PQI-5884 - Programação Inteira Mista Aplicada à Otimização de Processos
3º Período 2020

Data	Atividade	Conteúdo
17/09	Aula 1	Introdução, formulação, classes, representação
24/09	Aula 2	Condições de otimalidade
01/10	Aula 3	Condições KKT, multiplicadores
08/10	Aula 4	Otimização irrestrita
15/10	Aula 5	LP
22/10	Aula 6	NLP
29/10	Aula 7	MILP
05/10	Aula 8	MILP, problemas clássicos
12/11	**Aula 9**	**MILP, problema de scheduling**
19/11	Aula 10	MINLP, problema de síntese
26/11	-	-
03/12	Aula 11	Apresentações

MILP: Mixed Integer Linear Programming

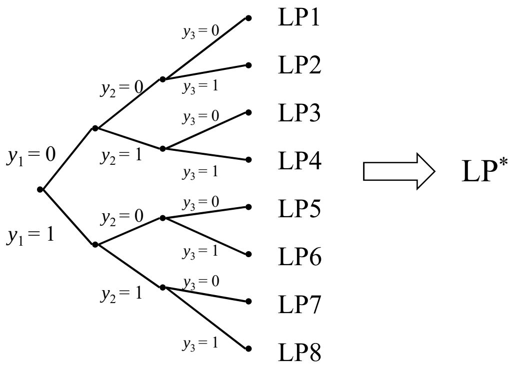
$$
\begin{aligned}
\text{min} \quad & z = \underline{c}_1^{\mathrm{T}}.\underline{x} + \underline{c}_2^{\mathrm{T}}.\underline{y} \\
\text{s.a.} \quad & \underline{\underline{A}}.\underline{x} + \underline{\underline{B}}.\underline{y} = \underline{d}_1 \\
& \underline{\underline{C}}.\underline{x} + \underline{\underline{D}}.\underline{y} \leq \underline{d}_2 \\
& \underline{x} \in \Re^n \\
& \underline{y} \in \{0, 1\}^m
\end{aligned}
$$

$$
y_i = \begin{cases} 1 \text{ se a proposição } \mathrm{Y}_i \text{ for verdadeira} \\ 0 \text{ caso contrário} \end{cases}
$$

2

Técnicas de solução MILP

ENUMERAÇÃO EXAUSTIVA



m	10	20	30	40	50	60	70
2^m	1,0E03	1,0E06	1,1E09	1,1E12	1,1E15	1,2E18	1,2E21
tempo	6E–7 s	6E–4 s	0,6 s	11 min	7,7 dias	21 anos	22 milênios

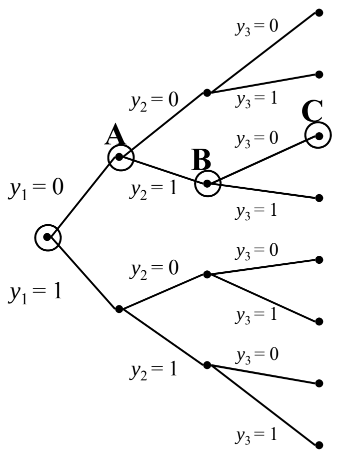
RELAXAÇÃO

$$y_i \in \{0,1\} \quad \Longrightarrow \quad \begin{array}{l} 0 \le y_i \le 1 \\ y_i \in \Re \end{array}$$

$$z_{relax}^* \le z^*$$
$$gap = z^* - z_{relax}^*$$

MILP	MILP relaxado como LP
min $z = -2{,}3.y_1 - 2{,}0.y_2$ s.a.: $y_1 + y_2 \le 1$ $1{,}2.y_1 + 0{,}4.y_2 \le 1$ $y_1, y_2 \in \{0,1\}$	min $z_{relax} = -2{,}3.y_1 - 2{,}0.y_2$ s.a.: $y_1 + y_2 \le 1$ $1{,}2.y_1 + 0{,}4.y_2 \le 1$ $0 \le y_1, y_2 \le 1$ $y_1, y_2 \in \Re^1$
Solução: $y_1^* = 0$ $y_2^* = 1$ $z^* = -1{,}00$	Solução: $y_1^* = 0{,}75$ $y_2^* = 0{,}25$ $z_{relax}^* = -2{,}23$ Arredondamento não garante viabilidade nem otimalidade

