Spaltengenerierungsbasierte Schranken für Ressourcennivellierungsprobleme

15. Doktorandenworkshop
an der Otto-von-Guericke-Universität Magdeburg

Tobias Paetz
Christoph Schwindt

Abteilung für Betriebswirtschaftslehre,
insbesondere Produktion und Logistik

TU Clausthal

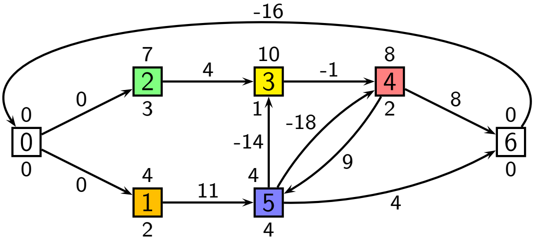
Agenda

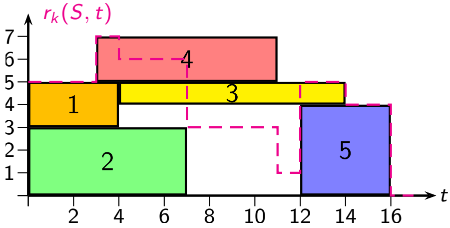
Ressourcennivellierungsprobleme

- Projekt besteht aus n Aktivitäten $i \in V$ der Dauern p_i
- Jede Aktivität i benötigt während Ausführung r_{ik} Einheiten erneuerbarer Ressourcen $k \in \mathcal{R}$
- Zwischen Startzeitpunkten von Aktivitäten zeitliche Mindest- und Höchstabstände gegeben
- Projekt muss $\overline{d}$ Zeiteinheiten nach Projektstart beendet sein
  Vereinbarung: Deadline $\overline{d}$ als zeitlicher Höchstabstand zwischen Projektstart 0 und Projektende $n+1$ abgebildet: $\delta_{(n+1)0} = -\overline{d}$
- Gesucht: Schedule $S = (S_i)_{i \in V}$ mit minimaler Summe der gewichteten Varianzen der Ressourcenprofile $r_k(S,t)$

$$(P)\begin{cases} \text{Min.} & \sum_{k \in \mathcal{R}} c_k \int_0^{\overline{d}} r_k^2(S,t)\,dt \\ \text{u. d. N.} & S_j \geq S_i + \delta_{ij} \quad ((i,j) \in E) \\ & S_0 = 0 \end{cases}$$

Beispiel





Antikette A	x_A	$x_A \cdot r_{Ak}^2$
$\{1, 2\}$	3	$3 \cdot 5^2 = 75$
$\{1, 2, 4\}$	1	$1 \cdot 7^2 = 49$
$\{2, 3, 4\}$	3	$3 \cdot 6^2 = 108$
$\{3, 4\}$	4	$4 \cdot 3^2 = 36$
$\{3\}$	1	$1 \cdot 1^2 = 1$
$\{3, 5\}$	2	$2 \cdot 5^2 = 50$
$\{5\}$	2	$2 \cdot 4^2 = 32$
Summe	16	351

Reformulierung

- Assoziiere mit jeder nichtleeren Antikette $A \in \mathcal{A}$ der strikten Halbordnung $\Theta(D) = \{(i,j) \mid d_{ij} \geq p_i\}$ eine Dauervariable x_A
- Repräsentiere Schedule als Folge von Antiketten A mit $x_A > 0$ und Ressourceninanspruchnahmen $r_{Ak} = \sum_{i \in A} r_{ik}$

$$
(P') \left\{
\begin{array}{lll}
\text{Min.} & \sum_{k \in \mathcal{R}} c_k \sum_{A \in \mathcal{A}} r_{Ak}^2 \cdot x_A & \\
\text{u. d. N.} & \sum_{A \in \mathcal{A}: i \in A} x_A = p_i & (i \in V) \\
& \sum_{A \in \mathcal{A}} x_A \leq \overline{d} & \\
& x_A \geq 0 & (A \in \mathcal{A}) \\
& \text{Side constraints} &
\end{array}
\right.
$$

