Das disjunktive Graphenmodell für Shop-Probleme

Definition 1: Ein disjunktiver Graph ist ein Graph $G = (V, E)$ mit

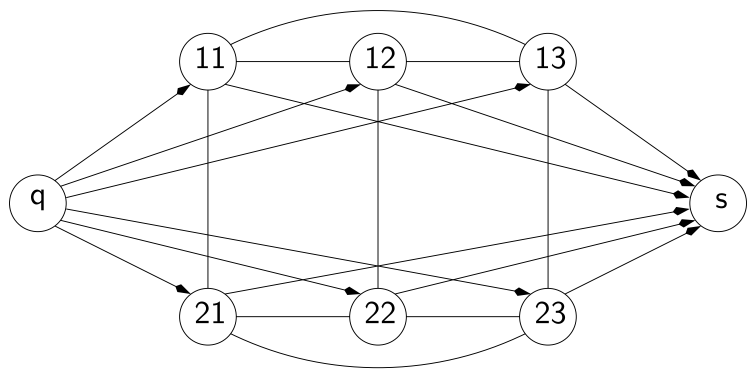
- $V = \{(i,j) \mid (i,j) \in SIJ\} \cup \{q\} \cup \{s\}$
  Die Knotenmenge ist die Menge der Operationen zuzüglich einer Quelle q und einer Senke s.
- $E = \mathbb{C} \cup \mathbb{D}$, wobei gilt:
  - $\mathbb{C}$ ist die Menge der **konjunktiven** (gerichteten) Kanten, die die Vorrangbedingungen zwischen den Operationen widerspiegeln. Dazu kommen gerichtete Kanten von der Quelle zu allen Operationen ohne Vorgänger und gerichtete Kanten von allen Operationen, die keinen Nachfolger haben, zur Senke.
  - $\mathbb{D}$ ist die Menge aller **disjunktiven** (ungerichteten) Kanten. Eine disjunktive Kante wird zwischen Operationen mit dem gleichen Auftrag oder der gleichen Maschine eingeführt, die nicht bereits durch eine Kante aus $\mathbb{C}$ verbunden sind.

Jeder Knoten (i,j) wird mit p_{ij} gewichtet, q und s erhalten das Gewicht 0. Ein Plangraph entspricht damit einem azyklisch gerichteten disjunktiven Graphen, aus dem Quelle und Senke mit den inzidierenden Kanten entfernt wurden.

Das disjunktive Graphenmodell - Beispiel

Beispiel 1. Betrachtet wird das Problem $O \mid\mid C_{max}$ mit der Bearbeitungszeitmatrix:

$$P = \begin{pmatrix} 2 & 3 & 3 \\ 4 & 1 & 2 \end{pmatrix}.$$



Alle Kanten sind disjunktiv, die den technologischen und organisatorischen Reihenfolgen entsprechen. Damit ist der von allen Knoten mit gleichem Auftrag (bzw. mit gleicher Maschine) aufgespannte Teilgraph eine Clique. Gesucht ist eine Orientierung der disjunktiven Kanten, so dass der entstehende Graph zyklenfrei ist und der kritische Weg (C_{max}) minimal wird!

Das Blockmatrizenmodell für Shop-Probleme

Pläne und Lateinische Rechtecke

Definition 2: Ein **lateinisches Rechteck** $LR[n, m, r] = (lr_{ij})$ ist eine $n \times m$ Matrix mit den Elementen lr_{ij} aus einer Belegungsmenge $B = \{1, \ldots, r\}$, wobei jede Zahl aus B in jeder Zeile und Spalte des Rechtecks höchstens einmal vorkommt.
$LQ[n] = LR[n, m, r]$ mit $n = m = r$ wird als **lateinisches Quadrat** der Ordnung n bezeichnet.

Satz 1. *Aus jedem lateinischen Rechteck $LR[n, m, r]$ lässt sich eindeutig eine zulässige Kombination von technologischen und organisatorischen Reihenfolgen für ein Open-Shop Problem mit n Aufträgen und m Maschinen konstruieren.*

Satz 2. *Jeder zulässigen Kombination von technologischen und organisatorischen Reihenfolgen eines Open-Shop Problems mit $SIJ = I \times J$, $|I| = n$, $|J| = m$, lässt sich eineindeutig ein lateinisches Rechteck $LR[n, m, r]$ zuordnen, das die folgende* **Sequence-Bedingung** *erfüllt:*
Zu jedem $lr_{ij} > 1$ existiert entweder in Zeile i oder in Spalte j oder in beiden die Zahl $lr_{ij} - 1$.

