Shop models: General Introduction -1-

Remark: Consider non preemptive problems with regular objectives

Notation Shop Problems:

- m machines, n jobs $1, \dots, n$
- operations $O = \{(i,j) | j = 1, \dots, n;\ i \in M^j \subset M := \{1, \dots, m\}\}$ with processing times p_{ij}
- M^j is the set of machines where job j has to be processed on
- $PREC$ specifies the precedence constraints on the operations

Shop models: General Introduction

Notation Shop Problems:

- m machines, n jobs $1, \dots, n$
- operations $O = \{(i,j) | j = 1, \dots, n;\ i \in M^j \subset M := \{1, \dots, m\}\}$ with processing times p_{ij}
- M^j is the set of machines where job j has to be processed on
- $PREC$ specifies the precedence constraints on the operations
- Flow shop: $M^j = M$ and $PREC = \{(i,j) \to (i+1,j) | i = 1, \dots, m-1; j = 1, \dots, n\}$

Shop models: General Introduction

Notation Shop Problems:

- m machines, n jobs $1, \ldots, n$
- operations $O = \{(i, j) | j = 1, \ldots, n;\ i \in M^j \subset M := \{1, \ldots, m\}\}$ with processing times p_{ij}
- M^j is the set of machines where job j has to be processed on
- $PREC$ specifies the precedence constraints on the operations
- Flow shop: $M^j = M$ and $PREC = \{(i, j) \to (i + 1, j) | i = 1, \ldots, m - 1; j = 1, \ldots, n\}$
- Open shop: $M^j = M$ and $PREC = \emptyset$
- Job shop: $PREC$ contain a chain $(i_1, j) \to \ldots, \to (i_{|M^j|}, j)$ for each j

Shop models: General Introduction

Disjunctive Formulation of the constraints

- C_{ij} denotes completion time of operation (i, j)
- $PREC$ have to be respected:

Shop models: General Introduction -2-

Disjunctive Formulation of the constraints

- C_{ij} denotes completion time of operation (i, j)
- $PREC$ have to be respected:

$$C_{ij} - p_{ij} \geq C_{kl} \quad \text{for all } (k, l) \to (i, j) \in PREC$$

- no two operations of the same job are processed at the same time:

Shop models: General Introduction -2-

<u>Disjunctive Formulation of the constraints</u>

- C_{ij} denotes completion time of operation (i, j)
- $PREC$ have to be respected:

$$C_{ij} - p_{ij} \geq C_{kl} \quad \text{for all } (k, l) \rightarrow (i, j) \in PREC$$

- no two operations of the same job are processed at the same time:

$$C_{ij} - p_{ij} \geq C_{kj} \text{ or } C_{kj} - p_{kj} \geq C_{ij} \quad \text{for all } i, k \in M^j; i \neq k$$

- no two operations are processed jointly on the same machine:

Shop models: General Introduction

Disjunctive Formulation of the constraints

- $PREC$ have to be respected:

$$C_{ij} - p_{ij} \geq C_{kl} \quad \text{for all } (k,l) \rightarrow (i,j) \in PREC$$

- no two operations of the same job are processed at the same time:

$$C_{ij} - p_{ij} \geq C_{kj} \text{ or } C_{kj} - p_{kj} \geq C_{ij} \quad \text{for all } i,k \in M^j; i \neq k$$

- no two operations are processed jointly on the same machine:

$$C_{ij} - p_{ij} \geq C_{il} \text{ or } C_{il} - p_{il} \geq C_{ij} \quad \text{for all } (i,j),(i,l) \in O; j \neq l$$

- $C_{ij} - p_{ij} \geq 0$
- the 'or' constraints are called disjunctive constraints
- some of the disjunctive constraints are 'overruled' by the $PREC$ constraints

Shop models: General Introduction

Disjunctive Formulation - makes pan objective

$$
\begin{array}{ll}
\min C_{max} & \\
s.t. & \\
C_{max} \geq C_{ij} & (i,j) \in O \\
C_{ij} - p_{ij} \geq C_{kl} & (k,l) \rightarrow (i,j) \in PREC \\
C_{ij} - p_{ij} \geq C_{kj} \text{ or } C_{kj} - p_{kj} \geq C_{ij} & i,k \in M^j; i \neq k \\
C_{ij} - p_{ij} \geq C_{il} \text{ or } C_{il} - p_{il} \geq C_{ij} & (i,j),(i,l) \in O; j \neq l \\
C_{ij} - p_{ij} \geq 0 & (i,j) \in O
\end{array}
$$