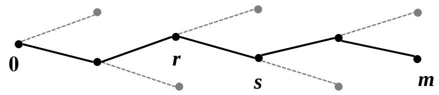
BRANCH-AND-BOUND



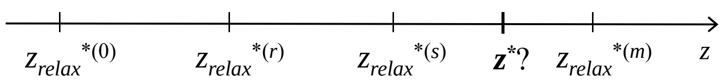
Resolução sequencial de problemas relaxados com especificação de variáveis binárias

Exemplo: Nó B consistes do problema relaxado com
$y_1 = 0$ fixo
$y_2 = 1$ fixo

Estratégias de exploração da árvore



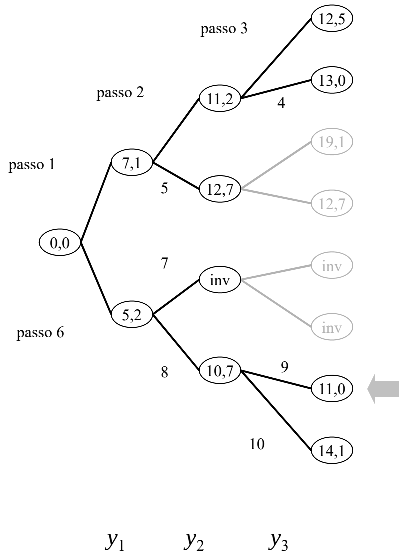
- Se inviável em '*r*', então inviável em '*s*'.
- Sendo '*r*' e '*s*' problemas viáveis: $z_{relax}{}^{*(r)} \geq z_{relax}{}^{*(s)}$
- Sendo '*m*' um nó final viável: $z_{relax}{}^{*(m)} \geq z^*$



Estratégia *depth-first* ou LIFO (*last in first out*): a busca é realizada em profundidade.

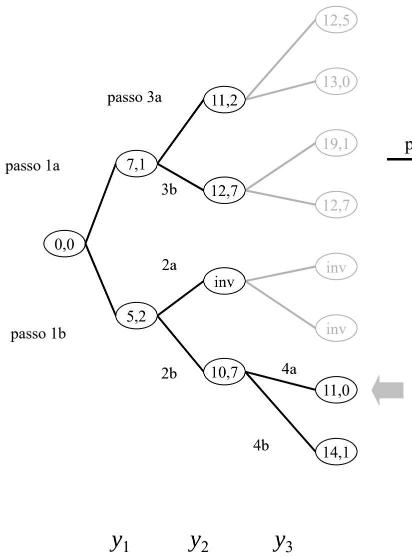
Estratégia *breadth-first*: a busca é realizada em amplitude (largura), expandindo ramos a partir de nós selecionados.

Exemplo de busca depth-first



passo	$z_{relax}^{*(m)}$
0	-
1	-
2	-
3	12,5
4	12,5
5	12,5
6	12,5
7	12,5
8	12,5
9	11,0
10	**11,0 ótimo**

Exemplo de busca breadth-first



passo	$z_{relax}^{*(k)}$
0	0,0
1	5,2 / 7,1
2	7,1 / 10,7 / inv
3	10,7 / 11,2 / 12,7 / inv
4	**11,0** / 11,2 / 12,7 / 14,1 / inv **ótimo**

MÉTODOS RIGOROSOS MILP

a) *Branch-and-Bound* (Land, Doig, 1960) – método clássico de busca implícita em uma árvore binária resolvendo uma sequência de MILPs relaxados.

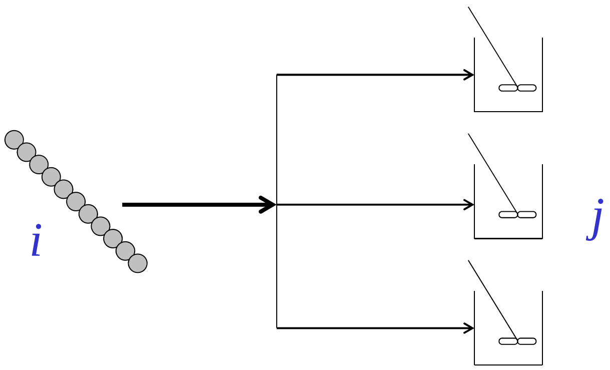
b) Decomposição de Benders (Benders, 1962) – decomposição do problema baseada na partição entre variáveis binárias e contínuas.

c) *Branch-and-Cut* (Crowder, Johnson, Padberg, 1983) – combinação do método *Branch-and-Bound* com o de planos cortantes (*cutting planes*) melhorando a solução do problema relaxado a cada iteração.

