frico
future research in combinatorial optimization

Gerechtigkeit am Fließband

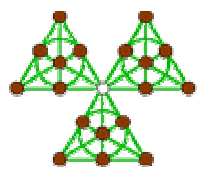
Sequenzierung von Bitvektoren unter Nebenbedingungen

Robert Nickel, Winfried Hochstättler

Lehrstuhl für Mathematische Grundlagen der Informatik

Brandenburgische Technische Universität Cottbus

BTU



Eine Anwendung aus der Automobilherstellung

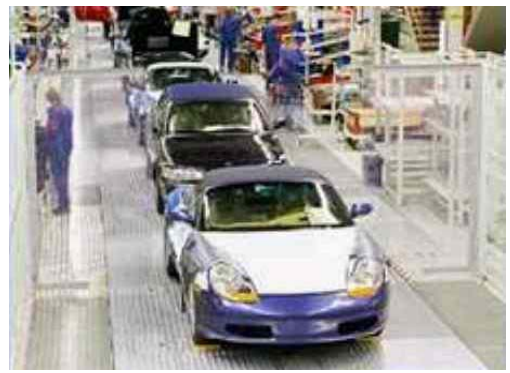
„Body Shop“

- Pressen, Montieren und Verschweißen der Karosserie

„Paint Shop“

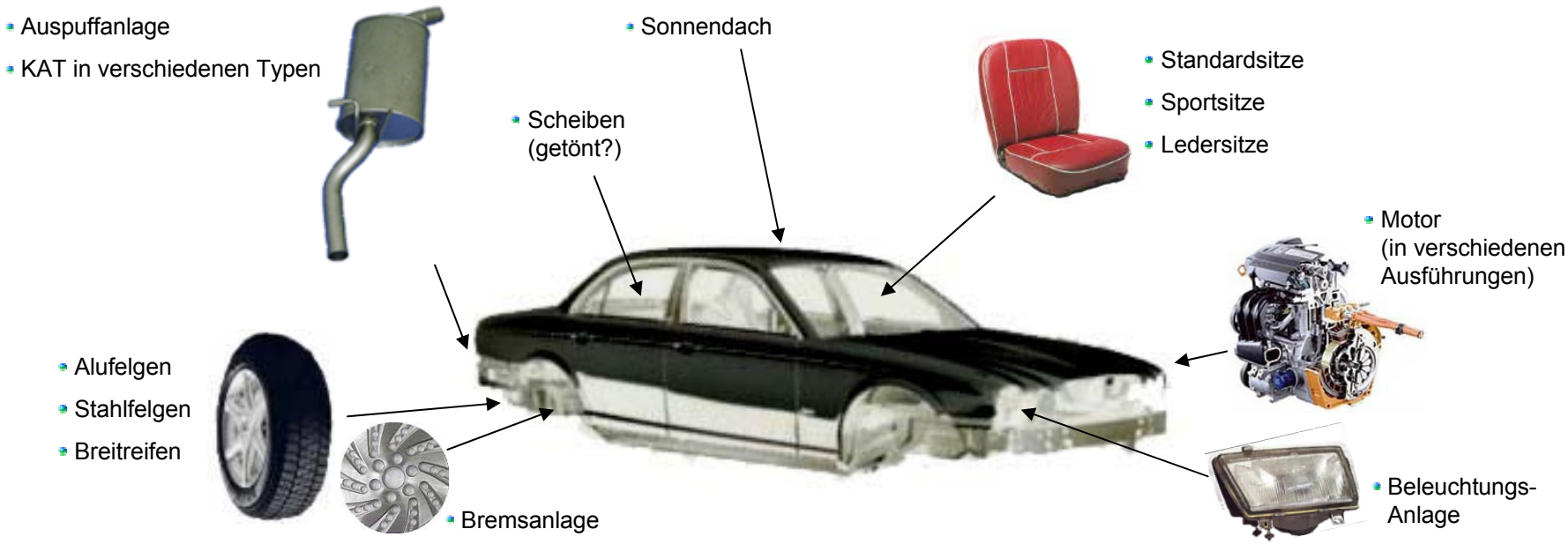
- Grundieren und Lackieren der Karosserie

