

Projektplanung mit partiell erneuerbaren Ressourcen

Jan Elseberg

6. November, 2007

Grenzen des RCPSP

Arbeitsplan für Gaststätte

- 7 Arbeitstage, je 2 Schichten.
- 3 Arbeiterklassen
 - Vollzeitangestellte, mit 4 Arbeitstagen nur Mo. - Fr.
 - Teilzeitangestellte, mit 2.5 Arbeitstagen
 - Manager

Probleme

- Gewerkschaft: Teilzeiter arbeiten nicht Sa. und So. zugleich
- Geschäftsleitung: Mindestens ein Manager muss eine Wochenendschicht übernehmen

RCPSP-Definition

Aktivitäten:

- J Aktivitäten, $J = \{1, \dots, J\}$
- Dummy-Aktivitäten 1, J
- Aktivitäts-Dauer d_j
- Ressourcen-Bedarf k_{ip}^ρ, k_{ip}^ν

Ressourcen:

- K^ρ erneuerbare Ressourcen, $K^\rho = \{1, \dots, K^\rho\}$
- Verfügbarkeit K_{pt}^ρ
- K^ν nicht erneuerbare Ressourcen, $K^\nu = \{1, \dots, K^\nu\}$
- Verfügbarkeit K_p^ν

Zeit-Horizont:

- T Zeitperioden, $T = \{1, \dots, T\}$

Präzedenzrelationen:

- partielle Ordnung \prec auf J

RCPSP / π - Definition

Aktivitäten:

- J Aktivitäten, $J = \{1, \dots, J\}$
- Dummy-Aktivitäten 1, J
- Aktivitäts-Dauer d_j
- Ressourcen-Bedarf k_{ip}

Ressourcen:

- P partiell erneuerbare Ressourcen, $P = \{1, \dots, P\}$
- Verfügbarkeit K_{ps}

Zeit-Horizont:

- T Zeitperioden, $T = \{1, \dots, T\}$
- S Teilmengen, $\Pi_s \subseteq T$

Präzedenzrelationen:

- partielle Ordnung \prec auf J

alternative RCPSP / π - Definition

Aktivitäten:

- J Aktivitäten, $J = \{1, \dots, J\}$
- Dummy-Aktivitäten 1, J
- Aktivitäts-Dauer d_j
- Ressourcen-Bedarf k_{ip}

Ressourcen:

- P partiell erneuerbare Ressourcen, $P = \{1, \dots, P\}$
- Verfügbarkeit K_p

Zeit-Horizont:

- T Zeitperioden, $T = \{1, \dots, T\}$
- S Teilmengen, $\Pi_p \subseteq T$

Präzedenzrelationen:

- partielle Ordnung \prec auf J

Inklusion der nichterneuerbaren Ressourcen

Ersetzung

- nichtern. Ressource n, mit Kapazität K_n'
- part. ern. Ressource p, mit $\Pi_p = \{1, \dots, T\}$ und $K_p = K_n'$

Arbeitsplan für Gaststätte

- Arbeitszeit der Teilzeitangestellten auf 2.5 Tage beschränkt
- part. ern. Ressource p mit $\Pi_p = \{1, \dots, 14\}$ und $K_p = 5$

Inklusion der erneuerbaren Ressourcen

Ersetzung

- ern. Ressource r, mit Kapazität K_{rt}^p
- T part. ern. Ressourcen p_t , mit Teilmengen $\Pi_{p_t} = \{t\}$ und $K_{p_t} = K_{rt}^p$

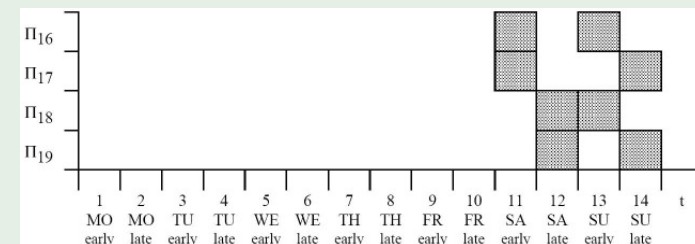
Arbeitsplan für Gaststätte

- Arbeitszeit der Vollzeitangestellten auf Mo. bis Fr. beschränkt
- part. ern. Ressource p mit $\Pi_p = \{11, 12, 13, 14\}$ und $K_p = 0$

Weiterführende Modellierungen

Arbeitsplan für Gaststätte

- Teilzeitangestellte dürfen nicht Sa. und So. zugleich arbeiten
- 4 part. ern. Ressource p mit $K_p = 1$
- Teilmenge Π_p für alle Kombinationen der Wochenendschichten



logische Relationen

Arbeitsplan für Gaststätte

- Mindestens ein Manager muss eine Wochenendschicht übernehmen.
- Oder auch:
 Manager_1 OR Manager_2 OR ... OR Manager_n
 müssen innerhalb einer Zeitperiode arbeiten.

