Implementation exercises for the course Heuristic Optimization

Christian L. Camacho-Villalón [1]
ccamacho@ulb.ac.be

IRIDIA, CoDE, ULB

March 2, 2023

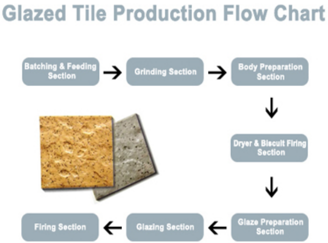
[1] Slides based on Franco Mascia's and Federico Pagnozzi's previous work.

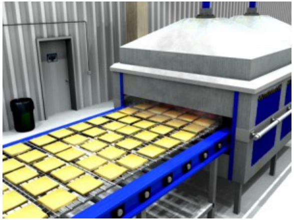
Exercise 1.1: Iterative Improvement for the PFSP

Goal: implement perturbative local search algorithms for the PFSP

1. Permutation Flowshop Scheduling Problem (PFSP)
2. First-improvement and best-improvement
3. Transpose, exchange and insert neighborhoods
4. Random initialization vs. simplified RZ heuristic
5. Statistical empirical analysis

The Permutation Flowshop Scheduling Problem (1/6)

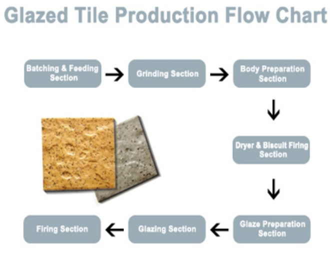


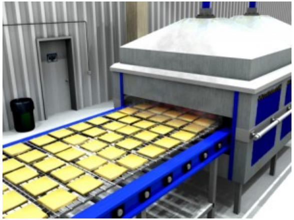


Example in ceramic tile production

- Tiles need several processing steps with different machines
- Tiles of different type require specific processing times for each machine
- Goal: find a schedule of the jobs that minimizes an objective function (makespan or total completion time)

The Permutation Flowshop Scheduling Problem (1/6)

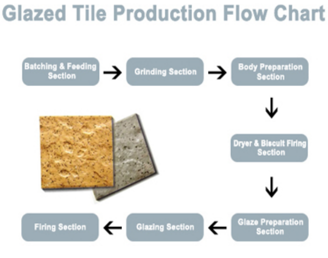


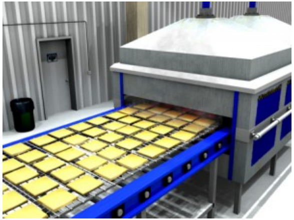


Example in ceramic tile production

- Tiles need several processing steps with different machines
- Tiles of different type require specific processing times for each machine
- Goal: find a schedule of the jobs that minimizes an objective function (makespan or total completion time)

The Permutation Flowshop Scheduling Problem (1/6)

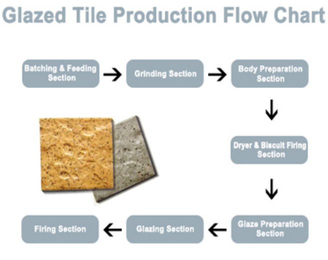


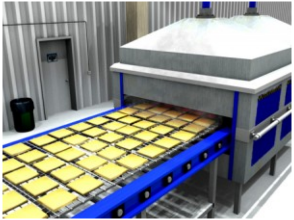


Example in ceramic tile production

- Tiles need several processing steps with different machines
- Tiles of different type require specific processing times for each machine
- Goal: find a schedule of the jobs that minimizes an objective function (makespan or total completion time)

The Permutation Flowshop Scheduling Problem (1/6)





Example in ceramic tile production

- Tiles need several processing steps with different machines
- Tiles of different type require specific processing times for each machine
- Goal: find a schedule of the jobs that minimizes an objective function (makespan or total completion time)

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (2/6)

Flowshop scheduling

- Several scheduling problems have been proposed with different formulations and constraints.
- In permutation flowshop problems:
  - jobs are composed of operations to be executed on several machines
  - all jobs pass through the machines in the same order
  - all jobs available at time zero
  - pre-emption is not allowed
  - each operation has to be performed on a specific machine
  - each job at most on one machine at a time
  - each machine at most one job at a time

Permutation flowshop scheduling problem with weighted tardiness (PFSP-WT)

- Jobs pass trough all machines in the same order (FCFS queues)
- There are infinite buffers between machines
- Constraints: *due dates*, *importance*

The Permutation Flowshop Scheduling Problem (3/6)

Given

A set of n jobs $J_1, \ldots, J_n$ jobs, where each job J_i consists of m operations $o_{i1}, \ldots, o_{im}$ performed on $M_1, \ldots, M_m$ machines (in that order), with processing time p_{ij} for operation o_{ij}.

Due dates

each job J_i has a due date d_i and a priority w_i. Let C_{ij} be the completion time of job J_i on machine M_j, and C_i the completion time of job J_i on the last machine.

The Permutation Flowshop Scheduling Problem (3/6)

Given

A set of n jobs $J_1, \ldots, J_n$ jobs, where each job J_i consists of m operations $o_{i1}, \ldots, o_{im}$ performed on $M_1, \ldots, M_m$ machines (in that order), with processing time p_{ij} for operation o_{ij}.

Due dates

each job J_i has a due date d_i and a priority w_i. Let C_{ij} be the completion time of job J_i on machine M_j, and C_i the completion time of job J_i on the last machine.

The Permutation Flowshop Scheduling Problem (3/6)

Given

A set of n jobs $J_1, \ldots, J_n$ jobs, where each job J_i consists of m operations $o_{i1}, \ldots, o_{im}$ performed on $M_1, \ldots, M_m$ machines (in that order), with processing time p_{ij} for operation o_{ij}.