„Assembly Line“



- Einbau aller Komponenten

Arbeit auf dem Fließband



- Einbau aller Features erfolgt auf der Assembly Line durch separate Teams
- Es gibt etwa 150 Features und mehrere hundert Zusammenstellungen pro Tagesproduktion
- Eine Tagesproduktion besteht aus etwa 2000 Modellen

Modellbildung

- Es gibt eine feste Menge einbaubarer Features und jedes Modell beinhaltet ein Teilmenge davon

  $\Longrightarrow$ Jedes Modell kann als 0-1-Vektor aufgefasst werden

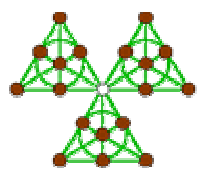
- Modelle mit gleicher Menge an Features werden als ein Modelltyp bezeichnet

  $\Longrightarrow$ Die Tagesproduktion ist eine Multimenge von Bitvektoren

$$F = \{b_1, \cdots, b_1, b_2, \cdots, b_2, \cdots, b_\alpha, \cdots, b_\alpha\}$$

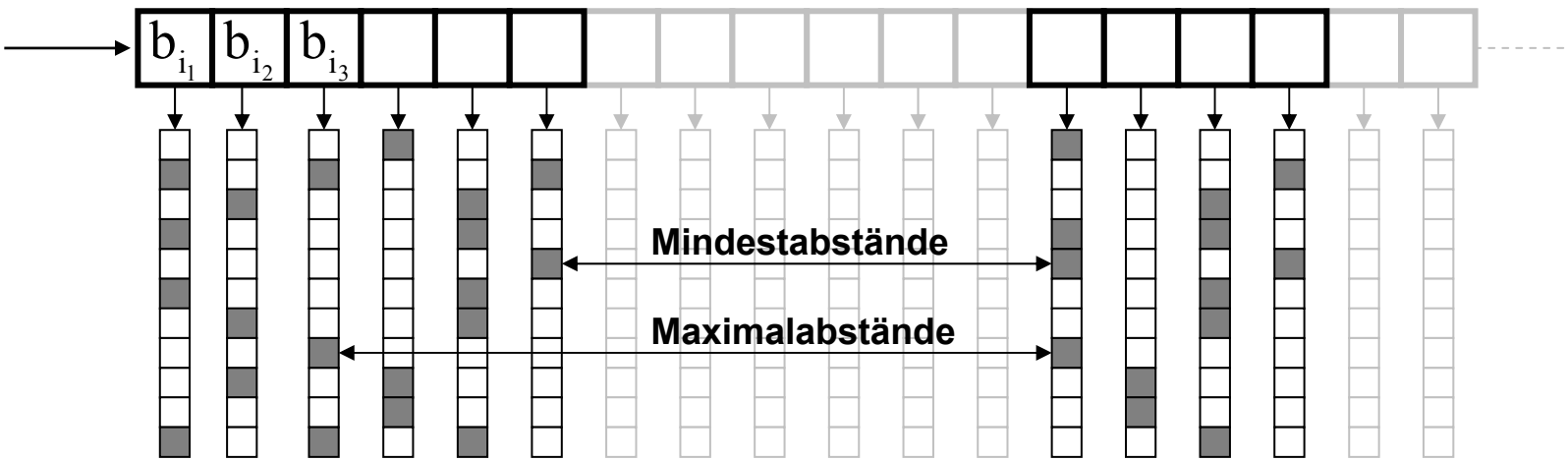
- Die Autos werden in regelmäßigen Abständen aufs Fließband gelegt

  $\Longrightarrow$ Wir betrachten vereinfacht eine Sequenz von Bitvektoren



Modellbildung II

- Der Einbau eines Features benötigt eine bestimmte Zeit
- Kein Team sollte zu lange nichts zu tun haben



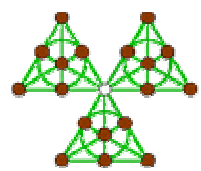
Bitvektorsequenzierung mit Abstandsbedingungen