Das Blockmatrizenmodell für Shop-Probleme

$$\underbrace{\begin{array}{cc} P = (p_{ij}) & LR = (lr_{ij}) \\ \text{Bearbeitungs-} & \text{Plan} \\ \text{zeiten} & \end{array}}_{} \qquad \begin{array}{c} \Longrightarrow \\ \Longleftarrow \\ \text{wenn } G(MO, JO) \\ \text{azyklisch ist} \end{array} \qquad \left\{ \begin{array}{c} MO = (mo_{ij}) \\ \text{machine orders} \\ JO = (jo_{ij}) \\ \text{job orders} \end{array} \right.$$

$$\Downarrow$$

$$C = (c_{ij})$$

Schedule

$H = (h_{ij}),\ T = (t_{ij})$ und $W = (w_{ij})$

Heads, Tails und Weights

h_{ij} ist die (kürzeste) Zeit zur Bearbeitung aller Vorgänger der Operation (i, j)

t_{ij} ist die (kürzeste) Zeit zur Bearbeitung aller Nachfolger der Operation (i, j)

w_{ij} ist das Gewicht eines maximal gewichteten Weges über die Operation (i, j)

Es gilt: $C = H + P$ und $W = H + P + T$

Algorithmen zum Blockmatrizenmodell (1)

Algorithmus zur Bestimmung von $LR[n, m, r]$ aus MO und JO (falls möglich)

Eingabe: n, m, SIJ, Matrizen MO und JO über der Operationenmenge SIJ;
Ausgabe: $LR[n, m, r]$ auf der Operationenmenge SIJ (falls $G(MO, JO)$ zyklenfrei).
BEGIN
  $k := 0$;
  REPEAT
    $k := k + 1$; Bestimme die Menge $MQ = \{(i, j) \in SIJ \mid mo_{ij} = jo_{ij} = 1\}$;
    IF $MQ = \emptyset$ **THEN** (MO, JO) ist unzulässig und **STOP**;
    FOR ALL$(i, j) \in MQ$ **DO**
      BEGIN $lr_{ij} = k$; Markiere in MO die Zeile i und in JO die Spalte j; **END**;
    $SIJ := SIJ \setminus MQ$;
    FOR ALL $(i, j) \in SIJ$ aus einer markierten Zeile in MO **DO** $mo_{ij} := mo_{ij} - 1$;
    FOR ALL $(i, j) \in SIJ$ aus einer markierten Spalte in JO **DO** $jo_{ij} := jo_{ij} - 1$;
  UNTIL $SIJ = \emptyset$;
END.

Algorithmen zum Blockmatrizenmodell (2)

Bezeichnungen:
a_i: kleinste Zahl, die für den Rang einer Operation von Auftrag A_i in MO zur Verfügung steht;
b_j: kleinste Zahl, die für den Rang einer Operation auf Maschine M_j in JO zur Verfügung steht.

Algorithmus zur Bestimmung der Matrizen MO und JO aus dem Plan $LR[n, m, r]$

Eingabe: n, m, r, I, J, SIJ, Plan $LR[n, m, r]$;
Ausgabe: Matrizen MO und JO;
BEGIN Setze für alle $i \in I$: $a_i = 1$ und für alle $j \in J$: $b_j = 1$;
  FOR $k := 1$ **TO** r **DO**
    FOR ALL $(i, j) \in SIJ$ mit $lr_{ij} = k$ **DO**
      BEGIN
        Setze $mo_{ij} = a_i$ und $a_i = a_i + 1$;
        Setze $jo_{ij} = b_i$ und $b_j = b_j + 1$;
      END;
END.

Algorithmen zum Blockmatrizenmodell (3)

Bezeichnungen:

r_i: früheste Verfügbarkeit von Auftrag A_i; $\overline{r}_j$: früheste Verfügbarkeit von Maschine M_j.

