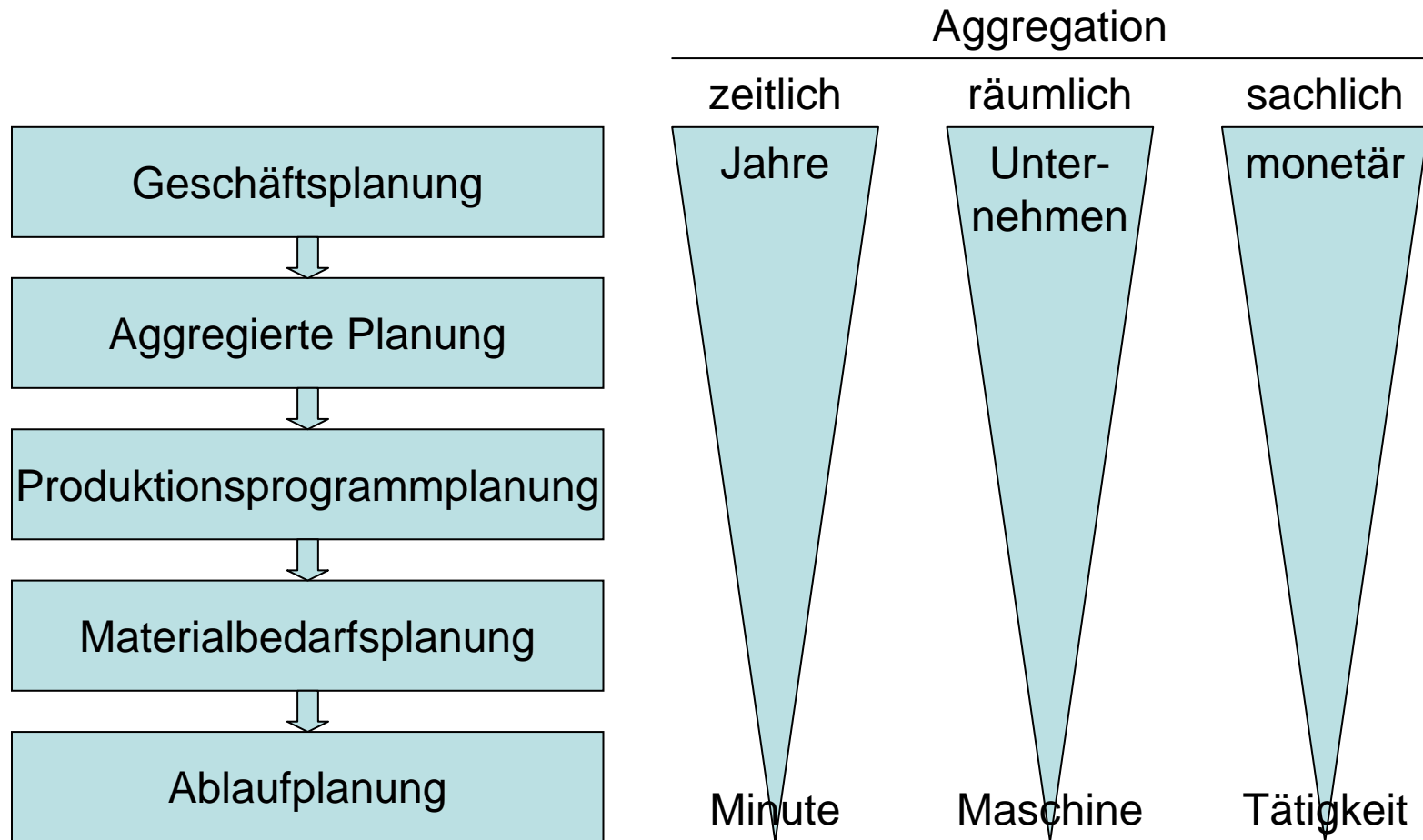


# Aufbau des MRP II



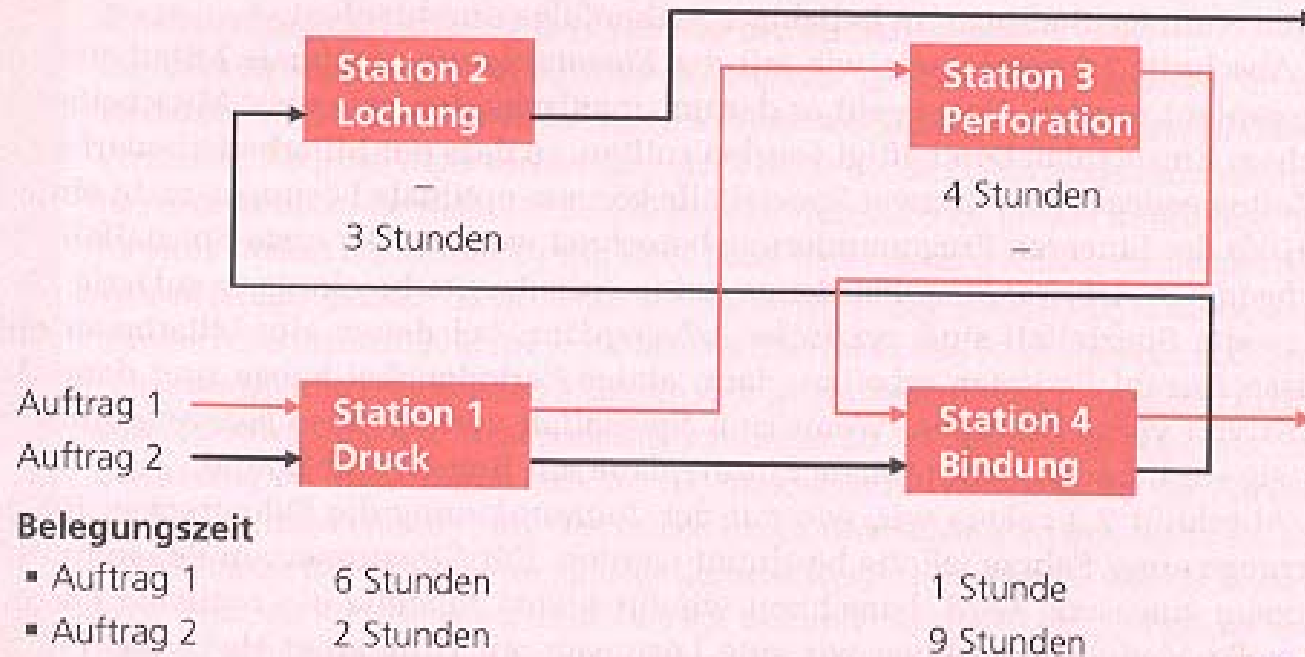
# Szenario



# Beispiel-Druckerei

Abbildung 7.1

Zwei Aufträge in einer Druckerei



# ■ Kennzahlen

# Wichtigste Kennzahlen

