Mathematical modeling

Jean-Louis Bouquard

Beijing Institute of Technology

Optimization Methods

Content

1. Mathematical modeling
2. Solving linear problems

1. Mathematical modeling

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

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 LINEAR
- Do not confuse variables (unknown, decisions) and data (known)
- Do not confuse math-model-variables and algorithm-variables

A first model for the $F2//\sum T_j$

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

Succession of the operations and tardinesses:

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

Objective function:

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

A first model for the F2//$\sum T_j$

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

Succession of the operations and tardinesses:

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

Objective function:

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

A first model for the F2//$\sum T_j$

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

Succession of the operations and tardinesses:

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

Objective function:

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

A first model for the F2//$\sum T_j$

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

Succession of the operations and tardinesses:

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

Objective function:

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

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \text{if } j_1 \text{ before } j_2 \\ 1 & \text{if } j_1 \text{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \textit{if } j_1 \textit{ before } j_2 \\ 1 & \textit{if } j_1 \textit{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \textit{if } j_1 \textit{ before } j_2 \\ 1 & \textit{if } j_1 \textit{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \textit{if } j_1 \textit{ before } j_2 \\ 1 & \textit{if } j_1 \textit{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \textit{if } j_1 \textit{ before } j_2 \\ 1 & \textit{if } j_1 \textit{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \text{if } j_1 \text{ before } j_2 \\ 1 & \text{if } j_1 \text{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

- 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 $C_{i,j_1} \leq C_{i,j_2} - p_{i,j_2}$
- 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
- $Z_{j_1,j_2} = \begin{cases} 0 & \text{if } j_1 \text{ before } j_2 \\ 1 & \text{if } j_1 \text{ after } j_2 \end{cases}$

A first model for the F2//$\sum T_j$

Non overlap constraints:

$$\forall (j_1, j_2), 1 \leq j_1 < j_2 \leq n, \quad \begin{array}{l} C_{1,j_1} \leq C_{1,j_2} - p_{1,j_2} \;\; \textit{if} \;\; Z_{j_1,j_2} = 0 \\ C_{1,j_2} \leq C_{1,j_1} - p_{1,j_1} \;\; \textit{if} \;\; Z_{j_1,j_2} = 1 \end{array}$$

They can be turned into:

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

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

A first model for the F2//$\sum T_j$

Non overlap constraints:

$$\forall(j_1,j_2), 1 \leq j_1 < j_2 \leq n, \quad \begin{array}{l} C_{1,j_1} \leq C_{1,j_2} - p_{1,j_2} \;\; \textit{if} \;\; Z_{j_1,j_2} = 0 \\ C_{1,j_2} \leq C_{1,j_1} - p_{1,j_1} \;\; \textit{if} \;\; Z_{j_1,j_2} = 1 \end{array}$$

They can be turned into:

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

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}):

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

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
3				0	1	0
4					1	1
5						0
6						

A first model for the F2//$\sum T_j$

Exclusion on each machine:

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

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \qquad (2)$$
$$C_{2,j} \geq C_{1,j} + p_{2,j} \qquad (3)$$
$$T_j \geq C_{2,j} - d_j \qquad (4)$$

Objective function

$$Minimize \quad \sum_{j=1}^{n} T_j \qquad (5)$$

A first model for the F2//$\sum T_j$

Exclusion on each machine:

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

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \qquad (2)$$
$$C_{2,j} \geq C_{1,j} + p_{2,j} \qquad (3)$$
$$T_j \geq C_{2,j} - d_j \qquad (4)$$

Objective function

$$Minimize \quad \sum_{j=1}^{n} T_j \qquad (5)$$

A first model for the F2//$\sum T_j$

Exclusion on each machine:

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

Succession of the operations and tardinesses:

$$\forall j, 1 \leq j \leq n, \quad C_{1,j} \geq p_{1,j} \quad (2)$$
$$C_{2,j} \geq C_{1,j} + p_{2,j} \quad (3)$$
$$T_j \geq C_{2,j} - d_j \quad (4)$$

Objective function

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

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

- 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 $j = S_k$?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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 $j = S_k$?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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 $j = S_k$?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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 $j = S_k$?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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 $j = S_k$?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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 $j = S_k$?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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* $j = S_k$ *?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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* $j = S_k$ *?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

