

Modellierung der Waferproduktion

Modellierungsseminar 2008/2009

Bearbeitung

Matthias Bremer, Sebastian Schoß

Kooperationspartner:

microFAB Bremen GmbH
Thomas Stärz



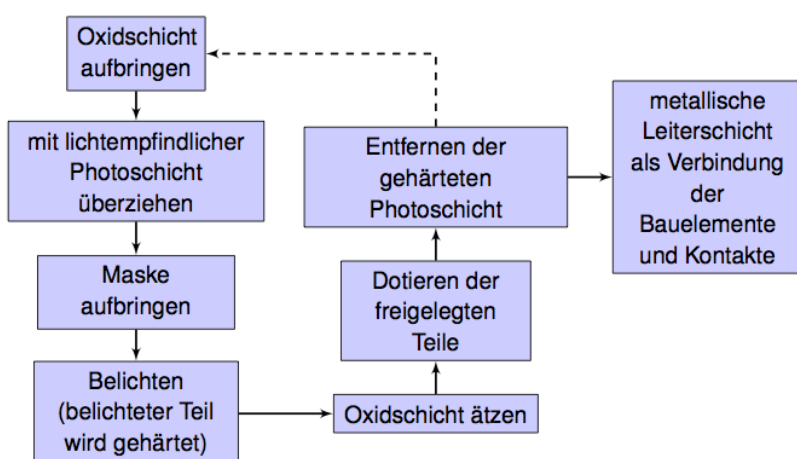
Betreuung:

Zentrum für Technomathematik
Dennis Wassel



Problemstellung

- Chargen passieren zyklisch mehrere Maschinen in vorgegebener Reihenfolge



- versch. Startzeitpunkte (release time) der Chargen
- 10% der Chargen müssen zum „rework“
- ca. 100 Produktionsschritte

⇒ In welcher Reihenfolge sollen die Chargen bearbeitet werden, um die mittlere Produktionsdauer zu minimieren?



Job-Shop-Scheduling-Problem

n Jobs bestehen aus je o_i Operationen:

$$J_i = \{O_{i,1}, O_{i,2}, \dots, O_{i,o_i}\} \quad \forall i \in \{1, \dots, n\}$$

t_0 - frühester allg. Startzeitpunkt

$p_{i,k} \in \mathbb{R}^+$ - Bearbeitungsdauer einer Operation $O_{i,k}$

$Z_{i,k} = [t_{i,k}^s, t_{i,k}^e]$ - Bearbeitungszeit einer Operation
($t_{i,k}^s \geq t_0, t_{i,k}^e = t_{i,k}^s + p_{i,k}$)

$$\Rightarrow \min f \begin{pmatrix} Z_{1,1} & Z_{1,2} & \dots & Z_{1,o_1} \\ Z_{2,1} & Z_{2,2} & \dots & Z_{2,o_2} \\ \vdots & \vdots & \ddots & \vdots \\ Z_{n,1} & Z_{n,2} & \dots & Z_{n,o_n} \end{pmatrix}$$

Bottleneck-Heuristik¹

Algorithmus:

- Sortierung der Maschinen-Warteschlangen nach:
$$PPTR = \frac{\text{verbleibende Prozesszeit der Jobs}}{\text{gesamte Prozesszeit der Jobs}}$$
- Bottleneck-Maschine (BN) ermitteln
- Warteschlangen in drei Gruppen teilen:
G1: Jobs, die durch die BN-Maschine müssen
G2: Jobs, die die BN-Maschine nicht passieren
G3: restliche Jobs
- Status der Bottleneck-Maschine ermitteln
- Scheduling-Algorithmus

Scheduling-Algorithmus:

- BN im Leerlauf und $G1 \neq \emptyset \Rightarrow$ Job mit kleinster PPTR
- BN nicht im Leerlauf und $G1 \neq \emptyset \Rightarrow$ Job aus G2, der **keinen** Leerlauf am BN erzeugt
- Erzeugen alle Jobs aus G2 Leerlauf und $G1 \neq \emptyset \Rightarrow$ Job aus G1, der **keinen** Leerlauf am BN erzeugt
- Erzeugen alle Jobs Leerlauf \Rightarrow Job mit kleinster PPTR
- $G1 = G2 = \emptyset \Rightarrow$ unterteile G3 und gehe zu 1.

Ergebnisse

Software:

- Implementierung in C++ (schnell, objektorientiert)
- Laufzeit: 10 Jobs à 99 Produktionsschritte in **0,3 sek.**
20 Jobs à 99 Produktionsschritte in **0,6 sek.**
- einsetzbar in Produktionsplanung und -analyse

Zeitschritt	Bearbeitungstag und -uhrzeit	Belegungen der Maschinen						
		(** markiert die BN-Maschine in jedem Zeitschritt)						
36	1, 10:0	6	7	5	10	0	2	**
37	1, 10:20	6	7	5	10	0	2	**
38	1, 10:40	4	6	9	8	3	1	**
39	1, 11:0	4	6	9	8	3	1	**
40	1, 11:20	2	4	7	5	10	0	**
41	1, 11:40	2	4	7	5	10	0	**
42	1, 12:0	1	2	6	9	8	3	**
43	1, 12:20	1	2	6	9	8	3	**
44	1, 12:40	0	1	4	7	5	10	**
45	1, 13:0	0	1	4	7	5	10	**
46	1, 13:20	3	0	2	6	9	5	**
47	1, 13:40	3	0	2	6	9	5	**

Programmausgabe (Maschinenbelegungsplan)

¹Sarin, Shikalgar: *Reduction of Average Cycle Time at a Wafer Fabrication Facility*, 2001