Jean-Louis Bouquard



Beijing Institute of Technology

Content

Mathematical modeling

2 Solving linear problems





- Why a mathematical modeling?

- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be I INE AR
- Do not confuse variables (unknown, decisions) and
- Do not confuse m

- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be I INE ARX
- Do not confuse variables (unknown, decisions) and the flavours
- The size continue (math-rende) and sixing

- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be LINEAR
- Do not confuse variables (unknown, decisions) and
- Do not confuse math model variables

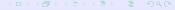
 alreadition variables

- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-mod

- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables



- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables



- Why a mathematical modeling?
- Define clearly the problem, the data
- Choose the variables
- Write the constraints
- and the objective function
- Restriction! constraints and objective function must be LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

- For every job j, C_{1,j} and C_{2,j} are the completion times of j
- T_j is the tardiness of j, $T_j \ge 0$

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j}$$
 $C_{2,j} \geq C_{1,j} + p_{2,j}$
 $T_j \geq C_{2,j} - d_j$

biactive tunction:

- For every job j, C_{1,j} and C_{2,j} are the completion times of j
- T_i is the tardiness of j, $T_i \ge 0$

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j}$$
 $C_{2,j} \geq C_{1,j} + p_{2,j}$
 $T_j \geq C_{2,j} - d_j$

Objective function:

Minimize
$$\sum_{j=1}^{n} T_j$$



- For every job j, C_{1,j} and C_{2,j} are the completion times of j
- T_j is the tardiness of j, $T_j \ge 0$

Succession of the operations and tardinesses:

$$orall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \ C_{2,j} \geq C_{1,j} + p_{2,j} \ T_j \geq C_{2,j} - d_j$$

Objective function:





- For every job j, C_{1,j} and C_{2,j} are the completion times of j
- T_j is the tardiness of j, $T_j \ge 0$

Succession of the operations and tardinesses:

$$orall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \ C_{2,j} \geq C_{1,j} + p_{2,j} \ T_j \geq C_{2,j} - d_j$$

Objective function:

Minimize
$$\sum_{j=1}^{n} T_j$$



- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- ullet Or j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1}
 ho_{i,j_1}$
- Idea: link this choice to a binary variable
- For every 2 jobs j_1 and j_2 ($j_1 < j_2$) $Z_{h,h} \text{ is a binary variable } (\{0,1\})$
 - indicating whether j_1 is before j_2



- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- Or j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1} p_{i,j_2}$
- Idea: link this choice to a binary variable
- $Z_{h,h}$ is a binary variable ({0,1}) indicating whether h is before h

- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- ullet Or j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1} p_{i,j_1}$
- Idea: link this choice to a binary variable
- For every 2 jobs j_1 and j_2 ($j_1 < j_2$)

- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- Or j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1} p_{i,j_1}$
- Idea: link this choice to a binary variable
- For every 2 jobs j_1 and j_2 ($j_1 < j_2$) Z_{j_1,j_2} is a binary variable ($\{0,1\}$) indicating whether j_1 is before j_2



- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1} p_{i,j_1}$
- Idea: link this choice to a binary variable



- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- ullet Or j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1} p_{i,j_1}$
- Idea: link this choice to a binary variable
- For every 2 jobs j_1 and j_2 ($j_1 < j_2$) Z_{j_1,j_2} is a binary variable ($\{0,1\}$) indicating whether j_1 is before j_2



- How to express the non overlap constraints?
- For every 2 jobs (j_1, j_2) and every machine i
- Either j_1 is before j_2 and $C_{i,j_1} \leq C_{i,j_2} p_{i,j_2}$
- Or j_1 is after j_2 and $C_{i,j_2} \leq C_{i,j_1} p_{i,j_1}$
- Idea: link this choice to a binary variable
- For every 2 jobs j_1 and j_2 ($j_1 < j_2$) Z_{j_1,j_2} is a binary variable ($\{0,1\}$) indicating whether j_1 is before j_2
- $Z_{j_1,j_2} = \begin{cases} 0 & \text{if } j_1 \text{ before } j_2 \\ 1 & \text{if } j_1 \text{ after } j_2 \end{cases}$



