

**Verfahren
zur
ressourcenbeschränkte Projektplanung**

Vorlesung vom 08.11.2005: Einführung (RCPS)

- Wiederholung
- Zeitplanung
- RCPS - Prioritätsregelverfahren

C.-U. Fündeling, Ch. Mellentien

LS Prof. Dr. K. Neumann

Institut für Wirtschaftstheorie und Operations Research

Universität Karlsruhe (TH)

Einführung

- Wiederholung
- Zeitplanung
- RCPS – Prio.

- Standardproblem

$$\left. \begin{array}{l}
 \text{Min.} \quad f(S) \\
 \text{u.d.N.} \quad r_k(S, t) \leq R_k \quad (k \in \mathcal{R}, 0 \leq t \leq \bar{d}) \\
 \quad \quad S_j - S_i \geq \delta_{ij} \quad (\langle i, j \rangle \in E) \\
 \quad \quad S_i \geq 0 \quad (i \in V) \\
 \quad \quad S_0 = 0
 \end{array} \right\} (PS|temp, \bar{d}|f)$$

- Zeitplanungsproblem (Ressourcenrelaxation)

$$\left. \begin{array}{l}
 \text{Min.} \quad f(S) \\
 \text{u.d.N.} \quad S_j - S_i \geq \delta_{ij} \quad (\langle i, j \rangle \in E) \\
 \quad \quad S_i \geq 0 \quad (i \in V) \\
 \quad \quad S_0 = 0
 \end{array} \right\} (PS_\infty|temp, \bar{d}|f)$$

- Problemschwere

- $PS|temp, \bar{d}|f$: NP-schwer
- $PS_\infty|temp, \bar{d}|f$: Abhängig von $f(S)$

Einführung

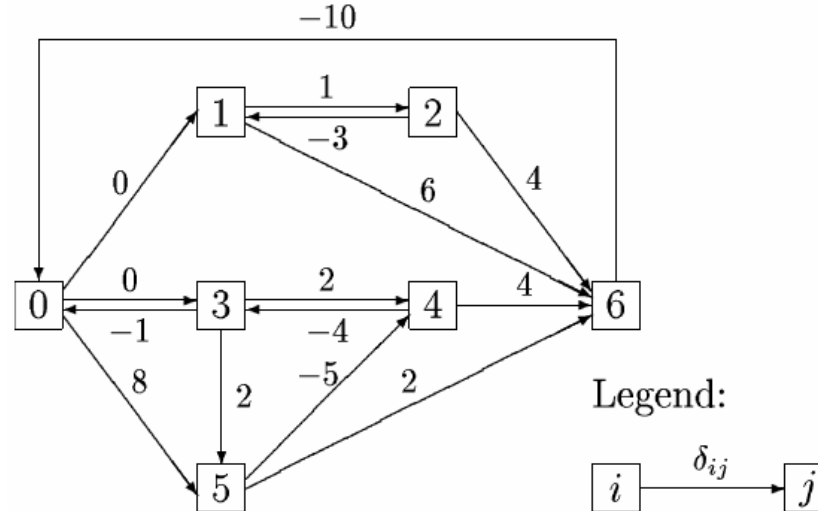
- Wiederholung
- **Zeitplanung**
- RCPS – Prio.

- Zulässigkeit eines Schedules
 - Zeit-zulässig = erfüllt zeitliche Nebenbedingungen
 - Ressourcen-zulässig = erfüllt Ressourcen-NB
 - Zulässig = zeit- und ressourcen-zulässig
- Optimalität eines Schedules
 - Zeit-optimal = zeit-zulässig und minimal bzgl. f
 - Optimal = zulässig und minimal bzgl. f
- Motivation für (reine) Zeitplanung
 - Zulässig immer auch zeit-zulässig
 - Zeitplanung u.U. „leichter“ als RCPS
 - ⇒ Häufiger Ansatz
 1. Bestimmung „zeit-optimaler“ Schedule
 2. Reparatur/Behebung von Ressourcenkonflikten

Einführung

- Wiederholung
- **Zeitplanung**
- RCPS – Prio.

- Netzwerk (mit allen zeitlichen NB)



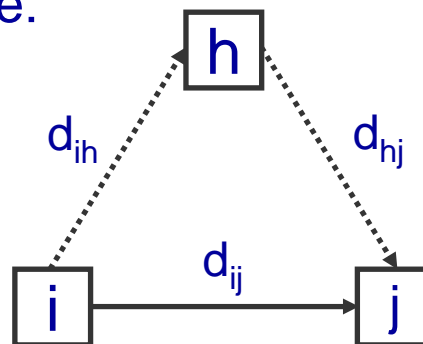
- Definition: Projektstart $S_0 \stackrel{!}{=} 0$
 - Frühester (zeit-zulässiger) Startzeitpunkt ES_i
 - (Induzierter) Mindestabstand zw. 0 und i
 - Länge des längsten Wegs d_{0i}
 - Hier: $ES = (0, 0, 1, 0, 3, 8, 10)$
 - Spätester (zeit-zulässiger) Startzeitpunkt LS_i
 - (Induzierter) Höchstabstand zw. 0 und i
 - (Negative) Länge des längsten Wegs $-d_{i0}$
 - Hier: $LS = (0, 4, 6, 1, 5, 8, 10)$
- ⇒ Zeit-zulässiges Zeitfenster $[ES_i; LS_i]$

Einführung

- Wiederholung
- **Zeitplanung**
- RCPS – Prio.

