathcal{C} = \{0, 1, 2\}$ ergibt sich zu

$$LB0(S^{\mathcal{C}}) = 2 + 2 = 4.$$

Da die Bedingung $LB0(S^{\mathcal{C}}) < UB$ erfüllt ist, fügen wir den Teilschedule $(\{0, 1, 2\}, (0, 0, 3))$ den Mengen Λ und Ψ hinzu.

Für die Knoten 0 und 3 ist Bedingung (i) und für die Knoten 2 und 3 ist Bedingung (iii) erfüllt. Wir erhalten $\mathcal{T}_3 := \{2, 9\}$ und erweitern den aktuellen Teilschedule um Knoten 3. Es ergeben sich die Teilschedules $S^{\mathcal{C}} = (0, 3, 2)$, $\mathcal{C} = \{0, 2, 3\}$ und $S^{\mathcal{C}} = (0, 3, 9)$, $\mathcal{C} = \{0, 2, 3\}$. Die unteren Schranken berechnen sich zu

$$\begin{aligned} LB0(S^{\mathcal{C}}) &= 3 + 4 = 7 \quad (S^{\mathcal{C}} = (0, 3, 2)) \\ LB0(S^{\mathcal{C}}) &= 2 + 2 = 4 \quad (S^{\mathcal{C}} = (0, 3, 9)). \end{aligned}$$

Da $LB0(S^{\mathcal{C}}) < UB$ nur für den Teilschedule $(\{0, 2, 3\}, (0, 3, 9))$ gilt, fügen wir nur diesen Teilschedule den Mengen Λ und Ψ hinzu.

Für die Knoten 2 und 4 sind die Bedingungen (i) und (iii) erfüllt. Damit ergibt sich $\mathcal{T}_4 := \{1, 9\}$. Die unteren Schranken der zugehörigen Teilschedules lassen sich wie folgt berechnen

$$\begin{aligned} LB0(S^{\mathcal{C}}) &= 1 + 2 = 3 \quad (S^{\mathcal{C}} = (0, 3, 1)) \\ LB0(S^{\mathcal{C}}) &= 1 + 2 = 3 \quad (S^{\mathcal{C}} = (0, 3, 9)). \end{aligned}$$

Da die untere Schranke beider Teilschedules kleiner als UB ist, werden beide Teilschedules in den Mengen Λ und Ψ gespeichert.

Nun betrachten wir die Knoten 2 und 5, für die (i) zutrifft. Wir erhalten $\mathcal{T}_5 := \{10\}$ und erweitern den aktuellen Teilschedule um Knoten 5. Die untere Schranke des entsprechenden Teilschedules ergibt sich zu

$$LB0(S^{\mathcal{C}}) = 1 + 4 = 5.$$

Da $LB0(S^C) < UB$ gilt, fügen wir Teilschedule $(\{0, 2, 5\}, (0, 3, 10))$ den Mengen Λ und Ψ hinzu.

Für die Knoten 2 und 6 sind ebenfalls die Bedingungen (i) und (iii) erfüllt und es ergibt sich die Menge $\mathcal{T}_6 := \{5, 9\}$. Durch Erweiterung des aktuellen Teilschedules erhalten wir $S^C = (0, 3, 5)$ mit $\mathcal{C} = \{0, 2, 6\}$ und $S^C = (0, 3, 9)$ mit $\mathcal{C} = \{0, 2, 6\}$. Die unteren Schranken berechnen sich zu

$$\begin{aligned} LB0(S^C) &= 3 + 3 = 6 \quad (S^C = (0, 3, 5)) \\ LB0(S^C) &= 2 + 2 = 4 \quad (S^C = (0, 3, 9)). \end{aligned}$$

Da beide untere Schranken kleiner als UB sind, werden die Teilschedules den Mengen Λ und Ψ hinzugefügt. Nun werden alle generierten Teilschedules aus der Menge Λ entfernt und dem Stapel Ω gemäß nicht wachsender Werte von $LB0$ hinzugefügt. Es ergibt sich $\Omega := \{(\{0, 2, 4\}, (0, 3, 1)), (\{0, 2, 4\}, (0, 3, 9)), (\{0, 1, 2\}, (0, 0, 3)), (\{0, 2, 3\}, (0, 3, 9)), (\{0, 2, 6\}, (0, 3, 9)), (\{0, 2, 5\}, (0, 3, 10)), (\{0, 2, 6\}, (0, 3, 5)), (\{0, 1\}, (0, 0)), (\{0, 3\}, (0, 2))\}$. Der in den beiden ersten Iterationen entstandene Enumerationsbaum ist in Abbildung 2.43 dargestellt. In der nächsten Iteration ist mit dem Knoten 4 fortzufahren.

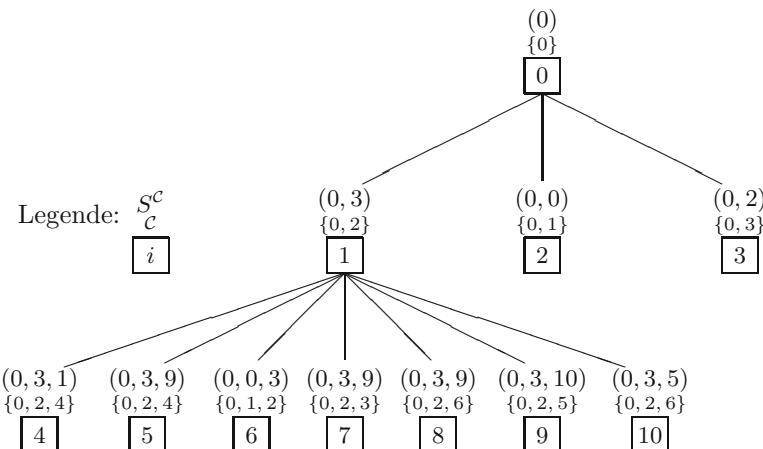


Abb. 2.43. Enumerationsbaum für das Beispiel 2.34

Anhand von Beispiel 2.34 erkennt man, dass die Anwendung des Branch-and-Bound-Verfahrens sehr aufwändig ist. Eine optimale Lösung wird für Ressourceninvestitionprobleme mit mehr als 50 Vorgängen und für Ressourcenabweichungs- sowie Ressourceninvestitionsprobleme mit mehr als 30 Vorgängen i.d.R. nicht in akzeptabler Zeit gefunden. Deshalb verwendet man zur Bestimmung einer akzeptablen Näherungslösung für Projektplanungsprobleme (2.1) mit Zielfunktion (RI), (RD) oder (RL) zumeist eine Heuristik. Im nächsten Kapitel gehen wir näher auf ein solches heuristisches Lösungsverfahren

für Ressourceninvestment-, Ressourcenabweichungs- und Ressourcennivellierungsprobleme ein.

2.3 Heuristisches Lösungsverfahren

In diesem Abschnitt führen wir ein *Prioritätsregelverfahren* zur Bestimmung einer Näherungslösung für Projektplanungsprobleme (2.1) ein. Das Verfahren wird bei Ressourceninvestment-, Ressourcenabweichungs- und Ressourcennivellierungsproblemen angewendet, da die Ermittlung einer exakten Lösung für diese Probleme sehr aufwändig ist. Eine Anwendung des Prioritätsregelverfahrens für Projektplanungsprobleme (2.1) mit den Zielfunktionen (*PD*), (*MFT*), (*WST*), (*E + T*) und (*NPV*) ist prinzipiell möglich, aber i.d.R. nicht sinnvoll, da für diese Probleme leistungsfähige exakte Verfahren existieren (vgl. Abschnitt 2.2).

Das Prioritätsregelverfahren generiert durch die sukzessive Einplanung von Vorgängen eine Näherungslösung und wird daher auch als *Konstruktionsverfahren* bezeichnet. Seien \mathcal{C} die Menge der bereits eingeplanten Vorgänge und $S^{\mathcal{C}}$ der zugehörige Teilschedule. Dann lässt sich das prinzipielle Vorgehen des Prioritätsregelverfahrens wie folgt beschreiben. Ausgehend vom Teilschedule $S^{\mathcal{C}} = (0)$ mit $\mathcal{C} = \{0\}$ und $S_0 = 0$ erweitern wir den aktuellen Teilschedule in jedem Schritt um mindestens einen Vorgang j , indem wir den Startzeitpunkt S_j festlegen. Dies wird solange wiederholt, bis allen Vorgängen des Projektes ein Startzeitpunkt zugewiesen wurde.

Sei $\bar{\mathcal{C}} := V \setminus \mathcal{C}$ die Menge der noch nicht eingeplanten Vorgänge. Wie bereits in Abschnitt 2.2.4 gesehen, ändern sich beim Einplanen einzelner Vorgänge i.d.R. die frühesten und spätesten Startzeitpunkte der übrigen Vorgänge $j \in \bar{\mathcal{C}}$ gemäß

$$ES_j(S^{\mathcal{C}}) := \max\{d_{0j}, \max_{i \in \mathcal{C}}(S_i^{\mathcal{C}} + d_{ij})\} \quad (2.9)$$

$$LS_j(S^{\mathcal{C}}) := \min\{-d_{j0}, \min_{i \in \mathcal{C}}(S_i^{\mathcal{C}} - d_{ji})\}. \quad (2.10)$$

Vorgang j kann somit nur innerhalb des Zeitfensters $W_j(S^{\mathcal{C}}) = [ES_j(S^{\mathcal{C}}), LS_j(S^{\mathcal{C}})]$ eingeplant werden.