Due dates

each job J_i has a due date d_i and a priority w_i. Let C_{ij} be the completion time of job J_i on machine M_j, and C_i the completion time of job J_i on the last machine.

The Permutation Flowshop Scheduling Problem (4/6)

Objective

Find a permutation (i.e., a schedule) π that minimizes the sum of the *total weighted tardiness*:

$$\sum_{i=1}^{n} w_i \cdot T_i,$$

where $T_i = \max\{C_i - d_i, 0\}$ is the tardiness of completing job i.

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \dots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \dots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \dots\ n, j = 2, \dots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j = 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k = 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k = 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j = 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k = 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k = 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots m \\
C_{\pi(k)1} &= \sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j = 1, \ldots m\\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k = 1, \ldots n\\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k = 2, \ldots n, j = 2, \ldots m\\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \dots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \dots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \dots n, j = 2, \dots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \dots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \dots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \dots\ n, j = 2, \dots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1,\ldots\ m\\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1,\ldots\ n\\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2,\ldots\ n, j = 2,\ldots\ m\\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

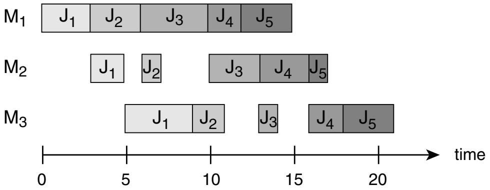
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

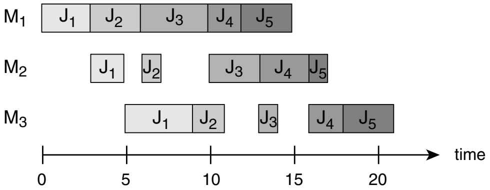
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \dots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \dots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \dots\ n, j = 2, \dots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

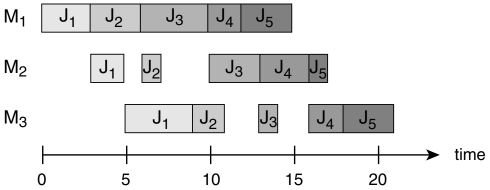
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \dots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \dots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \dots\ n, j = 2, \dots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

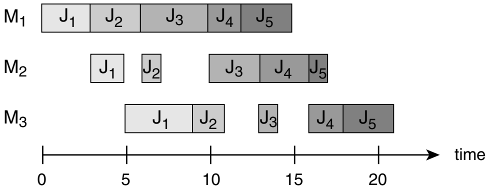
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

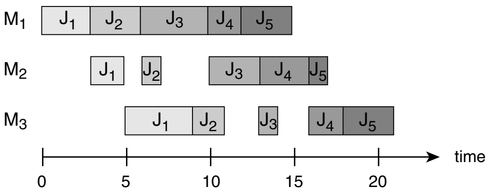
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

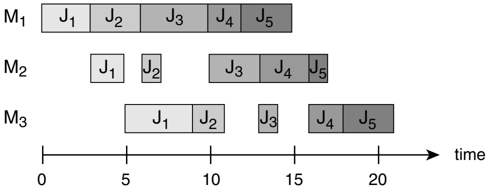
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

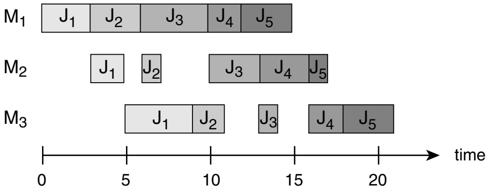
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

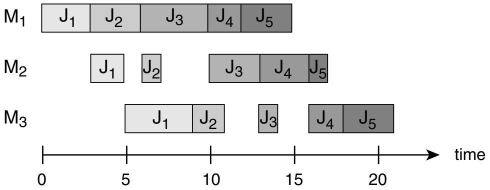
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots\ m\\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots\ n\\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots\ n, j = 2, \ldots\ m\\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

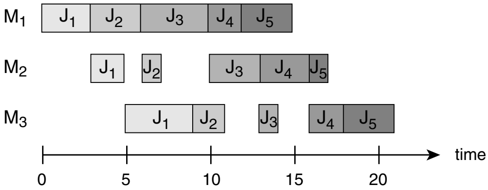
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

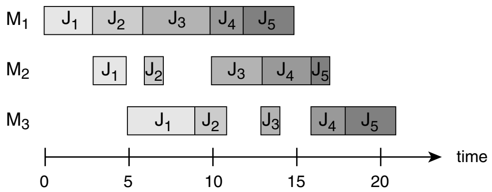
Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3



	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

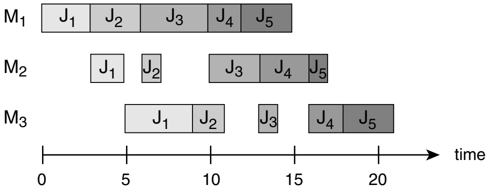
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

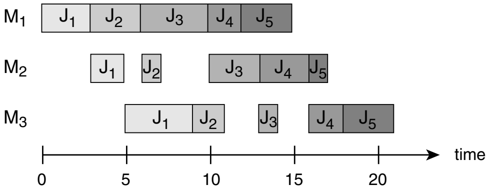
Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3



	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

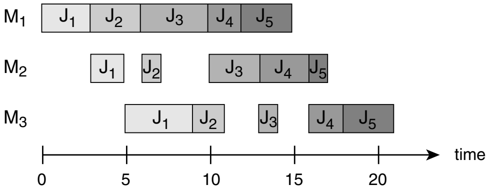
Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3



	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

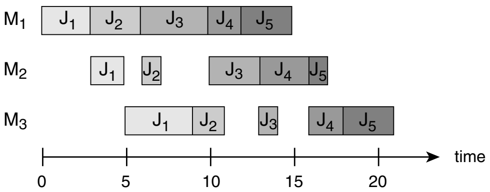
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} && j = 1, \ldots\ m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} && k = 1, \ldots\ n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} && k = 2, \ldots\ n, j = 2, \ldots\ m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