- Side constraints: Lösbarkeit des 1-Maschinen-Problems $1|temp|\bullet$ mit Jobmenge $J = \{A \in \mathcal{A} \mid x_A > 0\}$

Relaxation und Prinzip der Spaltengenerierung

- Relaxation der side constraints liefert lineares Programm mit exponentiell wachsender Anzahl an Entscheidungsvariablen

$$
(LP)\left\{\begin{array}{lllr}
\text{Min.} & \sum_{A\in\mathcal{A}} \underbrace{\left(\sum_{k\in\mathcal{R}} c_k r_{Ak}^2\right)}_{:=c_A} \cdot x_A & & \\
\text{u. d. N.} & \sum_{A\in\mathcal{A}: i\in A} x_A = p_i & (i\in V) & u_i \\
& \sum_{A\in\mathcal{A}} x_A + x_\emptyset = \overline{d} & & v \\
& x_A \geq 0,\ \ x_\emptyset \geq 0 & (A\in\mathcal{A}) &
\end{array}\right.
$$

- Löse (LP) durch Spaltengenerierung
  - Konstruiere zulässige Basislösung
  - Finde in jeder Iteration Nichtbasisvariable mit negativem reduzierten Zielfunktionskoeffizienten, und führe Pivotschritt durch
  - Breche ab, wenn alle reduzierten Zielfunktionskoeffizienten nichtnegativ

Reduzierte Zielfunktionskoeffizienten

- Dualisierung von (LP) liefert

$$(D)\begin{cases} \text{Max.} & \sum_{i\in V} p_i \cdot u_i + \overline{d}\cdot v \\ \text{u. d. N.} & \sum_{i\in A} u_i + v \le c_A \quad (A\in\mathcal{A}) \\ & v \le 0 \end{cases}$$

- Hieraus folgt für reduzierte Zielfunktionskoeffizienten

$$\zeta_A = c_A - \sum_{i\in A} u_i - v \quad (A\in\mathcal{A}), \qquad \zeta_\emptyset = -v$$

- Sei B Basismatrix zur aktuellen Basislösung x; dann gilt für Simplex-Multiplikatoren u, v

$$\begin{pmatrix} u \\ v \end{pmatrix} = (B^\top)^{-1}\begin{pmatrix} c_A^B \\ 0 \end{pmatrix}$$

- Hinreichende Optimalitätsbedingung: $\min_{A\in\mathcal{A}} \zeta_A \ge 0$ und $v \le 0$

Pricing-Problem

- Ermittle (Nichtbasis-)Index A^* mit kleinstem ζ_A
- Führe Indikatorvariable y_i ein mit

$$y_i = \begin{cases} 1, & \text{falls } i \in A^* \\ 0, & \text{sonst} \end{cases}$$

Pricing-Problem: MICP

$$(PP(u,v)) \begin{cases} \text{Min.} & \zeta_{A^*} = \sum_{k \in \mathcal{R}} c_k \left(\sum_{i \in V} r_{ik} y_i \right)^2 - \sum_{i \in V} u_i y_i - v \\ \text{u. d. N.} & y_i + y_j \leq 1 \quad ((i,j) \in \Theta(D)) \\ & y_i \in \{0,1\} \quad (i \in V) \end{cases}$$

- $(PP(u,v))$ ist konvexes Stabile-Mengen-Problem auf perfektem Graphen (Vergleichbarkeitsgraphen)