Beim Einplanen von Vorgang j wählen wir für den Startzeitpunkt S_j einen Zeitpunkt $t \in W_j(S^{\mathcal{C}})$, der bezüglich der zugrunde liegenden Zielfunktion am günstigsten erscheint, d.h. bei dem sich der Zielfunktionswert durch die Einplanung von Vorgang j zum Zeitpunkt $S_j := t$ am wenigsten verschlechtert. Die Bewertung eines möglichen Startzeitpunktes t erfolgt dabei aufgrund der Differenz der Zielfunktionswerte der Teilschedules $S^{\mathcal{C} \cup \{j\}}$ und $S^{\mathcal{C}}$, was voraussetzt, dass für einen beliebigen Teilschedule $S^{\mathcal{C}}$ ein zugehöriger Zielfunktionswert $f(S^{\mathcal{C}})$ bestimmt werden kann. Dies ist für alle (summen-)separierbaren Zielfunktionen der Form $f(S) := \sum_{i \in V} f_i(S_i)$, wie z.B. (*PD*), (*MFT*), (*WST*), (*E + T*) und (*NPV*), bzw. für alle ressourcengebrauchsabhängigen Zielfunktionen der Form $f(S) := f(r_k(S, \cdot)_{k \in \mathcal{R}})$, wie z.B.

(RI) , (RD) und (RL) , die prinzipiell für jedes Ressourcenprofil bzw. jede Menge von Ressourcenprofilen ausgewertet werden können, der Fall. Betrachten wir die Änderung des Zielfunktionswertes, die sich bei der Einplanung eines einzelnen Vorgangs j ergibt.

Definition 2.35 (Erweiterungskosten). Seien S^C ein Teilschedule und f eine separierbare oder eine ressourcengebrauchsabhängige Zielfunktion. Weiter seien j ein noch nicht eingeplanter Vorgang, $S_j \in W_j(S^C)$ und $S^{C \cup \{j\}}$ ein Teilschedule, für den $S_i^{C \cup \{j\}} = S_i^C$ für alle $i \in \mathcal{C}$ und $S_j^{C \cup \{j\}} = S_j$ gilt. Dann bezeichnen wir mit

$$f^a(S^C, j, S_j) := f(S^{C \cup \{j\}}) - f(S^C)$$

die *Erweiterungskosten* (additional cost), die entstehen, wenn wir den Teilschedule S^C um Vorgang j mit Startzeitpunkt S_j erweitern.

Beispiel 2.36. Betrachten wir den MPM-Netzplan in Abbildung 2.44 mit fünf Vorgängen und einer erneuerbaren Ressource, d.h. der Index k kann entfallen. Seien $\mathcal{C} = \{0, 1, 2, 4\}$ die Menge der bereits eingeplanten Vorgänge und $S^C = (0, 1, 4, 8)$ der entsprechende Teilschedule. Vorgang 3 kann innerhalb des Zeitfensters $W_3(S^C) = [2, 7]$ eingeplant werden. Im Folgenden wollen wir für die Zielfunktionen (RI) , (RD) und (RL) die Erweiterungskosten bestimmen, die sich ergeben, wenn wir Vorgang 3 zum Zeitpunkt $S_3 = 4$ einplanen (vgl. Abb. 2.44).

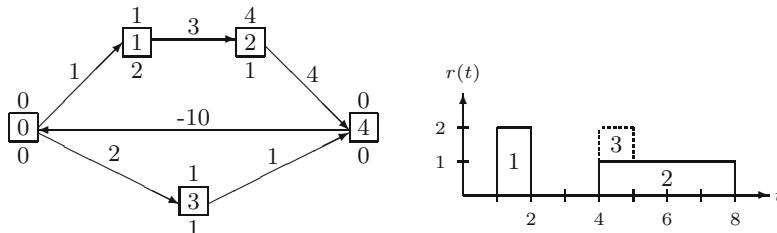


Abb. 2.44. Projektnetzplan und mögliches Ressourcenprofil

Für die Zielfunktion des Ressourceninvestmentproblems (RI) erhalten wir

$$\begin{aligned} f^a(S^C, 3, 4) &= c^P \max_{t \in [0, \bar{d}]} r_k(S^{C \cup \{3\}}, t) - c^P \max_{t \in [0, \bar{d}]} r_k(S^C, t) \\ &= (2 - 2) c^P = 0, \end{aligned}$$

für die Zielfunktion (RD) mit $Y = 1$ erhalten wir

$$\begin{aligned} f^a(S^C, 3, 4) &= c^D \int_{t \in [0, \bar{d}]} [r_k(S^{C \cup \{3\}}, t) - 1]^+ dt \\ &\quad - c^D \int_{t \in [0, \bar{d}]} [r_k(S^C, t) - 1]^+ dt = (2 - 1) c^D = c^D \end{aligned}$$

und für die Ressourcennivellierungszielfunktion (*RL*) ergibt sich

$$f^a(S^C, 3, 4) = \int_{t \in [0, \bar{d}]} r_k^2(S^C \cup \{3\}, t) - \int_{t \in [0, \bar{d}]} r_k^2(S^C, t) = 11 - 8 = 3.$$

Wie bereits erläutert, terminiert unser Konstruktionsverfahren einen Vorgang nach dem anderen, wobei der nächste einzuplanende Vorgang j immer zu einem Zeitpunkt $t \in W_j(S^C)$ eingeplant wird, an dem die Erweiterungskosten $f^a(S^C, j, t)$ minimal sind. Da die Anzahl der Zeitpunkte $t \in W_j(S^C)$ i.d.R. überabzählbar ist, definieren wir für die verschiedenen Zielfunktionen jeweils eine Menge von *Entscheidungszeitpunkten* $\mathcal{D}_j(S^C)$, die mindestens einen Minimalpunkt von $f^a(S^C, j, \cdot)$ auf $W_j(S^C)$ enthält.

Die Zielfunktion des Ressourceninvestmentproblems (*RI*) gehört, wie bereits in Abschnitt 2.1.5 gezeigt, zu den lokal regulären Funktionen und ist somit auf der Isoordnungsmenge jedes zeitzulässigen Schedules regulär. Insbesondere ist (*RI*) konstant auf jeder Isoordnungsmenge, was folgenden Rückschluss zulässt. Für die Zielfunktion (*RI*) sind die Erweiterungskosten $f^a(S^C, j, S_j)$ konstant auf jedem Intervall (eventuell offen oder halboffen) von Startzeitpunkten, für die die entsprechenden Teilschedules $S^C \cup \{j\}$ die gleiche Ordnung induzieren. In Abbildung 2.45 sind die Erweiterungskosten konstant auf den Intervallen $[0, t]$, $]t, t'$, $[t', t'$ und $]t', t''$.¹³

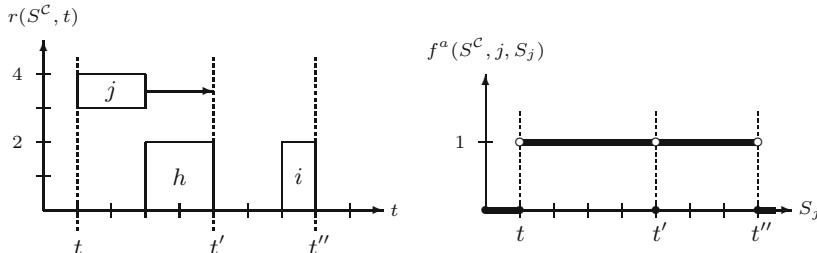


Abb. 2.45. Ressourcenprofil des Teilschedules S^C und Erweiterungskosten

Aus dieser Eigenschaft der Erweiterungskosten lässt sich folgern, dass bei der Einplanung eines Vorgangs $j \in \bar{\mathcal{C}}$ neben dem $ES_j(S^C)$ noch Zeitpunkte zu betrachten sind, an denen mindestens ein bereits eingeplanter Vorgang endet und die Erweiterungskosten fallen können (Vorrangbeziehung fällt weg). Sei

$$CT(S^C) := \{t \in [0, \bar{d}] \mid \text{es existiert } i \in \mathcal{C} : t = S_i^C + p_i\}$$

die Menge aller Zeitpunkte, zu denen gemäß S^C mindestens ein Vorgang endet (completion times). Dann erhalten wir für die Zielfunktion (*RI*) die folgende Menge von Entscheidungszeitpunkten

$$\mathcal{D}_j(S^C) := \{ES_j(S^C)\} \cup (W_j(S^C) \cap CT(S^C)).$$

¹³ In den Abbildungen 2.45 und 2.47 gehen wir davon aus, dass Vorgang j zu jedem Zeitpunkt im Planungszeitraum starten kann.