Weighted tardiness = 50

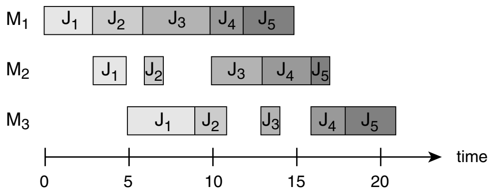
The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \dots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \dots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \dots n, j = 2, \dots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\} & &
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3

	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33



Makespan = 21

Sum of completion times = 73

Weighted sum of completion times = 186

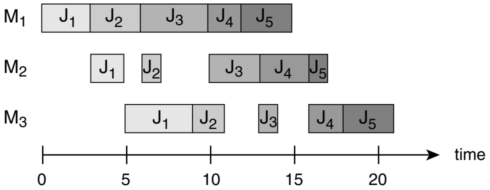
Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (5/6)

Computing completion times and tardiness

$$
\begin{aligned}
C_{\pi(1)j} &= \textstyle\sum_{h=1}^{j} p_{\pi(1)h} & j &= 1, \ldots m \\
C_{\pi(k)1} &= \textstyle\sum_{h=1}^{k} p_{\pi(h)1} & k &= 1, \ldots n \\
C_{\pi(k)j} &= \max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} & k &= 2, \ldots n, j = 2, \ldots m \\
T_{\pi(k)} &= \max\{C_{\pi(k),m} - d_i, 0\}
\end{aligned}
$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
d_i	8	11	12	14	10
w_i	1	2	4	2	3



	3	6	10	12	15
	5	7	13	16	17
C_i	9	11	14	18	21
T_i	1	0	2	4	11
$w_i \cdot C_i$	9	22	56	36	63
$w_i \cdot T_i$	1	0	8	8	33

Makespan = 21

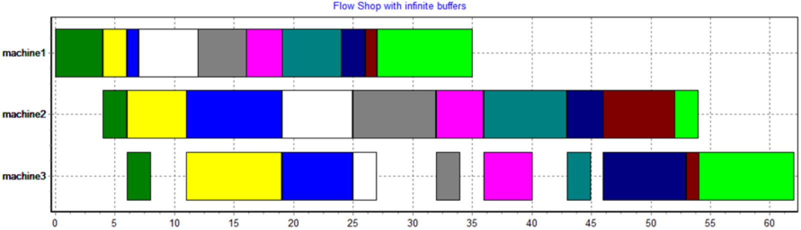
Sum of completion times = 73

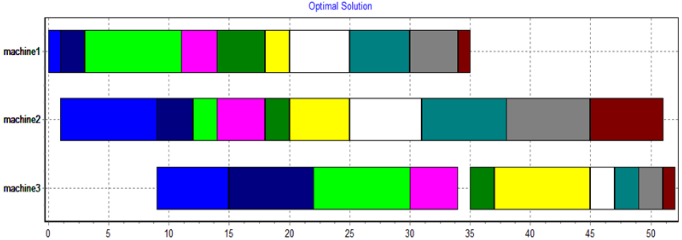
Weighted sum of completion times = 186

Weighted tardiness = 50

The Permutation Flowshop Scheduling Problem (6/6)

Example: random vs. optimal





Exercise 1.1: Iterative Improvement for the PFSP

Implement 12 iterative improvements algorithms for the PFSP-WT

Exercise 1.1: Iterative Improvement for the PFSP

Implement 12 iterative improvements algorithms for the PFSP-WT

- Pivoting rule:
  1. first-improvement
  2. best-improvement
- Neighborhood:
  1. Transpose
  2. Exchange
  3. Insert
- Initial solution:
  1. Random permutation
  2. Simplified RZ heuristic

Exercise 1.1: Iterative Improvement for the PFSP

Implement 12 iterative improvements algorithms for the PFSP-WT

- Pivoting rule:
  1. first-improvement
  2. best-improvement
- Neighborhood:
  1. Transpose
  2. Exchange
  3. Insert
- Initial solution:
  1. Random permutation
  2. Simplified RZ heuristic

2 pivoting rules $\times$ 3 neighborhoods $\times$ 2 initialization methods $=$
12 combinations

Exercise 1.1: Iterative Improvement for the PFSP

Implement 12 iterative improvements algorithms for the PFSP-WT

Don't implement 12 programs!
Reuse code and use command-line parameters

```
pfspwt-ii --first --transpose --srz
pfspwt-ii --best --exchange --random-init
...
```

Exercise 1.1: Iterative Improvement for the PFSP

Iterative Improvement

$\pi :=$ `GenerateInitialSolution()`

while π is not a local optimum **do**

  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$

  $\pi := \pi'$

Exercise 1.1: Iterative Improvement for the PFSP

Iterative Improvement

$\pi :=$ `GenerateInitialSolution()`
while π is not a local optimum **do**
  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$
  $\pi := \pi'$

Which neighbour to choose? Pivoting rule

- **Best Improvement:** choose best from all neighbours of π
  - ✔ Better quality
  - ✘ Requires evaluation of all neighbours in each step
- **First improvement:** evaluate neighbours in fixed order and choose first improving neighbour.
  - ✔ More efficient
  - ✘ Order of evaluation may impact quality / performance

Exercise 1.1: Iterative Improvement for the PFSP

Iterative Improvement

$\pi :=$ `GenerateInitialSolution()`
while π is not a local optimum **do**
  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$
  $\pi := \pi'$

Which neighbour to choose? Pivoting rule

- **Best Improvement:** choose best from all neighbours of π
  - ✔ Better quality
  - ✘ Requires evaluation of all neighbours in each step
- **First improvement:** evaluate neighbours in fixed order and choose first improving neighbour.
  - ✔ More efficient
  - ✘ Order of evaluation may impact quality / performance

