a cultura do rigor

Departamento de Informática

Programação Inteira

Métodos Quantitativos

LEI

2006/2007

Advertência

- Autores
  - João Moura Pires (jmp@di.fct.unl.pt)
  - Susana Nascimento (snt@di.fct.unl.pt)
- Este material pode ser livremente usado para uso pessoal ou académico e sem qualquer autorização prévia dos autores desde que acompanhado desta declaração dos autores.
- Para uso comercial (por exemplo em cursos pagos) o uso deste material requer a expressa autorização dos autores.

2

Programação Linear Inteira

- Programação Linear - PL
- Programação Linear Inteira - PLI (ou apenas PI)
  - PLI = PL + Todas as variáveis **inteiras**
- Programação Linear Binária - PLB (ou apenas PB)
  - Caso particular de PI em que o domínio das variáveis é {0, 1}
- Programação Linear Inteira Mista
  - Algumas variáveis são inteiras
  - Algumas variáveis são reais

Motivação e diferença

- Variáveis de decisão representam quantidades inteiras
  - Pessoas, máquinas, etc.
- Problemas evolvendo uma várias decisões "sim/não" inter-relacionadas

---

- O princípio da divisibilidade já não existe.

Um exemplo de Programação Inteira

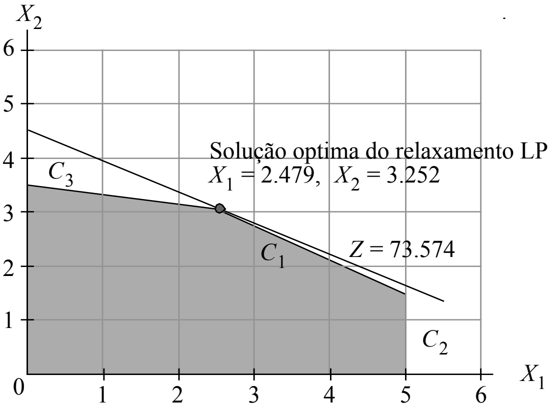
Maximizar $Z = 10\,x_1 + 15\,x_2$

C$_1$: $282\,x_1 + 400\,x_2 \leq 2000$
C$_2$: $x_1 \leq 5$
C$_3$: $4\,x_1 + 40\,x_2 \leq 140$

$x_1, x_2 \geq 0$ e inteiros

5

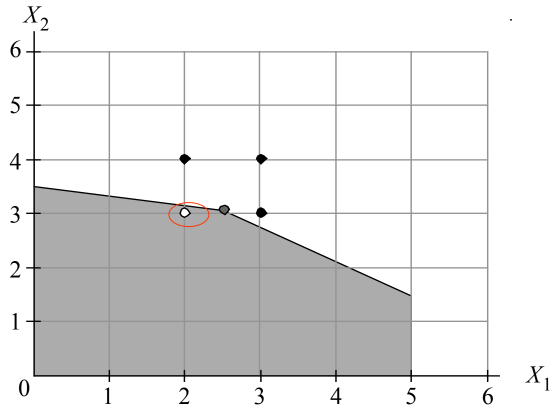
Relaxamento linear



C$_1$: $282\,x_1 + 400\,x_2 \leq 2000$
C$_2$: $x_1 \leq 5$
C$_3$: $4x_1 + 40\,x_2 \leq 140$

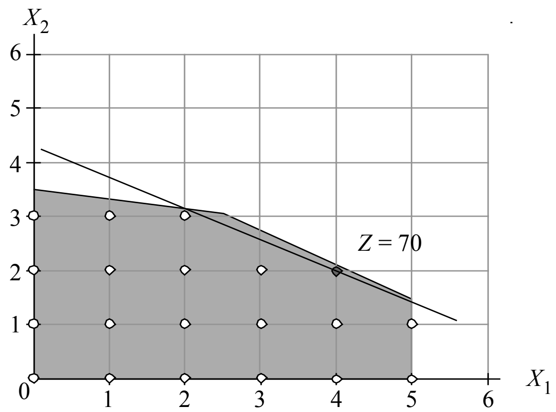
6

Arredondar a solução óptima do problema relaxado



X1	X2	Z
2.5	3.3	74
X1	X2	Z
2	3	65
2	4	80
3	3	75
3	4	90

Solução óptima do ILP



X1	X2	Z
2.5	3.3	74
X1	X2	Z
2	3	65
2	4	80
3	3	75
3	4	90
4	**2**	**70**

PL pode dar limite superior?

O óptimo (maximização) do problema de PI é menor ou igual do que o óptimo do problema relaxado (ex: 65 ≤ 74)

Primeiras impressões

- PI é muito mais difícil do que PL
  - PL : milhares de variáveis
  - PI: poucas centenas de variáveis
- Para se resolver um problema PI podemos começar por resolver a sua forma relaxada (retirando a restrição de as variáveis serem inteiras)
  - O valor óptimo do problema relaxado é melhor ou igual que o valor óptimo do problema inteiro.
  - Avaliar as soluções inteiras por "arredondamento" de uma solução real.

9

Um exemplo de decisões do tipo "sim/não"

Uma empresa pretende investir numa nova fábrica em Los Angeles ou em San Francisco (ou em ambas as cidades).

Considera além disso a construção de quando muito um armazém, o qual deverá ficar localizado na mesma cidade onde for construída uma nova fábrica.

O capital disponível para investir é de 10 milhões de dólares

Var	Questão	Retorno	Investimento
X_1	Fábrica em Los Angeles?	9	6
X_2	Fábrica em San Francisco?	5	3
X_3	Armazém em Los Angeles?	6	5
X_4	Armazém em San Francisco?	4	2

* Retorno já inclui o valor do Investimento e estão ambos em milhões de dólares

10

Um exemplo de decisões do tipo "sim/não" - variáveis binárias -

- X_j = {1 se a decisão j for sim; 0 se a decisão j for não}
- X_3 e X_4 são **mutuamente exclusivas** (apenas um armazém)

$$\boxed{X_3 + X_4 \leq 1}$$

- X_3 e X_4 são **decisões contingentes** das decisões de X_1 e X_2 (apenas se constrói um armazém numa cidade onde também se vai construir uma fábrica)
  - Se $X_1 = 0$ então $X_3 = 0$ $\boxed{X_3 \leq X_1}$
  - Se $X_2 = 0$ então $X_4 = 0$ $\boxed{X_4 \leq X_2}$

Um exemplo de decisões do tipo "sim/não" - BIP-

Formular o problema

Maximizar $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$ (investimento)

$x_3 + x_4 \leq 1$ (máximo um armazém)

$-x_1 + x_3 \leq 0$ (contingência)

$-x_2 + x_4 \leq 0$ (contingência)

e x_j é binário para j = 1, 2, 3, 4

Var	Ret	Inv
X_1	9	6
X_2	5	3
X_3	6	5
X_4	4	2

Ver outros exemplos secção 12.2

- Vamos introduzir variáveis binárias na formulação de modelos de PL ou PI com restrições.
  - Por exemplo, em problemas com variáveis inteiras ou reais com disparidades que envolvem relacções combinatórias de elementos do modelo.
- Vamos discutir alguns casos em que os x_j´s são variáveis de decisão originais, e em que os y_i´s são variáveis binárias auxiliares.

- Modelação de restrições disjuntas
- Funções objectivo com *N* valores possíveis
- Modelação de Problema de custo fixo
- Outros ...