CASO MILP: PROGRAMAÇÃO DE PRODUÇÃO EM UNIDADES DE BATELADA (PÁG.26)

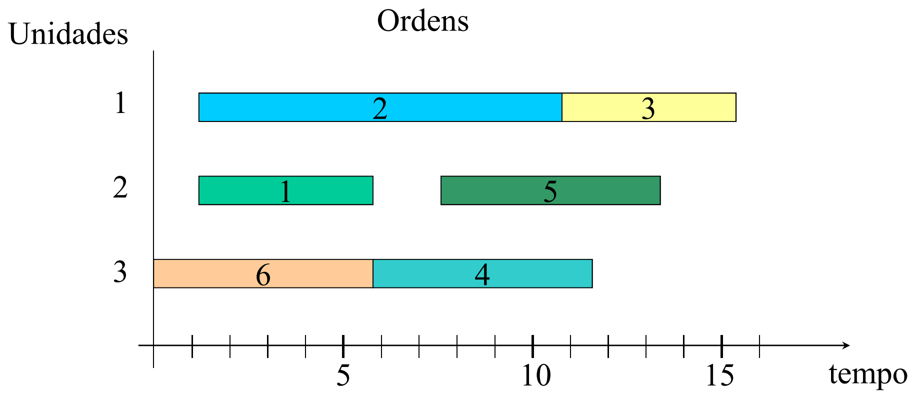
Mendez, Henning, Cerdá (2000)

Objetivo: Sequenciamento de *n* ordens de produção em uma planta multiproduto com *m* unidades em batelada



Unidades em paralelo
Estágio único de produção

Objetivo: Diagrama de Gantt



Variáveis binárias de decisão:

WF_{ij} ordem i é a primeira processada na unidade j

W_{ij} ordem i é processada na unidade j e não é a 1ª

$X_{ii'}$ ordem i' é processada logo após ordem i

Variáveis reais de decisão:

C_i instante de conclusão da ordem i

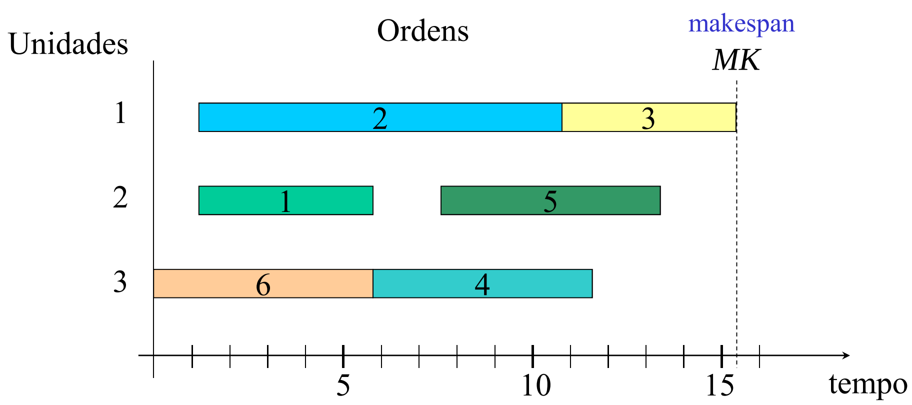
Parâmetros:

t_{ij} tempo de processamento de i na unidade j

$s_{ii'}$ tempo de preparação para processar i' após i

d_i data para entrega da ordem i

Exemplo:



Variáveis binárias verdadeiras:

WF_{21}, WF_{12}, WF_{63}

W_{31}, W_{52}, W_{43}

X_{23}, X_{15}, X_{64}

Variáveis reais

$C_1 = 5{,}8$ $\quad C_2 = 10{,}9$

$C_3 = 15{,}3$ $\quad C_4 = 11{,}6$

$C_5 = 13{,}3$ $\quad C_6 = 5{,}8$

Informações sobre incompatibilidades

Define-se como *FS* o grupo de sequências que não podem ser feitas.