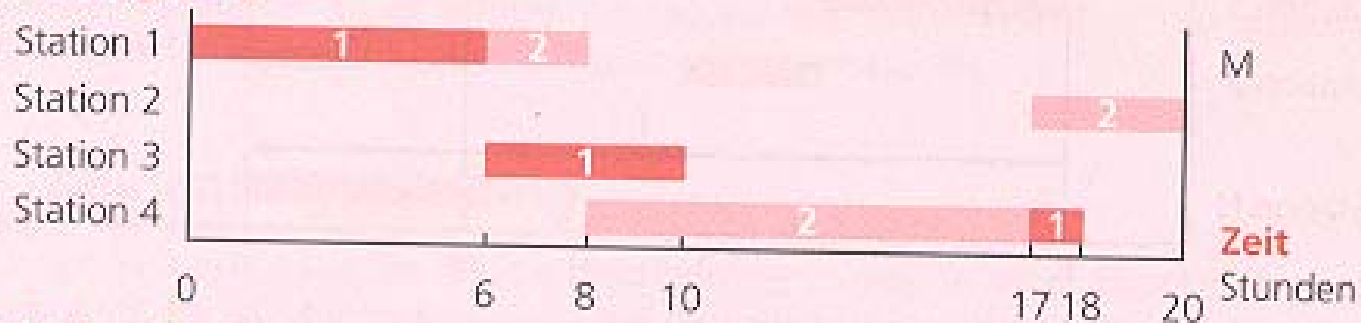
- Fertigstellungszeit
  - Auftrags-bezogen
  - Ende letzter Schritt
- Durchlaufzeit
  - Auftrags-bezogen
  - Beginn erster Schritt bis Ende letzter Schritt
- Verspätung
- Gesamtbearbeitungszeit