Algorithmus zur Bestimmung der Matrix C aus den Matrizen P und $LR[n, m, r]$

Eingabe: n, m, r, I, J, SIJ, Bearbeitungszeitmatrix P und Plan $LR[n, m, r]$;
Ausgabe: Matrix $C = (c_{ij})$ der Fertigstellungszeiten der Operationen $(i, j) \in SIJ$.
BEGIN

Setze für alle $i \in I$: $r_i = 0$ und für alle $j \in J$: $\overline{r}_j = 0$;

FOR $k := 1$ **TO** r **DO FOR ALL** $(i, j) \in SIJ$ mit $lr_{ij} = k$ **DO**

BEGIN

$c_{ij} := \max\{r_i, \overline{r}_j\} + p_{ij}$;

$r_i := c_{ij}$; $\overline{r}_j := c_{ij}$;

END;

END.

Algorithmen zum Blockmatrizenmodell (4)

Bezeichnungen: $r_i, \overline{r}_j$: früheste Verfügbarkeit von Auftrag A_i bzw. von Maschine M_j;
$s_i, \overline{s}_j$: früheste Verfügbarkeit von Auftrag A_i bzw. von Maschine M_j bei der Rückwärtsrechnung.

Bestimmung der Heads h_{ij} und Tails t_{ij} für alle Operationen $(i, j) \in SIJ$

Eingabe: n, m, r, I, J, SIJ, Matrix P und Plan $LR[n, m, r]$;
Ausgabe: Matrizen H und T.
BEGIN Setze für alle $i \in I$: $r_i = 0$ und für alle $j \in J$: $\overline{r}_j = 0$;
Setze für alle $i \in I$: $s_i = 0$ und für alle $j \in J$: $\overline{s}_j = 0$;
FOR $k := 1$ **TO** r **DO**
BEGIN FOR ALL $(i, j) \in SIJ$ mit $lr_{ij} = k$ **DO**
BEGIN $h_{ij} := \max\{r_i, \overline{r}_j\}$; $r_i := h_{ij} + p_{ij}$; $\overline{r}_j := h_{ij} + p_{ij}$; **END**;
FOR ALL $(i, j) \in SIJ$ mit $lr_{ij} = r - k + 1$ **DO**
BEGIN $t_{ij} := \max\{s_i, \overline{s}_j\}$; $s_i := t_{ij} + p_{ij}$; $\overline{s}_j := t_{ij} + p_{ij}$; **END**;
END;
END.

Shop-Probleme mit Unterbrechungen

Sei $z_{ij} - 1$, $z_{ij} \geq 1$, die Anzahl der Unterbrechungen der Operation (i, j). Dann wird jede Operation aus SIJ in eine geordnete Menge $pSIJ$ von Teiloperationen $\{(i, j)_k \mid k = 1, \ldots, z_{ij}\}$ aufgespalten, wobei $(i, j)_k$ bearbeitet sein muss, bevor die Bearbeitung von $(i, j)_{k+1}$ starten kann, $k = 1, \ldots, z_{ij} - 1$.

Analog zu den Problemen ohne $pmtn$ werden die folgenden Digraphen definiert:

- $G^p(MO) = (pSIJ, A(pMO))$ enthält gerichtete Kanten, gegeben durch die Vorrangbedingungen der Teiloperationen für den gleichen Auftrag A_i, $i = 1, \ldots, n$, in allen technologischen Reihenfolgen.
- $G^p(JO) = (pSIJ, A(pJO))$ enthält gerichtete Kanten, gegeben durch die Vorrangbedingungen der Teiloperationen auf der gleichen Maschine M_j, $j = 1, \ldots, m$, in allen organisatorischen Reihenfolgen.
- $G^p(MO, JO) = (pSIJ, A(pMO, pJO))$ enthält alle gerichteten Kanten aus $A(pMO) \cup A(pJO)$.