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

- 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* $j = S_k$ *?*"
- we associate the binary variable $X_{k,j}$
- $X_{k,j} = \begin{cases} 1 & \textit{if } j \textit{ has rank } k \\ 0 & \textit{otherwise} \end{cases}$

A second model
Constraints

- Every rank is assigned to some job:

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

- Every job has some rank:

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

A second model
Constraints

- Every rank is assigned to some job:

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

- Every job has some rank:

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

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

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

A second model

- 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, $G_k \geq 0$

A second model

- 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, $G_k \geq 0$

A second model

- 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, $G_k \geq 0$

A second model

- 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, $G_k \geq 0$

A second model

- 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, $G_k \geq 0$

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the $S_k{}^{th}$ position
- Thus p_{1,S_k} is $\sum\limits_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum\limits_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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
- Thus p_{1,S_k} is $\sum\limits_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum\limits_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the $S_k{}^{th}$ position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the ${S_k}^{th}$ position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the ${S_k}^{th}$ position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the ${S_k}^{th}$ position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the ${S_k}^{th}$ position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model

- $C_{1,S_k} \geq C_{1,S_{k-1}} + p_{1,S_k}$ 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 $(X_{k,1}, X_{k,2}, \ldots, X_{k,n})$
- $(X_{k,1}, X_{k,2}, \ldots, X_{k,n}) = (0, \ldots, 0, 1, 0, \ldots, 0)$
- The 1 has the ${S_k}^{th}$ position
- Thus p_{1,S_k} is $\sum_{j=1}^{n} p_{1,j} X_{k,j}$
- $\sum_{j=1}^{n} p_{1,j} X_{k,j}$ is linear

A second model
Constraints

The first job:

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

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

The other jobs on machine 1:

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

The other jobs on machine 2:

$$\forall k = 2..n \qquad F_{2,k} \geq F_{2,k-1} + \sum_{j=1}^{n} p_{2,j} X_{k,j} \tag{11}$$

A second model
Constraints

The first job:

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

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

The other jobs on machine 1:

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

The other jobs on machine 2:

$$\forall k = 2..n \qquad F_{2,k} \geq F_{2,k-1} + \sum_{j=1}^{n} p_{2,j} X_{k,j} \tag{11}$$

A second model
Constraints

The first job:

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

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

The other jobs on machine 1:

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

The other jobs on machine 2:

$$\forall k = 2..n \qquad F_{2,k} \geq F_{2,k-1} + \sum_{j=1}^{n} p_{2,j} X_{k,j} \tag{11}$$

A second model

The other jobs from machine 1 to machine 2:

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

The tardinesses:

$$\forall k = 1..n \qquad G_k \geq F_{2,k} - \sum_{j=1}^{n} d_j X_{k,j} \tag{13}$$

The criterion:

$$Minimize \quad \sum_{k=1}^{n} G_k \tag{14}$$

A second model

The other jobs from machine 1 to machine 2:

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

The tardinesses:

$$\forall k = 1..n \qquad G_k \geq F_{2,k} - \sum_{j=1}^{n} d_j X_{k,j} \tag{13}$$

The criterion:

$$Minimize \quad \sum_{k=1}^{n} G_k \tag{14}$$

A second model

The other jobs from machine 1 to machine 2:

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

The tardinesses:

$$\forall k = 1..n \qquad G_k \geq F_{2,k} - \sum_{j=1}^{n} d_j X_{k,j} \tag{13}$$

The criterion:

$$Minimize \quad \sum_{k=1}^{n} G_k \tag{14}$$

2 Solving linear problems

Linear problems

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

Linear problems

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

Linear problems

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

Linear problems

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

Linear problems

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

Linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

Solving linear problems

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

The solver GLPK

- 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

The solver GLPK

- 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

The solver GLPK

- 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

The solver GLPK

- 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

The solver GLPK

- 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

Solving linear problems
Using GLPK

- 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

Solving linear problems
Using GLPK

- 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

Solving linear problems
Using GLPK

- 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

Solving linear problems
Using GLPK

- 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

Solving linear problems

Using GLPK

- 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

Solving linear problems
Using GLPK

- 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