Exercise 1.1: Iterative Improvement for the PFSP

Iterative Improvement

$\pi :=$ `GenerateInitialSolution()`
while π is not a local optimum **do**
  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$
  $\pi := \pi'$

Which neighbour to choose? Pivoting rule

- **Best Improvement:** choose best from all neighbours of π
  - ✔ Better quality
  - ✘ Requires evaluation of all neighbours in each step
- **First improvement:** evaluate neighbours in fixed order and choose first improving neighbour.
  - ✔ More efficient
  - ✘ Order of evaluation may impact quality / performance

Exercise 1.1: Iterative Improvement for the PFSP

Iterative Improvement

$\pi := \texttt{GenerateInitialSolution()}$
while π is not a local optimum **do**
  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$
  $\pi := \pi'$

Initial solution

- Random uniform permutation
- Simplified RZ heuristic

Exercise 1.1: Iterative Improvement for the PFSP

Iterative Improvement

$\pi :=$ `GenerateInitialSolution()`
while π is not a local optimum **do**
  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$
  $\pi := \pi'$

Simplified RZ heuristic

Start by ordering the jobs in ascending order of their weighted sum of processing times.
Construct the solution by inserting **one job at a time** in the position that minimize the WCT.

The weighted sum of processing times of job J_i is computed as $\frac{1}{w_i} \cdot \sum_1^m p_{ij}$
Note: the solution is constructed incrementally, and at each iteration C_i corresponds to the makespan of the partial solution.

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\quad j = 1, \ldots\ m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\quad k = 1, \ldots\ n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\quad k = 2, \ldots\ n, j = 2, \ldots\ m$

$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$ $\quad WCT = 65$

$\{J_5\ J_3\}$ $\quad WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$ $\quad WCT = 98$

$\{J_3\ J_2\ J_5\}$ $\quad WCT = 94$

$\{J_3\ J_5\ J_2\}$ $\quad WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 123$

$\{J_3\ J_4\ J_5\ J_2\}$ $\quad WCT = 130$

$\{J_3\ J_5\ J_4\ J_2\}$ $\quad WCT = 125$

$\{J_3\ J_5\ J_2\ J_4\}$ $\quad WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 167$

$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $\quad WCT = 161$

$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $\quad WCT = 163$

$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $\quad WCT = 151$

$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $\quad WCT = 144$

Simplified RZ heuristic: an example

$$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h} \qquad j = 1, \ldots m$$
$$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1} \qquad k = 1, \ldots n$$
$$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} \qquad k = 2, \ldots n, j = 2, \ldots m$$
$$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$	$WCT = 65$
$\{J_5\ J_3\}$	$WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$	$WCT = 98$
$\{J_3\ J_2\ J_5\}$	$WCT = 94$
$\{J_3\ J_5\ J_2\}$	$WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$	$WCT = 123$
$\{J_3\ J_4\ J_5\ J_2\}$	$WCT = 130$
$\{J_3\ J_5\ J_4\ J_2\}$	$WCT = 125$
$\{J_3\ J_5\ J_2\ J_4\}$	$WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$	$WCT = 167$
$\{J_4\ J_1\ J_3\ J_5\ J_2\}$	$WCT = 161$
$\{J_4\ J_3\ J_1\ J_5\ J_2\}$	$WCT = 163$
$\{J_4\ J_3\ J_5\ J_1\ J_2\}$	$WCT = 151$
$\{J_4\ J_3\ J_5\ J_2\ J_1\}$	$WCT = 144$

Simplified RZ heuristic: an example

$$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h} \qquad j = 1, \ldots m$$
$$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1} \qquad k = 1, \ldots n$$
$$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} \qquad k = 2, \ldots n, j = 2, \ldots m$$
$$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$ $WCT = 65$

$\{J_5\ J_3\}$ $WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$ $WCT = 98$

$\{J_3\ J_2\ J_5\}$ $WCT = 94$

$\{J_3\ J_5\ J_2\}$ $WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$ $WCT = 123$

$\{J_3\ J_4\ J_5\ J_2\}$ $WCT = 130$

$\{J_3\ J_5\ J_4\ J_2\}$ $WCT = 125$

$\{J_3\ J_5\ J_2\ J_4\}$ $WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $WCT = 167$

$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $WCT = 161$

$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $WCT = 163$

$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $WCT = 151$

$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $WCT = 144$

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\quad j = 1, \ldots m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\quad k = 1, \ldots n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\quad k = 2, \ldots n, j = 2, \ldots m$

$D_i = \frac{1}{w_i} \cdot \sum_{1}^{m} p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$	$WCT = 65$
$\{J_5\ J_3\}$	$WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$	$WCT = 98$
$\{J_3\ J_2\ J_5\}$	$WCT = 94$
$\{J_3\ J_5\ J_2\}$	$WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$	$WCT = 123$
$\{J_3\ J_4\ J_5\ J_2\}$	$WCT = 130$
$\{J_3\ J_5\ J_4\ J_2\}$	$WCT = 125$
$\{J_3\ J_5\ J_2\ J_4\}$	$WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$	$WCT = 167$
$\{J_4\ J_1\ J_3\ J_5\ J_2\}$	$WCT = 161$
$\{J_4\ J_3\ J_1\ J_5\ J_2\}$	$WCT = 163$
$\{J_4\ J_3\ J_5\ J_1\ J_2\}$	$WCT = 151$
$\{J_4\ J_3\ J_5\ J_2\ J_1\}$	$WCT = 144$

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\quad j = 1, \ldots m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\quad k = 1, \ldots n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\quad k = 2, \ldots n, j = 2, \ldots m$