Shop models: General Introduction

Disjunctive Formulation - sum objective

$$
\begin{array}{ll}
\min \sum w_j L_j & \\
s.t. & \\
L_j \geq C_{ij} - d_j & (i,j) \in O \\
C_{ij} - p_{ij} \geq C_{kl} & (k,l) \to (i,j) \in PREC \\
C_{ij} - p_{ij} \geq C_{kj} \text{ or } C_{kj} - p_{kj} \geq C_{ij} & i,k \in M^j; i \neq k \\
C_{ij} - p_{ij} \geq C_{il} \text{ or } C_{il} - p_{il} \geq C_{ij} & (i,j),(i,l) \in O; j \neq l \\
C_{ij} - p_{ij} \geq 0 & (i,j) \in O
\end{array}
$$

Remark:

- also other constraints, like e.g. release dates, can be incorporated
- the disjunctive constraints make the problem hard (lead to an ILP formulation)

Shop models: General Introduction

Disjunctive Graph Formulation

- graph representation used to represent instances and solutions of shop problems
- can be applied for regular objectives only

Shop models: General Introduction

Disjunctive Graph $G = (V, C, D)$

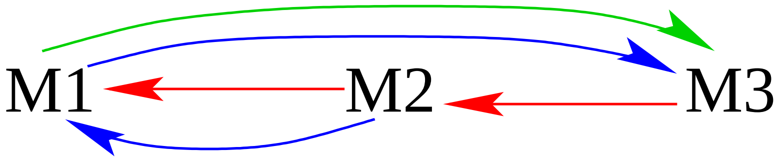
- V set of vertices representing the operations O
- a vertex is labeled by the corresponding processing time;
- Additionally, a source node 0 and a sink node $*$ belong to V; their weights are 0
- C set of conjunctive arcs reflecting the precedence constraints: for each $(k, l) \to (i, j) \in PREC$ a directed arc belongs to C
- additionally $0 \to O$ and $O \to *$ are added to C
- D set of disjunctive arcs representing 'conflicting' operations: between each pair of operations belonging to the same job or to be processed on the same machine, for which no order follows from $PREC$, an undirected arc belongs to D

Shop models: General Introduction

Disjunctive Graph - Example Job Shop

- Data: 3 jobs, 3 machines;

Jobs:	1	$(3,1) \longrightarrow (2,1) \longrightarrow (1,1)$	$p_{31} = 4, p_{21} = 2, p_{11} = 1$
	2	$(1,2) \longrightarrow (3,2)$	$p_{12} = 3, p_{32} = 3$
	3	$(2,3) \longrightarrow (1,3) \longrightarrow (3,3)$	$p_{23} = 2, p_{13} = 4, p_{33} = 1$