Beispiel 2.37. Wir betrachten die in Abbildung 2.46 dargestellten Ressourcenprofile. Die Menge der bereits eingeplanten Vorgänge sei $\mathcal{C} = \{0, 1, 2, 3\}$ mit $S^{\mathcal{C}} = (0, 0, 3, 3)$. Der als Nächstes zu betrachtende Vorgang 4 mit $p_4 = 3$ kann innerhalb des Zeitfensters $W_4(S^{\mathcal{C}}) = [1, 4]$ eingeplant werden. Die Erweiterungskosten der Funktion (RI) sind auf dem Intervall $[1, 2[$ konstant, es gilt $f^a(S^{\mathcal{C}}, 4, t) = 1$ für $t \in [1, 2[$. Die durch die Teilschedules $S^{\mathcal{C} \cup \{4\}}$ mit $S_4 \in [1, 2[$ implizierte strenge Ordnung (Menge von Vorrangbeziehungen) ist $O(S^{\mathcal{C}}) = \{(1, 2), (1, 3)\}$ (vgl. Abb. 2.46a). Weiterhin sind die Erweiterungskosten auf dem Intervall $[2, 4[$ konstant. Wir erhalten $f^a(S^{\mathcal{C}}, 4, t') = 1$ mit $t' \in [2, 4[$ und die durch die Teilschedules $S^{\mathcal{C} \cup \{4\}}$ mit $S_4 \in [2, 4[$ implizierte strenge Ordnung ist $O(S^{\mathcal{C}}) = \{(1, 2), (1, 3), (1, 4)\}$ (vgl. Abb. 2.46b). Für den Zeitpunkt $t = 4$ ergibt sich $f^a(S^{\mathcal{C}}, 4, 4) = 0$ und die durch $S^{\mathcal{C} \cup \{4\}} = (0, 0, 3, 3, 4)$ implizierte strenge Ordnung ist $O(S^{\mathcal{C}}) = \{(1, 2), (1, 3), (1, 4), (3, 4)\}$ (vgl. Abb. 2.46c). Abbildung 2.46d zeigt zusammenfassend den Verlauf der Erweiterungskosten. Die Menge der Entscheidungszeitpunkte ist $\mathcal{D}_4(S^{\mathcal{C}}) = \{1, 2, 4\}$, wobei für $S_4 = 4$ die Erweiterungskosten minimal sind.

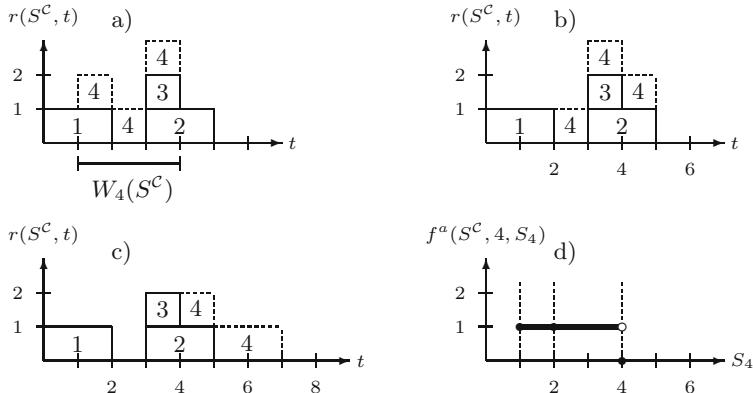


Abb. 2.46. Ressourcenprofile des Teilschedules $S^{\mathcal{C}}$ und Erweiterungskosten

Bei den Zielfunktionen (RD) und (RL) handelt es sich um lokal konkav Funktionen, die auf der Isoordnungsmenge jedes zeitzulässigen Schedules konkav sind (vgl. Abschnitt 2.1.5). Für diese Funktionen sind die Erweiterungskosten $f^a(S^{\mathcal{C}}, j, S_j)$ konkav auf jedem Intervall von Startzeitpunkten, für die die entsprechenden Teilschedules $S^{\mathcal{C} \cup \{j\}}$ die gleiche Ordnung induzieren. Da sich diese Ordnung mit S_j ändert, falls sich der Startzeitpunkt von Vorgang j hinter den Endzeitpunkt eines Vorgang $i \in \mathcal{C}$ verschiebt (Vorrangbeziehung (i, j) kommt hinzu) oder wenn sich der Endzeitpunkt von Vorgang j hinter den Startzeitpunkt von Vorgang $l \in \mathcal{C}$ verschiebt (Vorrangbeziehung (j, l) fällt weg), müssen wir bei der Einplanung von Vorgang $j \in \mathcal{C}$ neben dem $ES_j(S^{\mathcal{C}})$ und dem $LS_j(S^{\mathcal{C}})$ alle Zeitpunkte t betrachten, an denen ein Vorgang $i \in \mathcal{C}$ endet und alle Zeitpunkte t' , für die zum Zeitpunkt $t' + p_j$ ein

Vorgang $l \in \mathcal{C}$ startet. In Abbildung 2.47 ist der Sachverhalt veranschaulicht, wir geben die Erweiterungskosten für die Zielfunktion (RD) mit $Y = 1$ an.

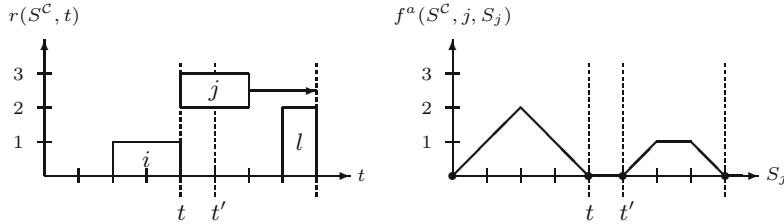


Abb. 2.47. Ressourcenprofil des Teilschedules S^C und Erweiterungskosten

Sei

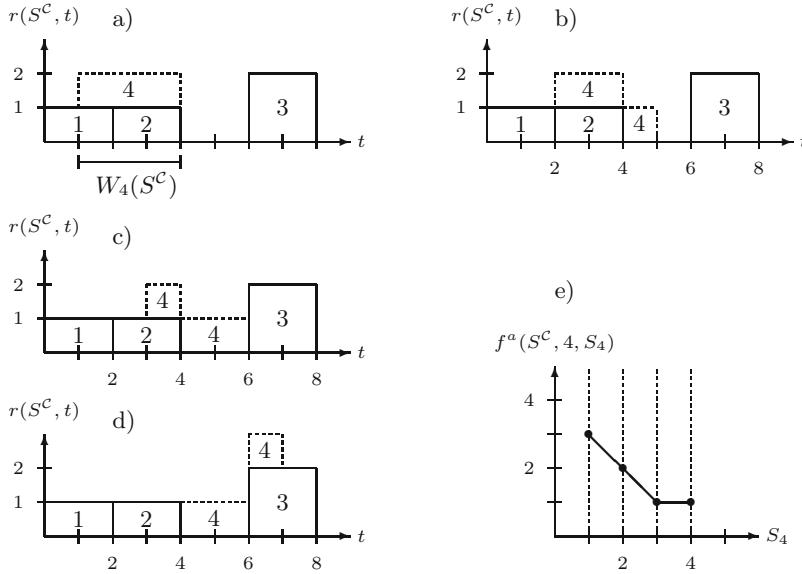
$$ST(S^C) := \{t \in [0, \bar{d}] \mid \text{es existiert } i \in \mathcal{C} : t = S_i^C\}$$

die Menge aller Zeitpunkte, zu denen gemäß S^C mindestens ein Vorgang startet (start times). Dann erhalten wir für die Zielfunktionen (RD) und (RL) die folgende Menge von Entscheidungszeitpunkten

$$\begin{aligned} \mathcal{D}_j(S^C) := & \{ES_j(S^C), LS_j(S^C)\} \cup (W_j(S^C) \cap CT(S^C)) \\ & \cup \{t \in W_j(S^C) \mid t + p_j \in ST(S^C)\}. \end{aligned}$$

Beispiel 2.38. Betrachten wir die in Abbildung 2.48 dargestellten Ressourcenprofile. Die Menge der bereits eingeplanten Vorgänge ist $\mathcal{C} = \{0, 1, 2, 3\}$ mit $S^C = (0, 0, 2, 6)$. Als Nächstes ist der Vorgang 4 mit $p_4 = 3$ im Intervall $W_4(S^C) = [1, 4]$ einzuplanen. Planen wir Vorgang 4 zum Zeitpunkt $ES_4 = 1$ ein, so ist die zu $S^{C \cup \{4\}}$ gehörige Ordnung $O(S^{C \cup \{4\}}) = \{(1, 2), (1, 3), (2, 3), (4, 3)\}$. Diese Ordnung ändert sich, falls wir den Einplanungszeitpunkt von 4 erhöhen. Zu den Zeitpunkten $t = 2, t = 3 + \varepsilon$ mit $0 < \varepsilon \ll 1$ und $t = 4$ kommen Vorrangbeziehungen hinzu oder fallen weg. Auf den resultierenden Intervallen $[1, 2], [2, 3], [3, 4]$ und $[4, 4]$ verlaufen die Erweiterungskosten konkav und da die Zielfunktionen (RD) und (RL) stetig sind, nehmen die Erweiterungskosten jeweils auf einem der beiden Intervallrandpunkte ihr Minimum an (vgl. Abb. 2.48a bis d). Abbildung 2.48e zeigt den Verlauf der Erweiterungskosten für die Zielfunktion (RD) mit $Y = 1$. Die Menge der Entscheidungszeitpunkte ist $\mathcal{D}_4(S^C) = \{1, 2, 3, 4\}$, wobei die Erweiterungskosten für $S_4 = 3$ und $S_4 = 4$ minimal sind. Im Folgenden wählen wir immer die größte Minimalstelle, d.h. $S_4 = 4$.