# Ablaufalternativen

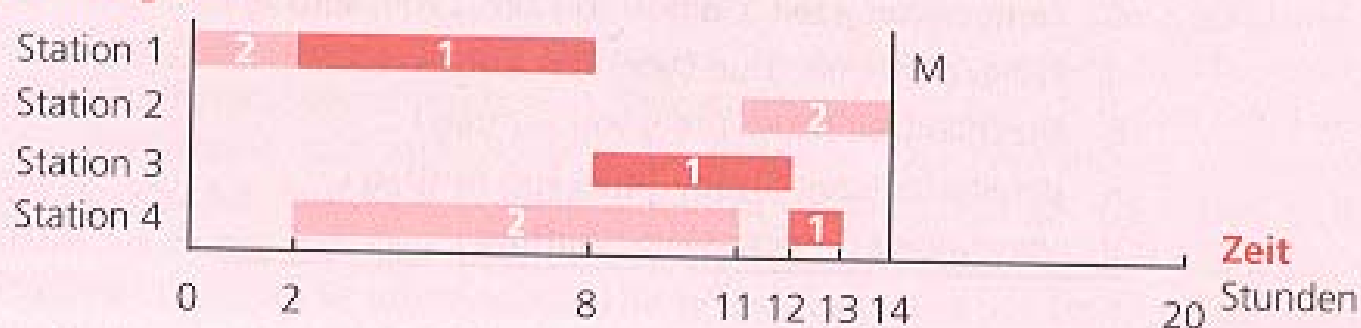
Abbildung 7.3

Gantt-Diagramme für zwei mögliche Reihenfolgen

**A Auftrag 1 → 2**



**B Auftrag 2 → 1**



# Aggregation von Kennzahlen

**Tabelle 7.1**

## Aggregierte Kennzahlen

Kennzahl	Reihenfolge	Auftrag		Durchschnitt Stunden	Maximum Stunden	Anzahl
		1	2			
		Stunden	Stunden			
Fertigstellungszeit	A	18	20	19,0	20	–
	B	13	14	13,5	14	–
Durchlaufzeit	A	18	20	19,0	20	–
	B	13	14	13,5	14	–
Verspätung	A	3	0	1,5	3	1
	B	0	0	0,0	0	0

# ■ Prioritätsregeln



# Optimierung der Reihenfolge

- Ziel: Auswahl der bzgl. der benutzten Kennzahl besten Reihenfolge
- Komplexitätsproblem
  - I Aufträge
  - Beispiel: I=100 Aufträge,  $10^{158}$  Reihenfolgen
- einfache Prioritätsregeln
  - geringer Aufwand bei der Reihenfolgebestimmung
  - akzeptierbare Lösungsqualität

# Prioritätsregeln

- Ankunftsreihenfolge
  - First Come First Serve (FCFS)
  - Beispiel: Supermarktkasse
  - “gerechte” Regel
- Kürzeste Belegungszeit
  - Shortest Processing Time (SPT)
  - flexible Umplanung bei neuen Aufträgen möglich
- Frühester Fertigstellungstermin
  - Earliest Due Date (EDD)
  - Sortierung nach Fälligkeitszeiten

# Szenario-Beschreibung

- eine Arbeitsstation
- zum Zeitpunkt  $t=0$  stehen fünf Aufträge bereit
- Auftrag  $i=1$  war zuerst da, dann Auftrag  $i=2$ , ...
- Terminieren Sie die Aufträge mit
  - FCFS
  - SPT
  - EDD

Auftrag	Belegungszeit	Fälligkeitstermin
$i$	$t_i$	$d_i$
1	4	8
2	7	12
3	1	4
4	6	13
5	3	14

# Terminierung mit FCFS

Auftrag	Belegungszeit	Fälligkeits-termin	Fertigstellungs-zeit	Verspätung
$i$	$t_i$	$d_i$	$C_i$	$V_i$
1	4	8	4	0
2	7	12	11	0
3	1	4	12	8
4	6	13	18	5
5	3	14	21	7
			66	