Non overlap constraints:

$$\forall (j_1, j_2), 1 \leq j_1 < j_2 \leq n, \ C_{1,j_1} \leq C_{1,j_2} - p_{1,j_2} \ \ \text{if} \ \ Z_{j_1,j_2} = 0 \ C_{1,j_2} \leq C_{1,j_1} - p_{1,j_1} \ \ \text{if} \ \ Z_{j_1,j_2} = 1$$

They can be turned into

$$\forall i \in \{1, 2\}, \forall (j_1, j_2), \quad C_{i, j_1} + p_{i, j_2} \leq C_{i, j_2} + HV \times Z_{j_1, j_2}$$

$$1 \leq j_1 < j_2 \leq n, \quad C_{i, j_1} + p_{i, j_2} \leq C_{i, j_1} + HV \times (1 - Z_{j_1, j_2})$$

Where HV is a *high value* bigger than any $C_{i,j}$



Non overlap constraints:

$$\forall (j_1, j_2), 1 \leq j_1 < j_2 \leq n, \ C_{1,j_1} \leq C_{1,j_2} - p_{1,j_2} \ \ \text{if} \ \ Z_{j_1,j_2} = 0 \ C_{1,j_2} \leq C_{1,j_1} - p_{1,j_1} \ \ \text{if} \ \ Z_{j_1,j_2} = 1$$

They can be turned into:

$$\forall i \in \{1,2\}, \forall (j_1,j_2), \quad C_{i,j_1} + p_{i,j_2} \leq C_{i,j_2} + HV \times Z_{j_1,j_2} \\ 1 \leq j_1 < j_2 \leq n, \quad C_{i,j_1} + p_{i,j_2} \leq C_{i,j_1} + HV \times (1 - Z_{j_1,j_2})$$

Where HV is a high value bigger than any $C_{i,j}$



Example of a permutation

The permutation (253614) is represented by the (triangular) matrix (Z_{j_1,j_2}):

Example of a permutation

The permutation (253614) is represented by the (triangular) matrix (Z_{j_1,j_2}):

	1	2	3	4	5	6
1		1	1	0	1	1
2			0	0	0	0
				0	1	0
4					1	1
4 5						0
6						

Exclusion on each machine:

$$\forall i \in \{1,2\}, \forall (j_1,j_2), 1 \leq j_1 < j_2 \leq n, \\ p_{i,j_2} \leq C_{i,j_2} - C_{i,j_1} + HV \times Z_{j_1,j_2} \leq HV - p_{i,j_1}$$

Succession of the operations and tardinesses:

$$\forall j, 1 \le j \le n, \quad C_{1,j} \ge p_{1,j}$$
 (2)

$$C_{2,j} \ge C_{1,j} + p_{2,j} \tag{3}$$

$$T_j \ge C_{2,j} - d_j \tag{4}$$

Objective function

$$Minimize \sum_{i=1}^{n} T_{i}$$
 (5)



9 / 24

Exclusion on each machine:

$$\forall i \in \{1,2\}, \forall (j_1,j_2), 1 \leq j_1 < j_2 \leq n, \\ p_{i,j_2} \leq C_{i,j_2} - C_{i,j_1} + HV \times Z_{j_1,j_2} \leq HV - p_{i,j_1}$$

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \tag{2}$$

$$C_{2,j} \geq C_{1,j} + p_{2,j}$$
 (3)

$$T_j \geq C_{2,j} - d_j \tag{4}$$

Objective function









Exclusion on each machine:

$$\forall i \in \{1,2\}, \forall (j_1,j_2), 1 \leq j_1 < j_2 \leq n, \\ p_{i,j_2} \leq C_{i,j_2} - C_{i,j_1} + HV \times Z_{j_1,j_2} \leq HV - p_{i,j_1}$$