Betrachten wir nun die Vorgehensweise des Konstruktionsverfahrens im Einzelnen. In jedem Schritt weisen wir einem Vorgang j einen Startzeitpunkt $S_j^+ \in \mathcal{D}_j(S^C)$ zu, wobei S_j^+ die größte Minimalstelle der Erweiterungskosten $f^a(S^C, j, \cdot)$ auf $\mathcal{D}_j(S^C)$ darstellt. Auf diese Weise wird zum Beispiel für ein Ressourceninvestmentproblem sukzessive ein Outtree eines Ordnungsnetzplans $N(O(S))$ erzeugt, d.h. ein Minimalpunkt eines Schedulepolytops

Abb. 2.48. Ressourcenprofile des Teilschedules S^C

$\mathcal{S}_T(O(S)), S \in \mathcal{S}_T$. Für Probleme mit den Zielfunktionen (RD) oder (RL) wird sukzessive ein Gerüst, d.h. ein Extrempunkt eines Schedulepolytops konstruiert.

Den nächsten einzuplanenden Vorgang $j \in \bar{\mathcal{C}}$ bestimmen wir gemäß einer *Prioritätsregel*. Dabei bestimmen wir zunächst für alle Vorgänge $j \in \bar{\mathcal{C}}$ den entsprechenden Prioritätsregelwert und wählen dann unter den Vorgängen mit maximalem Prioritätsregelwert denjenigen mit kleinstem Vorgangsnummer aus. Die Wahl der Prioritätsregel hängt von der zugrunde liegenden Zielfunktion ab. In der Literatur werden viele verschiedene Prioritätsregeln angeführt. Prioritätsregeln, die für die Zielfunktionen (RI), (RD) und (RL) „gute“ Lösungen liefern, sind:

GRD-Regel (Greatest Resource Demand): Wähle Vorgang j mit dem größten Ressourcenbedarf, d.h. $j := \min\{i \in \bar{\mathcal{C}} \mid \max_{h \in \bar{\mathcal{C}}} \sum_{k \in \mathcal{R}} r_{hk}\}$.

GRDT-Regel (GRD per Time unit): Wähle Vorgang j mit dem größten Ressourcenbedarf pro Zeiteinheit, d.h. $j := \min\{i \in \bar{\mathcal{C}} \mid \max_{h \in \bar{\mathcal{C}}} \sum_{k \in \mathcal{R}} r_{hk}\}$.

LST-Regel (Latest Start Time): Wähle Vorgang j mit dem aktuell kleinsten spätesten Startzeitpunkt, d.h. $j := \min\{i \in \bar{\mathcal{C}} \mid \min_{h \in \bar{\mathcal{C}}} LS_h(S^C)\}$.

MST-Regel (Minimum Slack Time): Wähle Vorgang j mit der aktuell kleinsten Gesamtpufferzeit, d.h. $j := \min\{i \in \bar{\mathcal{C}} \mid \min_{h \in \bar{\mathcal{C}}} TF_h(S^C)\}$.

Allgemein unterscheidet man zwischen *statischen* und *dynamischen* Prioritätsregeln. Bei den statischen Prioritätsregeln (GRD und GRDT) werden

die Prioritätsregelwerte nur einmal zu Beginn des Verfahrens berechnet. Bei dynamischen Prioritätsregeln werden die Prioritätsregelwerte jedesmal wenn ein Vorgang eingeplant wurde, aktualisiert. Die *LST*-Regel und die *MST*-Regel können statisch oder dynamisch verwendet werden, wobei die dynamischen Versionen mit *LSTd* bzw. *MSTd* bezeichnet werden. Bei der Auswahl von Vorgang $j \in \bar{\mathcal{C}}$ kann anstelle einer einzigen Prioritätsregel auch eine Kombination von Prioritätsregeln angewendet werden. Die Prioritätsregelkombination *MST-GRD* bedeutet beispielsweise, dass zuerst alle Vorgänge $j \in \bar{\mathcal{C}}$ mit kleinsten Gesamtpufferzeit ausgewählt werden und dann unter diesen derjenige mit dem größten Ressourcenbedarf. Da es i.d.R. keine Prioritätsregel bzw. Prioritätsregelkombination gibt, die alle anderen dominiert, wendet man Prioritätsregelverfahren häufig nicht nur einmal (Single Pass), sondern mehrmals mit unterschiedlichen Prioritätsregelkombinationen an (Multi Pass) und wählt dann die beste Lösung aus.

Ist der nächste einzuplanende Vorgang $j \in \bar{\mathcal{C}}$ bestimmt, berechnen wir in Abhängigkeit der Zielfunktion die Menge der Entscheidungszeitpunkte $\mathcal{D}_j(S^C)$ von Vorgang j . $\mathcal{D}_j(S^C)$ enthält mindestens eine Minimalstelle S_j^+ der Erweiterungskosten $f^a(S^C, j, \cdot)$ auf $W_j(S^C) = [ES_j(S^C), LS_j(S^C)]$. Wir bestimmen aus Gründen der Eindeutigkeit die größte Minimalstelle S_j^+ von $f^a(S^C, j, \cdot)$ und setzen $S_j := S_j^+$. Ferner entfernen wir j aus der Menge $\bar{\mathcal{C}}$ der noch nicht eingeplanten Vorgänge und fügen j in die Menge \mathcal{C} der bereits eingeplanten Vorgänge ein. Danach müssen die *ES*- und *LS*-Werte für alle Vorgänge $i \in \bar{\mathcal{C}}$ gemäß (2.9) und (2.10) aktualisiert werden. Gilt $ES_i = LS_i$ so werden die jeweiligen Vorgänge $i \in \bar{\mathcal{C}}$ zu ES_i terminiert. In dieser Art und Weise fahren wir fort, bis allen Vorgängen $i \in V$ ein Startzeitpunkt zugewiesen ist. Algorithmus 2.39 fasst die einzelnen Schritte des Prioritätsregelverfahrens zur Bestimmung einer Näherungslösung für Projektplanungsprobleme (2.1) mit Zielfunktion (*RI*), (*RD*) und (*RL*) zusammen.

Algorithmus 2.39 (Prioritätsregelverfahren für Problem (2.1) mit Zielfunktion (*RI*), (*RD*) bzw. (*RL*)).

Initialisierung:

Setze $\mathcal{C} := \{0\}$, $S^C := (0)$ und $\bar{\mathcal{C}} := V \setminus \{0\}$.

Bestimme die Längen längster Wege d_{ij} für alle $i, j \in V$.

Setze $ES_i := d_{0i}$ und $LS_i := -d_{i0}$ für alle $i \in V \setminus \{0\}$.

Für alle $i \in \bar{\mathcal{C}}$ mit $LS_i = ES_i$:

Setze $S_i := ES_i$, $\mathcal{C} := \mathcal{C} \cup \{i\}$ und $\bar{\mathcal{C}} := \bar{\mathcal{C}} \setminus \{i\}$.

Hauptschritt:

Solang $\bar{\mathcal{C}} \neq \emptyset$:

Wähle Vorgang $j \in \bar{\mathcal{C}}$ mit höchster Priorität.

Bestimme $\mathcal{D}_j(S^C)$ in Abhängigkeit von der Zielfunktion.

Bestimme die größte Minimalstelle S_j^+ von $f^a(S^C, j, \cdot)$ auf $\mathcal{D}_j(S^C)$.

Setze $S_j := S_j^+$, $\mathcal{C} := \mathcal{C} \cup \{j\}$ und $\overline{\mathcal{C}} := \overline{\mathcal{C}} \setminus \{j\}$.

Für alle $i \in \overline{\mathcal{C}}$:

Setze $ES_i := \max(d_{0i}, S_j + d_{ji})$ und $LS_i := \min(-d_{i0}, S_j - d_{ij})$.

Für alle $i \in \overline{\mathcal{C}}$ mit $LS_i = ES_i$:

Setze $S_i := ES_i$, $\mathcal{C} := \mathcal{C} \cup \{i\}$ und $\overline{\mathcal{C}} := \overline{\mathcal{C}} \setminus \{i\}$

Rückgabe S .

Beispiel 2.40. Wir betrachten den in Abbildung 2.49 dargestellten Projektnetzplan mit sechs realen Vorgängen, einer erneuerbaren Ressource und einer maximalen Projektdauer von $\bar{d} = 9$. Zur Minimierung der Zielfunktion (RL) wenden wir Algorithmus 2.39 mit der Prioritätsregel GRD an.

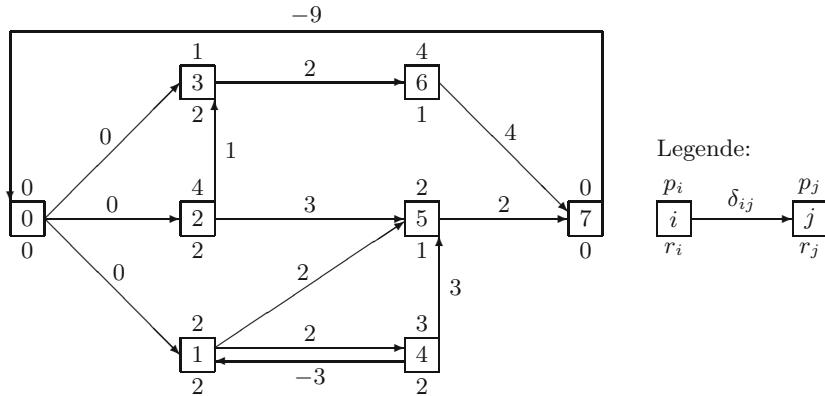


Abb. 2.49. Projektnetzplan mit sechs realen Vorgängen

Tabelle 2.15 zeigt die frühesten und spätesten Startzeitpunkte sowie die Ressourcenbedarfe der Vorgänge $i \in V$.