13

Modelação de restrições disjuntas - apenas 1 de entre 2 restrições -

- Apenas uma de entre duas restrições deve ser satisfeita (embora possam as duas ser satisfeitas)

$$C_1: f_1(x_1, x_2, \ldots, x_n) \le d_1$$
$$C_2: f_2(x_1, x_2, \ldots, x_n) \le d_2$$

$\boxed{C_1 \vee C_2}$

- Seja *M* um número inteiro positivo muito grande.

$$C'_1: f_1(x_1, x_2, \ldots, x_n) \le d_1 + M.y$$
$$C'_2: f_2(x_1, x_2, \ldots, x_n) \le d_2 + M.(1 - y)$$
$$C'_3: y \in \{0, 1\}$$

$\boxed{C_1' \wedge C_2' \wedge C_3'}$

$y = 0$ => a restrição C'_2 *verifica-se automaticamente*
$y = 1$ => a restrição C'_1 *verifica-se automaticamente*

14

K de N restrições devem ser satisfeitas

$f_1(x_1, x_2, \ldots, x_n) \le d_1$
$\ldots$
$f_N(x_1, x_2, \ldots, x_n) \le d_N$

($K < N$)

- K restrições **devem** ser satisfeitas.
- Soluções **podem** satisfazer acidentalmente mais do que K restrições

$$f_1(x_1, x_2, \ldots, x_n) \le d_1 + M.y_1$$
$$\ldots$$
$$f_N(x_1, x_2, \ldots, x_n) \le d_N + M.y_N$$
$$\sum_{i=1}^{N} y_i = N - K$$
$$y_i \in \{0,1\} \quad (i = 1,\ldots,N)$$

K restrições mantêm-se
N-K restrições automaticamente eliminadas.

$y_i = 0$ => a restrição i mantém-se
$y_i = 1$ => a restrição i é automaticamente satisfeita

Objectivo: Escolher uma combinação de K restrições que permita atingir o melhor valor possível da função objectivo (resolver globalmente a formulação inteira do problema).

Funções com N valores possíveis

$$f_1(x_1, x_2, \ldots, x_n) = d_1 \text{ ou } d_2 \text{ ou } \ldots \text{ou } d_N$$

$$\Downarrow$$

$$f(x_1,\ldots,x_n) = \sum_{i=1}^{N} d_i.y_i$$
$$\sum_{i=1}^{N} y_i = 1$$
$$y_i \in \{0,1\} \ (i = 1, 2, \ldots, N)$$

Asseguram que apenas um y_i tomará o valor 1

Alteração no Wyndor Glass Co problem:
o tempo de produção pode ser de 18 ou 12 ou 6h /sema

$3x_1 + 2x_2 \le 18$,
$3x_1 + 2x_2 = 6$ ou 12 ou …ou 18

$$\Downarrow$$

$3x_1 + 2x_2 = 18y_1 + 12y_2 + 6y_3$
$y_1 + y_2 + y_3 = 1$
e y_j vars. binárias para $j = 1, 2, 3$

Problema de custo fixo

Um processo produtivo j tem duas componentes de custo:

- um custo fixo k_j (quando o processo é utilizado) e mais
- um custo variável proporcional ao número de actividades produzidas x_j, sendo c_j a constante de proporcionalidade (custo unitário).

O custo total da actividade j pode ser representado pela função:

$$f_j(x_j) = \begin{cases} k_j + c_j x_j & se \;\; x_j > 0 \\ 0 & se \;\; x_j = 0 \end{cases}$$

17

Problema de custo fixo (cont)

- Modelo completo original para n actividades

  Minimizar $Z = f_1(x_1) + \ldots + f_n(x_n)$

  sujeito a conjunto de restrições lineares

- Formulação PLI

$$Minimizar\ Z = \sum_{j=1}^{n} (c_j x_j + k_j y_j)$$

sujeito a

$$\text{restrições lineares originais}$$

$$x_j \leq M . y_j$$

$$y_j \in \{0,1\}$$

Vars. y_j podem ser vistas como decisões contingentes

$$y_j = \begin{cases} 1 & se\, x_j > 0 \\ 0 & se\, x_j = 0 \end{cases}$$

Deverá a actividade j ser considerada? (i.e. $x_j > 0$)

18

Exemplos de PLI
- Selecção de projectos de Investimento (1) -

Dispõe-se de capital D para investir em n projectos.

O projecto j necessita de c_j unidades de capital e tem uma rentabilidade r_j.

Quais os projectos a seleccionar de modo a maximizar a rentabilidade total?

$$\begin{aligned} &\text{Maximizar} \quad Z = \sum_{j=1}^{n} r_j . x_j \\ &\text{sujeito a} \quad \sum_{j=1}^{n} c_j . x_j \le D \\ &\qquad\qquad x_j \in \{0,1\} \ \ (j = 1, \ldots, n) \end{aligned}$$

Problema da mochila 0-1
(*Knapsack* 0-1)

(1) - Versão simplificada

19

Exemplos de PLI
- Localização de Armazéns -

Um sistema de distribuição tem como principais componentes a localização dos armazéns, os custos de transporte e os centros de exploração.

b_j - quantidades a fornecer a cada cliente (j = 1, …, n)

f_i - custos de exploração de armazém (i = 1, …, m)

c_{ij} - custos de transporte (de uma unidade de produto), do armazém i para o cliente j

---

Variáveis de decisão:

x_{ij} - quantidade a enviar do armazém i para o cliente j

y_i - i-ésimo armazém aberto ou não

20

Exemplos de PLI
- Localização de Armazéns (cont) -

x_{ij} - quantidade a enviar do armazém i para o cliente j

y_i - i-ésimo armazém aberto ou não

Custo total de transporte

Custo de exploração dos armazéns abertos

$$\text{Minimizar} \quad Z = \sum_{j=1}^{n} \sum_{i=1}^{m} c_{ij}.x_{ij} + \sum_{i=1}^{m} f_i.y_i$$

$$\text{sujeito a} \quad \sum_{i=1}^{n} x_{ij} = b_j \qquad (j = 1,\ldots,n)$$

$$\sum_{j=1}^{n} x_{ij} - y_i \sum_{j=1}^{n} b_j \le 0 \quad (i = 1,\ldots,m)$$

$$x_{ij} \ge 0 \qquad (i = 1,\ldots,m;\ j = 1,\ldots,n)$$

$$y_i \in \{0,1\} \qquad (i = 1,\ldots,m)$$

21

Exemplos de PLI
- Localização de Armazéns (cont) -

- O número de unidades entregues a cada cliente j deve ser igual às suas necessidades

$$\sum_{i=1}^{m} x_{ij} = b_j \quad (j = 1,\ldots,n)$$

- O número de unidades saídas de cada armazém i deve ser zero quando o armazém não é activado ($y_i = 0$) e pode ser quando igual às necessidades de todos os clientes.

$$\sum_{j=1}^{n} x_{ij} \le y_i \sum_{j=1}^{n} b_j \quad (i = 1,\ldots,m)$$

22

Exemplos de PLI
-Fazer Escolhas e variáveis de decisão continuas -

Uma empresa industrial desenvolveu 3 possíveis produtos novos e considera as seguintes restrições:

a) Evitar diversificação: produzir quando muito 2 novos produtos

b) Gestão de recursos: embora tenham 2 fábricas capazes de produzir os novos produtos apenas uma delas deverá ser usada.