Algorithmus 1 Spaltengenerierung

Input: Lineares Programm (LP)
Output: Optimale Lösung x

bestimme frühesten Schedule ES unter Anordnungsbeziehungen $(i, j) \in \Theta(D)$ und zugehörige Basismatrix B;
repeat
  berechne Basislösung $x = B^{-1} \cdot (p, \overline{d})^\top$;
  berechne Simplex-Multiplikatoren $(u, v)^\top = (B^\top)^{-1} \cdot (c^B, 0)^\top$;
  löse Pricing-Problem $(PP(u, v))$;
  if $\min\{\zeta_{A^*}, -v\} < 0$ **then**
    berechne $\gamma = B^{-1} \cdot ((\mathbb{1}_{A^*}(i))_{i \in V}, 1)^\top$;
    bestimme $A' \in \text{argmin}\{\frac{x_A}{\gamma_A} \mid \gamma_A > 0\}$;
    tausche in B Spalte $(A^*, 1)^\top$ gegen Spalte $(A', 1)^\top$ aus;
  end if
until $\min\{\zeta_{A^*}, -v\} \geq 0$

Preprocessing I

1. Ersetze positive Ende-Start-Abstände $\delta_{ij} - p_i > 0$ durch Dummy-Aktivitäten der Dauer $\delta_{ij} - p_i > 0$
2. Identifiziere unvermeidbare Antiketten A, die sich in jedem zulässigen Schedule in Ausführung befinden müssen, d. h., $x_A > 0$

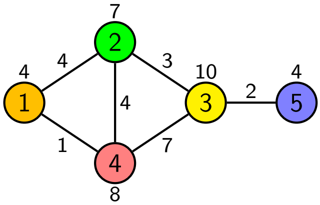
Proposition

Sei $\emptyset \neq A \subseteq V$. Dann befinden sich alle Aktivitäten $i \in A$ mindestens während der Dauer $p(A) = \max\{0, \min_{i,j \in A}(d_{ij} + p_j)\}$ gleichzeitig in Ausführung. Die Schranke ist scharf, d. h., es existiert stets ein zulässiger Schedule mit $x_A = p(A)$.

- Wegen $p(A) = \min_{i \in A} p(A \setminus \{i\})$ können die Antiketten A mit $p(A) > 0$ als Cliquen im Graphen $G = (V, E')$ mit Kantenmenge $E' = \{\{i, j\} \mid p(\{i, j\}) > 0\}$ berechnet werden

Preprocessing II

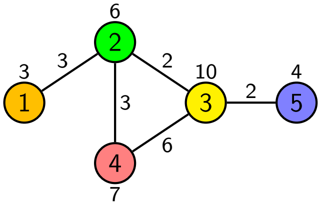
- Beginne mit einer $\subseteq$-maximalen unvermeidbaren Antikette A
- Füge Antikette A als Aktivität i_A mit $p_{i_A} = p(A)$ in Projekt ein
- Reduziere Dauern $p(A')$ aller Antiketten A' mit $A \cap A' \neq \emptyset$ um $\Delta = [\min\{p(A), p(A'), -d(A, A')\}]^+$
  Anmerkung: $d(A, A') = \max_{i \in A, j \in A'} \max\{d_{ij} + p_i, d_{ji} + p_j\}$
- Fahre mit nächster Antikette A fort, bis $p(A) = 0$ für alle $A : |A| \geq 2$
- Nehme in Pricing-Problem (dualisierte) Inkompatibilitätsbedingung $y_{i_A} + y_{i_{A'}} \leq 1$ für eingefügte Antiketten $A \neq A'$ mit $A \cap A' \neq \emptyset$ auf