logische Relationen

Identität

Aktivität j mit $d_j \leq n$ soll innerhalb $T' = \{t_i, \dots, t_n\}$ bearbeitet werden:

$$\Pi_p = T \setminus T', \quad K_p = 0, \quad k_{jp} = 1$$

$k_{jp} > K_p$ ausreichend,

oder

$$\Pi_p = T', \quad K_p = -d_j, \quad k_{jp} = -1$$

$k_{jp}d_j > K_p$ ausreichend.

logische Relationen

Negation - NOT

Aktivität j soll innerhalb T' nicht bearbeitet werden:

$$\Pi_p = T', \quad K_p = 0, \quad k_{jp} = 1$$

$k_{jp} > K_p$ ausreichend.

logische Relationen

Konjunktion - AND

Mehrere Aktivitäten J' mit jeweils $d_j \leq n$ sollen innerhalb von $T' = \{t_i, \dots, t_n\}$ bearbeitet werden:

$$\Pi_p = T \setminus T', \quad K_p = 0, \quad k_{jp} = 1 \quad k_{jp} > K_p \text{ ausreichend,}$$

oder

$$\Pi_p = T', \quad K_p = -\sum_{j \in J'} d_j, \quad k_{jp} = -1$$

$k_{jp} \sum_{j \in J'} d_j = K_p < 0$ ausreichend.

logische Relationen

Disjunktion - OR

Eine von mehreren Aktivitäten J' soll innerhalb von T' bearbeitet werden:

$$\Pi_p = T', \quad K_p = -1, \quad k_{jp} = -1, \quad k_{jp} = K_p < 0 \text{ ausr.}$$

Arbeitsplan für Gaststätte

- Mindestens ein Manager muss eine Wochenendschicht übernehmen.
- $k_{jp} = -1$ für alle Jobs, $K_p = -1$
für Teilmenge $\Pi_p = \{11, 12, 13, 14\}$

logische Relationen

Negierte Konjunktion - NAND

Nicht alle Aktivitäten J' mit jeweils $d_j \leq n$ dürfen innerhalb von $T' = \{t_i, \dots, t_n\}$ bearbeitet werden:

$$\Pi_p = T', \quad K_p = \sum_{j \in J'} d_j - 1, \quad k_{jp} = 1$$

logische Relationen

Negierte Disjunktion - NOR

Keine der Aktivitäten J' dürfen innerhalb von T' bearbeitet werden:

$$\Pi_p = T', \quad K_p = 0, \quad k_{jp} = 1$$

$$k_{jp} > K_p \text{ ausreichend.}$$

logische Relationen

Exklusive Disjunktion - XOR

Genau eine der Aktivitäten J' muss innerhalb von $T' = \{t_i, \dots, t_n\}$ bearbeitet werden:

- Ressource p

$$\Pi_p = T', \quad K_p = -1, \quad k_{jp} = -1$$

$$k_{jp} = K_p < 0 \text{ ausreichend.}$$

- Ressource p'

$$\Pi_{p'} = T', \quad K_{p'} = 1, \quad k_{jp'} = 1$$

$$k_{jp'} = K_{p'} \text{ ausreichend.}$$

logische Relationen

Exklusive Disjunktion - XOR

Genau m Aktivitäten aus J' müssen innerhalb von $T' = \{t_i, \dots, t_n\}$ bearbeitet werden:

- Ressource p

$$\Pi_p = T', \quad K_p = -m, \quad k_{jp} = -1$$

$k_{jp} = K_p/m < 0$ ausreichend.

- Ressource p'

$$\Pi_{p'} = T', \quad K_{p'} = m, \quad k_{jp'} = 1$$

$k_{jp'} = K_{p'}/m$ ausreichend.

Grenzen der logischen Modellierung

- Für alle logischen Ausdrücke benötigt:
 - Negation
 - Konjunktion oder Disjunktion
- Komposition der Relationen ist Konjunktion
- Negation aber nicht möglich
- Nicht alle logischen Ausdrücke lassen sich somit darstellen

Übersicht

Optimale Algorithmen:

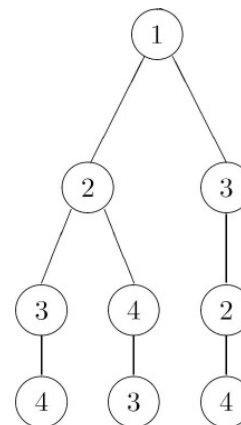
- Branch & Bound Verfahren
- lineare (Integer) Programmierung

heuristische Algorithmen:

- Regel-basierte Planung
- heuristische Branch & Bound Verfahren
- lokale Suchverfahren

Viele Verfahren für RCPSP sind schwer, bzw. gar nicht übertragbar.