c) O tempo de produção nas duas fábricas não é o mesmo, nem a capacidade de produção disponível

		Produto			Horas
		1	2	3	disponíveis
Fábrica	1	3	4	6	30
	2	4	6	2	40
Lucro unitário		5	7	3	(M$)
Mercado Potencial		7	5	9	(unid/semana)

23

Exemplos de PLI
-Escolhas e variáveis de decisão continuas (cont) -

		1	2	3	disponíveis
Fábrica	1	3	4	6	30
	2	4	6	2	40
Lucro unitário		5	7	3	(M$)
Mercado Potencial		7	5	9	(unid/semana)

- Modelo de PL

Maximizar $Z = 5x_1 + 7x_2 + 3x_3$

sujeito a

$3x_1 + 4x_2 + 6x_3 \leq 30$ (capacidade de produção da fábrica 1)

$4x_1 + 6x_2 + 2x_3 \leq 40$ (capacidade de produção da fábrica 2)

$x_1 \leq 7$ (mercado potencial para o produto 1)

$x_2 \leq 5$ (mercado potencial para o produto 2)

$x_3 \leq 9$ (mercado potencial para o produto 3)

$x_1, x_2, x_3 \geq 0$

24

Exemplos de PLI
-Escolhas e variáveis de decisão continuas (cont) -

- A restrição *a*) não foi considerada no modelo PL
  - Apenas duas variáveis (x_1, x_2, x_3) podem ter valores não nulos

$y_i = 1$ (produto *i* é produzido); $y_i = 0$ (produto *i* não é produzido);

Novas restrições

$$x_i \leq M.y_i \qquad (i = 1, 2, 3)$$

$$y_1 + y_2 + y_3 \leq 2 \qquad \text{(só serão produzidos até 2 produtos)}$$

$$y_i \in \{0, 1\}$$

25

Exemplos de PLI
-Escolhas e variáveis de decisão contínuas (cont) -

- A restrição *b*) não foi considerada no modelo PL
  - Apenas uma das fábricas deve ser usada, ou seja apenas uma das restrições de capacidade devem ser consideradas

    $3x_1 + 4x_2 + 6x_3 \leq 30$ (capacidade de produção da fábrica 1) **OU**

    $4x_1 + 6x_2 + 2x_3 \leq 40$ (capacidade de produção da fábrica 2)

$y_4 = 0$ (usar fábrica 1); $y_4 = 1$ (usar fábrica 2);

Novas restrições

$$3x_1 + 4x_2 + 6x_3 \leq 30 + M.y_4$$

$$4x_1 + 6x_2 + 2x_3 \leq 40 + M.(1 - y_4)$$

$$y_4 \in \{0, 1\}$$

26

Exemplos de PLI
-Escolhas e variáveis de decisão contínuas (cont) -

Maximizar $Z = 5x_1 + 7x_2 + 3x_3$

sujeito a

$3x_1 + 4x_2 + 6x_3 - M.y_4 \leq 30$ (capacidade da fábrica 1)

$4x_1 + 6x_2 + 2x_3 + M.y_4 \leq 40 + \mathrm{M}$ (capacidade da fábrica 2)

$x_1 \leq 7; x_2 \leq 5; x_3 \leq 9$ (mercado potencial)

$x_1 \leq \mathrm{M}\, y_1 ; x_2 \leq \mathrm{M}\, y_2; x_3 \leq \mathrm{M}\, y_3;$

$y_1 + y_2 + y_3 \leq 2$ (até 2 produtos)

$x_1, x_2, x_3 \geq 0$ (variáveis reais de decisão)

$y_j \in \{0, 1\}$ (j= 1,2,3,4) (variáveis binárias auxiliares)

Introdução à resolução de problemas de Programação Linear Inteira

- Introdução
- *Branch and Bound* (caso binário)
- Pesquisa de soluções quasi-óptimas
- *Branch and Bound* (caso mixto)
- Aspectos complementares
  - Pre-processamento dos problemas
  - Branch and Cut
- Algumas referências
- Exercícios

Alguns enganos

- Problemas puros de PLI têm um número finito de soluções (inteiras) enquanto PL têm um número infinito.
  - O número de soluções a considerar cresce exponencialmente com o número de variáveis. (Ex: com n variáveis binárias - 2^n soluções distintas a considerar).
- A remoção de algumas soluções (as soluções não inteiras) torna o problema mais fácil
  - Pelo contrário, é porque todas as soluções estão presentes em PL que se pode garantir que um dos vértices da região admissível é uma solução óptima. Esta é a razão da eficiência do SIMPLEX.

PLI v.s. PL

- Dado um problema de PLI, o correspondente problema de PL é referido como **relaxamento linear.**
- Em geral resolver um problema PLI é muito mais difícil do que resolver o seu relaxamento linear
- A ideia central de muitos algoritmos é resolver um problema de PLI através da resolução de uma sequência de relaxamentos lineares de partes do problema original

Factores da complexidade computacional

- Para um problema de PLI os factores determinantes são
  - Número de variáveis inteiras
  - Alguma estrutura especial

- Para um problema de PL o número de restrições é muito mais importante do que o número de variáveis

31

Casos particulares

- Embora em geral não aconteça, um caso particular em que um problema PLI não é mais difícil do que a seu relaxamento linear é quando as soluções deste último satisfazem as restrições inteiras.
- Casos especiais de PL em que devido à sua estrutura particular se garante soluções óptimas inteiras:
  - Minimun cost flow problem (com parâmetros inteiros)
    - Problema de transporte (transportation problem)
    - Problema de afectação (assignment problem)
    - Problema do caminho mais curto (shortest-path problem)
    - Problema do fluxo máximo (maximum flow problem)

32

Arredondar as soluções do relaxamento linear

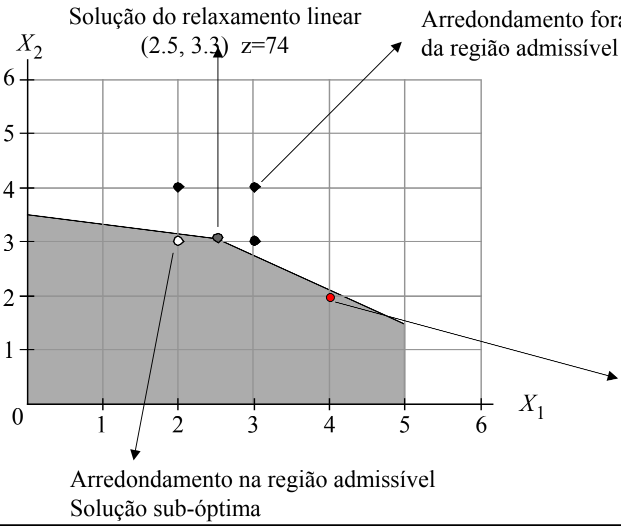
- O arredondamento de uma solução óptima do relaxamento linear de um problema de PLI pode (frequentemente) não pertencer à região admissível do problema PLI.

- Mesmo quando o arredondamento de uma solução óptima do relaxamento linear está na região admissível não existe garantia de que ela seja óptima para o problema PLI.