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \tag{2}$$

$$C_{2,j} \geq C_{1,j} + p_{2,j}$$
 (3)

$$T_j \geq C_{2,j} - d_j \tag{4}$$

Objective function

$$Minimize \sum_{j=1}^{n} T_{j}$$
 (5)

- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k as a variable ?
- We would have: $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- It is not a linear constraint
- ullet p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- ullet To the ves-no auestion: "Is $i=S_{\mathbb R}$?
- we associate the binary variable K.
- \wedge $X_{i,j} = \begin{cases} 1 & \text{if } j \\ 1 & \text{if } j \end{cases}$

10 / 24

- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k as a variable ?
- We would have: $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$
- It is not a linear constraint
- $\triangleright p_{1.S_k}$ is depending on S_k but not linearly
- Again, we use binary variables:
- ullet To the ves-no auestion: " $ls~i=S_{arkappa}$?"
- we associate the binary variable X.

<ロ> < 回 > < 回 > < 巨 > < 巨 > 三 の < ○

- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k as a variable ?
- We would have: $C_{1,\mathcal{S}_k} \geq C_{1,\mathcal{S}_{k-1}} + p_{1,\mathcal{S}_k}$
- It is not a linear constraint
- p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the ves-no question: "Is $i = S_{k}$?"
- we accordate the binary variable X.
- \bullet $X_{\nu} = \begin{cases} 1 & \text{if } j \text{ fi} \end{cases}$

- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k as a variable ?
- We would have: $C_{1,\mathcal{S}_k} \geq C_{1,\mathcal{S}_{k-1}} + p_{1,\mathcal{S}_k}$
- It is not a linear constraint
- ullet p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the ves-no question: "Is $i = S_i$. ?"
- we associate the hinary variable.

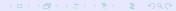
- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k
 as a variable ?
- We would have: $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- It is not a linear constraint
- p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the yes-no question: "Is $j = S_k$?"



- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k
 as a variable ?
- We would have: $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$
- It is not a linear constraint
- p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the yes-no question: "Is $j = S_k$?"
- we associate the binary variable X_k



- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k as a variable ?
- We would have: $C_{1,\mathcal{S}_k} \geq C_{1,\mathcal{S}_{k-1}} + p_{1,\mathcal{S}_k}$
- It is not a linear constraint
- p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the yes-no question: "Is $j = S_k$?"
- \bullet we associate the binary variable X_k
- $X_{k,j} = \begin{cases} 1 & \text{if } j \text{ has rank } k \\ 0 & \text{otherwise} \end{cases}$



A second model for the F2// $\sum T_i$

- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k as a variable ?
- We would have: $C_{1,\mathcal{S}_k} \geq C_{1,\mathcal{S}_{k-1}} + p_{1,\mathcal{S}_k}$
- It is not a linear constraint
- p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the yes-no question: "Is $j = S_k$?"
- ullet we associate the binary variable $X_{k,j}$





A second model for the F2// $\sum T_i$

- The main decision is the choice of the permutation
- Can we choose S_k = number of the job in position k
 as a variable ?
- We would have: $C_{1,\mathcal{S}_k} \geq C_{1,\mathcal{S}_{k-1}} + p_{1,\mathcal{S}_k}$
- It is not a linear constraint
- p_{1,S_k} is depending on S_k but not linearly
- Again, we use binary variables:
- To the yes-no question: "Is $j = S_k$?"
- ullet we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \text{if } j \text{ has rank } k \\ 0 & \text{otherwise} \end{cases}$



• Every rank is assigned to some job:

$$\forall k = 1..n$$
 $\sum_{j=1}^{n} X_{k,j} = 1$ (6)

Every job has some rank:

$$\forall j = 1..n$$
 $\sum_{k=1}^{n} X_{k,j} = 1$ (7)