Antikette $A = \{1, 2, 4\}$
$p(A) = 1$

Preprocessing II

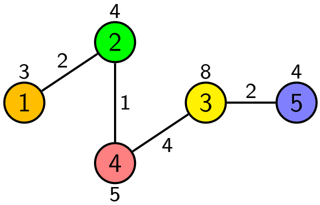
- Beginne mit einer $\subseteq$-maximalen unvermeidbaren Antikette A
- Füge Antikette A als Aktivität i_A mit $p_{i_A} = p(A)$ in Projekt ein
- Reduziere Dauern $p(A')$ aller Antiketten A' mit $A \cap A' \neq \emptyset$ um $\Delta = [\min\{p(A), p(A'), -d(A, A')\}]^+$
  Anmerkung: $d(A, A') = \max_{i \in A, j \in A'} \max\{d_{ij} + p_i, d_{ji} + p_j\}$
- Fahre mit nächster Antikette A fort, bis $p(A) = 0$ für alle $A : |A| \geq 2$
- Nehme in Pricing-Problem (dualisierte) Inkompatibilitätsbedingung $y_{i_A} + y_{i_{A'}} \leq 1$ für eingefügte Antiketten $A \neq A'$ mit $A \cap A' \neq \emptyset$ auf



Antikette $A = \{2, 3, 4\}$
$p(A) = 2$

Preprocessing II

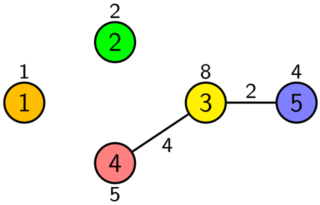
- Beginne mit einer $\subseteq$-maximalen unvermeidbaren Antikette A
- Füge Antikette A als Aktivität i_A mit $p_{i_A} = p(A)$ in Projekt ein
- Reduziere Dauern $p(A')$ aller Antiketten A' mit $A \cap A' \neq \emptyset$ um $\Delta = [\min\{p(A), p(A'), -d(A, A')\}]^+$
  Anmerkung: $d(A, A') = \max_{i \in A, j \in A'} \max\{d_{ij} + p_i, d_{ji} + p_j\}$
- Fahre mit nächster Antikette A fort, bis $p(A) = 0$ für alle $A : |A| \geq 2$
- Nehme in Pricing-Problem (dualisierte) Inkompatibilitätsbedingung $y_{i_A} + y_{i_{A'}} \leq 1$ für eingefügte Antiketten $A \neq A'$ mit $A \cap A' \neq \emptyset$ auf



Antikette $A = \{1, 2\}$
$p(A) = 2$

Preprocessing II

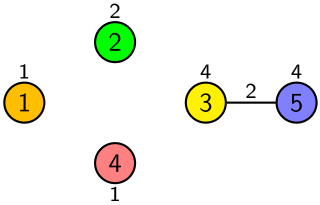
- Beginne mit einer $\subseteq$-maximalen unvermeidbaren Antikette A
- Füge Antikette A als Aktivität i_A mit $p_{i_A} = p(A)$ in Projekt ein
- Reduziere Dauern $p(A')$ aller Antiketten A' mit $A \cap A' \neq \emptyset$ um $\Delta = [\min\{p(A), p(A'), -d(A, A')\}]^+$
  Anmerkung: $d(A, A') = \max_{i \in A, j \in A'} \max\{d_{ij} + p_i, d_{ji} + p_j\}$
- Fahre mit nächster Antikette A fort, bis $p(A) = 0$ für alle $A : |A| \geq 2$
- Nehme in Pricing-Problem (dualisierte) Inkompatibilitätsbedingung $y_{i_A} + y_{i_{A'}} \leq 1$ für eingefügte Antiketten $A \neq A'$ mit $A \cap A' \neq \emptyset$ auf



Antikette $A = \{3, 4\}$
$p(A) = 4$

Preprocessing II

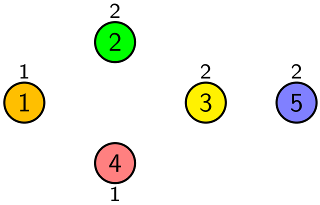
- Beginne mit einer $\subseteq$-maximalen unvermeidbaren Antikette A
- Füge Antikette A als Aktivität i_A mit $p_{i_A} = p(A)$ in Projekt ein
- Reduziere Dauern $p(A')$ aller Antiketten A' mit $A \cap A' \neq \emptyset$ um $\Delta = [\min\{p(A), p(A'), -d(A, A')\}]^+$
  Anmerkung: $d(A, A') = \max_{i \in A, j \in A'} \max\{d_{ij} + p_i, d_{ji} + p_j\}$
- Fahre mit nächster Antikette A fort, bis $p(A) = 0$ für alle $A : |A| \geq 2$
- Nehme in Pricing-Problem (dualisierte) Inkompatibilitätsbedingung $y_{i_A} + y_{i_{A'}} \leq 1$ für eingefügte Antiketten $A \neq A'$ mit $A \cap A' \neq \emptyset$ auf