Tabelle 2.15. ES_i , LS_i und $r_i p_i$ für $i = 0, \dots, 7$

i	0	1	2	3	4	5	6	7
ES_i	0	0	0	1	2	5	3	7
LS_i	0	2	2	3	4	7	5	9
$r_i p_i$	0	4	8	2	6	2	4	0

Im Initialisierungsschritt planen wir Vorgang 0 zum Zeitpunkt 0 ein, d.h. wir setzen $\mathcal{C} := \{0\}$ und $S^{\mathcal{C}} := (0)$. Der Vorgang mit dem größten Ressourcenbedarf ist Vorgang 2. Da außer Vorgang 0 noch kein Vorgang eingeplant wurde, erhalten wir als Menge der Entscheidungszeitpunkte für Vorgang 2 den frühesten und spätesten Startzeitpunkt von 2, d.h. $\mathcal{D}_2(S^{\mathcal{C}}) = \{0, 2\}$. Die Erweiterungskosten für die Zeitpunkte $t = 0$ und $t = 2$ sind $f^a(S^{\mathcal{C}}, 2, 0) =$

$f^a(S^C, 2, 2) = 16 - 0 = 16$. Folglich planen wir Vorgang 2 zu $S_2 = 2$ ein, der größten Minimalstelle von f^a . Da Vorgang 2 zu seinem spätesten Startzeitpunkt eingeplant wurde, aktualisieren wir die frühesten Startzeitpunkte der Vorgänge 3, 6 und 7, indem wir $ES_3 := 3$, $ES_6 := 5$ und $ES_7 := 9$ setzen. Damit gilt für die Vorgänge $i = 3, 6, 7$ nun $ES_i = LS_i$ und wir setzen $S_3 := 3$, $S_6 := 5$, $S_7 := 9$ (vgl. Abb. 2.50a).

Im zweiten Hauptschritt wählen wir Vorgang 4 als nächsten einzuplanenden Vorgang. Die Menge der Entscheidungszeitpunkte von Vorgang 4 besteht aus $\mathcal{D}_4(S^C) = \{2, 4\}$. Für die Erweiterungskosten gilt $f^a(S^C, 4, 2) = 80 - 36 = 44$ und $f^a(S^C, 4, 4) = 72 - 36 = 36$. Wir planen Vorgang 4 somit zu $S_4 = 4$ ein. Nun sind die frühesten Startzeitpunkte der Vorgänge 1 und 5 zu aktualisieren, dabei erhalten wir $ES_1 := 1$ und $ES_5 := 7$. Für Vorgang 5 gilt $ES_5 = LS_5$ und wir setzen $S_5 := 7$ (vgl. Abb. 2.50b).

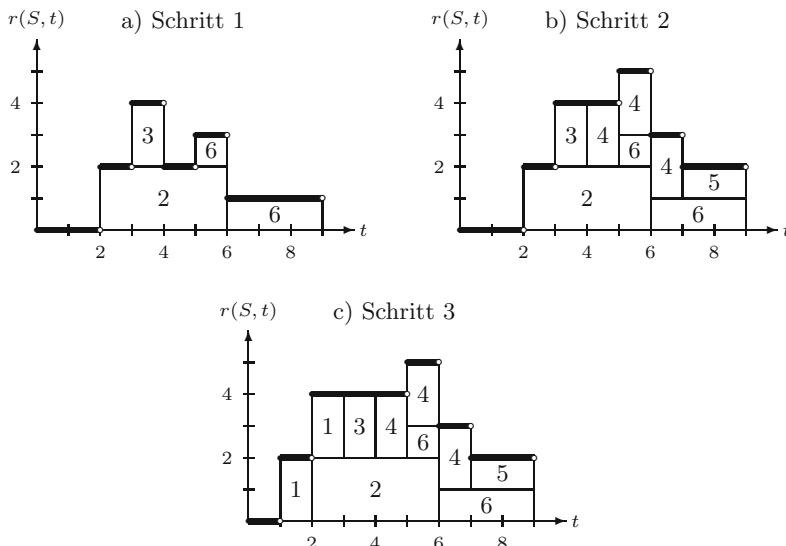


Abb. 2.50. Ressourcenprofile der einzelnen Teilschedules

Im letzten Schritt wählen wir Vorgang 1 als nächsten einzuplanenden Vorgang. Die Menge der Entscheidungszeitpunkte von Vorgang 1 ist $\mathcal{D}_1(S^C) = \{1, 2\}$. Für die Erweiterungskosten gilt $f^a(S^C, 1, 1) = 94 - 78 = 16$ und $f^a(S^C, 1, 2) = 110 - 78 = 32$. Vorgang 1 wird daher zu $S_1 = 1$ eingeplant. In Abbildung 2.50c ist das entstandene Ressourcenprofil mit Zielfunktionswert 94 veranschaulicht.

Bei den Zielfunktionen (RI), (RD) und (RL) hängen die Erweiterungskosten $f^a(S^C, j, S_j)$ vom aktuellen Ressourcengebrauch ab. Dies bedeutet, dass sich die Erweiterungskosten $f^a(S^C, j, S_j)$ ändern können, wenn wir vor der

Einplanung von Vorgang j einen anderen Vorgang $h \in \bar{\mathcal{C}}$ einplanen. Im Folgenden zeigen wir, wie man schon vor der Einplanung eines Vorgangs $h \in \bar{\mathcal{C}}$ einen Teil seines Ressourcenbedarfs bei der Bestimmung der Erweiterungskosten für Vorgang j berücksichtigen kann. Hierzu führen wir den Begriff des Basisintervalls eines Vorgangs ein, das angibt, zu welchen Zeitpunkten sich ein noch nicht eingeplanter Vorgang in Ausführung befindet, und zwar unabhängig davon, welche Startzeitpunkte wir ihm und den übrigen noch nicht eingeplanten Vorgängen zuweisen.

Seien $EC_i = ES_i + p_i$ der früheste Zeitpunkt, zu dem Vorgang i enden kann, und LS_i der späteste Startzeitpunkt von Vorgang i unter Berücksichtigung der maximalen Projektdauer \bar{d} . Vorgang i muss dann unabhängig von seinem Startzeitpunkt $S_i \in [ES_i, LS_i]$ im Zeitintervall $[LS_i, EC_i]$ ausgeführt werden. Das Intervall $[LS_i, EC_i]$ wird daher als *Basisintervall* von Vorgang i bezeichnet. Gilt für einen Vorgang $i \in V$ beispielsweise $p_i = 9$, $ES_i = 2$ und $LS_i = 4$, dann muss Vorgang i im Basisintervall $[4, 11]$ ausgeführt werden unabhängig davon, zu welchen Zeitpunkten er und die übrigen Vorgänge $j \in V$ eingeplant werden (vgl. Abb. 2.51).

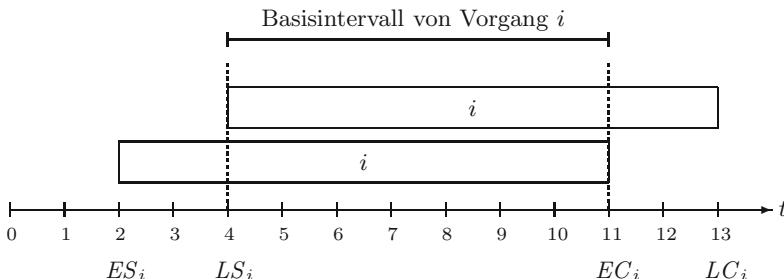


Abb. 2.51. Basisintervall von Vorgang i

Ist $TF_i = 0$ für einen Vorgang $i \in V$ und damit $S_i = ES_i = LS_i$, so nennen wir Vorgang i *fixiert*. Ist für einen Vorgang i die Ungleichung $0 < TF_i < p_i$ erfüllt, so nennen wir den Vorgang *teilfixiert*. Ein teilfixierter Vorgang i wird unabhängig von seinem Startzeitpunkt immer im Basisintervall $[LS_i, EC_i]$ ausgeführt. Da das zugrunde liegende Projekt zum Zeitpunkt 0 startet ($S_0 = 0$), hängt die Menge der fixierten und teilfixierten Vorgänge von der maximalen Projektdauer \bar{d} ab. Mit wachsendem \bar{d} wird die Anzahl der teilfixierten Vorgänge kleiner.