$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$
- $\{J_3\ J_5\}$ $\quad WCT = 65$
- $\{J_5\ J_3\}$ $\quad WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$
- $\{J_2\ J_3\ J_5\}$ $\quad WCT = 98$
- $\{J_3\ J_2\ J_5\}$ $\quad WCT = 94$
- $\{J_3\ J_5\ J_2\}$ $\quad WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$
- $\{J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 123$
- $\{J_3\ J_4\ J_5\ J_2\}$ $\quad WCT = 130$
- $\{J_3\ J_5\ J_4\ J_2\}$ $\quad WCT = 125$
- $\{J_3\ J_5\ J_2\ J_4\}$ $\quad WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$
- $\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 167$
- $\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $\quad WCT = 161$
- $\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $\quad WCT = 163$
- $\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $\quad WCT = 151$
- $\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $\quad WCT = 144$

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\quad j = 1, \dots m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\quad k = 1, \dots n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\quad k = 2, \dots n, j = 2, \dots m$

$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$ $\quad WCT = 65$

$\{J_5\ J_3\}$ $\quad WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$ $\quad WCT = 98$

$\{J_3\ J_2\ J_5\}$ $\quad WCT = 94$

$\{J_3\ J_5\ J_2\}$ $\quad WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 123$

$\{J_3\ J_4\ J_5\ J_2\}$ $\quad WCT = 130$

$\{J_3\ J_5\ J_4\ J_2\}$ $\quad WCT = 125$

$\{J_3\ J_5\ J_2\ J_4\}$ $\quad WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 167$

$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $\quad WCT = 161$

$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $\quad WCT = 163$

$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $\quad WCT = 151$

$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $\quad WCT = 144$

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\quad j = 1, \dots m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\quad k = 1, \dots n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\quad k = 2, \dots n, j = 2, \dots m$

$D_i = \frac{1}{w_i} \cdot \sum_{1}^{m} p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$ $\quad WCT = 65$

$\{J_5\ J_3\}$ $\quad WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$ $\quad WCT = 98$

$\{J_3\ J_2\ J_5\}$ $\quad WCT = 94$

$\{J_3\ J_5\ J_2\}$ $\quad WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 123$

$\{J_3\ J_4\ J_5\ J_2\}$ $\quad WCT = 130$

$\{J_3\ J_5\ J_4\ J_2\}$ $\quad WCT = 125$

$\{J_3\ J_5\ J_2\ J_4\}$ $\quad WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 167$

$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $\quad WCT = 161$

$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $\quad WCT = 163$

$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $\quad WCT = 151$

$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $\quad WCT = 144$

Simplified RZ heuristic: an example

$$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h} \qquad j = 1, \ldots m$$
$$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1} \qquad k = 1, \ldots n$$
$$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} \qquad k = 2, \ldots n, j = 2, \ldots m$$
$$D_i = \frac{1}{w_i} \cdot \sum_{1}^{m} p_{ij}$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$
Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$
$\{J_3\ J_5\}$ $WCT = 65$
$\{J_5\ J_3\}$ $WCT = 65$
Step 2 $\pi = \{J_3\ J_5\}$
$\{J_2\ J_3\ J_5\}$ $WCT = 98$
$\{J_3\ J_2\ J_5\}$ $WCT = 94$
$\{J_3\ J_5\ J_2\}$ $WCT = 91$
Step 3 $\pi = \{J_3\ J_5\ J_2\}$
$\{J_4\ J_3\ J_5\ J_2\}$ $WCT = 123$
$\{J_3\ J_4\ J_5\ J_2\}$ $WCT = 130$
$\{J_3\ J_5\ J_4\ J_2\}$ $WCT = 125$
$\{J_3\ J_5\ J_2\ J_4\}$ $WCT = 125$
Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$
$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $WCT = 167$
$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $WCT = 161$
$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $WCT = 163$
$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $WCT = 151$
$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $WCT = 144$

Simplified RZ heuristic: an example

$$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h} \qquad j = 1, \ldots m$$
$$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1} \qquad k = 1, \ldots n$$
$$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j} \qquad k = 2, \ldots n, j = 2, \ldots m$$
$$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$	$WCT = 65$
$\{J_5\ J_3\}$	$WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$	$WCT = 98$
$\{J_3\ J_2\ J_5\}$	$WCT = 94$
$\{J_3\ J_5\ J_2\}$	$WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$	$WCT = 123$
$\{J_3\ J_4\ J_5\ J_2\}$	$WCT = 130$
$\{J_3\ J_5\ J_4\ J_2\}$	$WCT = 125$
$\{J_3\ J_5\ J_2\ J_4\}$	$WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$	$WCT = 167$
$\{J_4\ J_1\ J_3\ J_5\ J_2\}$	$WCT = 161$
$\{J_4\ J_3\ J_1\ J_5\ J_2\}$	$WCT = 163$
$\{J_4\ J_3\ J_5\ J_1\ J_2\}$	$WCT = 151$
$\{J_4\ J_3\ J_5\ J_2\ J_1\}$	$WCT = 144$

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\quad j = 1, \dots m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\quad k = 1, \dots n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\quad k = 2, \dots n, j = 2, \dots m$

$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$ $\quad WCT = 65$

$\{J_5\ J_3\}$ $\quad WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$ $\quad WCT = 98$

$\{J_3\ J_2\ J_5\}$ $\quad WCT = 94$

$\{J_3\ J_5\ J_2\}$ $\quad WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 123$

$\{J_3\ J_4\ J_5\ J_2\}$ $\quad WCT = 130$

$\{J_3\ J_5\ J_4\ J_2\}$ $\quad WCT = 125$

$\{J_3\ J_5\ J_2\ J_4\}$ $\quad WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 167$