$$X_{ii'} = 0 \qquad \text{para } (i,i') \in FS \qquad (1)$$

Define-se como *FP* o grupo das alocações proibidas de ordem *i* em unidade *j*.

$$WF_{ij} + W_{ij} = 0 \qquad \text{para } (i,j) \in FP \qquad (2)$$

Modelagem

Cada ordem i deve ser alocada a uma única unidade de processamento j :

$$\sum_j WF_{ij} + \sum_j W_{ij} = 1 \qquad \forall \text{ ordem } i \qquad (3)$$

Uma ordem, e sua sucessora direta, devem ser processadas na mesma unidade. Sempre que a ordem i for sucedida pela ordem i' ($X_{ii'} = 1$), ambas devem ser processadas na mesma unidade:

$$\left[\left(WF_{ij} \vee W_{ij}\right) \wedge X_{ii'}\right] \Rightarrow W_{i'j} \qquad (4)$$

$$WF_{ij} + W_{ij} \leq W_{i'j} + 1 - X_{ii'} \qquad \forall\, j, (ii') \qquad (5)$$

Cada unidade j só pode ter no máximo uma ordem inicial:

$$\sum_i WF_{ij} \leq 1 \qquad \forall \text{ unidade } j \qquad (6)$$

Toda ordem deve: ser a primeira a ser processada, ou então ser precedida por outra ordem na sequência:

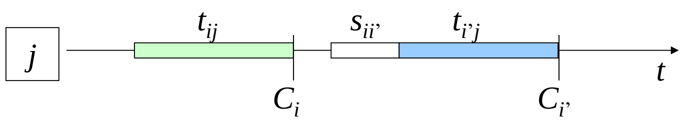
$$\sum_j WF_{i'j} + \sum_i X_{ii'} = 1 \qquad \forall \text{ ordem } i' \qquad (7)$$

Toda ordem *i* deve ser sucedida por no máximo uma ordem na sequência de processamento:

$$\sum_{i'} X_{ii'} \le 1 \qquad \forall \text{ ordem } i \qquad (8)$$

Uma ordem *i'* só pode ser processada após a conclusão da ordem anterior *i*

$$C_{i'} \ge C_i + \left[\sum_j \left(s_{ii'} + t_{i'j}\right) W_{i'j}\right] + M\left(X_{ii'} - 1\right) \qquad \forall\ (i, i') \qquad (9)$$



O processamento da ordem *i* na unidade *j* só pode ser iniciada se as duas estiverem disponíveis:

$$C_i \ge \sum_j \left(\max\left\{ru_j, ro_i\right\} + t_{ij}\right) \cdot \left(WF_{ij} + W_{ij}\right) \qquad \forall \text{ ordem } i \qquad (10)$$

Parâmetros adicionais:

ro_i instante de liberação da ordem *i*

ru_j instante de liberação da unidade *j*

Medidas de desempenho

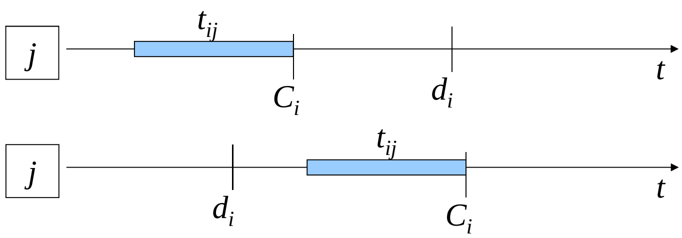
1) Adiantamento da ordem:

$E_i \geq d_i - C_i \qquad \forall\ i \qquad (11)$

2) Atraso da ordem:

$T_i \geq C_i - d_i \qquad \forall\ i \qquad (12)$

d_i = data de entrega da ordem i (parâmetro)
C_i = instante de conclusão da ordem i



3) Número de atrasos:

$T_i \leq M \,.\, NT_i \qquad \forall\ i \qquad (13)$

M = parâmetro big M
NT_i = variável binária para existência de atraso na ordem i

4) Makespan (MK)