Antikette $A = \{3, 5\}$
$p(A) = 2$

Preprocessing II

- Beginne mit einer $\subseteq$-maximalen unvermeidbaren Antikette A
- Füge Antikette A als Aktivität i_A mit $p_{i_A} = p(A)$ in Projekt ein
- Reduziere Dauern $p(A')$ aller Antiketten A' mit $A \cap A' \neq \emptyset$ um $\Delta = [\min\{p(A), p(A'), -d(A, A')\}]^+$
  Anmerkung: $d(A, A') = \max_{i \in A, j \in A'} \max\{d_{ij} + p_i, d_{ji} + p_j\}$
- Fahre mit nächster Antikette A fort, bis $p(A) = 0$ für alle $A : |A| \geq 2$
- Nehme in Pricing-Problem (dualisierte) Inkompatibilitätsbedingung $y_{i_A} + y_{i_{A'}} \leq 1$ für eingefügte Antiketten $A \neq A'$ mit $A \cap A' \neq \emptyset$ auf



Experimentelle Performance-Analyse

- Testsets j10, j20, j30 mit jeweils 270 Instanzen (Kolisch et al. 1999)
- Variation des Deadline-Faktors: $DF \in \{1.0, 1.1, 1.5\}$
- Versionen der Spaltengenerierung
  - SG1: ohne Preprocessing
  - SG2: Ersetzung positiver Ende-Start-Abstände
  - SG3: Identifikation von unvermeidbaren Antiketten
  - SG4: Kombination von SG2 und SG3

Mittlere relative Abweichungen von optimalen Zielfunktionswerten (Rieck et al. 2012):

	j10			j20			j30	
DF	1.0	1.1	1.5	1.0	1.1	1.5	1.0	1.1
SG1	7.78 %	6.06 %	1.86 %	8.85 %	5.43 %	1.85 %	9.64 %	5.79 %
SG2	4.43 %	5.23 %	1.83 %	5.51 %	4.73 %	1.83 %	6.16 %	4.91 %
SG3	2.66 %	4.42 %	1.01 %	3.38 %	4.20 %	0.79 %	4.75 %	5.15 %
SG4	1.93 %	3.96 %	0.99 %	2.49 %	3.67 %	0.78 %	3.27 %	4.30 %

Zusammenfassung

- Antiketten-Formulierung von Ressourcennivellierungsproblemen
- Lösung linearer Relaxation mit Spaltengenerierung
- Verschärfung der Relaxation durch Preprocessing
- Mittlere relative Abweichungen $< 5\,\%$ für alle Szenarien
- Güte der Schranken nimmt mit wachsender Problemschwere zu

Ausblick

- Übertragung auf andere Nivellierungszielfunktionen, z. B. Total Overload Cost
- Untersuchung des Komplexitätsstatus des Pricing-Problems
- Branch-and-Bound-Algorithmen für Ressourcennivellierungsprobleme auf Grundlage der Antiketten-Formulierung

Literatur

Rieck J, Zimmermann J, Gather T (2012)
Mixed-integer linear programming for resource leveling problems
European Journal of Operational Research 221: 27–37

Kolisch R, Schwindt C, Sprecher A (1999)
Benchmark instances for project scheduling problems
In: Węglarz J (ed) Project Scheduling: Recent Models, Algorithms and Applications. Kluwer, Boston, pp 197–212