- Zu bestimmender zeit-zulässiger Startzeitpunkt $S_i \in [ES_i; LS_i]$
 - Abhängig von
 - Zielfunktion f
 - Startzeitpunkten anderer Aktivitäten S_j
 - $\Rightarrow d_{ji}, d_{ij}$ notwendig
 - Folge: $ES_i \nearrow, LS_i \searrow$

- Tripel-Algorithmus zur Berechnung längster Wege zw. allen Knoten $D := (d_{ij})_{i,j \in V}$
 - OR-Einführung (Kürzeste Wege)
 - Idee:



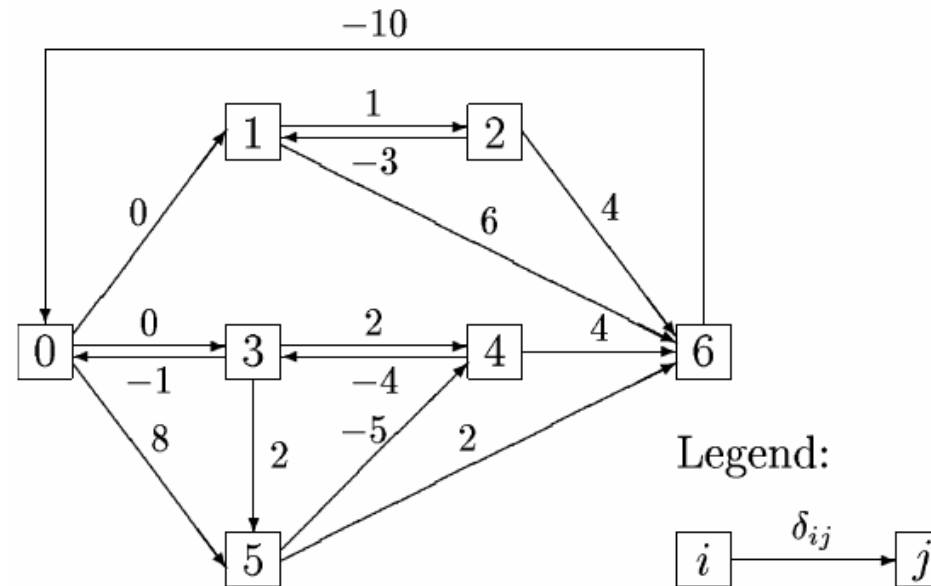
$$d_{ij} < d_{ih} + d_{hj} ?$$

$$\text{Ja} \Rightarrow d_{ij} := d_{ih} + d_{hj}$$

Einführung

- Wiederholung
- **Zeitplanung**
- RCPS – Prio.

- **Beispiel**



$i \setminus j$	0	1	2	3	4	5	6
0	0	0	1	0	3	8	10
1	-4	0	1	-4	-1	4	6
2	-6	-3	0	-6	-3	2	4
3	-1	-1	0	0	2	2	6
4	-5	-5	-4	-4	0	3	4
5	-8	-8	-7	-8	-5	0	2
6	-10	-10	-9	-10	-7	-2	0

Einführung

- *Wiederholung*
- **Zeitplanung**
- *RCPS – Prio.*

- Polynomial
 - Project Duration (ES zeit-optimal)
 - Maximum Lateness (ES zeit-optimal)
 - **Weighted Earliness-Tardiness** (Verfahren des steilsten Abstiegs)
- NP-schwer
 - Resource Investment Cost
 - **Resource Adjustment Cost**
- Offen: Net Present Value

Einführung

- Wiederholung
- Zeitplanung
- RCPS – Prio.