# Terminierung mit SPT

Auftrag	Belegungszeit	Fälligkeits-termin	Fertigstellungs-zeit	Verspätung
$i$	$t_i$	$d_i$	$C_i$	$V_i$
3	1	4	1	0
5	3	14	4	0
1	4	8	8	0
4	6	13	14	1
2	7	12	21	9
			48	



# Terminierung mit EDD

Auftrag	Belegungszeit	Fälligkeits-termin	Fertigstellungs-zeit	Verspätung
$i$	$t_i$	$d_i$	$C_i$	$V_i$
3	1	4	1	0
1	4	8	5	0
2	7	12	12	0
4	6	13	18	5
5	3	14	21	7
			57	



# Vergleich der Verfahren

- SPT minimiert durchschnittliche Fertigungsstellungszeit
- EDD minimiert maximale Verspätung
- Wie kann die Anzahl der verspäteten Aufträge minimiert werden?
  - Moore's Algorithmus

# Moore's Algorithmus

- (1) Initialisierung des Algorithmus
  - Sortierung der Aufträge nach EDD-Regel
  - Reihenfolge R, leere Menge J (für verspätete Aufträge)
- (2) Auswahl nächster Auftrag
  - falls kein Auftrag verspätet: gehe zu (4)
  - wähle ersten verspäteten Auftrag i in R
- (3) Entfernung Auftrag
  - wähle Auftrag j, der
    - bezüglich R nicht hinter i liegt
    - die längste Bearbeitungszeit hat
  - entferne j aus R und füge in in J ein
  - gehe zu (2)
- (4) Beende den Algorithmus
  - Bearbeite zunächst die Aufträge in der durch den Rest von R vorgegebenen Reihenfolge
  - Bearbeite dann die Aufträge aus J in einer beliebigen Reihenfolge



# Anwendung von Moore's Verfahren

- (1) Initialisierung
  - $R = (3,1,2,4,5)$
  - $J = \{\}$
- (2) Auswahl des Referenz-Auftrags
  - erster verspäteter Auftrag in R:  $i=4$ , also gehe zu (3)
- (3) Entfernung eines Auftrags aus R
  - längster Auftrag (nicht nach Auftrag 4 in R):  $j=2$
  - aktualisiere R:  $R=(3,1,4,5)$
  - aktualisiere J:  $J=\{2\}$
  - berechne mit EDD unter Verwendung von R die neuen Zeiten
- (2a) Auswahl des nächsten Referenz-Auftrags
  - es gibt keinen verspäteten Auftrag mehr, gehe zu (4)
- (4) die Bearbeitungsreihenfolge lautet  $(R,J) = (3,1,4,5,2)$ 
  - nur Auftrag 2 ist verspätet.

# Terminierung mit Moore

Auftrag	Belegungszeit	Fälligkeits-termin	Fertigstellungs-zeit	Verspätung
$i$	$t_i$	$d_i$	$C_i$	$V_i$
3	1	4	1	0
1	4	8	5	0
4	6	13	11	0
5	3	14	14	0
2	7	12	21	9
			52	

## ■ Zwei-Stationen-Szenarien

# Szenarien

## ■ Fließfertigung

- Maschinen-Reihenfolge für alle Aufträge gleich
- ungetaktete Fließfertigung (Flow Shop)
  - Aufträge können zeit-unabhängig voneinander gestartet werden
- getaktete Fließfertigung (Assembly Line)
  - nur zu bestimmten Zeitpunkten können an allen Stationen Aufträge gleichzeitig gestartet werden