- O arredondamento pode ser adequado em problemas de grande dimensão em que os valores das variáveis são grandes e o arredondamento produz erros pequenos

33

Um exemplo do arredondamento



X1	X2	Z
2.5	3.3	74
X1	X2	Z
2	3	65
2	4	80
3	3	75
3	4	90
4	**2**	**70**

Solução óptima do problema PLI

34

Branch and Bound (Partição e Avaliação)

- Três Fases
  - Dividir o problema original em subproblemas mais simples de resolver (***Branching***).
  - Avaliar os subproblemas de forma a determinar qual é na melhor das hipóteses o valor do óptimo (***Bounding***).
  - Eliminar os subproblemas que não podem de certeza conduzir a uma solução óptima (***Fathoming***).

Como o problema de BIP puro tem um nº finito de soluções admissíveis, então deve usar-se procedimento de enumeração para encontrar uma solução óptima.
O *Branch and Bound* é um método de divisão e conquista

35

Exemplo a desenvolver no BB

Maximizar $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$
sujeito a

$$\begin{array}{l} 6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10 \\ x_3 + x_4 \leq 1 \\ -x_1 + x_3 \leq 0 \\ -x_2 + x_4 \leq 0 \\ x_i \text{ binárias } (i = 1, 2, 3, 4) \end{array}$$

O valor óptimo do problema de PLIB terá $Z \leq 16$

A solução óptima do relaxamento linear do problema é:
$(x_1, x_2, x_3, x_4) = (5/6, 1, 0, 1)$ e o valor de $Z = 16.5$

36

Partição (branching)

- Dividir um problema num conjunto de subproblemas (mais pequenos que o problema original)
- A divisão é feita criando uma partição do conjunto de soluções fazíveis
  - Partição a partir de variáveis binárias
    {soluções com $x_i = 0$}; {soluções com $x_i = 1$}
  - Partição a partir de variáveis inteiras
    a) {soluções com $x_i \leq$ a}; {soluções com $x_i \geq$ a + 1}
    b) {soluções com $x_i = 0$}; … ; {soluções com $x_i =$ n}
- Existem métodos sofisticados para a selecção da variável de partição (**branching variable**)
  - Por simplificação, vamos seleccionar pela ordem das variáveis: x_1, x_2, x_3, ...

Exemplo de partição binária

Maximizar $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

$$\begin{aligned} 6x_1 + 3x_2 + 5x_3 + 2x_4 &\leq 10 \\ x_3 + x_4 &\leq 1 \\ -x_1 + x_3 &\leq 0 \\ -x_2 + x_4 &\leq 0 \end{aligned}$$

x_j binária ($j=1,2,3,4$)

$X_1 = 1$

Subproblema 1

Maximizar $Z = 9 + 5x_2 + 6x_3 + 4x_4$

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 4 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 1 \\ -x_2 + x_4 &\leq 0 \end{aligned}$$

$X_1 = 0$

Subproblema 2

Maximizar $Z = 5x_2 + 6x_3 + 4x_4$

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\leq 10 \\ x_3 + x_4 &\leq 1 \\ x_3 &\leq 0 \\ -x_2 + x_4 &\leq 0 \end{aligned}$$

Bounding (avaliação)

- Para cada um dos subproblemas é necessário obter uma avaliação de qual é o melhor valor que o óptimo pode ter (um majorante nos problemas de maximização e um minorante nos problemas de minimização - em qualquer dos casos é um limite - (**bound**)).
- A forma de o fazer é resolver uma forma relaxada do problema e que seja de fácil resolução. Embora se possa considerar outras formas de relaxamento o mais usual é considerar o relaxamento linear (i.e. relaxar as restrições que impõem que variáveis sejam inteiras).
- e usar o simplex para resolver o correspondente problema de PL.

39

Exemplo de avaliação

Maximizar $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$

$$\begin{aligned} 6x_1 + 3x_2 + 5x_3 + 2x_4 &\le 10 \\ x_3 + x_4 &\le 1 \\ -x_1 + x_3 &\le 0 \\ -x_2 + x_4 &\le 0 \end{aligned}$$

$0 \le x_j \le 1$ $(j=1, 2, 3, 4)$

$Z \le 16$

$(5/6, 1, 0, 1)$, $Z = 16.5$

$X_1 = 1$ — Subproblema 1 — $Z \le 16$

Maximizar $Z = 9 + 5x_2 + 6x_3 + 4x_4$

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\le 4 \\ x_3 + x_4 &\le 1 \\ x_3 &\le 1 \\ -x_2 + x_4 &\le 0 \end{aligned}$$

$0 \le x_j \le 1$ $(j=2, 3, 4)$

(**1**, 4/5, 0, 4/5), $Z = 16.2$

$X_1 = 0$ — Subproblema 2 — $Z \le 9$

Maximizar $Z = 5x_2 + 6x_3 + 4x_4$

$$\begin{aligned} 3x_2 + 5x_3 + 2x_4 &\le 10 \\ x_3 + x_4 &\le 1 \\ x_3 &\le 0 \\ -x_2 + x_4 &\le 0 \end{aligned}$$

$0 \le x_j \le 1$ $(j=2, 3, 4)$

(**0**, 1, 0, 1), $Z = 9$

40

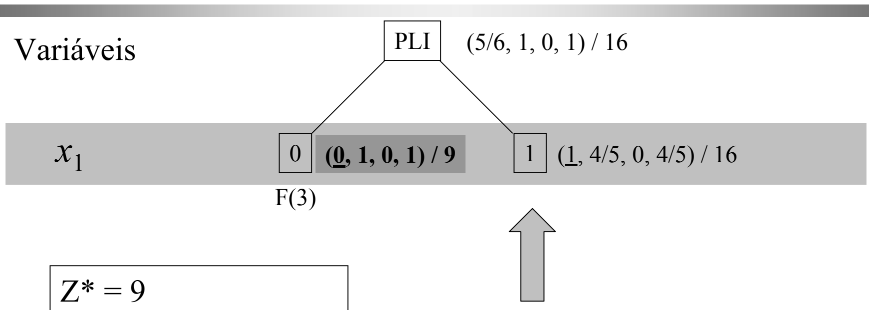
Fathoming (conquistar)

- Vamos designar por *Z** o valor da melhor solução inteira até aí encontrada e por **semente** (incumbent) essa solução (ou seja, *Z** é o valor de *Z* da semente)
- Um subproblema pode ser "conquistado" se:
  - **F(1)** - A estimativa do óptimo (obtida por resolução do relaxamento linear) for pior que Z^* (se bound $\leq Z^*$ para maximização).
  - **F(2)** - Se a região admissível for vazia.
  - **F(3)** - For encontrada uma solução inteira cujo óptimo seja melhor que a melhor solução inteira encontrada até então (Z^*).

A pesquisa é conduzida pela solução óptima retendo para exploração os problemas que possam ter solução admissível melhor que a solução corrente.

41

Fathoming: exemplo



$Z^* = 9$
Semente = (0, 1, 0, 1)

É necessário desenvolver este subproblema

42

Estrutura geral do BB

- Inicialização

  $Z^* = -\infty$; Resolver o relaxamento linear do problema original e repetir as iterações se não existir solução inteira

- Passos de cada iteração

  1 *Branching*: Escolha de um problema; escolha de uma variável

  2 *Bounding*: Determinação das estimativas por resolução de relaxamento linear. Se necessário actualizar Z*

  3 *Fathoming*: Determinar quais os problemas a eliminar.

- Parar quando não existirem mais subproblemas. A semente actual é a solução óptima. Senão, repetir.