In Abhängigkeit von Teilschedule S^C ergibt sich das *planungsabhängige Basisintervall* eines noch nicht eingeplanten Vorgangs $j \in \bar{\mathcal{C}}$ zu $[LS_j(S^C), EC_j(S^C)]$. Setzen wir $ES_i := LS_i := S_i$ für alle eingeplanten Vorgänge $i \in \mathcal{C}$ (deren Startzeitpunkt somit schon fixiert ist), so stellt

$$\mathcal{A}^b(S^C, t) := \{i \in V \mid LS_i(S^C) \leq t < EC_i(S^C)\}$$

mit $t \in [0, \bar{d}]$ die Menge der Vorgänge dar, die sich zum Zeitpunkt t in Ausführung befinden, und zwar unabhängig davon zu welchem Zeitpunkt die noch nicht eingeplanten Vorgänge $j \in \bar{\mathcal{C}}$ eingeplant werden. Vorgänge mit Dauer 0 finden dabei keine Berücksichtigung, da sich die Ressourcenprofile durch ihre Einplanung nicht ändern. Für einen Teilschedule S^C ergibt sich die Menge an Ressource k , die zum Zeitpunkt $t \in [0, \bar{d}]$ benötigt wird, unter Berücksichtigung aller fixierten und teilfixierten Vorgänge zu

$$r_k^b(S^C, t) := \sum_{i \in \mathcal{A}^b(S^C, t)} r_{ik}.$$

Die Erweiterungskosten für die Zielfunktionen (RI) , (RD) und (RL) lassen sich unter Berücksichtigung fixierter und teilfixierter Vorgänge modifizieren. Unter Einbeziehung des Ressourcenprofils $r_k^b(S^C, \cdot)$ ergeben sich für das Ressourceninvestmentproblem die *modifizierten Erweiterungskosten* f^b

$$f^b(S^C, j, S_j) := \sum_{k \in \mathcal{R}} c_k^P \left(\max_{t \in [0, \bar{d}]} r_k^b(S^{C \cup \{j\}}, t) - \max_{t \in [0, \bar{d}]} r_k^b(S^C, t) \right),$$

für das Ressourcenabweichungsproblem

$$f^b(S^C, j, S_j) := \sum_{k \in \mathcal{R}} c_k^D \int_{t \in [0, \bar{d}]} \left([r_k^b(S^{C \cup \{j\}}, t) - Y_k]^+ - [r_k^b(S^C, t) - Y_k]^+ \right) dt$$

und für das Ressourcennivellierungsproblem

$$f^b(S^C, j, S_j) := \sum_{k \in \mathcal{R}} \int_{t \in [0, \bar{d}]} \left((r_k^b)^2(S^{C \cup \{j\}}, t) - (r_k^b)^2(S^C, t) \right) dt.$$

Um die Effizienz unseres Prioritätsregelverfahrens (vgl. Algorithmus 2.39) basierend auf dem unvermeidbaren Ressourceneinsatz fixierter und teilfixierter Vorgänge zu steigern, ersetzen wir f^a durch f^b . Im Folgenden erläutern wir die Vorgehensweise des modifizierten Prioritätsregelverfahrens an einem Beispiel.

Beispiel 2.41. Betrachten wir wieder den in Abbildung 2.49 dargestellten Projektnetzplan und nehmen an, dass die Zielfunktion (RL) zu minimieren sei. Wir bestimmen eine Näherungslösung für das zugrunde liegende Problem mit Algorithmus 2.39 und der Prioritätsregel GRD , wobei wir zur Bestimmung der Erweiterungskosten die Funktion f^b verwenden. Die frühesten und spätesten Startzeitpunkte sowie die Ressourcenbedarfe der Vorgänge $i \in V$ sind in Tabelle 2.15 dargestellt.

Im Initialisierungsschritt setzen wir $\mathcal{C} := \{0\}$, $S_0 := 0$ und $S^C := (0)$. Für alle Vorgänge $i \in V \setminus \{0\}$ beträgt die Gesamtpufferzeit $TF_i = 2$ und für die Vorgänge $i = 2, 4, 6$ gilt $TF_i < p_i$. Die Vorgänge 2, 4 und 6 sind somit teilfixiert. Das Ressourcenprofil ist in Abbildung 2.52a dargestellt.

Der Vorgang mit dem größtem Ressourcenbedarf ist Vorgang 2, er ist als Erstes einzuplanen. Wir erhalten als Menge der Entscheidungszeitpunkte von Vorgang 2 den frühesten und spätesten Startzeitpunkt von Vorgang 2, d.h. $\mathcal{D}_2(S^C) = \{0, 2\}$. Die Erweiterungskosten f^b zu den Zeitpunkten 0 und 2 sind $f^b(S^C, 2, 0) = 22 - 14 = 8$ und $f^b(S^C, 2, 2) = 34 - 14 = 20$. Somit planen wir Vorgang 2 zu $S_2 = 0$ ein. Die Aktualisierung der ES - und LS -Werte der noch nicht eingeplanten Vorgänge führt zu keinen Änderungen. Das Ressourcenprofil ist in Abbildung 2.52b dargestellt, wobei der fixierte Vorgang 2 mit durchgezogener Linie eingezeichnet wurde.

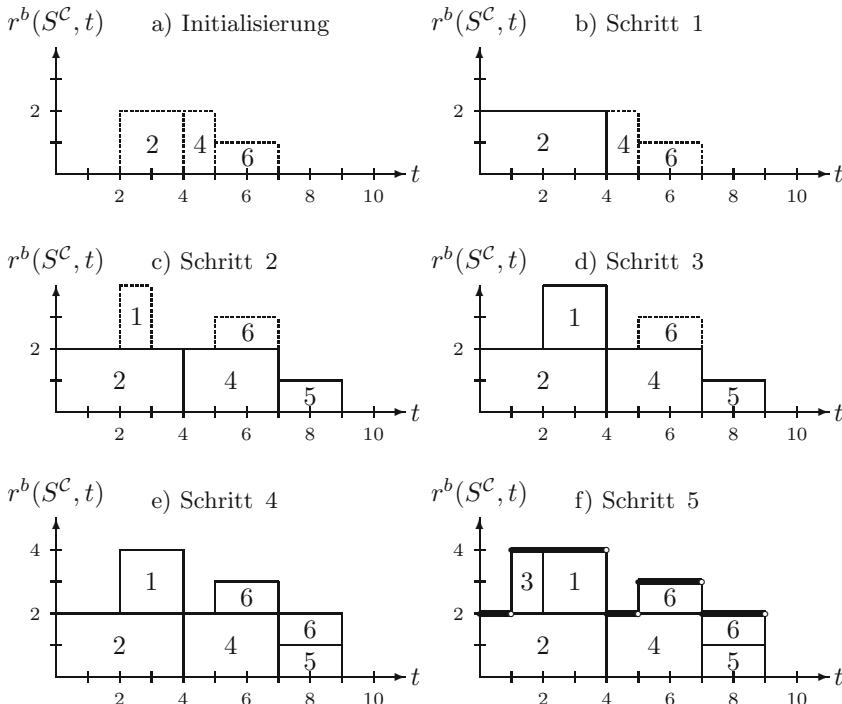


Abb. 2.52. Ressourcenprofile der einzelnen Schritte

Als Nächstes wählen wir Vorgang 4 und erhalten für die Menge der Entscheidungszeitpunkte $\mathcal{D}_4(S^C) = \{2, 4\}$. Die Erweiterungskosten zu den Zeitpunkten 2 und 4 sind $f^b(S^C, 4, 2) = 46 - 22 = 24$ und $f^b(S^C, 4, 4) = 38 - 22 = 16$. Wir planen Vorgang 4 zu $S_4 = 4$ ein. Da wir Vorgang 4 zu seinem spätesten Startzeitpunkt eingeplant haben, aktualisieren wir die frühesten Startzeitpunkte der Vorgänge 1, 5 und 7 auf $ES_1 := 1$, $ES_5 := 7$ und $ES_7 := 9$. Damit gilt für die Vorgänge 5 und 7 $ES_5 = LS_5$ und $ES_7 = LS_7$; sie sind somit fixiert. Wir setzen $S_5 := 7$ sowie $S_7 := 9$. Für Vorgang 1 gilt $TF_1 = 1 < p_1 = 2$,

daher ist Vorgang 1 nun teilfixiert. Abbildung 2.52c zeigt das entsprechende Ressourcenprofil.

Im dritten Hauptschritt wählen wir den Vorgang 1 als nächsten einzuplanenden Vorgang. Die Menge der Entscheidungszeitpunkte von Vorgang 1 ist $\mathcal{D}_1(S^C) = \{1, 2\}$. Für die Erweiterungskosten zu den Zeitpunkten 1 und 2 erhalten wir $f^b(S^C, 1, 1) = 64 - 52 = 12$ und $f^b(S^C, 1, 2) = 12$. Wir wählen die größte Minimalstelle und planen Vorgang 1 zu $S_1 := 2$ ein. Eine Aktualisierung der *ES*- und *LS*-Werte ist nicht erforderlich. Das Ressourcenprofil ist in Abbildung 2.52d dargestellt.

Im vierten Schritt wählen wir den Vorgang 6 als nächsten einzuplanenden Vorgang. Wir erhalten als Menge von Entscheidungszeitpunkten $\mathcal{D}_6(S^C) = \{3, 4, 5\}$. Es ergeben sich die Erweiterungskosten $f^b(S^C, 6, 3) = 78 - 64 = 14$, $f^b(S^C, 6, 4) = 72 - 64 = 8$ und $f^b(S^C, 6, 5) = 70 - 64 = 6$. Wir planen Vorgang 6 zu $S_6 = 5$ ein. Das Ressourcenprofil ist in Abbildung 2.52e dargestellt.

Im letzten Schritt wählen wir den Vorgang 3. Für die Menge von Entscheidungszeitpunkten gilt $\mathcal{D}_3(S^C) = \{1, 3\}$, und es ergeben sich die Erweiterungskosten $f^b(S^C, 3, 1) = 82 - 70 = 12$ und $f^b(S^C, 3, 3) = 90 - 70 = 20$. Somit planen wir Vorgang 3 zu $S_3 = 1$ ein. Das entstandene Ressourcenprofil mit Zielfunktionswert 82 ist in Abbildung 2.52f veranschaulicht.