- Gegeben:
  - Eine Multimenge F von β-dimensionalen Bitvektoren

$$F = \{b_1, \cdots, b_1, b_2, \cdots, b_2, \cdots, b_\alpha, \cdots, b_\alpha\}$$

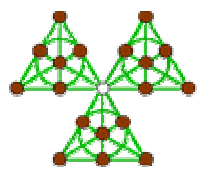
  - sowie Zahlen $\bar{m}_j$ und $\underline{m}_j$ mit $\underline{m}_j < \bar{m}_j \qquad \forall j = 1..\beta$

- Gesucht:
  - Finde eine Sequenzierung der Vektoren von F, so dass zwischen zwei Einsen in der Komponente j immer mindestens $\underline{m}_j$ und maximal $\bar{m}_j$ Nullen stehen!

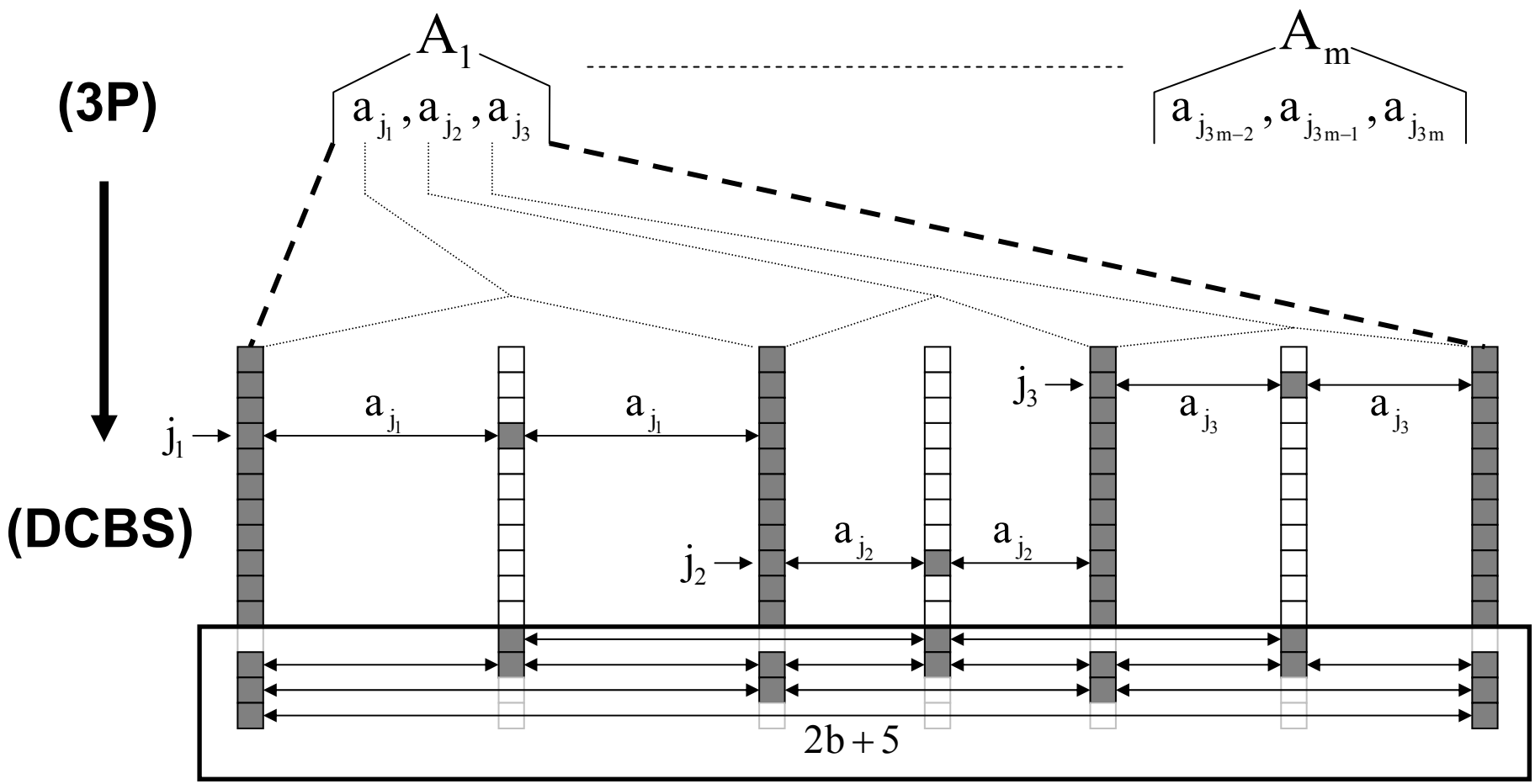


Komplexitätsanalyse