11 / 24

• Every rank is assigned to some job:

$$\forall k = 1..n$$
 $\sum_{j=1}^{n} X_{k,j} = 1$ (6)

Every job has some rank:

$$\forall j = 1..n$$
 $\sum_{k=1}^{n} X_{k,j} = 1$ (7)

11 / 24

Example of a permutation

The permutation (253614) is represented by the matrix $(X_{k,j})$:

Example of a permutation

The permutation (253614) is represented by the matrix $(X_{k,j})$:

	1	2	3	4	5	6
1	0	1	0	0	0	0
2	0	0	0	0	1	0
3	0	0	1	0	0	0
4	0	0	0	0	0	1
5	1	0	0	0	0	0
6	0	0	0	1	0	0

- In $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- we will replace C_{1,S_k} with $F_{1,k}$
- For every rank k, F_{1,k} and F_{2,k} are the completion times of the kth job
- Similarly T₆, is renamed C
- \circ G, is the randinger of the $k^{\prime\prime\prime}$ tota G, > 0

- In $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- we will replace C_{1,S_k} with $F_{1,k}$
- For every rank k, F_{1,k} and F_{2,k} are the completion times of the kth job
- Similarly $T_{S_{\nu}}$ is renamed G_k
- G_k is the tandinose of the k''' tob. $G_k > 0$

- In $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- we will replace C_{1,S_k} with $F_{1,k}$
- For every rank k, $F_{1,k}$ and $F_{2,k}$ are the completion times of the k^{th} job
- Similarly T_{S_k} is renamed G_k
- G_k is the tardiness of the k^{th} job, $G_k \ge 0$

- In $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- we will replace C_{1,S_k} with $F_{1,k}$
- For every rank k, $F_{1,k}$ and $F_{2,k}$ are the completion times of the k^{th} job
- Similarly T_{S_k} is renamed G_k

• G_k is the tardiness of the k^{th} job, $G_k \ge 0$

- In $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$
- we will replace C_{1,S_k} with $F_{1,k}$
- For every rank k, $F_{1,k}$ and $F_{2,k}$ are the completion times of the k^{th} job
- Similarly T_{S_k} is renamed G_k
- G_k is the tardiness of the k^{th} job, $G_k \ge 0$

- $C_{1.S_k} \ge C_{1.S_{k-1}} + p_{1.S_k}$ is written



14 / 24

- $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$ is written
- $F_{1,k} \geq F_{1,k-1} + p_{1,S_k}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})
- S_k is replaced by the row $(X_{k,1}, X_{k,2}, \dots, X_{k,n})$
- \bullet $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the S_k^{th} position
- Thus ρ_{1,S_k} is $\sum \rho_{1,j} X_{k,j}$
- Σ ρ₁ X_{n,1} is linear

- $C_{1.S_k} \ge C_{1.S_{k-1}} + p_{1.S_k}$ is written
- \bullet $F_{1,k} \geq F_{1,k-1} + p_{1,S_{k}}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})

- - Mathematical modeling

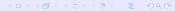
14 / 24

- $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$ is written
- $F_{1,k} \geq F_{1,k-1} + p_{1,S_k}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})
- S_k is replaced by the row $(X_{k,1}, X_{k,2}, \dots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \dots, X_{k,n}) = (0, \dots, 0, 1, 0, \dots, 0)$
- The 1 has the S_k^{th} position

- $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$ is written
- $F_{1,k} \geq F_{1,k-1} + p_{1,S_k}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})
- S_k is replaced by the row $(X_{k,1}, X_{k,2}, \dots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the S_k^{th} position
 - Thus p_{1,S_k} is $\sum_{i=1}^n p_{1,j} X_{k,j}$
-) p_{1,1}X_{x,1} is linear