## ■ Werkstattfertigung (Job Shop)

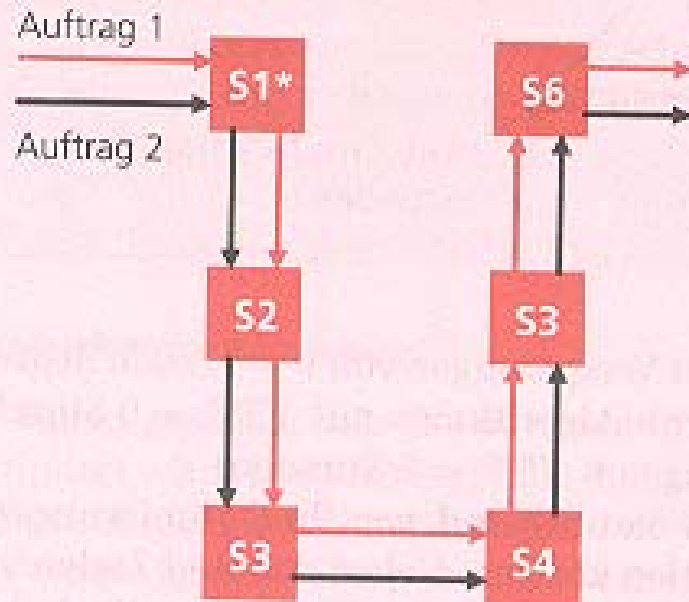
- jeder Auftrag hat eigene Maschinen-Reihenfolge

# Beispiel Jobshop & Flowshop

Abbildung 7.6

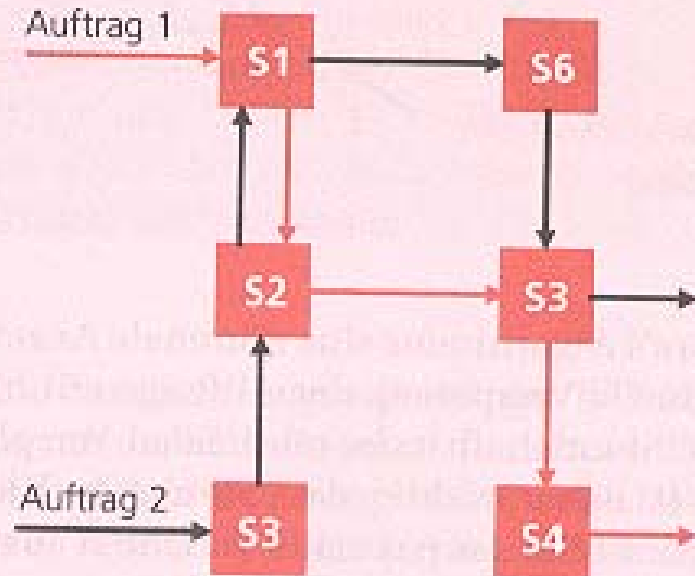
## Fließfertigung und Werkstattfertigung

### Fließfertigung



\* S = Station

### Werkstattfertigung



# Zwei-Stationen Flowshop

Abbildung 7.7

Belegungszeiten mehrerer Aufträge auf zwei Stationen



# Auftrags-Permutationen

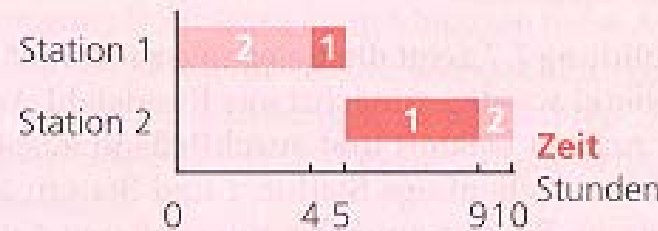
Abbildung 7.8

Gantt-Diagramme vier möglicher Bearbeitungsreihenfolgen

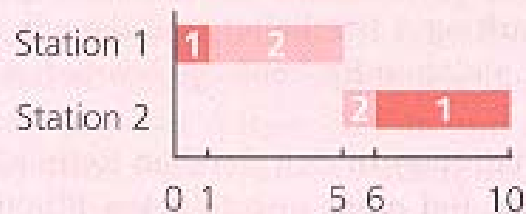
**A** Station 1: 1 → 2  
Station 2: 1 → 2



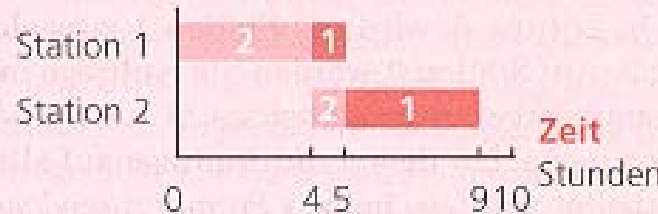
**B** Station 1: 2 → 1  
Station 2: 1 → 2



**C** Station 1: 1 → 2  
Station 2: 2 → 1



**D** Station 1: 2 → 1  
Station 2: 2 → 1



- Es reicht, die Einlastungsreihenfolge in Station 1 festzulegen!