2.4 Kundenauftragsfertigung

In diesem Abschnitt stellen wir die Terminierung von Fertigungsaufträgen bei der *Kundenauftragsfertigung* als Anwendungsbeispiel für die Projektplanung unter Zeitrestriktionen vor. Typischerweise ist die Kundenauftragsfertigung in der Einzel- und Kleinserienfertigung anzutreffen. Ein Kundenauftrag, der u.U. mehrere Endprodukte umfasst, bzw. der zur Erledigung des Kundenauftrags durchzuführende Produktionsprozess lässt sich als Projekt auffassen. Alle benötigten Vor-, Zwischen- und Endprodukte werden aufgrund von vorliegenden Kundenaufträgen gefertigt. Für jeden Kundenauftrag liegen die Mengen der gewünschten Endprodukte (Primärbedarfe) und zugehörige Liefertermine vor. Für jeden Produktionsprozess bzw. für jedes Zwischen- und Endprodukt ist ein so genannter *Arbeitsplan* gegeben, der die einzelnen *Arbeitsgänge* (Vorgänge) zur Herstellung des betrachteten Produktes sowie die Reihenfolge enthält, in der die Arbeitsgänge durchzuführen sind. Dabei werden ein oder mehrere aufeinander folgende Arbeitsgänge auf einer bestimmten Maschine ausgeführt. Im Rahmen der Kundenauftragsfertigung sind die Startzeitpunkte der einzelnen Arbeitsgänge so zu bestimmen, dass die einzelnen Produkte zu den gewünschten Lieferterminen fertiggestellt werden können. In der Praxis ist man häufig bestrebt, einen Kundenauftrag so schnell wie möglich zu bearbeiten, damit frühzeitig freie Produktionskapazitäten für zukünftige Kundenaufträge zur Verfügung stehen.

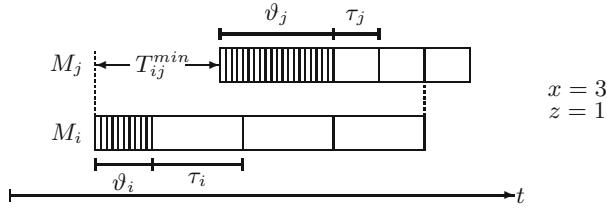
Wir beschreiben im Folgenden die Fertigung eines Kundenauftrags im Rahmen einer offenen Fertigung als Projektplanungsproblem. Dabei bezeichnen

wir die Fertigung des periodenbezogenen Gesamtbedarfs (Bruttobedarf) für ein Produkt als *Job*. Die Bearbeitung eines Jobs auf einer Maschine wird als *Operation* bezeichnet. In der Praxis wird meist gefordert, dass eine Operation in ihrer Durchführung nicht unterbrochen werden darf, um eine gleichmäßige Auslastung der Betriebsmittel zu erreichen. Im Folgenden werde jede (nicht unterbrechbare) Operation, die einem oder mehreren Arbeitsgängen entspricht, durch einen Projektvorgang repräsentiert. In der *offenen Fertigung* können aufeinanderfolgende Operationen zeitlich überlappen. Hierbei wird ein Produkt bereits zu einer gemäß des zugrunde liegenden Arbeitsplans nächsten Maschine weitergeleitet, während die Bearbeitung auf der ersten Maschine noch läuft. Die Weitergabe erfolgt dann in Transportlosen, die kleiner als das jeweilige Fertigungslos sind. Die offene Fertigung stellt eine wichtige Maßnahme zur Verkürzung von Durchlaufzeiten in der Produktion dar.

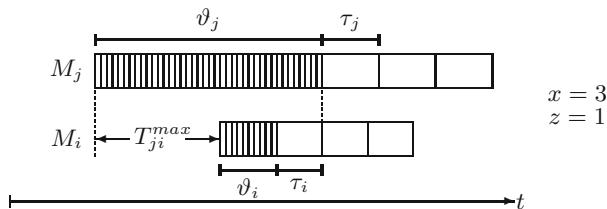
Da die Gesamtdauer der Bearbeitung aller Kundenaufträge minimiert werden soll, wählen wir die Minimierung der Projektdauer als Zielfunktion. Als Restriktionen sind die Befriedigung des Primärbedarfs an allen Endprodukten, die Einhaltung einer durch den Arbeitsplan vorgegebenen Bearbeitungsreihenfolge der Arbeitsgänge bzw. Operationen zur Herstellung eines Produktes und die Einhaltung der Lieferzeiten für die vorliegenden Kundenbestellungen zu berücksichtigen. Die Einhaltung einer vorgeschriebenen Lieferzeit für ein Produkt kann durch Einführung eines Höchstabstandes zwischen dem Projektstart und der letzten Operation des Projektes gewährleistet werden. Zur Gewährleistung der vorgegebenen Bearbeitungsreihenfolge der Operationen eines Jobs fügen wir zeitliche Mindestabstände ein, die sich wie folgt ergeben. Bezeichne O_i eine Operation, die der Bearbeitung eines Jobs auf der Maschine M_i entspricht. Weiterhin sei ϑ_i die Rüstzeit für Operation O_i auf Maschine M_i und τ_i die Bearbeitungszeit pro Einheit des betrachteten Produktes auf M_i . Dann ist $p_i = \vartheta_i + \tau_i x$ die Ausführungszeit bzw. die Dauer der Operation O_i , wobei x den Bruttobedarf des jeweiligen Produktes darstellt. Betrachten wir den Fall, dass ein Produkt zunächst auf Maschine M_i bearbeitet und danach direkt an Maschine M_j zur Weiterbearbeitung übergeben werden soll. Ist die Transportlosgröße z des Produktes kleiner als der Bruttobedarf x , dann können einige Einheiten von M_i nach M_j überführt werden, bevor die Bearbeitung des gesamten Bruttobedarfs an M_i abgeschlossen ist. Betrachten wir hierzu das folgende Beispiel.

Beispiel 2.42. Gegeben sei ein (Zwischen-)Produkt mit einem Bruttobedarf von $x = 3$ und einer Transportlosgröße von $z = 1$. Abbildung 2.53 zeigt den Fall, dass die Bearbeitungszeit pro Produkteinheit auf Maschine M_i größer ist als auf Maschine M_j , d.h. $\tau_i > \tau_j$. Dann müssen wir einen Mindestabstand T_{ij}^{min} zwischen dem Start der Operation O_i und der Operation O_j einführen, um zu gewährleisten, dass O_j nicht unterbrochen wird.

Des Weiteren betrachten wir den Fall, dass die Bearbeitungszeit des (Zwischen-)produktes auf M_i kleiner oder gleich der Bearbeitungszeit auf M_j ist, d.h. es gilt $\tau_i \leq \tau_j$. Abbildung 2.54 veranschaulicht diese Situation, wo-

Abb. 2.53. Bearbeitung auf M_i und M_j mit Mindestabstand

bei o.B.d.A. angenommen wird, dass die Rüstzeit auf Maschine M_i sehr viel kleiner ist als auf Maschine M_j , d.h. es gilt $\vartheta_i \ll \vartheta_j$. In diesem Fall müssen wir einen Höchstabstand T_{ji}^{max} zwischen dem Start der Operationen O_j und O_i einfügen, um eine möglichst zügige, unterbrechungsfreie Bearbeitung der Operationen O_i und O_j zu erreichen.

Abb. 2.54. Bearbeitung auf M_i und M_j mit Höchstabstand

Sei x der Bruttobedarf und z die Transportlosgröße eines Produktes, wobei wir annehmen, dass x/z ganzzahlig ist. Dann garantiert die Einführung eines Pfeils $\langle i, j \rangle$ mit der Bewertung

$$\delta_{ij} := \begin{cases} \vartheta_i + x\tau_i - \vartheta_j - (x-z)\tau_j, & \text{für } \tau_i > \tau_j \\ \vartheta_i + z\tau_i - \vartheta_j, & \text{für } \tau_i \leq \tau_j \end{cases} \quad (2.11)$$

zwischen dem Start von zwei aufeinanderfolgenden Operationen O_i und O_j , dass O_j ohne Unterbrechung ausgeführt werden kann. Für $\delta_{ij} \geq 0$ entspricht der Pfeil $\langle i, j \rangle$ einem Mindestabstand $T_{ij}^{min} := \delta_{ij}$ zwischen dem Start von O_i und O_j . Ist $\delta_{ij} < 0$, dann entspricht $\langle i, j \rangle$ einem Höchstabstand $T_{ji}^{max} := -\delta_{ij}$ zwischen dem Start von O_j und O_i . Existiert eine Transportzeit t_{ij} für den Transfer des Transportloses von Maschine M_i nach Maschine M_j , so addieren wir t_{ij} zu δ_{ij} hinzu. Anstatt wie eben beschrieben zwei Operationen desselben Jobs zu überlappen, können auch die letzte Operation O_i eines Produktes h und die erste Operation O_j eines anderen Produktes l überlappen, falls das (Zwischen-)Produkt h in das Produkt l eingeht.