$\mathcal{NPC}$	$\mathcal{P}$
• Allgemeiner Fall: Pseudo-polynomielle Reduktion vom THREE-PARTITION-PROBLEM • Fixierte Schranken: Reduktion vom HAMILTONIAN-PATH-PROBLEM	• fixierte Schranken, beschränkte Anzahl Features: Reduktion auf SHORTEST-PATH • Allgemeine Schranken, beschränkte Anzahl Features: Reduktion auf SHORTEST-PATH

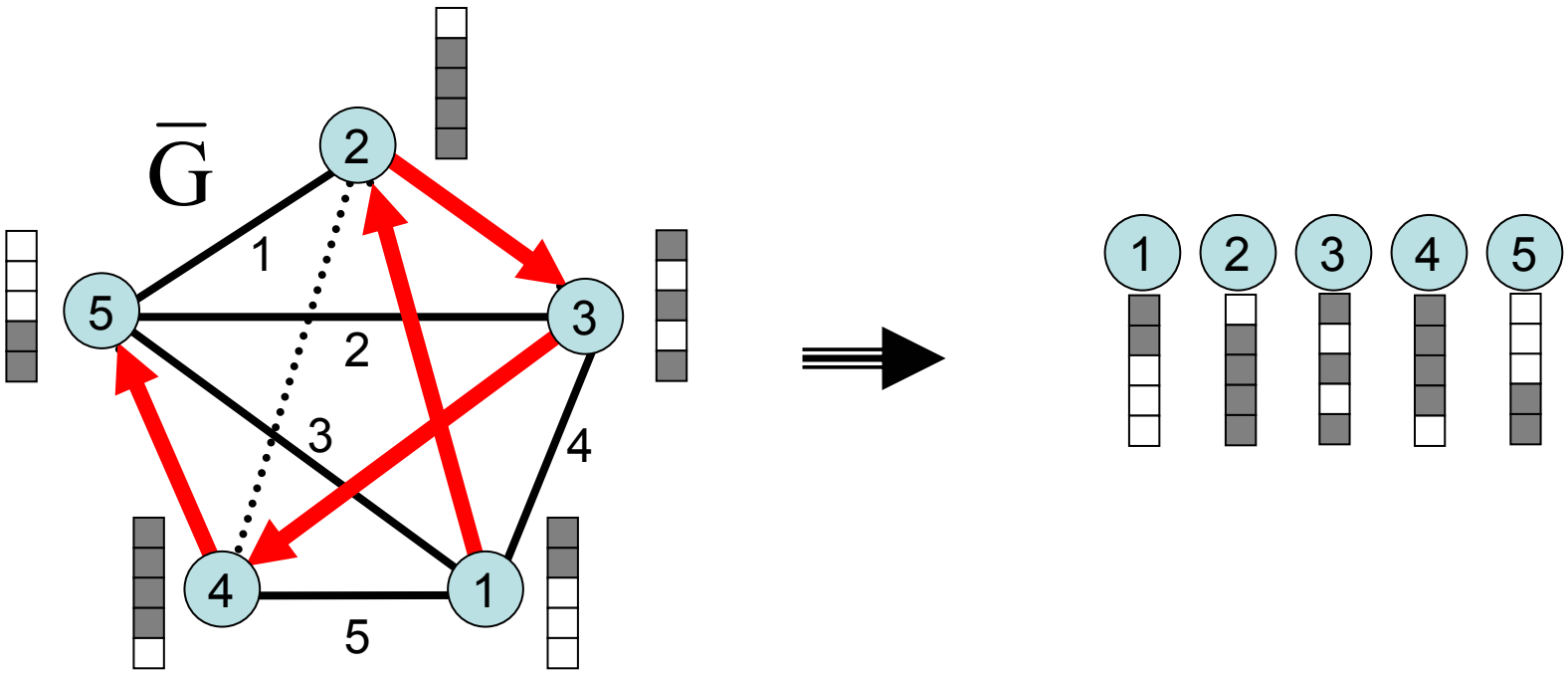


Der allgemeine Fall - Beweisskizze



Ein einfacher Spezialfall

- Zwei Einsen dürfen durch maximal eine Null getrennt sein
- Reduktion vom HAMILTONIAN-PATH-PROBLEM