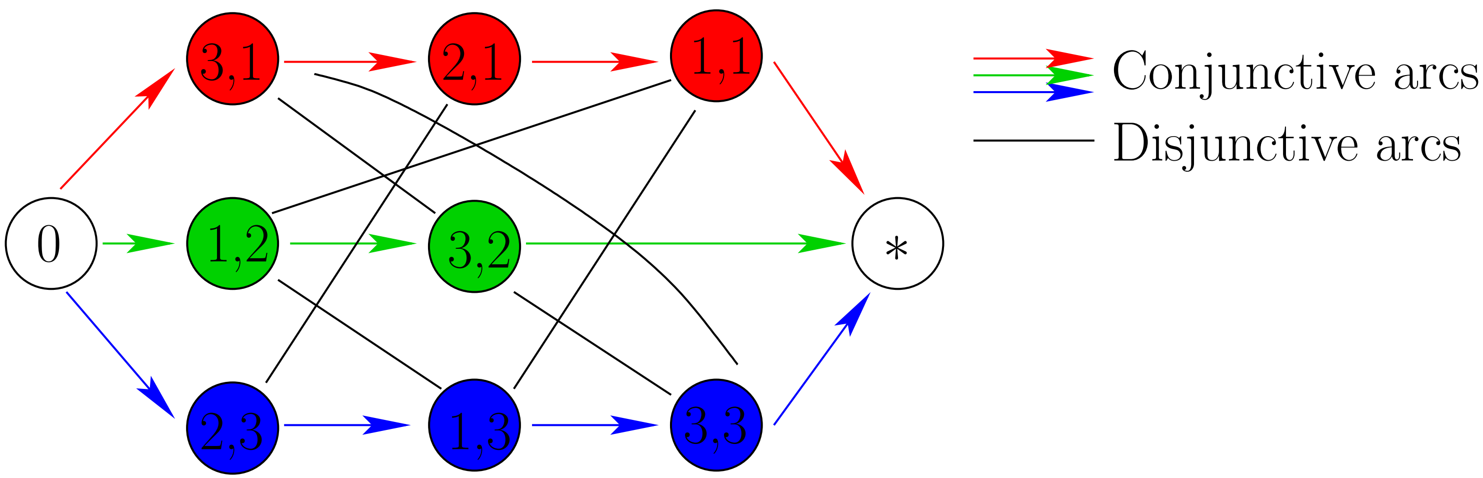


Shop models: General Introduction

Disjunctive Graph - Example Job Shop

Jobs: 1	(red)	$(3,1) \longrightarrow (2,1) \longrightarrow (1,1)$	$p_{31} = 4, p_{21} = 2, p_{11} = 1$
2	(green)	$(1,2) \longrightarrow (3,2)$	$p_{12} = 3, p_{32} = 3$
3	(blue)	$(2,3) \longrightarrow (1,3) \longrightarrow (3,3)$	$p_{23} = 2, p_{13} = 4, p_{33} = 1$

- Graph:



Shop models: General Introduction

Disjunctive Graph - Example Open Shop

- Data: 3 jobs, 3 machines;

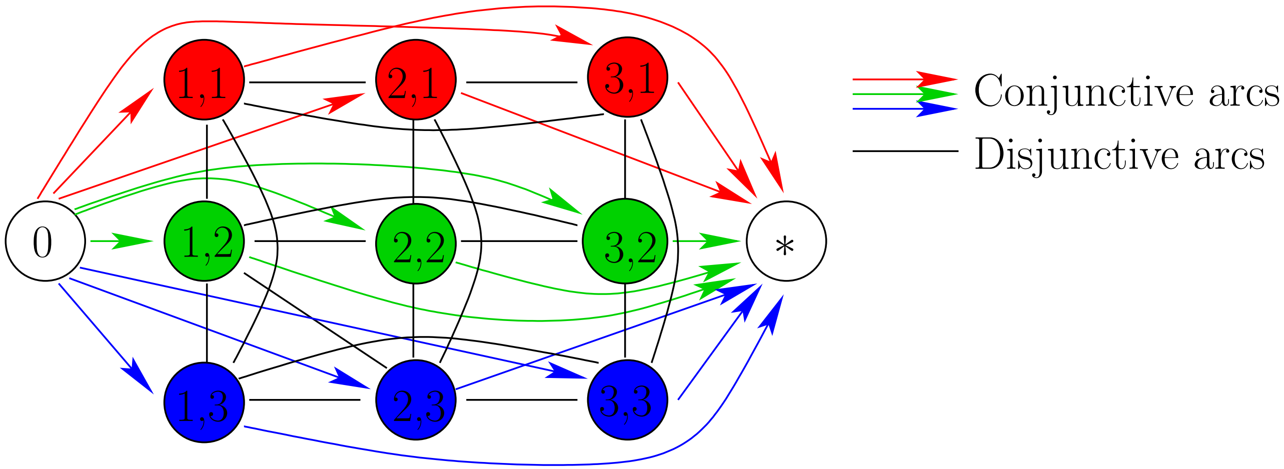
Jobs: 1	$(1,1),(2,1),(3,1)$	$p_{11}=4, p_{21}=2, p_{31}=1$
2	$(1,2),(2,2),(3,2)$	$p_{12}=3, p_{22}=1, p_{32}=3$
3	$(1,3),(2,3),(3,3)$	$p_{13}=2, p_{23}=4, p_{33}=1$

Shop models: General Introduction

Disjunctive Graph - Example Open Shop

Jobs: 1	red	$(1,1),(2,1),(3,1)$	$p_{11}=4, p_{21}=2, p_{31}=1$
2	green	$(1,2),(2,2),(3,2)$	$p_{12}=3, p_{22}=1, p_{32}=3$
3	blue	$(1,3),(2,3),(3,3)$	$p_{13}=2, p_{23}=4, p_{33}=1$

- Graph:



Shop models: General Introduction

Disjunctive Graph - Selection

- basic scheduling decision for shop problems (see disj. formulation): define an ordering for operations connected by a disjunctive arc
- $\rightarrow$ turn the undirected disjunctive arc into a directed arc
- selection S: a set of directed disjunctive arcs (i.e. $S \subset D$ together with a chosen direction for each $a \in S$)
- disjunctive arcs which have been directed are called 'fixed'
- a selection is a complete selection if
  - each disjunctive arc has been fixed
  - the graph $G(S) = (V, C \cup S)$ is acyclic

Shop models: General Introduction

Selection - Remarks

- a feasible schedule induces a complete selection
- a complete selection leads to sequences in which operations have to be processed on machines
- a complete selection leads to sequences in which operations of a job have to be processed
- Does each complete selection leads to a feasible schedule?