$MK \geq C_i \qquad \forall\ i \qquad (14)$

Funções objetivo

1) Minimização do adiantamento ponderado

$$\text{Min} \quad \sum_i p_i E_i \qquad (15)$$

2) Minimização do atraso ponderado

$$\text{Min} \quad \sum_i p_i T_i \qquad (16)$$

p_i = peso da ordem i

Funções objetivo

3) Minimização do número de ordens atrasadas:

$$\text{Min} \sum_i NT_i \qquad (17)$$

4) Minimização de desvios ponderados: este objetivo visa manter valores razoáveis de adiantamento e estoque intermediário.

$$\text{Min} \quad \sum_i p_i \cdot \left(\frac{E_i}{n+1} + T_i \right) \qquad (18)$$

5) Minimização do makespan

$$\text{Min } MK \qquad (19)$$

$$MK \geq C_i \qquad \forall\, i \longrightarrow MK = \max\{C_i\}$$

EM RESUMO

Variáveis

C_i	instante de conclusão da ordem *i* (dias)
E_i	adiantamento da ordem *i* (dias)
T_i	atrasado da ordem *i* (dias)
MK	tempo total de conclusão das ordens, *makespan* (dias)
WF_{ij}	binária, vale 1 se a ordem *i* é a primeira a ser processada na unidade *j*
W_{ij}	binária, vale 1 se a ordem *i* é processada na unidade *j*, não sendo a primeira
$X_{ii'}$	binária, vale 1 se a ordem *i'* é processada após a ordem *i* na mesma unidade
NT_i	binária, vale 1 se a ordem *i* está atrasada

Parâmetros

n	número de ordens
m	número de unidades
t_{ij}	tempo de processamento da ordem *i* na unidade *j* (dias)
d_i	data para entrega da ordem *i* (dias)
M	parâmetro big-M (dias)
p_i	peso relativo da ordem *i* (prioridade)
ro_i	tempo para liberação da ordem *i* (dias)
ru_j	tempo para liberação da unidade *j* para a programação(dias)
$s_{ii'}$	tempo de *setup* para processar a ordem *i'* após a ordem *i* (dias)

Exemplo Numérico

Ordem *i*	Tempo para liberação (dias)	Data de entrega (dias)	Tempo de processamento na unidade 1 (dias)	Tempo de processamento na unidade 2 (dias)	Peso p_i
1	0,0	14	6,8	6,1	1
2	5,0	14	7,0	7,0	1
3	0,0	21	5,0	X	1
4	6,0	14	X	5,1	2
5	0,0	21	4,9	5,6	1

Setup

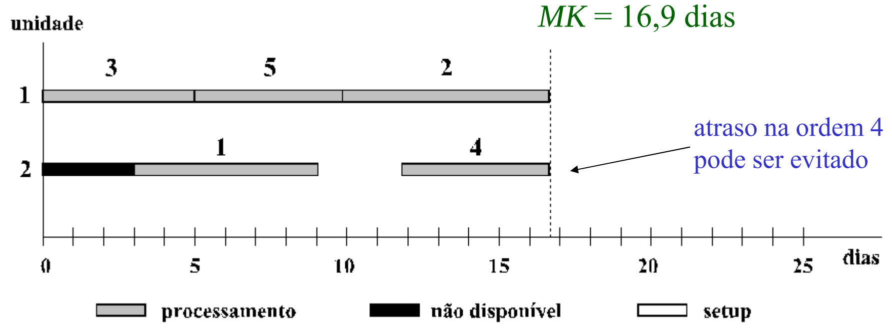
$i \to i'$	1	2	3	4	5
1	-		X		X
2	X	-	1,1		0,7
3	1,0	0,15	-		
4		X		-	X
5	1,4			0,3	-

Unidades:
1) $r_{u1} = 0$ dias
2) $r_{u2} = 3$ dias

Minimização do Makespan

Unidade	Ordem	Tempo de proces. (dias)	Tempo de setup (dias)	Início da ordem (dias)	Final da ordem (dias)	Data de entrega (dias)	Atraso (dias)	Adiant. (dias)
1	3	5,0	-	0,0	5,0	21	-	16,0
	5	4,9	0	5,0	9,9	21	-	11,1
	2	7,0	0	9,9	16,9	14	2,9	-
2	1	6,1	-	3,0	9,1	14	-	4,9
	4	5,1	0	11,8	16,9*	14	2,9	-