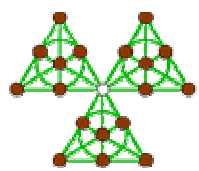


Polynomielle Fälle

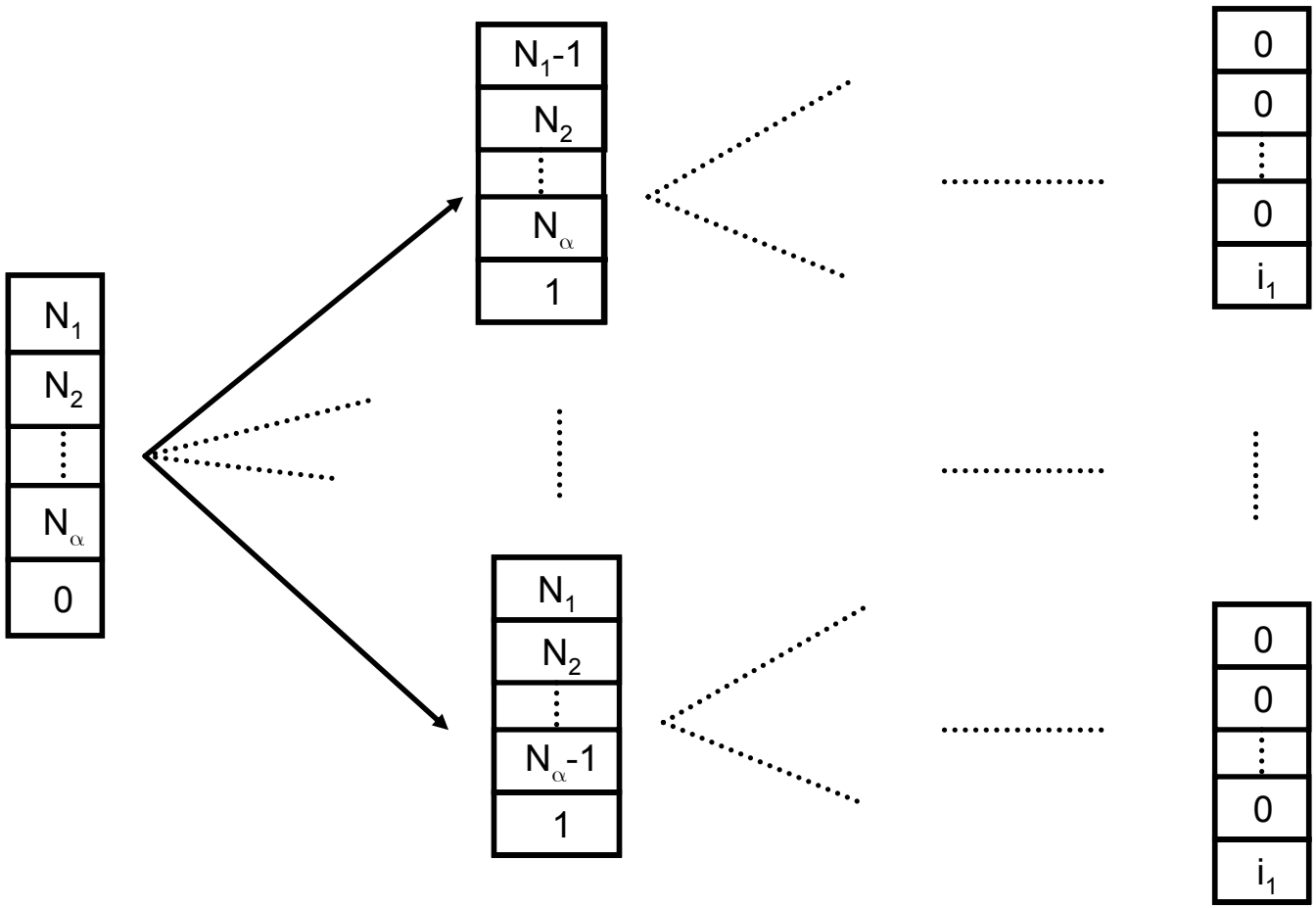
- Wir betrachten wieder den einfachen Fall $\underline{m} = 0, \overline{m} = 1$
- Reduktion auf die Suche eines Weges in einer Arboreszenz von der Wurzel zu einem Blatt
- Beschränkt man die Anzahl der vorkommenden Varianten α durch eine Konstante $\overline{\alpha}$ und bezeichnet man mit N_i das Vorkommen von Variante i in F für $i = 1..\alpha$ so ist