Präzedenz Baum



Erweiterung

- Knoten beinhalten zusätzlich die Startzeit t_i des Jobs j_i
- Bei Backtracking wird t_i zunächst inkrementiert
- Backtracking falls:
 - Teilplan nicht erweiterbar
 - Teilplan länger als obere Makespan Schranke

- Local-Left-Shift Rule nicht anwendbar
- neue Bounding Rules

Feasability Bound 1

Ziel

- Präzedenz-Baum weiter zurückschneiden

Idee

- Ressourcenbedarf der verbliebenen Aktivitäten abschätzen

Feasability Bound 1

Ressourcenbedarf der verbliebenen Aktivitäten abschätzen

- minimaler Verbrauch der Ressource p von Job i :

$$MinCons_{ipt} = \min\{Cons_{ipt} | t \leq \tau \leq LST_i\}$$

- verbliebene Kapazität einer Ressource p im Teilplan:

$$K_p^0 = K_p - \sum_{i=1}^n Cons_{j_i,pt_i}$$

Feasability Bound 1

Ressourcenbedarf der verbliebenen Aktivitäten abschätzen

- Minimaler Verbrauch aller Nachfolger Aktivitäten muss unter Kapazität liegen:

$$K_p^0 \geq \sum_{i \in Succ_j} MinCons_{ipEST_i}$$

Feasability Bound 2

Ziel

- Präzedenz-Baum noch weiter zurückschneiden

Idee

- Lerne aus fehlerhaften Teilplänen
- Für jede nicht einplanbare Aktivität j speichere:

$$\varsigma_j = (j, \tau_j, MinCons_{j1\tau_j}, \dots, MinCons_{jP\tau_j})$$

- Zukünftige Pläne müssen alle ς_j aufheben
 - Wenn zum Beispiel j vor τ_j eingeplant werden kann

Greedy Randomized Adaptive Search Procedures (GRASP)

- Heuristisches Verfahren
- Anpassung des Serial Schedule Scheme SSS
- In J Schritten wird jeweils eine Aktivität eingeplant
- bereits eingeplante Aktivitäten in C
- verfügbare Aktivitäten + **ihre möglichen Startzeiten** $(j, t) \in D$
 - $j \notin C$, j noch nicht eingeplant
 - alle Vorgänger von j sind bereits eingeplant
 - $t \in \{EST_j, \dots, LST_j\}$, alle einplanbaren Zeitpunkte
 - genügend Kapazität für die Startzeit t

Greedy Randomized Adaptive Search Procedures (GRASP)

- In jedem Schritt des SSS muss aus D gewählt werden
- Prioritätsregel ω :

$$\omega = (j, t) \in D \rightarrow \mathbb{R}_{\geq 0}$$

- Reue-Wert ρ_{jt} :

$$\rho_{jt} = \omega_{jt} - \min\{\omega_{i,\tau} | (i, \tau) \in D\} + 1$$

- Auswahlwahrscheinlichkeit $P(j, t)$:

$$P(j, t) = \frac{\rho_{jt}^\alpha}{\sum_{(i,\tau) \in D} \rho_{i\tau}^\alpha}$$

- Bias-Wert α

statische Prioritätsregeln für GRASP

MINEFT	Minimale früheste Endzeit
MINLFT	Minimale späteste Endzeit
MINSLK	Minimaler Slack $s_j = LST_j - EST_j$
MTSUCC	Maximale Anzahl Nachfolger
MAXSRU	Maximale statischer Ressourcen-Bedarf
MINSRU	Minimaler statischer Ressourcen-Bedarf

$$SRU_j = \sum_{p \in P} k_{jp}$$

statische Prioritätsregeln für GRASP

MAXTRU	Maximaler zeitabh. Ressourcen-Bedarf
MINTRU	Maximaler zeitabh. Ressourcen-Bedarf

$$TRU_{jt} = \sum_{p \in P} (Cons_{jpt} + MinCons_{jpt})$$

MAXRRU	Maximaler rel. Ressourcen-Bedarf
MINRRU	Minimaler rel. Ressourcen-Bedarf

$$RRU_j = \sum_{p \in P} (Cons_{jpt} + MinCons_{jpt}) / K_p$$

dynamische Prioritätsregeln für GRASP

MAXDRRU Maximaler dyn. rel. Ressourcen-Bedarf

MINDRRU Minimaler dyn. rel. Ressourcen-Bedarf

$$DRRU_{jt} = \sum_{p \in P} (Cons_{jpt} + MinCons_{jpt}) / K_p^0$$

MAXTRC Maximale totale verbliebene Kapazität

MINTRC Minimale totale verbliebene Kapazität

$$TRC_j = \sum_{p \in P} (K_p^0 - Cons_{jpt} - MinCons_{jpt})$$

Experimenteller Aufbau

- $J = 10$
- $P = 30$
- $d_j \in [1, 10]$
- 10 Instanzen für alle Parameter Kombinationen \Rightarrow 2160 Instanzen