MK = 16,9 dias

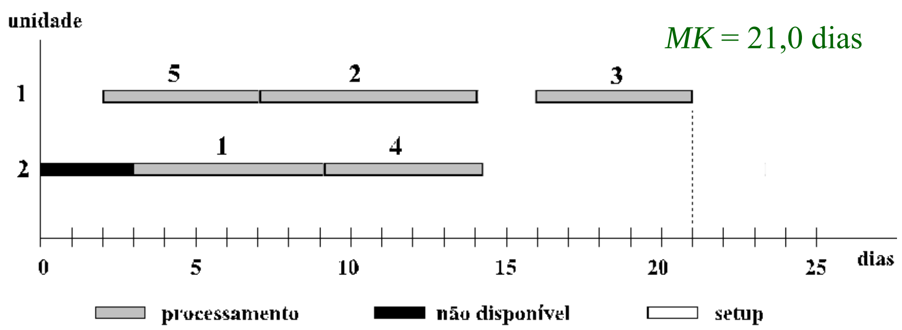


desvio = 14,0 dias

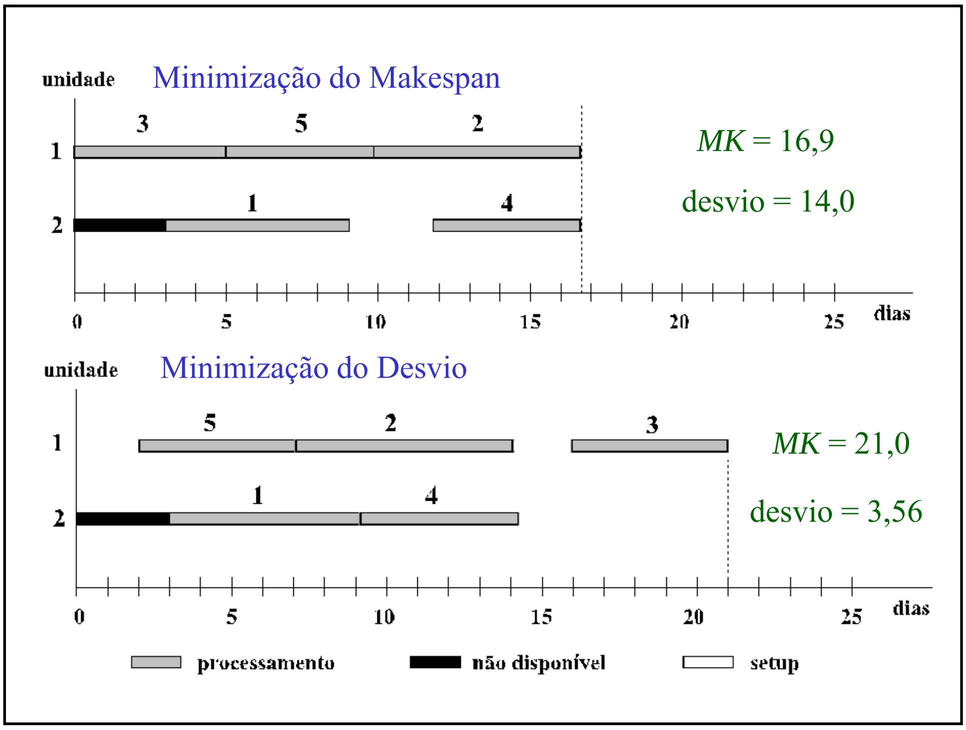
Minimização do Desvio

Unidade	Ordem	Data de entrega	Atraso
1	5	21	–14
	2	14	0
	3	21	0
2	1	14	–4,9
	4	14	+0,2

MK = 21,0 dias



desvio = 3,56


unidade
Minimização do Makespan
3
5
2
1
2
1
4
MK = 16,9
desvio = 14,0
0
5
10
15
20
25
dias
unidade
Minimização do Desvio
5
2
3
1
2
1
4
MK = 21,0
desvio = 3,56
0
5
10
15
20
25
dias
processamento
não disponível
setup