# Johnson-Algorithmus

- Initialisierung

- $J_1$ : Menge aller Aufträge, deren Belegungszeit auf Maschine 1 kleiner oder gleich der Belegungszeit auf Maschine 2 sind
- $J_2$ : Menge aller Aufträge, deren Belegungszeit auf Maschine 1 größer als auf Maschine 2 ist

- Sortierung der Aufträge

- in  $J_1$ : Sortierung nach aufsteigender Belegungszeit auf Station 1
- in  $J_2$ : Sortierung nach absteigender Belegungszeit auf Station 2

- Verkettung der Reihenfolgen

- die Verkettung der beiden Reihenfolgen (erst Station 1) ist die optimale Lösung.



# Anwendung Johnson-Algorithmus

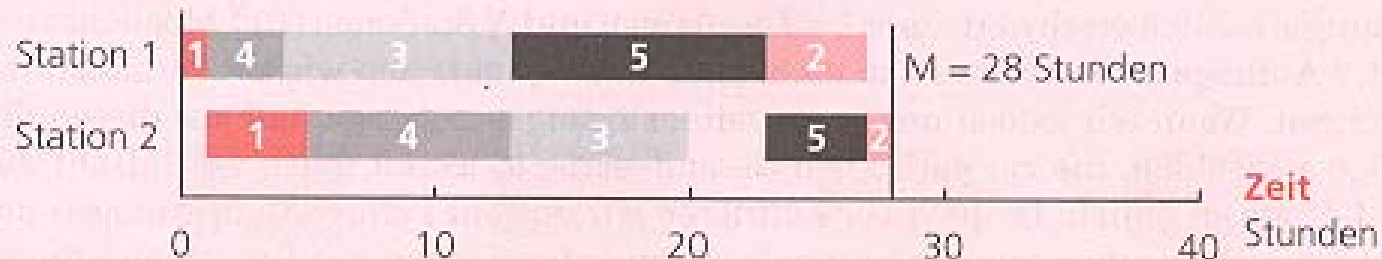
- Initialisierung
  - $J_1 = \{1,4\}$
  - $J_2 = \{2,3,5\}$
- Reihenfolgebildung
  - $j_1 = (1,4)$
  - $j_2 = (3,5,2)$
- Verkettung
  - $j = (1,4,3,5,2)$  ist optimale Permutation

# Vergleich FCFS und Johnson

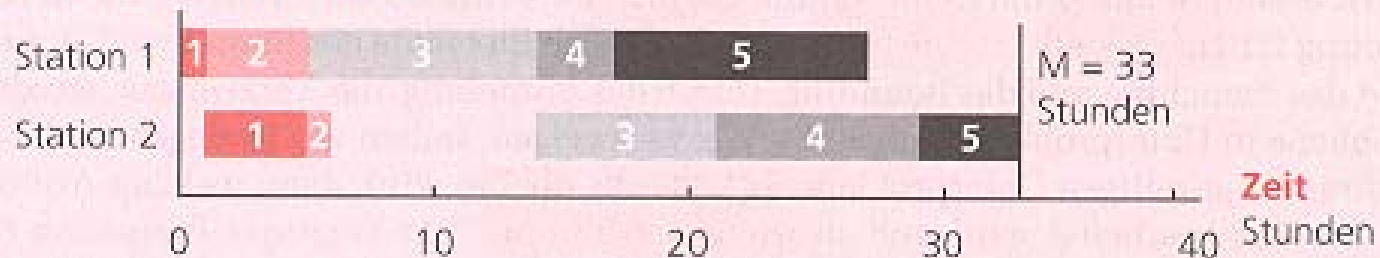
Abbildung 7.9

Vergleich der optimalen Reihenfolge mit der FCFS-Lösung

## Johnson-Algorithmus



## First-Come-First-Serve



# Zusammenfassung

- Problemstellungen
  - Fließ- und Werkstattfertigung
- Verfahren
  - Prioritätsregeln
  - Moore's Algorithmus
  - Johnson-Verfahren
- Zur Vertiefung
  - Abschnitt 7.1 in Thonemann (2005)
  - Übungsaufgaben 1-4 ("Hausaufgabe": Aufgabe 2)