$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $\quad WCT = 161$

$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $\quad WCT = 163$

$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $\quad WCT = 151$

$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $\quad WCT = 144$

Simplified RZ heuristic: an example

$C_{\pi(1)j} = \sum_{h=1}^{j} p_{\pi(1)h}$ $\qquad j = 1, \dots\ m$

$C_{\pi(k)1} = \sum_{h=1}^{k} p_{\pi(h)1}$ $\qquad k = 1, \dots\ n$

$C_{\pi(k)j} = max\{C_{\pi(k-1)j}, C_{\pi(k)(j-1)}\} + p_{\pi(k)j}$ $\qquad k = 2, \dots\ n, j = 2, \dots\ m$

$D_i = \frac{1}{w_i} \cdot \sum_1^m p_{ij}$

Job	J_1	J_2	J_3	J_4	J_5
p_{i1}	3	3	4	2	3
p_{i2}	2	1	3	3	1
p_{i3}	4	2	1	2	3
w_i	1	2	4	2	3
D_i	9	3	2	3.5	2.3

Starting sequence $= \{J_3\ J_5\ J_2\ J_4\ J_1\}$

Initial Solution $= \{J_4\ J_3\ J_5\ J_2\ J_1\}$

Step 1 $\pi = \{\}$

$\{J_3\ J_5\}$ $\quad WCT = 65$

$\{J_5\ J_3\}$ $\quad WCT = 65$

Step 2 $\pi = \{J_3\ J_5\}$

$\{J_2\ J_3\ J_5\}$ $\quad WCT = 98$

$\{J_3\ J_2\ J_5\}$ $\quad WCT = 94$

$\{J_3\ J_5\ J_2\}$ $\quad WCT = 91$

Step 3 $\pi = \{J_3\ J_5\ J_2\}$

$\{J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 123$

$\{J_3\ J_4\ J_5\ J_2\}$ $\quad WCT = 130$

$\{J_3\ J_5\ J_4\ J_2\}$ $\quad WCT = 125$

$\{J_3\ J_5\ J_2\ J_4\}$ $\quad WCT = 125$

Step 4 $\pi = \{J_4\ J_3\ J_5\ J_2\}$

$\{J_1\ J_4\ J_3\ J_5\ J_2\}$ $\quad WCT = 167$

$\{J_4\ J_1\ J_3\ J_5\ J_2\}$ $\quad WCT = 161$

$\{J_4\ J_3\ J_1\ J_5\ J_2\}$ $\quad WCT = 163$

$\{J_4\ J_3\ J_5\ J_1\ J_2\}$ $\quad WCT = 151$

$\{J_4\ J_3\ J_5\ J_2\ J_1\}$ $\quad WCT = 144$

Exercise 1.1: Iterative Improvement for the PFSP

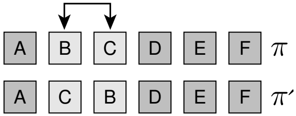
Iterative Improvement

$\pi :=$ `GenerateInitialSolution()`
while π is not a local optimum **do**
  choose a neighbour $\pi' \in \mathcal{N}(\pi)$ such that $F(\pi') < F(\pi)$
  $\pi := \pi'$

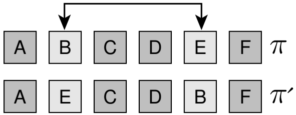
Which neighborhood $\mathcal{N}(\pi)$?

- Transpose
- Exchange
- Insert

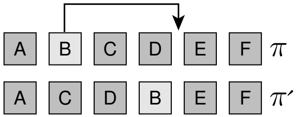
Exercise 1.1: Iterative Improvement for the PFSP



transpose neighbourhood

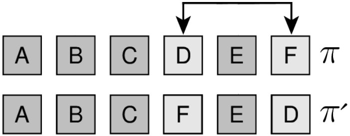


exchange neighbourhood



insert neighbourhood

Exercise 1.1: Iterative Improvement for the PFSP



exchange neighbourhood

Example: Exchange π_i and π_j ($i < j$), $\pi' = \texttt{Exchange}(\pi, i, j)$

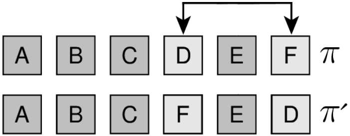
Only jobs after i are affected!

Do not recompute the evaluation function from scratch!

Equivalent speed-ups with Transpose and Insertion

(NOTE: Implementing speed-ups will get you extra points in the exercise)

Exercise 1.1: Iterative Improvement for the PFSP



exchange neighbourhood

Example: Exchange π_i and π_j $(i < j)$, $\pi' = \texttt{Exchange}(\pi, i, j)$

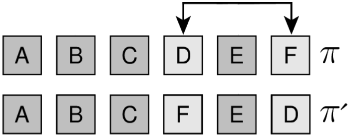
Only jobs after i are affected!

Do not recompute the evaluation function from scratch!

Equivalent speed-ups with Transpose and Insertion

(NOTE: Implementing speed-ups will get you extra points in the exercise)

Exercise 1.1: Iterative Improvement for the PFSP



exchange neighbourhood

Example: Exchange π_i and π_j ($i < j$), $\pi' = \texttt{Exchange}(\pi, i, j)$

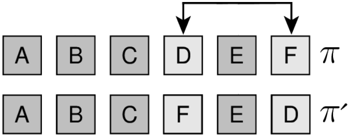
Only jobs after i are affected!

Do not recompute the evaluation function from scratch!

Equivalent speed-ups with Transpose and Insertion

(NOTE: Implementing speed-ups will get you extra points in the exercise)

Exercise 1.1: Iterative Improvement for the PFSP



exchange neighbourhood

Example: Exchange π_i and π_j ($i < j$), $\pi' = \texttt{Exchange}(\pi, i, j)$

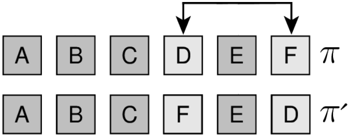
Only jobs after i are affected!

Do not recompute the evaluation function from scratch!