43

Passos de cada iteração do BB

- Passos de cada iteração

  1 *Branching*: entre os subproblemas em aberto escolher o que foi criado mais recentemente (em caso de empate escolher o de melhor (maior) “bound” (estimativa). Escolher uma variável para realizar a partição.

  2 *Bounding*: Para cada novo subproblema obter uma estimativa pela resolução do correspondente relaxamento linear, com arredondamento por defeito.

  3 *Fathoming*: Para cada novo problema aplicar as 3 regras deste passo e eliminar todos os problemas que tenham sido conquistados

44

BB: continuar o exemplo
2ª Iteração

$X_1 = 1$

Maximizar $Z = 9 + 5x_2 + 6x_3 + 4x_4$
$3x_2 + 5x_3 + 2x_4 \leq 4$
$x_3 + x_4 \leq 1$
$x_3 \leq 1$
$-x_2 + x_4 \leq 0$
$0 \leq x_j \leq 1$ $(j=2, 3, 4)$

$Z \leq 16$

(**1**, 4/5, 0, 4/5), $Z = 16.2$

Expandir ambas

Z=16 > Z=13 => vai ser expandido

$X_2 = 0$ — $Z \leq 13$

Subproblema 3

Maximizar $Z = 9 + 6x_3 + 4x_4$
$5x_3 + 2x_4 \leq 4$
$x_3 + x_4 \leq 1$
$x_3 \leq 1$
$x_4 \leq 0$
$0 \leq x_j \leq 1 (j=3, 4)$

(**1**, **0**, 4/5,0), $Z = 13.8$

$X_2 = 1$ — $Z \leq 16$

Subproblema 4

Maximizar $Z = 14 + 6x_3 + 4x_4$
$5x_3 + 2x_4 \leq 1$
$x_3 + x_4 \leq 1$
$x_3 \leq 1$
$x_4 \leq 1$
$0 \leq x_j \leq 1 (j=3, 4)$

(**1**, **1**, 0, 0.5), $Z = 16$

BB: continuar o exemplo
3ª Iteração

$X_1 = 1, X_2 = 1$

Maximizar $Z = 14 + 6x_3 + 4x_4$
$5x_3 + 2x_4 \leq 1$
$x_3 + x_4 \leq 1$
$x_3 \leq 1$
$x_4 \leq 1$
$0 \leq x_j \leq 1 (j=3, 4)$

$Z \leq 16$

(**1**, **1**, 0, 0.5), $Z = 16$

Conquistado por F(2)

$X_3 = 0$ — $Z = 16$

Subproblema 5

Maximizar $Z = 14 + 4x_4$
$2x_4 \leq 1$
$x_4 \leq 1$
$0 \leq x_4 \leq 1$

(**1**, **1**, **0**,0.5), $Z = 16$

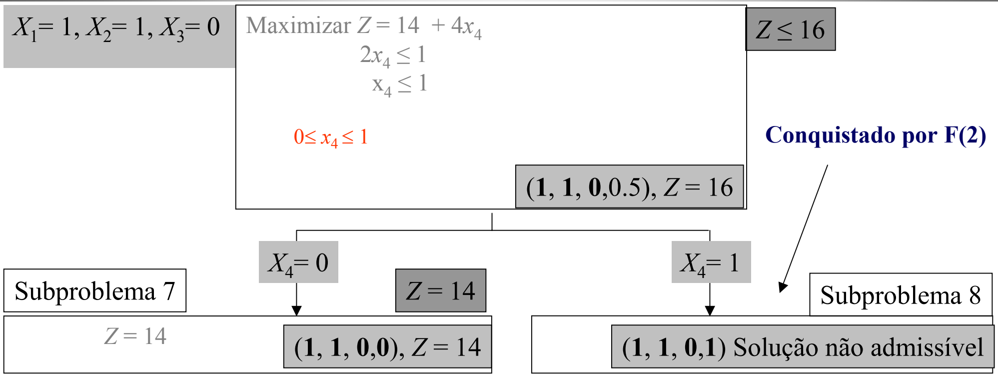
$X_3 = 1$ — $Z \leq 16$

Subproblema 6

Maximizar $Z = 20 + 4x_4$
$2x_4 \leq -4$
$x_4 \leq 0$
$x_4 \leq 1$
$0 \leq x_4 \leq 1$

Solução não admissível

BB: continuar o exemplo
4ª Iteração



Solução final será $Z^* = 14$ (que é melhor do que $Z^* = 9$)

BB: terminar o exemplo

Subproblema 3 $\quad Z \le 13$

Maximizar $Z = 9 + 6x_3 + 4x_4$

$5x_3 + 2x_4 \le 4$

$x_3 + x_4 \le 1$

$x_3 \le 1$

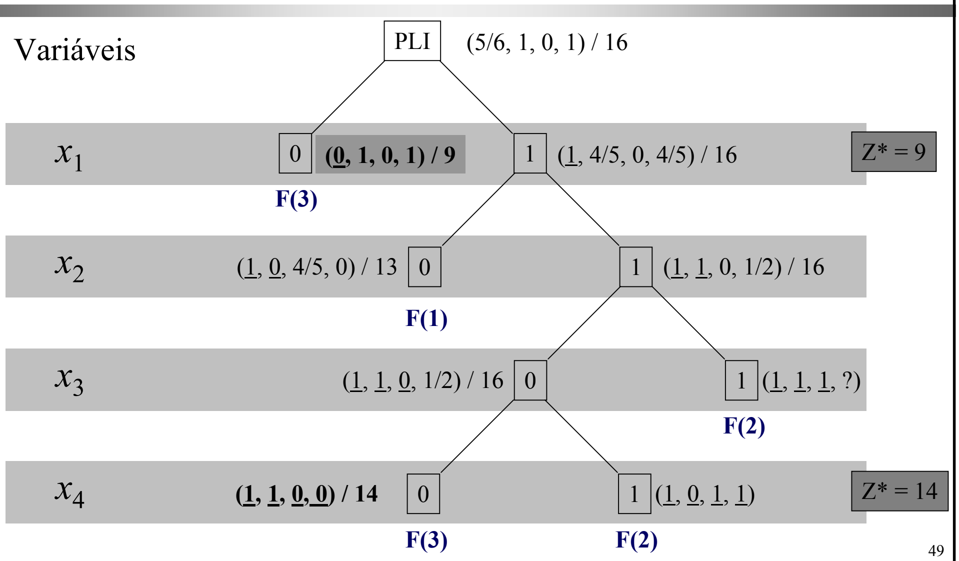
$x_4 \le 0$

$0 \le x_j \le 1 (j = 3, 4)$

(**1**, **0**, 4/5,0), $Z = 13.8$

- Bound= 13 ≤ $Z^* = 14$, e portanto este subproblema agora é conquistado (F(1)).
- O problema termina com a semente (**1**, **1**, **0**,**0**) e $Z^* = 14$

BB: árvore completa



Variantes do BB - PLIB

- As variantes do branch and bound têm a haver de como é que cada um dos passos ‘branching’, ‘bound’ e ‘fathoming’ são explorados.