- Grundprinzip für $PS(\infty)|temp, \bar{d}|f$
 1. Bestimme Menge einplanbarer Aktivitäten
 2. Wähle eine Aktivität aus (**Prioritätsregel**)
 3. Plane Aktivität (zeit-)zulässig und „**kostenminimal**“ (lokal optimal!) ein unter Berücksichtigung bisher geplanter Aktivitäten
 4. Aktualisiere Zeitfenster für noch zu planende Aktivitäten
 5. Fahre mit 1. fort, wenn noch Aktivitäten zu planen
- Rückplanung
 - Wenn Aktivität nicht zulässig planbar (3.), plane bereits geplante Aktivitäten **teilweise** (d.h. wenn nötig) wieder aus, weiter mit 1.
 - Notwendig aufgrund zeitlicher Höchstabstände
- Vorteil: Schnell
- Nachteil: Heuristik
- Einsatzbereich
 - schwierige und/oder große Probleme
 - Hier: $n > 300$ Aktivitäten

Einführung

- *Wiederholung*
- *Zeitplanung*
- **RCPS – Prio.**

- **Zeitbasiert**
 - Latest Start Time
 - Minimum Slack Time (LS – ES)
 - Most Total Successors
 - Maximum Weighted Earliness-Tardiness
- **Ressourcenbasiert**
 - Greatest Resource Demand
 - Greatest Resource Requirements
- **Statisch: Berechnung zu Beginn des Verfahrens**
- **Dynamisch: Aktualisierung während des Verfahrens**

Einführung

- Wiederholung
- Zeitplanung
- RCPS – Prio.

- Varianten des Grundprinzips

- Direkte Methode: Betrachte stets Gesamtproblem

- Serielles Generierungsschema: Aktualisiere Menge einplanbarer Aktivitäten nach jeder Einplanung (siehe Grundprinzip)
- Paralleles GS
 - Plane alle Aktivitäten aus 1. nach Prioritätsregeln (wdh. 2. bis 4.)
 - Aktualisiere anschließend erst Menge einplanbarer Aktivitäten (1.)

- Dekomposition

1. Zerlege Problem in Teilprobleme z.B. nach Zyklenstrukturen
2. Löse (kleinere) Teilprobleme, z.B. direkt
3. Konstruiere (Gesamt-)Lösung aus Teillösungen

Einführung

- Wiederholung
- Zeitplanung
- RCPS – Prio.

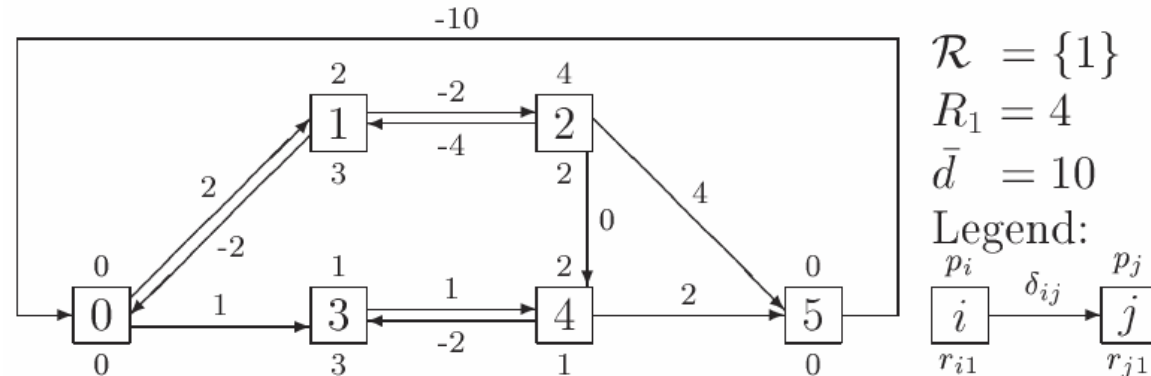
- Varianten des Grundprinzips
 - Reparaturmethode:
 - Löse $PS_{\infty} | temp, \bar{d} | f$
 - „Repariere“ auftretende Ressourcenkonflikte durch Verschiebung (nach links, rechts) einzelner (direkt, indirekt) beteiligter Aktivitäten

- Themen im SW-Praktikum: **Eine** Gruppe
 - „Klassisches“ Prio.verfahren: Direkte Methode, serielles GS für $PS | temp, \bar{d} | WET, RAC$
 - Reparaturmethode
 - Direkte Methode, serielles GS für $PS_{\infty} | temp, \bar{d} | WET, RAC$
 - Reparaturmechanismen

Einführung

- Wiederholung
- Zeitplanung
- RCPS – Prio.

- Beispielproblem



- Bestimme zeit-zulässigen Schedule bei folgenden Vorgaben

$$f = RAC = \sum \sum c_k^+ (\Delta r_{kt})^+ + c_k^- (-\Delta r_{kt})^+$$

$$c_1^+ = 2, c_1^- = 3$$

- Direkte Methode, serielles GS
- Prioritätsregelkombination (s)GRD – (d)MST

$$(s)GRD(i) := p_i \sum_{k \in \mathcal{R}} r_{ik}$$

$$(d)MST(i) := TF_i^c = LS_i^c - ES_i^c$$

- Wähle unter gleichwertigen Startzeitpunkten den frühesten (Kleinste-Index-Regel)