$$\{1,...,N_1\} \times ... \times \{1,...,N_\alpha\} \times \{1,...,\alpha\}$$

  eine Menge mit in α polynomieller Größe
- Fügt man zu dieser Knotenmenge einen Bogen genau dann ein, wenn eine der ersten α Komponenten um eins reduziert wird und die korrespondierenden Vektoren aufeinander folgen dürfen, so erhält man einen Baum (der nicht alle Knoten enthält)



Polynomielle Fälle II

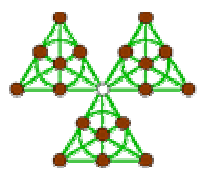


Ein dynamisches Programm

- Seien $\gamma^{k}(...)$ die Knoten der Höhe k, dann kann eine gegebene Instanz mit folgendem dynamischen Programm gelöst werden:

$$
\begin{array}{l}
\textbf{create } \gamma^0\,(N_1, ..., N_\alpha, 0) \\
\textbf{for } k = 1..n \textbf{ do} \\
\quad \textbf{for all } \gamma^{k-1} \textbf{ do} \\
\quad\quad \textbf{for } i = 1..\alpha \textbf{ do} \\
\quad\quad\quad \textbf{if } (\gamma_i^{k-1} > 0 \textbf{ and } (b_i \textit{ and } b_{\gamma_{\alpha+1}^{k-1}} \textit{ do not conflict})) \textbf{ then} \\
\quad\quad\quad\quad \textbf{create } \gamma^k = \gamma^{k-1}(..., \gamma_i^{k-1} - 1, ..., i) \\
\quad\quad\quad\quad \textbf{add arc } \textit{from } \gamma^{k-1} \textit{ to } \gamma^k \\
\quad\quad\quad\quad \textbf{if } (k = n) \textbf{ then return true} \\
\quad k = k + 1 \\
\textbf{return false}
\end{array}
$$

- Komplexität: $O\left(n^{\alpha}\right)$
- Läßt sich leicht auf den allgemeinen Fall übertragen



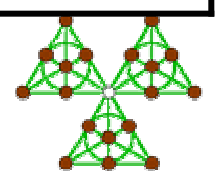
Ein Vergleich

Column Generation + CPLEX : (vergl. Drexl, Kimms: Sequencing JIT Mixed Model Assembly Lines 03/2001)

T \ O	3	5	7	3	5	7
10	0.03	0.06	0.10	0.03	0.03	0.03
15	0.19	0.27	0.62	0.08	0.18	0.18
20	0.88	1.11	2.36	0.31	0.53	1.05
30	9.31	22.17	24.14	2.41	5.97	9.28
40	119.78	124.15	256.86	19.56	28.45	39.19
50	1073.94	2447.47	415.50	355.15	161.90	147.25

Dynamische Programmierung:

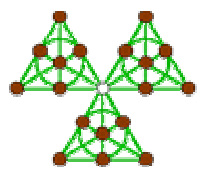
T \ O	3	5	7	3	5	7
10	0.000	0.000	0.000	0.000	0.000	0.000
15	0.000	0.000	0.000	0.000	0.010	0.010
20	0.000	0.010	0.010	0.000	0.070	0.080
30	0.000	X	46.827	2.183	X	X
40	2.123	X	X	X	X	X