- Considerem-se algumas variantes

Opções ao BB - PLIB

- Branching
  - Seleccionar o subproblema mais recente
    - Para ser mais eficiente o processo de recalcular a nova solução do relaxamento linear. Continuar a execução do simplex em vez de recomeçar do início.
  - Seleccionar o subproblema com melhor estimativa
    - Porque tende a mais rapidamente encontrar melhores estimativas e dessa forma eliminar mais subproblemas (diminuir a parte visitada da árvore de pesquisa)
  - Efectuar a partição
    - Qual é a variável a seleccionar?
    - Como particionar no caso de variáveis não binárias?

51

Opções ao BB - PLIB (cont)

- Bounds
  - A forma habitual de cálculo de estimativas é através do relaxamento linear e usar o simplex.
  - Alternativas podem ser usadas tais como relaxamento Lagrangiano
    - Deita-se fora o conjunto total de restrições funcionais $\mathbf{Ax} \leq \mathbf{b}$
    - A função objectivo *max* Z= **cx é substituída por** *max* $Z_R = \mathbf{cx} - \lambda(\mathbf{Ax} - \mathbf{b})$
    - O valor de Z_R *é* um limite válido.

- Eliminação
  - Quando o relaxamento linear tem uma solução inteira então ela também é óptima para o problema inteiro. Com o método do relaxamento Lagrangiano é necessária uma análise mais cuidada

52

Uma ou todas as soluções óptimas

- Quando há empates na solução óptima:
  - Nesse caso é desejável identificar todas as soluções óptimas para escolher a melhor
- A regra 3 para teste de eliminação de um problema passa a ser:
  - Estimativa do subproblema é pior que Z* (em vez de pior ou igual)
- Quando a solução é inteira e Z = Z* então deve-se guardar mais essa nova semente (mantendo todas as outras). Além disso é necessário determinar se a solução inteira encontrada para esse subproblema é única ou se existem outras.
- Quando não existirem mais problemas então todas as sementes actuais são soluções óptimas

53

Procura de soluções quasi-óptimas - maximização -

- Em muitos problemas reais basta determinar as soluções perto do óptimo → soluções **quasi-óptimas.**

54

Procura de soluções quasi-óptimas - maximização -

- O que é uma solução quasi-óptima
  - Se Z^{**} é o valor (desconhecido) de uma solução óptima então uma solução de valor Z é quasi-óptima se

$$(1 - \alpha)\, Z^{**} \leq Z$$

onde α é um valor menor que 1.

- Se fosse conhecido que o valor de Z de uma nova semente (Z^*) satisfaz $(1 - \alpha)\, Z^{**} \leq Z^*$ então o algoritmo poderia terminar com esta semente.

55

Procura de soluções quasi-óptimas - maximização - (cont)

- O teste para eliminar um subproblema passaria a ser

$$(1 - \alpha)\, bound \leq Z^*$$

onde bound é o valor óptimo do relaxamento linear do subproblema

Novas regras:

2- Se a região admissível for vazia.

3- For encontrada uma solução inteira cujo óptimo for melhor que a melhor solução inteira até então encontrada (Z^*).

1- A estimativa do óptimo (obtida por resolução do relaxamento linear) for pior que $Z^* / (1 - \alpha)$

56

Branch and Bound para PLI-mista

57

Branch and Bound para PLI-mista

Maximizar $\sum_{j=1}^{n} c_j x_j$

sujeito a

$$\sum_{j=1}^{n} a_{ij} x_j \quad \text{para} \quad i = 1,\ldots,m$$

e

$$x_j \geq 0 \quad \text{para} \quad j = 1,\ldots n$$

$$x_j \quad \text{inteiros para} \quad j = 1,\ldots I : I \leq n$$

As primeiras I variáveis são inteiras

58

Escolha da variável de partição

- Depois de escolhido um problema para dividir em subproblemas é necessário escolher uma variável para realizar a partição.
  – De entre as variáveis inteiras
  ✓ Escolher uma de entre aquelas cujo valor da solução do relaxamento linear não é inteiro.
- Exemplo:
  – Solução do relaxamento linear do problema a dividir em subproblemas é (1, 2.3, 0, 5.2, 10.2, …) devendo ser inteiras as 4 primeiras variáveis. Poderiam ser escolhidas para variáveis de partição x_2 ou x_4.

59

Modo de realizar a partição

- Seja
  – x_j a variável escolhida para partição
  – x_j^* o valor (não inteiro) da variável x_j da solução óptima do relaxamento linear do problema a subdividir.
- Partição em dois subproblemas
  – Problema 1: $x_j \leq \lfloor x_j^* \rfloor$
  – Problema 2: $x_j \geq \lfloor x_j^* \rfloor + 1$

  Cada uma destas desigualdades torna-se numa restrição adicional
- Exemplo
  – $x^*_j = 3.1/2$
  - Problema 1: $x_j \leq 3$
  - Problema 2: $x_j \geq 4$

60

Cálculo das estimativas (bounding)

- No caso BIP (se os coeficientes de Z fossem todos inteiros) a estimativa resulta do arredondamento do valor óptimo real do relaxamento linear.
- A existência de variáveis contínuas não permite garantir soluções óptimas de valor inteiro pelo que neste caso não se pode arredondar
  - Ex: se a solução óptima do relaxamento linear é $Z = 22.25$ então a estimativa será $Z \leq 22.25$.

61

Novas sementes

- Teremos uma nova semente quando a solução do relaxamento linear de um problema for tal que os valores de todas as variáveis inteiras forem inteiros e o valor Z for **melhor do que** Z^* (i.e. superior no caso de maximização)
- Exemplo com 3 variáveis inteiras e 2 contínuas)
  $Z^* = 9.1$
  - (1, 0, 2, 3.5 1.2) / Z = 9
  - (1, 0, 2.1, 3.5 1.2) / Z = 10.2
  - (1, 0, 2, 3.5 1.2) / Z = 9.8 **Nova semente**
  - (1, 0.5, 2, 4 1) / Z = 12

62

Um exemplo

Maximizar $Z = 4x_1 - 2x_2 + 7x_3 - x_4$
sujeito a

$$
\begin{cases}
x_1 + 5x_3 \leq 10 \\
x_1 + x_2 - x_3 \leq 1 \\
-6x_1 - 5x_2 \leq 0 \\
-x_1 + 2x_3 - 2x_4 \leq 3 \\
\\
x_i \geq 0 \ (i = 1, 2, 3, 4) \\
x_i \text{ inteiro } (i = 1, 2, 3)
\end{cases}
$$

$\boxed{x_4 \text{ - contínua}}$

$Z^* = -\infty$
(5/4, 3/2, 7/4, 0) / 14.25

Qualquer das três variáveis inteiras pode ser usada para partição

63

Primeira partição

Variáveis

All (5/4, 3/2, 7/4, 0) / 14.25

x_1 | $x_1 \geq 2$ (?, ?, ?, ?) | $x_1 \leq 1$ (1, 6/5, 9/5, 0) / 14.2

F(2)

$x_1^* = 5/4 = 1.25$

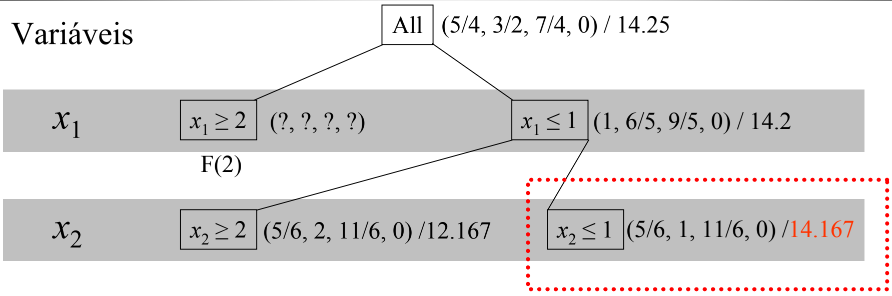
$\lfloor x_1^* \rfloor = 1 \longrightarrow x_1 \leq 1$ Subproblema 1