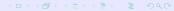


- $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$ is written
- $F_{1,k} \geq F_{1,k-1} + p_{1,S_k}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})
- S_k is replaced by the row $(X_{k,1}, X_{k,2}, \dots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the S_k^{th} position
- Thus p_{1,S_k} is $\sum_{i=1}^{n} p_{1,i} X_{k,i}$
- $\sum_{i=1}^n p_{1,j} X_{k,j}$ is linear



- $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$ is written
- $F_{1,k} \geq F_{1,k-1} + p_{1,S_k}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})
- S_k is replaced by the row $(X_{k,1}, X_{k,2}, \dots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the S_k^{th} position
- Thus p_{1,S_k} is $\sum_{i=1}^{n} p_{1,i} X_{k,i}$





- $C_{1,S_k} \ge C_{1,S_{k-1}} + p_{1,S_k}$ is written
- $F_{1,k} \geq F_{1,k-1} + p_{1,S_k}$
- We need p_{1,S_k} (and p_{2,S_k} and d_{S_k})
- S_k is replaced by the row $(X_{k,1}, X_{k,2}, \dots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the S_k^{th} position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\bullet \sum_{j=1}^{n} p_{1,j} X_{k,j} \text{ is linear}$



The first job:

Constraints

$$F_{1,1} = \sum_{j=1}^{n} p_{1,j} X_{1,j}$$
 (8)

$$F_{2,1} = \sum_{j=1}^{n} (p_{1,j} + p_{2,j}) X_{1,j}$$
 (9)

The other jobs on machine 1:

$$\forall k = 2..n$$
 $F_{1,k} = F_{1,k-1} + \sum_{j=1}^{n} p_{1,j} X_{k,j}$ (10)

The other jobs on machine 2

$$\forall k = 2..n$$
 $F_{2,k} \ge F_{2,k-1} + \sum_{j=1}^{n} p_{2,j} X_{k,j}$ (11)

The first job:

Constraints

$$F_{1,1} = \sum_{i=1}^{n} p_{1,j} X_{1,j}$$
 (8)

$$F_{2,1} = \sum_{j=1}^{n} (p_{1,j} + p_{2,j}) X_{1,j}$$
 (9)

The other jobs on machine 1:

$$\forall k = 2..n$$
 $F_{1,k} = F_{1,k-1} + \sum_{j=1}^{n} p_{1,j} X_{k,j}$ (10)

The other jobs on machine 2

$$\forall k = 2..n$$
 $F_{2,k} \ge F_{2,k-1} + \sum_{j=1}^{n} p_{2,j} X_{k,j}$ (11)

Constraints

The first job:

$$F_{1,1} = \sum_{j=1}^{n} p_{1,j} X_{1,j}$$
 (8)

$$F_{2,1} = \sum_{j=1}^{n} (p_{1,j} + p_{2,j}) X_{1,j}$$
 (9)

The other jobs on machine 1:

$$\forall k = 2..n$$
 $F_{1,k} = F_{1,k-1} + \sum_{j=1}^{n} p_{1,j} X_{k,j}$ (10)

The other jobs on machine 2:

$$\forall k = 2..n$$
 $F_{2,k} \ge F_{2,k-1} + \sum_{j=1}^{n} p_{2,j} X_{k,j}$ (11)

The other jobs from machine 1 to machine 2:

$$\forall k = 2..n \qquad F_{2,k} \ge F_{1,k} + \sum_{j=1}^{n} p_{2,j} X_{k,j}$$
 (12)

$$\forall k = 1..n$$
 $G_k \ge F_{2,k} - \sum_{j=1}^n d_j X_{k,j}$ (13)

$$Minimize \sum_{k=1}^{n} G_k$$
 (14)

The other jobs from machine 1 to machine 2:

$$\forall k = 2..n \qquad F_{2,k} \ge F_{1,k} + \sum_{j=1}^{n} p_{2,j} X_{k,j}$$
 (12)