Equivalent speed-ups with Transpose and Insertion

(NOTE: Implementing speed-ups will get you extra points in the exercise)

Exercise 1.1: Iterative Improvement for the PFSP



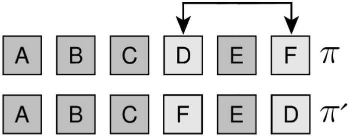
exchange neighbourhood

Example: Exchange π_i and π_j $(i < j)$, $\pi' = \texttt{Exchange}(\pi, i, j)$

Only jobs after i are affected!
Do not recompute the evaluation function from scratch!
Equivalent speed-ups with Transpose and Insertion

(NOTE: Implementing speed-ups will get you extra points in the exercise)

Exercise 1.1: Iterative Improvement for the PFSP



exchange neighbourhood

Example: Exchange π_i and π_j ($i < j$), $\pi' = \texttt{Exchange}(\pi, i, j)$

Only jobs after i are affected!
Do not recompute the evaluation function from scratch!
Equivalent speed-ups with Transpose and Insertion

(NOTE: Implementing speed-ups will get you extra points in the exercise)

Exercise 1.1: Iterative Improvement for the PFSP

Instances

- PFSP instances with 50 and 100 jobs, and 20 machines.
- A full description is provided in the project document on TEAMS

Experiments

Apply each algorithm k once to each instance i and compute:

1. Relative percentage deviation $\Delta_{ki} = 100 \cdot \dfrac{\text{cost}_{ki} - \text{best-known}_i}{\text{best-known}_i}$
2. Computation time (t_{ki})

Repeat **5 times**

Report for each algorithm k

- Average relative percentage deviation *per instance size* (50 and 100) per algorithm (12 algorithms)
- Average computation time *per instance size* (50 and 100) per algorithm (12 algorithms)

Exercise 1.1: Iterative Improvement for the PFSP

Instances

- PFSP instances with 50 and 100 jobs, and 20 machines.
- A full description is provided in the project document on TEAMS

Experiments

Apply each algorithm *k* once to each instance *i* and compute:

1. Relative percentage deviation $\Delta_{ki} = 100 \cdot \frac{\text{cost}_{ki} - \text{best-known}_i}{\text{best-known}_i}$
2. Computation time (t_{ki})

Repeat **5 times**

Report for each algorithm *k*

- Average relative percentage deviation *per instance size* (50 and 100) per algorithm (12 algorithms)
- Average computation time *per instance size* (50 and 100) per algorithm (12 algorithms)

Exercise 1.1: Iterative Improvement for the PFSP

Instances

- PFSP instances with 50 and 100 jobs, and 20 machines.
- A full description is provided in the project document on TEAMS

Experiments

Apply each algorithm k once to each instance i and compute:

1. Relative percentage deviation $\Delta_{ki} = 100 \cdot \dfrac{\text{cost}_{ki} - \text{best-known}_i}{\text{best-known}_i}$
2. Computation time (t_{ki})

Repeat **5 times**

Report for each algorithm k

- Average relative percentage deviation *per instance size* (50 and 100) per algorithm (12 algorithms)
- Average computation time *per instance size* (50 and 100) per algorithm (12 algorithms)

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Statistical test

- Paired t-test
- Wilcoxon signed-rank test

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (1)

- *Statistical hypothesis tests* are used to assess the validity of statements about properties of or relations between sets of statistical data.
- The statement to be tested (or its negation) is called the *null hypothesis* (H_0) of the test.
  Example: For the Wilcoxon signed-rank test, the null hypothesis is that 'the median of the differences is zero'.
- The *significance level* (α) determines the maximum allowable probability of incorrectly rejecting the null hypothesis.
  Typical values of α are 0.05 or 0.01.

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (1)

- *Statistical hypothesis tests* are used to assess the validity of statements about properties of or relations between sets of statistical data.
- The statement to be tested (or its negation) is called the *null hypothesis* (H_0) of the test.
  Example: For the Wilcoxon signed-rank test, the null hypothesis is that 'the median of the differences is zero'.
- The *significance level* (α) determines the maximum allowable probability of incorrectly rejecting the null hypothesis.
  Typical values of α are 0.05 or 0.01.

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (1)

- *Statistical hypothesis tests* are used to assess the validity of statements about properties of or relations between sets of statistical data.
- The statement to be tested (or its negation) is called the *null hypothesis* (H_0) of the test.
  Example: For the Wilcoxon signed-rank test, the null hypothesis is that 'the median of the differences is zero'.
- The *significance level* (α) determines the maximum allowable probability of incorrectly rejecting the null hypothesis.
  Typical values of α are 0.05 or 0.01.

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (1)

- *Statistical hypothesis tests* are used to assess the validity of statements about properties of or relations between sets of statistical data.
- The statement to be tested (or its negation) is called the *null hypothesis* (H_0) of the test.
  Example: For the Wilcoxon signed-rank test, the null hypothesis is that 'the median of the differences is zero'.
- The *significance level* (α) determines the maximum allowable probability of incorrectly rejecting the null hypothesis.
  Typical values of α are 0.05 or 0.01.

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (1)

- *Statistical hypothesis tests* are used to assess the validity of statements about properties of or relations between sets of statistical data.
- The statement to be tested (or its negation) is called the *null hypothesis* (H_0) of the test.
  Example: For the Wilcoxon signed-rank test, the null hypothesis is that 'the median of the differences is zero'.
- The *significance level* (α) determines the maximum allowable probability of incorrectly rejecting the null hypothesis.
  Typical values of α are 0.05 or 0.01.

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (2)

- The application of a test to a given data set results in a *p-value*, which represents the probability that the null hypothesis is incorrectly rejected.
- The null hypothesis is rejected iff this p-value is smaller than the previously chosen significance level.
- Most common statistical hypothesis tests are already implemented in statistical software such as the *R software environment* (http://www.r-project.org/).