$\lfloor x_1^* \rfloor + 1 = 2 \longrightarrow x_1 \geq 2$ Subproblema 2

Novas restrições de cada subproblema

64

Segunda partição



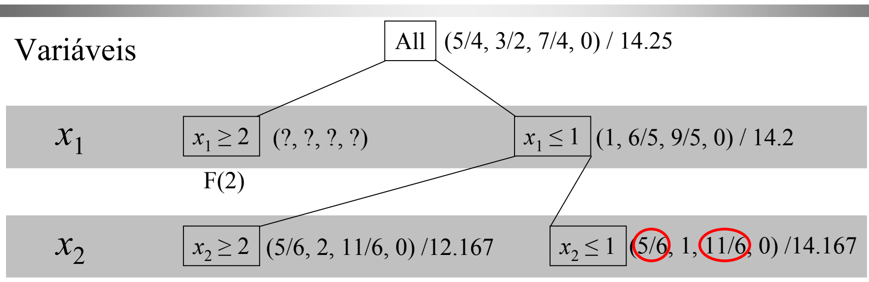
$\boxed{x_2^* = 6/5 = 1.2}$

$\lfloor x_2^* \rfloor = 1 \longrightarrow x_2 \leq 1$ Subproblema 3

$\lfloor x_2^* \rfloor + 1 = 2 \longrightarrow x_2 \geq 2$ Subproblema 4

65

Terceira partição



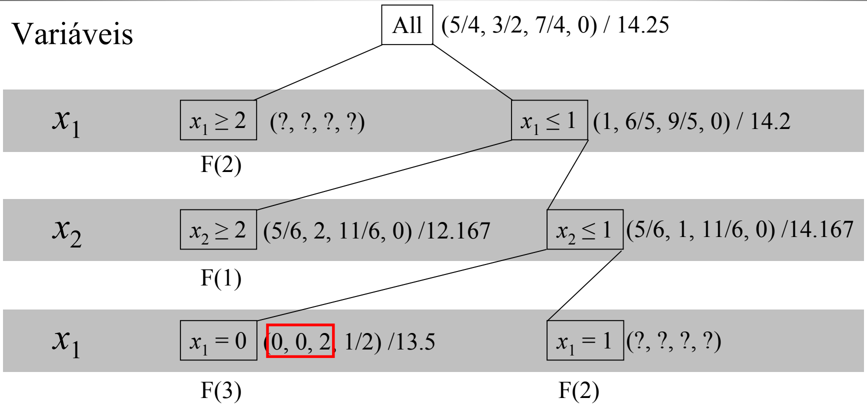
$\boxed{x_1^* = 5/6 = 0.83}$

$\lfloor x_1^* \rfloor = 0 \longrightarrow x_1 \leq 0 \Rightarrow x_1 = 0$ SP 5

$\lfloor x_1^* \rfloor + 1 = 1 \longrightarrow x_1 \geq 1 \Rightarrow x_1 = 1$ SP 6

66

Terceira partição (cont)



$Z^{**} = 13.5$ (0, 0, 2, 1/2)

Pre-processamento do problema

- Eliminação de variáveis
  - Identificar variáveis cujo valor pode ser fixado por ser o único que pode pertencer à região de soluções admissíveis
  - ✓ Redução da árvore de pesquisa
- Eliminar restrições redundantes
  - Identificar as restrições que estão automaticamente satisfeitas se as outras restrições forem satisfeitas
  - ✓ Reduz o número de restrições e melhora o desempenho do SIMPLEX sobre os problemas de relaxamento linear
- Constrangimento de restrições
  - Tornar certas restrições mais estritas de forma a diminuir a região admissível do relaxamento sem eliminar soluções inteiras.

Eliminação de variáveis (PIB)

- Se um valor de uma variável não poder satisfazer uma restrição, quaisquer que sejam os valores das outras variáveis, então a variável só pode ter o outro valor.
- Procedimentos/regras
  - Numa restrição $\leq$ procurar a variável de maior coeficiente positivo; e se a soma desse coeficiente com qualquer coeficiente negativo for maior que o lado direito da desigualdade então essa variável pode ser fixada em ZERO.
  - etc.

Um problema com 2756 variáveis conseguiram fixar 1341 variáveis

69

Exemplos de eliminação de variáveis binárias

- $3x_1 \leq 2$ $\Rightarrow x_1 = 0$, pois $3(1) > 2$
- $5x_1 + x_2 - 2x_3 \leq 2$ $\Rightarrow x_1 = 0$, pois $5(\mathbf{1}) + 1(0) -2(\mathbf{1}) > 2$
- $3x_1 + x_2 - 2x_3 \geq 2$ $\Rightarrow x_1 = 1$, pois $3(\mathbf{0}) + 1(1) -2(\mathbf{0}) < 2$
- $x_1 + x_2 - 2x_3 \geq 1$ $\Rightarrow x_3 = 0$, pois $1(1) + 1(1) -2(\mathbf{1}) < 1$
- $3x_1 + x_2 - 3x_3 \geq 2$ $\Rightarrow x_1 = 1$, pois $3(\mathbf{0}) + 1(1) -3(0) < 2$
  $\Rightarrow x_3 = 0$, pois $3(1) + 1(1) -3(\mathbf{1}) < 2$
- $3x_1 - 2x_2 \leq -1$ $\Rightarrow x_1 = 0$, pois $3(\mathbf{1}) -2(1) > -1$
  $\Rightarrow x_2 = 1$, pois $3(\mathbf{0}) -2(\mathbf{0}) > -1$

70

Propagar a eliminação de variáveis

- $3x_1 + x_2 - 2x_3 \geq 2 \quad \Rightarrow \boxed{x_1 = 1}$, pois $3(0) + 1(1) -2(0) < 2$

então

- $x_1 + x_4 + x_5 \leq 1 \quad \Rightarrow x_4 = 0, \boxed{x_5 = 0}$

então

- $-x_5 + x_6 \leq 0 \quad \Rightarrow x_6 = 0$

...

71

Eliminação de restrições redundantes

- Existe uma restrição sobre as variáveis binárias:
  - $x_i \in \{0, 1\}$
- Qualquer restrição funcional que restrinja os valores de uma variável a um conjunto de valores que contenha (ou seja igual a) $\{0, 1\}$ é uma restrição redundante
- Exemplos:
  - $3x_1 + 2x_2 \leq 6$ é redundante pois $3(1) + 2(1) \leq 6$ / idem (0,0), ...
  - $3x_1 - 2x_2 \leq 3$ é redundante pois $3(1) - 2(0) \leq 3$ / idem (1,1), ...
  - $3x_1 - 2x_2 \geq -3$ é redundante pois $3(0) - 2(1) \geq -3$ / idem (0,0), ...