The tardinesses:

$$\forall k = 1..n$$
 $G_k \ge F_{2,k} - \sum_{j=1}^n d_j X_{k,j}$ (13)

The criterion

$$Minimize \sum_{k=1}^{n} G_k$$
 (14)



The other jobs from machine 1 to machine 2:

$$\forall k = 2..n \qquad F_{2,k} \ge F_{1,k} + \sum_{j=1}^{n} p_{2,j} X_{k,j}$$
 (12)

The tardinesses:

$$\forall k = 1..n$$
 $G_k \ge F_{2,k} - \sum_{j=1}^n d_j X_{k,j}$ (13)

The criterion:

$$Minimize \sum_{k=1}^{n} G_k$$
 (14)

• The problem defined by:

- Variables are $(X_1, \ldots, X_n) = X$
- Constraints are $\sum_{j=1}^{n} a_{i,j} X_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_j$
- Minimize (or Maximize) $f(X) = \sum_{i=1}^{n} c_i X_i$
- is a linear problem if all the X/s are in IR
- It is said a mixed integer linear problem if
 - some X/s are in

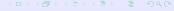
- The problem defined by:
- Variables are $(X_1, \ldots, X_n) = X$

• Constraints are
$$\sum_{j=1}^{n} a_{i,j} X_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_j$$

- Minimize (or Maximize) $f(X) = \sum_{j=1}^{n} c_j X_j$
- Is a linear pro

- The problem defined by:
- Variables are $(X_1, \ldots, X_n) = X$
- Constraints are $\sum_{j=1}^{n} a_{i,j} X_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_j$
- Minimize (or Maximize) $f(X) = \sum_{j=1}^{n} c_j X_j$
- ullet is a linear problem if all the X_i 's are in $\mathbb R$

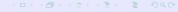
- The problem defined by:
- Variables are $(X_1, \ldots, X_n) = X$
- Constraints are $\sum_{j=1}^{n} a_{i,j} X_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_j$
- Minimize (or Maximize) $f(X) = \sum_{j=1}^{n} c_j X_j$
- \circ is a linear problem if all the X_i 's are in $\mathbb R$
- It is said a mixed integer linear problem if some X_i's are in Z



- The problem defined by:
- Variables are $(X_1, \ldots, X_n) = X$
- Constraints are $\sum_{j=1}^{n} a_{i,j} X_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_j$
- Minimize (or Maximize) $f(X) = \sum_{j=1}^{n} c_j X_j$
- is a linear problem if all the X_i 's are in $\mathbb R$
- It is said a mixed integer linear problem if some X_i's are in Z



- The problem defined by:
- Variables are $(X_1, \ldots, X_n) = X$
- Constraints are $\sum_{j=1}^{n} a_{i,j} X_j \begin{cases} \leq \\ = \\ \geq \end{cases} b_j$
- Minimize (or Maximize) $f(X) = \sum_{j=1}^{n} c_j X_j$
- is a linear problem if all the X_i 's are in $\mathbb R$
- It is said a mixed integer linear problem if some X_i 's are in \mathbb{Z}



- For the LP problems (continuous variables)
- proved polynomial (Khachiyan 1979)
- very efficient methods:
- Simplex method (Dantzig, 1947)
- Interior-point method (Karmarcar, 1984)

- For the LP problems (continuous variables)
- proved polynomial (Khachiyan 1979)
- very efficient methods:
- Simplex method (Dantzig, 1947)
- о интентог-роинт итентом (кантнансан, тэоч)

- For the LP problems (continuous variables)
- proved polynomial (Khachiyan 1979)
- very efficient methods:
- Simplex method (Dantzig, 1947)
- Interior-point method (Karmarcar, 1984)

- For the LP problems (continuous variables)
- proved polynomial (Khachiyan 1979)
- very efficient methods:
- Simplex method (Dantzig, 1947)
- Interior-point method (Karmarcar, 1984)

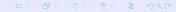
- For the LP problems (continuous variables)
- proved polynomial (Khachiyan 1979)
- very efficient methods:
- Simplex method (Dantzig, 1947)
- Interior-point method (Karmarcar, 1984)

- For the ILP problems (integer variables)
- NP-hard
- Branch & bound, branch & cut, branch & price

- For the ILP problems (integer variables)
- NP-hard
- Branch & bound, branch & cut, branch & price

. . . .