Bei der eben beschriebenen Vorgehensweise zur Erzeugung zeitlicher Mindest- und Höchstabstände zwischen zwei Operationen sind wir davon ausge-

gangen, dass genau eine Einheit des aus Operation O_i resultierenden Zwischenproduktes für die Herstellung einer Einheit des aus Operation O_j resultierenden (Zwischen-)Produktes benötigt wird. Tatsächlich herrschen in Unternehmen wesentlich komplexere Produktstrukturen vor. Die *Produktstruktur* eines Unternehmens gibt die strukturellen und mengenmäßigen Zusammenhänge zwischen den Produkten wieder und lässt sich durch einen so genannten Gozintographen abbilden. Ein Gozintograph ist ein Digraph, dessen Knoten die Produkte darstellen, wobei zwischen End- und Zwischenprodukten sowie Rohmaterialien unterschieden wird. Endprodukte werden durch Senken, und Rohmaterialien, die extern bezogen werden, durch Quellen im Gozintographen repräsentiert. Existiert ein Pfeil $\langle h, l \rangle$ von Knoten h zu Knoten l mit dem Gewicht a_{hl} , dann werden a_{hl} Einheiten von Produkt h benötigt, um eine Einheit von Produkt l herzustellen. Man bezeichnet a_{hl} daher als *Inputkoeffizient*. Wir unterscheiden an dieser Stelle der Einfachheit halber zwischen zwei grundlegenden Produktstrukturen. Bei einer *linearen* Produktstruktur besitzt jedes Produkt genau einen direkten Vorgänger und genau einen direkten Nachfolger im Gozintographen (vgl.

Abb. 2.55a). Bei einer *konvergenten* Produktstruktur besitzt jedes Produkt genau einen Nachfolger im Gozintographen, d.h. der Gozintograph stellt einen Intree dar (vgl.

Abb. 2.55b). Der Fall, dass eine *divergente Produktstruktur* vorliegt, d.h. der Gozintograph ist ein Outtree, wird in NEUMANN und SCHWINDT (1997) behandelt.

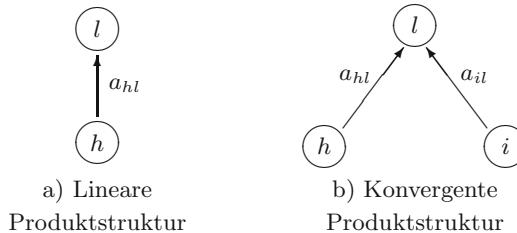


Abb. 2.55. Produktstrukturen

Existiert bei einer linearen oder konvergenten Produktstruktur ein Pfeil $\langle h, l \rangle$ von Knoten h zu Knoten l im Gozintographen, dann können die letzte Operation O_i des (Zwischen-) Produktes h und die erste Operation O_j des Produktes l überlappen. Operation O_i wird auf der Maschine M_i und Operation O_j auf der Maschine M_j ausgeführt. Falls a_{hl} Einheiten von (Zwischen-) Produkt h für eine Einheit des Produktes l benötigt werden, dann ergibt sich der Zeitabstand δ_{ij}^{hl} gemäß

$$\delta_{ij}^{hl} := \begin{cases} \vartheta_i + a_{hl}x_l\tau_i - \vartheta_j - (x_l - z_h/a_{hl})\tau_j, & \text{für } a_{hl}\tau_i > \tau_j \\ \vartheta_i + z_h\tau_i - \vartheta_j, & \text{für } a_{hl}\tau_i \leq \tau_j. \end{cases} \quad (2.12)$$

In Gleichung (2.12) repräsentiert x_l den Bruttobedarf von Produkt l und z_h die Transportlosgröße von Produkt h . Ist $a_{hl} = 1$, so entspricht (2.12) gerade (2.11).

Zum Abschluss beschreiben wir die Konstruktion des Projektnetzplans für das beschriebene Problem der Kundenauftragsfertigung. Wir nehmen an, dass die Produktstruktur durch einen Gozintographen gegeben ist. Jeder Knoten des Gozintographen wird durch eine Sequenz von Operationen ersetzt, die den jeweiligen Job abbilden. Auf diese Weise entspricht ein Job dem Weg von seinem ersten Operations-Knoten zu seinem letzten Operations-Knoten im resultierenden Netzwerk. Die Bewertungen der Pfeile entsprechen den oben eingeführten Mindest- und Höchstabständen.

Beispiel 2.43. Betrachten wir den in Abbildung 2.56 dargestellten Gozintographen einer Produktstruktur. Wir nehmen an, dass Zwischenprodukt A aus der Sequenz der Operationen A_1, A_2 und Zwischenprodukt C aus der Sequenz C_1, C_2, C_3 besteht, während Zwischenprodukt B nur auf einer Maschine gefertigt wird. Endprodukt I bestehe aus der Sequenz I_1, I_2 und Endprodukt II werde nur auf einer Maschine gefertigt. Die Arbeitspläne für die einzelnen Produkte sind ebenfalls in Abbildung 2.56 angegeben, wobei die Dauer der Operationen jeweils für das gesamte Fertigungslos angegeben ist, d.h. wir führen in unserem Beispiel der Einfachheit halber zeitliche Mindestabstände der Form $T_{ij}^{min} = p_i$ zwischen den Startzeitpunkten zweier aufeinanderfolgender Operationen O_i und O_j eines Jobs ein.

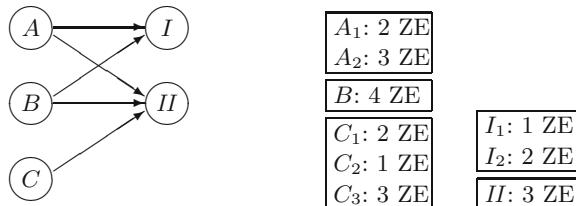


Abb. 2.56. Gozintograph und Arbeitspläne

Wir erhalten ein Netzwerk N' , das in Abbildung 2.57 innerhalb des gestrichelten Rechtecks dargestellt ist. Die Mindestabstände T_{ij}^{min} zwischen den Vorgängen i und j entsprechen der Dauer der zugehörigen Operation O_i . Nun ist eine neue Quelle α und eine neue Senke ω einzuführen, wobei α den Start und ω das Ende des Projektes repräsentieren. Ausgehend von der Quelle α fügen wir Pfeile zu den Quellen des Netzwerks N' mit der Bewertung 0 ein. Weiterhin sind Pfeile hinzuzufügen, die von allen Senken s des Netzwerks N' zu der Senke ω verlaufen und deren Bewertung gleich der Dauer der zu s gehörigen Operation ist. Die zusätzlichen Pfeile sind in Abbildung 2.57 fett ausgezeichnet.

Wir nehmen außerdem an, dass für einige Endprodukte h eine Lieferzeit \bar{d}_h festgelegt ist. Sei O_j die letzte Operation von Produkt h mit der Dauer p_j .

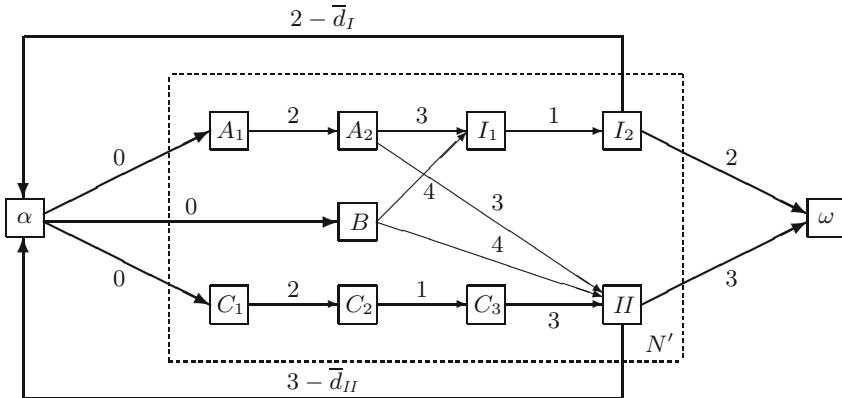


Abb. 2.57. Projektnetzplan N von Beispiel 2.43

Dann ist ein Höchstabstand $T_{\alpha,j}^{max} := \bar{d}_h - p_j$ zwischen dem Projektstart und dem Start von Operation O_j gegeben. Seien in Beispiel 2.43 die Lieferzeiten \bar{d}_I und \bar{d}_{II} für die Endprodukte I und II gegeben, dann sind zwei Rückwärtspfeile einzuführen, zum einen $\langle I_2, \alpha \rangle$ und zum anderen $\langle II, \alpha \rangle$ (vgl. Abb. 2.57).

Die Länge eines längsten Weges von α nach ω in N entspricht nun gerade der kürzestmöglichen Bearbeitungszeit des betrachteten Kundenauftrages, und die zugehörigen Startzeitpunkte der einzelnen Operationen entsprechen den frühesten Startzeitpunkten der entsprechenden Vorgänge in N .

Ergänzende Literatur

- | | |
|-------------------------------------|-------------------------------|
| BRUCKER ET AL. (1999) | MÖHRING (1984) |
| DEMEULEMEESTER (1995) | NEUMANN und SCHWINDT (1997) |
| DEMEULEMEESTER und HERROELEN (2002) | NEUMANN und ZIMMERMANN (1999) |
| DE REYCK (1998) | NEUMANN und ZIMMERMANN (2000) |
| HERROELEN ET AL. (1997) | NEUMANN ET AL. (2003) |
| KIMMS (2001) | NÜBEL (1999) |
| KOLISCH (2001) | ÖZDAMAR und ULUSOY (1995) |
| KOLISCH und PADMAN (2001) | VANHOUCKE ET AL. (2001) |
| MAYER (1998) | WEGLARZ (1999) |
| MODER ET AL. (1983) | ZIMMERMANN (2001) |