Eine parametrisierte Heuristik

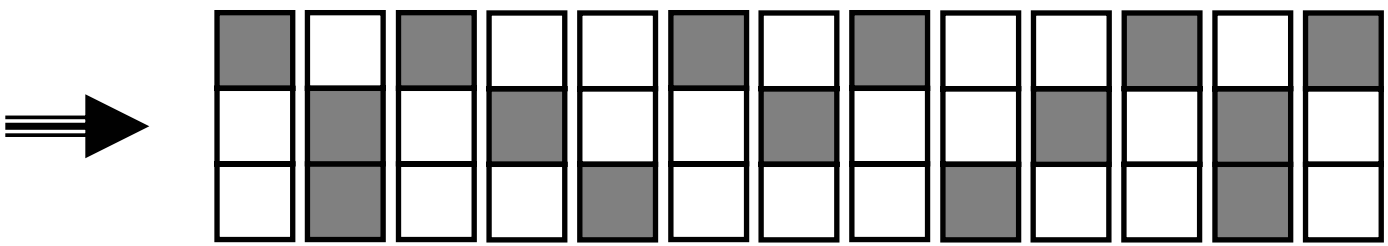
- Anforderungen an eine Sequenz:
  - möglichst wenige Regelverstöße
  - gleichmäßiger Feature-Fluss (Y. Monden: *Toyota Production System*)

- Strategie:
  - baue die Sequenz von links nach rechts auf
  - wähle in jedem Iterationsschritt aus der Menge der Vektoren mit den geringsten Regelverstößen denjenigen aus, der am besten die momentanen Anforderungen an den Feature-Fluss erfüllt
  - benutze diese Startsequenz für eine lokale Verbesserung



Beispiel

6x	3x	2x	2x	Anzahl	MinDist	MaxDist	∅
				6	1	2	1.2
				5	1	2	1.6
				4	2	3	2.25



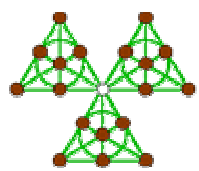
Parametrisierung und Erweiterung

- benutze eine gewichtete euklidische Norm beim Goal Chasing
- führe unterschiedliche Strafen für Regelverletzungen ein

- ersetze die Abstandsrestriktionen durch Restriktionen der Form

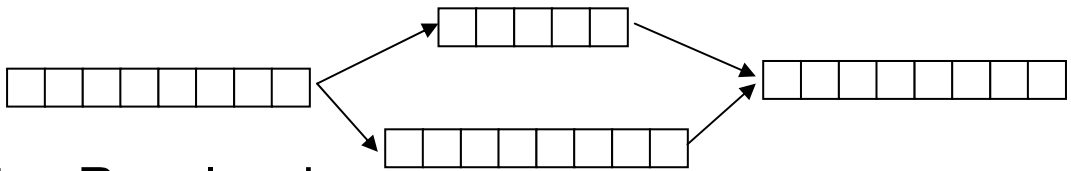
$$\sum_{j=1}^{\beta} W_j \left(X_{j,k} - X_{j,k-L} \right) \leq M \qquad \forall k = L+1..n$$

- dadurch können weitere Restriktionen sowohl im linearen Programm als auch in der Heuristik verwendet werden



Ausblick

- paralleles Zonensystem mit ungleicher Verteilung auf parallele Zonen (Regeln werden hier für jede einzelne Zone aufgestellt)



- erweitertes Regelwerk
  - BANNING : verbieten von Sequenzpositionen und Zonen für einzelne Features
  - CLUSTERING : Die Sequenz wird in y gleiche Teile geteilt und ein Feature darf nur in x davon vorkommen
  - GROUPING : Ein Feature soll nur in Gruppen (mit Mindest- und Maximalgröße) auftauchen und zwischen den Gruppen gilt ein Mindestabstand
  - DELAY : Ein Feature verspätet sich auf einer Zone erwartungsgemäß (trotzdem müssen auf nachfolgenden Zonen die Regeln gelten)