Shop models: General Introduction

Calculate a Schedule for a Complete Selection S

- calculated longest paths from 0 to all other vertices in $G(S)$
- Technical description:
  - length of a path $i_1, i_2, \ldots, i_r$ = sum of the weights of the vertices $i_1, i_2, \ldots, i_r$
  - calculate length l_{ij} of the longest path from 0 to (i, j) (using e.g. Dijkstra)
  - start operation (i, j) at time $l_{ij} - p_{ij}$ (i.e. $C_{ij} = l_{ij}$)
  - the length of a longest path from 0 to $*$ (such paths are called **critical paths**) is equal to the makespan of the schedule
- resulting schedule is the semiactive schedule which respects all precedence given by C and S

Shop models: General Introduction

Reformulation Shop Problem

find a complete selection for which the corresponding schedule minimizes the given (regular) objective function

Flow Shop models:

Makespan Minimization

- Lemma: For problem $F||C_{max}$ an optimal schedule exists with
  - the job sequence on the first two machines is the same
  - the job sequence on the last two machines is the same

  (Proof as Exercise)
- Consequence: For $F2||C_{max}$ and $F3||C_{max}$ an optimal solution exists which is a permutation solution
- For $Fm||C_{max}$, $m \geq 4$, instances exist where no optimal solution exists which is a permutation solution
  (Exercise)

Flow Shop models: -2-

Problem $F2||C_{max}$

- solution can be described by a sequence π
- problem was solved by Johnson in 1954

Johnson's Algorithm:

1. L = set of jobs with $p_{1j} < p_{2j}$;
2. R = set of remaining jobs;
3. sort L by SPT w.r.t. the processing times on first machine (p_{1j})
4. sort R by LPT w.r.t. the processing times on second machine (p_{2j})
5. sequence L before R (i.e. $\pi = (L, R)$ where L and R are sorted)

Flow Shop models:

Example solution problem $F2||C_{max}$

- $n = 5; p = \begin{pmatrix} 4 & 3 & 3 & 1 & 8 \\ 8 & 3 & 4 & 4 & 7 \end{pmatrix}$

Flow Shop models: -3-

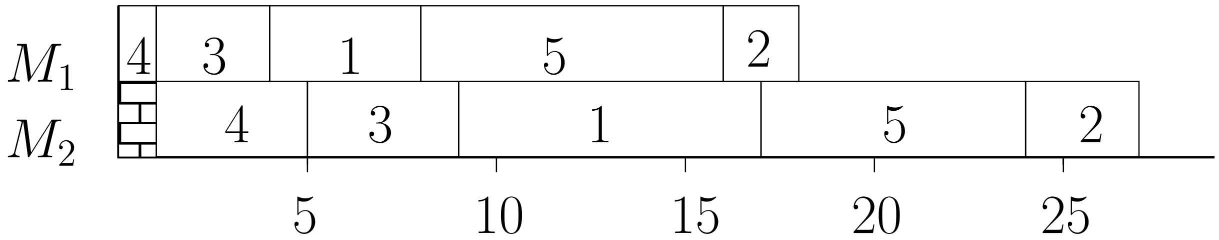
Example solution problem $F2||C_{max}$

- $n = 5; p = \begin{pmatrix} 4 & 3 & 3 & 1 & 8 \\ 8 & 3 & 4 & 4 & 7 \end{pmatrix}$
- $L = \{1, 3, 4\};\ R = \{2, 5\}$
- sorting leads to $L = \{4, 3, 1\};\ R = \{5, 2\}$

Flow Shop models:

Example solution problem $F2||C_{max}$

- $n = 5; p = \begin{pmatrix} 4 & 3 & 3 & 1 & 8 \\ 8 & 3 & 4 & 4 & 7 \end{pmatrix}$
- $L = \{1, 3, 4\};\ R = \{2, 5\}$
- sorting leads to $L = \{4, 3, 1\};\ R = \{5, 2\}$
- $\pi = (4, 3, 1, 5, 2)$



Flow Shop models:

Problem $F2||C_{max}$

- Lemma 1: If
$$\min\{p_{1i}, p_{2j}\} < \min\{p_{2i}, p_{1j}\}$$
then job i is sequenced before job j by Johnson's algorithm.
- Lemma 2: If job j is scheduled immediately after job i and
$$\min\{p_{1j}, p_{2i}\} < \min\{p_{2j}, p_{1i}\}$$
then swapping job i and j does not increase C_{max}.
- Theorem: Johnson's algorithm solves problem $F2||C_{max}$ optimal in $O(n\log(n))$ time.

(Proofs on the board)

Flow Shop models:

Problem $F3||C_{max}$

- $F3||C_{max}$ is NP-hard in the strong sense
- Reduction using 3-PARTITION
- Proof on the board