Automatisierte Instanzgenerierung

- Progen auf RCPSP zugeschnitten
 - Netzwerk Komplexität NC , bestimmt Präzedenz-Relationen
 - Ressourcen Faktor RF , bestimmt Ressourcen-Bedarf
 - Fälligkeits Faktor DF , bestimmt Zeithorizont
- part. ern. Ressourcen erfordern zusätzliche Werte
 - Kardinalitätsfaktor $CF \in [0, 1]$, bestimmt Kardinalität der Teilmengen Π_s

$$\text{ROUND}(1 - CF + T \cdot CF)$$

- Partitions Faktor $PF \in [0, 1]$, bestimmt Anzahl an Intervallen in Π_s

$$\text{ROUND}(1 - PF + \min\{|\Pi_s|, T - |\Pi_s|\} \cdot CF)$$

- part. ern. Ressourcen erfordern modifizierten Wert RS
 - Ressourcen Stärke $RS \in [0, 1]$, bestimmt Kapazität der Ressourcen

Parameter Test

PAR	VAL	# of Leaves		CPU-Time in Sec.	
		AVE	STD	AVE	STD
NC	1.5	22,265	153,651	3.462	20.332
	2.0	33,661	218,166	5.067	31.974
RF	0.1	3,858	26,125	0.818	4.268
	0.5	16,879	147,076	2.520	17.328
	0.9	63,152	287,569	9.455	42.380
RS	0.25	3,063	24,959	0.611	3.157
	0.5	33,818	167,662	5.448	27.428
	0.75	47,008	277,853	6.735	37.059
DF	0.3	2,724	12,918	0.629	2.110
	0.6	53,203	264,250	7.900	37.498
CF	0.2	2,461	20,612	0.621	3.090
	0.5	22,879	155,176	3.894	25.756
	0.8	58,549	284,317	8.279	38.131
PF	0.0	27,594	182,555	4.115	25.346
	0.5	28,332	194,791	4.413	28.187

Feasibility Bound Test

Table 5 Averages Over All Instances

	# of Leaves			CPU-Time in Sec.		
	AVE	STD	FAC	AVE	STD	FAC
BV	27,963	188,729	1.0	4.3	26.8	1.0
FB 1	4,441	50,873	6.3	1.0	8.1	4.3
FB 2	3,470	35,144	8.1	1.9	16.1	2.3
FB 1 & 2	580	6,696	48.1	0.7	5.5	6.0

Table 6 Averages Over 40 Instances with Highest Speed-up

	# of Leaves			CPU-Time in Sec.		
	AVE	STD	FAC	AVE	STD	FAC
BV	157,489	235,251	1.0	22.3	35.8	1.0
FB 1	5,840	18,293	27.0	1.2	3.2	18.3
FB 2	4,743	10,372	33.2	2.5	4.9	8.9
FB 1 & 2	124	336	1267.0	0.5	0.3	47.1

Table 7 Averages Over 40 Very Hard Instances

	# of Leaves			CPU-Time in Sec.		
	AVE	STD	FAC	AVE	STD	FAC
BV	556,981	863,217	1.0	77.9	119.8	1.0
FB 1	123,825	314,838	4.5	20.1	49.6	3.9
FB 2	66,360	151,350	8.4	31.2	68.3	2.5
FB 1 & 2	15,243	39,745	36.5	12.9	33.3	6.1

Prioritätsregel Test

Priority Rule ω	217 Instances			14 Instances			250 Instances
	Z = 10	100	1,000	Z = 10	100	1,000	CPU (Z = 1,000)
MINEFT	12.23	9.17	6.38	134.74	133.10	100.40	10.6
MINLFT	1.63	0.81	0.81	7.87	7.87	6.50	8.8
MINSLK	8.70	7.53	6.50	75.13	74.83	40.25	11.3
MTSUC	4.71	3.31	2.77	55.52	40.06	39.86	11.0
MAXTRC	49.48	9.41	5.26	4.93	3.84	1.63	27.2
MINTRU	49.48	10.01	5.60	4.71	3.95	1.63	19.6
MINRRU	50.15	10.38	5.93	5.04	3.73	1.73	22.3
MINDRRU	50.15	10.38	6.04	5.37	3.73	1.52	32.5