Shop-Probleme mit Unterbrechungen

Definition: Ein **preemptiver Plangraph** ist ein azyklischer Digraph $G^p(MO, JO)$. Zur Beschreibung der Graphen werden **preemptive Matrizen** verwendet, deren Elemente geordnete Mengen sind:

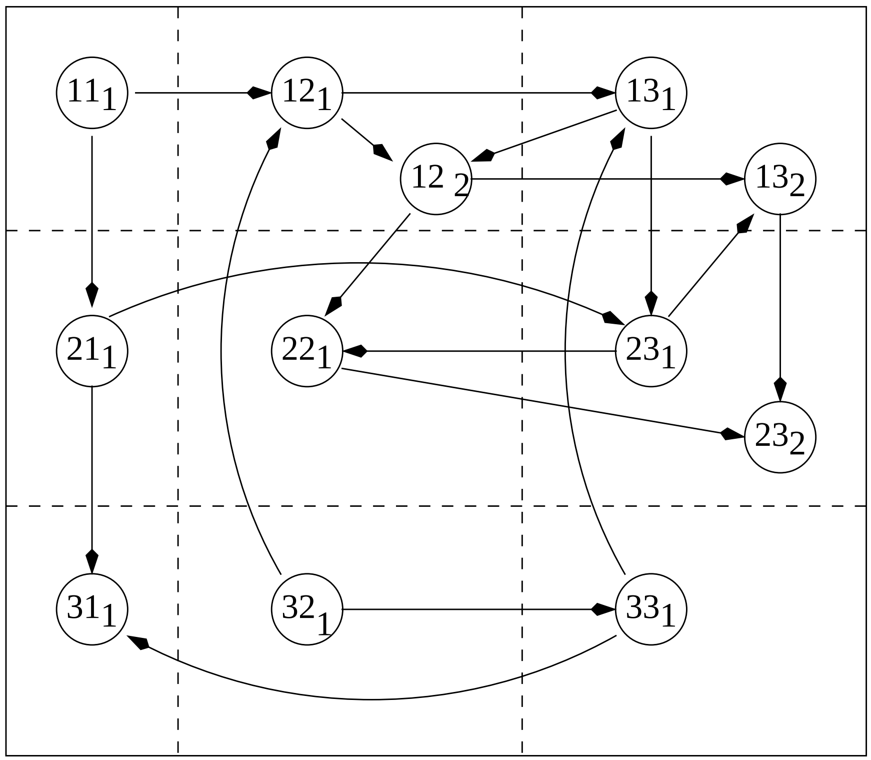
- die preemptive Matrix $pMO = (\{mo_{ij}^1, \ldots, mo_{ij}^{z_{ij}}\})$, wobei mo_{ij}^k der Rang der Teiloperation $(i, j)_k$ in $G^p(MO)$ ist;
- die preemptive Matrix $pJO = (\{jo_{ij}^1, \ldots, jo_{ij}^{z_{ij}}\})$, wobei jo_{ij}^k der Rang der Teiloperation $(i, j)_k$ in $G^p(JO)$ ist;
- der **premptive Plan** $pLR = (\{lr_{ij}^1, \ldots, lr_{ij}^{z_{ij}}\})$, wobei lr_{ij}^k der Rang der Operation $(i, j)_k$ im preemptiven Plangraphen $G^p(MO, JO)$ ist.

Jetzt: Zerlegung von p_{ij} in z_{ij} Teile, wobei $p_{ij} = \sum\limits_{k=1}^{z_{ij}} p_{ij}^k$ gelten muss. Dann erhält man:

- die **preemptive Bearbeitungszeitmatrix** $pP = (\{p_{ij}^1, \ldots, p_{ij}^{z_{ij}}\})$, wobei p_{ij}^k die Bearbeitungzeit der Teiloperation $(i, j)_k$ ist.
- den **preemptiven Schedule** $pC = (\{c_{ij}^1, \ldots, c_{ij}^{z_{ij}}\})$, wobei c_{ij}^k die Fertigstellungszeit der Teiloperation $(i, j)_k$ ist.

Shop-Probleme mit Unterbrechungen - Beispiel

Beispiel 2. Betrachtet wird folgender preemptiver Plangraph mit $n = 3$ und $m = 3$, wobei $z_{ij} = 2$ für $(i, j) \in \{(1, 2), (1, 3), (2, 3)\}$ und $z_{ij} = 1$ für alle anderen Operationen.