- For the ILP problems (integer variables)
- NP-hard
- Branch & bound, branch & cut, branch & price



Optim. Meth.

- For the ILP problems (integer variables)
- NP-hard
- Branch & bound, branch & cut, branch & price
-

- Solvers
- GLPK
- CPLEX
- MATLAB
- 0
- Shreadsheats: OnenOffice Event

- Solvers
- GLPK
- CPLEX
- MATLAB
- 0
- Sinreadsheats: OpenOffice Eyean

- Solvers
- GLPK
- CPLEX
- MATLAB
-
- Smeadsheater ChenCiffice Evos

- Solvers
- GLPK
- CPLEX
- MATLAB

. . . .

Spreadsheets: OpenOffice, Exce

- Solvers
- GLPK
- CPLEX
- MATLAB
- . . .
- Spreadsheets: OpenOffice, Exce

- Solvers
- GLPK
- CPLEX
- MATLAB
-
- Spreadsheets: OpenOffice, Excel

- This provides an exact method

- This provides an exact method
- therefore only fo small instances
- otherwise provides an approximation method with a time limit

- This provides an exact method
- therefore only fo small instances
- otherwise provides an approximation method with a time limit

- Models are written in the language GNU MathProg
- Data as well
- Then run GLPSOL (model1 or model2) to get an optimal solution
- otherwise, with a time limit, get an suboptimal solution
- or with the relaxed LP problem, get a lower bound

- Models are written in the language GNU MathProg
- Data as well
- Then run GLPSOL (model1 or model2) to get an optimal solution
- otherwise, with a time limit, get an suboptimal solution
- on with the relaxed LP problem, get a lower bound

- Models are written in the language GNU MathProg
- Data as well
- Then run GLPSOL (model1 or model2) to get an optimal solution
- otherwise, with a time limit, get an suboptimal solution
- or with the relaxed LP problem, get a lower bound

- Models are written in the language GNU MathProg
- Data as well
- Then run GLPSOL (model1 or model2) to get an optimal solution
- otherwise, with a time limit, get an suboptimal solution
- or with the relaxed LP problem, get a lower bound

- Models are written in the language GNU MathProg
- Data as well
- Then run GLPSOL (model1 or model2) to get an optimal solution
- otherwise, with a time limit, get an suboptimal solution
- or with the relaxed LP problem, get a lower bound

- Solvers can also be handled with a programming language
- thanks to a library
- C++, C, Python, Java, . . .
- A glance to Python and its PULP module
- Similarly to C++ and its CPLEX library
- Beside to OpenOffice

- Solvers can also be handled with a programming language
- thanks to a library
- C++, C, Python, Java, ...
- A glance to Python and its PULP module
- Similarly to C++ and its CPLEX library
- Beside to OpenOffice

- Solvers can also be handled with a programming language
- thanks to a library
- C++, C, Python, Java, . . .

- Solvers can also be handled with a programming language
- thanks to a library
- C++, C, Python, Java, . . .
- A glance to Python and its PULP module
- Similarly to C++ and its CPLEX library
- Beside, to OpenOffice

- Solvers can also be handled with a programming language
- thanks to a library
- C++, C, Python, Java, ...
- A glance to Python and its PULP module
- Similarly to C++ and its CPLEX library
- Beside, to OpenOffice

- Solvers can also be handled with a programming language
- thanks to a library
- C++, C, Python, Java, . . .
- A glance to Python and its PULP module
- Similarly to C++ and its CPLEX library
- Beside, to OpenOffice