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (2)

- The application of a test to a given data set results in a *p-value*, which represents the probability that the null hypothesis is incorrectly rejected.
- The null hypothesis is rejected iff this p-value is smaller than the previously chosen significance level.
- Most common statistical hypothesis tests are already implemented in statistical software such as the *R software environment* (http://www.r-project.org/).

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Background: Statistical hypothesis tests (2)

- The application of a test to a given data set results in a *p-value*, which represents the probability that the null hypothesis is incorrectly rejected.
- The null hypothesis is rejected iff this p-value is smaller than the previously chosen significance level.
- Most common statistical hypothesis tests are already implemented in statistical software such as the *R software environment* (http://www.r-project.org/).

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Example in R

```
best.known <- read.csv ("bestSolutions.txt")
a.cost <- read.table("ii-best-ex-rand.dat")$V1
a.cost <- 100 * (a.cost - best.known) / best.known$BS
b.cost <- read.table("ii-best-ins-rand.dat")$V1
b.cost <- 100 * (b.cost - best.known) / best.known$BS
t.test (a.cost, b.cost, paired=T)$p.value
[1] 0.8819112
wilcox.test (a.cost, b.cost, paired=T)$p.value
[1] 0.0019212
```

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Example in R

```
best.known <- read.csv ("bestSolutions.txt")
a.cost <- read.table("ii-best-ex-rand.dat")$V1
a.cost <- 100 * (a.cost - best.known) / best.known$BS
b.cost <- read.table("ii-best-ins-rand.dat")$V1
b.cost <- 100 * (b.cost - best.known) / best.known$BS
t.test (a.cost, b.cost, paired=T)$p.value
[1] 0.8819112
wilcox.test (a.cost, b.cost, paired=T)$p.value
[1] 0.0019212
```

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Example in R

```
best.known <- read.csv ("bestSolutions.txt")
a.cost <- read.table("ii-best-ex-rand.dat")$V1
a.cost <- 100 * (a.cost - best.known) / best.known$BS
b.cost <- read.table("ii-best-ins-rand.dat")$V1
b.cost <- 100 * (b.cost - best.known) / best.known$BS
t.test (a.cost, b.cost, paired=T)$p.value
[1] 0.8819112
wilcox.test (a.cost, b.cost, paired=T)$p.value
[1] 0.0019212
```

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Example in R

```
best.known <- read.csv ("bestSolutions.txt")
a.cost <- read.table("ii-best-ex-rand.dat")$V1
a.cost <- 100 * (a.cost - best.known) / best.known$BS
b.cost <- read.table("ii-best-ins-rand.dat")$V1
b.cost <- 100 * (b.cost - best.known) / best.known$BS
t.test (a.cost, b.cost, paired=T)$p.value
[1] 0.8819112
wilcox.test (a.cost, b.cost, paired=T)$p.value
[1] 0.0019212
```

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Example in R

```
best.known <- read.csv ("bestSolutions.txt")
a.cost <- read.table("ii-best-ex-rand.dat")$V1
a.cost <- 100 * (a.cost - best.known) / best.known$BS
b.cost <- read.table("ii-best-ins-rand.dat")$V1
b.cost <- 100 * (b.cost - best.known) / best.known$BS
t.test (a.cost, b.cost, paired=T)$p.value
[1] 0.8819112
wilcox.test (a.cost, b.cost, paired=T)$p.value
[1] 0.0019212
```

Exercise 1.1: Iterative Improvement for the PFSP

Is there a statistically significant difference between the solution quality generated by the different algorithms?

Example in R

```
best.known <- read.csv ("bestSolutions.txt")
a.cost <- read.table("ii-best-ex-rand.dat")$V1
a.cost <- 100 * (a.cost - best.known) / best.known$BS
b.cost <- read.table("ii-best-ins-rand.dat")$V1
b.cost <- 100 * (b.cost - best.known) / best.known$BS
t.test (a.cost, b.cost, paired=T)$p.value
[1] 0.8819112
wilcox.test (a.cost, b.cost, paired=T)$p.value
[1] 0.0019212
```

Exercise 1.2 VND algorithms for the PFSP

Implement 4 VND algorithms for the PFSP

- Pivoting rule: first-improvement
- Neighborhood order:
  1. transpose $\rightarrow$ exchange $\rightarrow$ insert
  2. transpose $\rightarrow$ insert $\rightarrow$ exchange
- Initial solution:
  1. Random permutation
  2. Simplified RZ heuristic

Exercise 1.2 VND algorithms for the PFSP

Variable Neighbourhood Descent (VND)

k neighborhoods $\mathcal{N}_1, \ldots, \mathcal{N}_k$
$\pi :=$ `GenerateInitialSolution()`
$i := 1$
repeat
  choose the first improving neighbor $\pi' \in \mathcal{N}_i(\pi)$
  if $\nexists \pi'$ **then**
    $i := i + 1$
  else
    $\pi := \pi'$
    $i := 1$
until $i > k$

Exercise 1.2 VND algorithms for the PFSP

Implement 4 VND algorithms for the PFSP

- Instances: Same as for exercise 1.1
- Experiments: **one run** of each algorithm per instance
- Report: Same as for exercise 1.1
- Statistical tests: Same as for exercise 1.1

- Instances and “skeleton” code are available on TEAMS

- Some of the deliverables you need to provide in a zip folder with your name via TEAMS are:
  - your implementation in `C`, `C++` or `Java`. `Python` is also possible, but **not** recommended
  - a README file explaining how to run your implementation from the command line on Linux
  - a report describing the implementation of the algorithms and the results you obtained more detail on TEAMS)
  - see the full description of the deliverables in the pdf on TEAMS

- Deadline is April 14, 2023 (23:59)

- Questions?
  Use TEAMS to post them in the channel of the assignment