Die preemptiven Matrizen

$$pMO = \begin{pmatrix} \{1\} & \{2,4\} & \{3,5\} \\ \{1\} & \{3\} & \{2,4\} \\ \{3\} & \{1\} & \{2\} \end{pmatrix}, \quad pJO = \begin{pmatrix} \{1\} & \{2,3\} & \{2,4\} \\ \{2\} & \{4\} & \{3,5\} \\ \{3\} & \{1\} & \{1\} \end{pmatrix},$$

$$pLR = \begin{pmatrix} \{1\} & \{2,4\} & \{3,5\} \\ \{2\} & \{5\} & \{4,6\} \\ \{3\} & \{1\} & \{2\} \end{pmatrix}.$$

Unter Verwendung der preemptiven Bearbeitungszeitmatrix pP wird der zugehörende preemptive Schedule berechnet:

$$pP = \begin{pmatrix} \{3\} & \{2,2\} & \{4,3\} \\ \{4\} & \{3\} & \{3,1\} \\ \{6\} & \{2\} & \{3\} \end{pmatrix}, \quad pC = \begin{pmatrix} \{3\} & \{5,11\} & \{9,15\} \\ \{7\} & \{15\} & \{12,16\} \\ \{13\} & \{2\} & \{5\} \end{pmatrix}.$$

Alle Zielfunktionswerte $f(C_1, \ldots, C_n)$ für diesen Schedule können sofort mittels $C_1 = 15$, $C_2 = 16$ and $C_3 = 13$ berechnet werden, also z.B. $C_{max} = 16$.

Die Sequence-Bedingung im Fall von Unterbrechungen

Der preemptive Plan pLR mit dem maximalem Rang r hat die folgenden Eigenschaften:

(a) Jedes Element von $\{1, \ldots, r\}$ kommt höchstens einmal in jeder Zeile und Spalte vor;

(b) **Sequence-Bedingung:** Zu jedem Element $lr_{ij}^k = l > 1$ existiert ein Eintrag $l - 1$ in Zeile i oder Spalte j.

Bemerkung: Die Matrizen H, T und W können im Fall von Unterbrechungen zu den preemptiven Matrizen pH, pT und pW modifiziert werden.

Die Algorithmen 1 - 4 lassen sich für den Fall von Unterbrechungen modifizieren.

Das Blockmatrizenmodell lässt sich auch auf Job-Shop, Flow-Shop und Mixed-Shop Probleme anwenden, da sich Vorrangbedingungen zwischen den Operationen einfach einarbeiten lassen. Dies gilt auch für den Fall von Unterbrechungen.

Pläne und Schedules für Einmaschinenprobleme

Einmaschinenprobleme: Einschränkung des B-Modells auf eine Maschine

Plan: Organisatorische Reihenfolge der Aufträge ist gesucht:

(1) $A_{i_1} \to A_{i_2} \to \ldots \to A_{i_n}$ entspricht der Reihenfolge $(i_1, i_2, \ldots, i_n)$ oder

(2) $\Pi = JO^T = (\pi_1 \pi_2 \ldots \pi_n)$, wobei π_μ die Position von Auftrag A_μ ist.

Schedule: Bearbeitungsendzeitpunkte C_i werden erfasst:

zu (1): $(C_{i_1} C_{i_2} \ldots C_{i_n})$ oder zu (2): $(C_1 C_2 \ldots C_n)$

Daneben wird mit dem Graphen $G(JO)$ und mit dem Ganttdiagramm gearbeitet.

Pläne und Schedules für Parallelmaschinenprobleme

Generell: B-Modell ineffizient, weil die Matrizen zu schwach besetzt sind.

Plan: Wird durch die Matrix $\overline{JO} = (\overline{jo}_{ij})$ beschrieben, wobei $\overline{jo}_{ij}$ den i-ten Auftrag auf Maschine M_j bezeichnet.

Schedule: Wird durch die Matrix $\overline{C} = (\overline{c}_{ij})$ beschrieben, wobei $\overline{c}_{ij}$ die Fertigstellungszeit des i-ten Auftrags auf Maschine M_j bezeichnet.

Bemerkung: In der Literatur werden Schedules auch durch die Startzeiten der einzelnen Operationen beschrieben. Oftmals wird nicht zwischen Plänen und Schedules unterschieden, so dass auch Pläne als Schedules verwendet werden.