72

Restringir as restrições

Consideremos um exemplo:

Maximize $Z = 3x_1 + 2x_2$

sujeito a

$2x_1 + 3x_2 \leq 4$

e

x_1 e x_2 binárias

Relaxamento Linear

Maximize $Z = 3x_1 + 2x_2$

sujeito a

$2x_1 + 3x_2 \leq 4$

e

$0 \leq x_1 \leq 1$

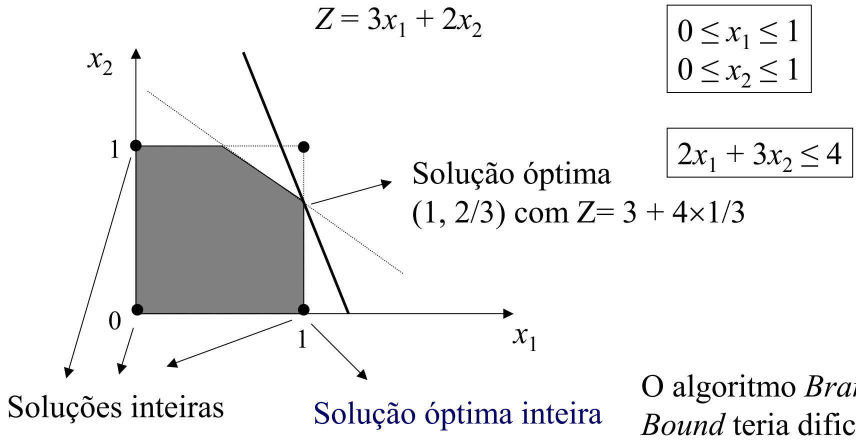
$0 \leq x_2 \leq 1$

O problema PIB tem 3 soluções admissíveis (0,0), (1,0), (0,1).

Solução óptima é $(x^*_1, x^*_2)= (1,0)$ com $Z^*=3$

73

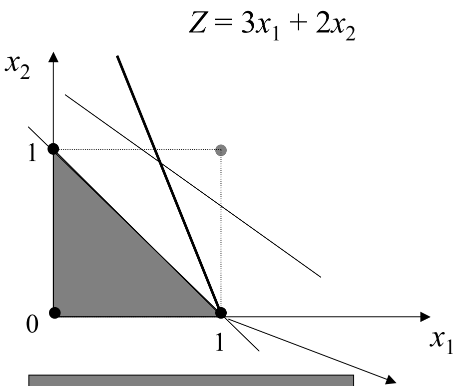
Restringir as restrições (cont)



O algoritmo *Branch and Bound* teria dificuldade em encontrar a solução óptima BIP.

74

Restringir as restrições (cont)



$0 \le x_1 \le 1$
$0 \le x_2 \le 1$

$\cancel{2x_1 + 3x_2 \le 4}$

Restringi-la

$x_1 + x_2 \le 1$

(não se eliminaram soluções inteiras)

O que é que se ganhou? **A Solução do relaxamento é inteira**

Este exemplo mostra como é que restringir uma restrição reduz a região admissível (i.e. Espaço de pesquisa), sem eliminar as soluções admissíveis do problema BIP.

Procedimento para restringir restrições $\le$

Seja a restrição $a_1x_1 + \ldots + a_nx_n \le b$

- 1 – Calcula-se S = soma dos a_j positivos
- 2 – Identificar qualquer $a_j \neq 0$ tal que $S < b + |a_j|$
  - (a) Se não existir, parar; não é possível restringi-la
  - (b) Se $a_j > 0$ seguir para 3
  - (c) Se $a_j < 0$ seguir para 4
- 3 ($a_j > 0$) Calcular $a_j^* = S - b$ e $b^* = S - a_j$. $a_j \leftarrow a_j^*$; $b \leftarrow b^*$
- 4 ($a_j < 0$) Aumentar o valor de a_j segundo $a_j \leftarrow b - S$;
- Voltar ao passo 1.

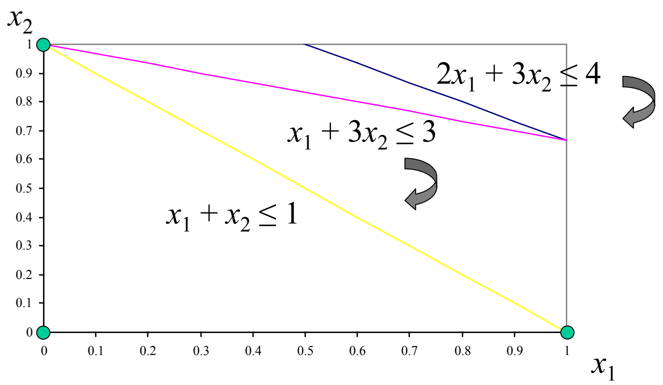
76

Exemplo de restringir uma restrição

- $2x_1 + 3x_2 \leq 4$ ($a_1 = 2$, $a_2 = 3$, $b = 4$)
  S = 2 + 3 = 5
  a_1 e a_2 satisfazem $S < b + |a_i|$. Escolhemos a_1
  – $a_1^* = 5 - 4 = 1$; $b^* = 5 - 2 = 3$; $b \leftarrow 3$, $a_1 \leftarrow 1$
- $x_1 + 3x_2 \leq 3$ ($a_1 = 1$, $a_2 = 3$, $b = 3$)
  S = 1 + 3 = 4
  a_2 satisfaz $S < b + |a_2|$.
  – $a_2^* = 4 - 3 = 1$; $b^* = 4 - 3 = 1$; $b \leftarrow 1$, $a_2 \leftarrow 1$
- $x_1 + 1x_2 \leq 1$ ($a_1 = 1$, $a_2 = 1$, $b = 1$)
  S = 1 + 1 = 2. Nenhum a_i satisfaz $S < b + |a_i|$.

77

Exemplo de restringir uma restrição



78

Vamos restringir uma restrição

$4x_1 - 3x_2 + x_3 + 2x_4 \leq 5$

$S = 4 + 1 + 2 = 7;\ b = 5;$ $S < b + |a_j|$ para a_1 e a_2

$a_1^* = S - b = 2;\ b^* = S - a_1 = 7 - 4 = 3$

$2x_1 - 3x_2 + x_3 + 2x_4 \leq 3$

$S = 2 + 1 + 2 = 5;\ b = 3;$ $S < b + |a_j|$ para a_2

$a_2^* = b - S = 3 - 5 = -2$

$2x_1 - 2x_2 + x_3 + 2x_4 \leq 3$

1 – Calcula-se S = soma dos a_j positivos

2 – Identificar qualquer $a_j \neq 0$ tal que $S < b + |a_j|$

(a) Se não existir, parar; não é possível restringi-la

(b) Se $a_j > 0$ seguir para 3

(c) Se $a_j < 0$ seguir para 4

3 – Calcular $a_j^* = S - b$ e $b^* = S - a_j$. $a_j \leftarrow a_j^*$; $b \leftarrow b^*$

4 – Aumentar o valor de a_j segundo $a_j \leftarrow b - S$;

79

Geração de planos de corte para BIP

- Um plano de corte para um problema de PLI é uma nova restrição funcional que reduz a região admissivel para o correspondente relaxamento linear sem eliminar qualquer solução admissível do problema PLI
- A restrição de uma restrição existente é um caso particular de um plano de corte.
- Exemplo: $x_1 + x_2 \leq 1$ é um plano de corte para o problema PLI anterior.
- O objectivo da técnica é acelarar o processo do Branch and Bound e encontrar uma solução óptima para o problema PIB

80

Um procedimento para a geração de planos de corte caso binário

- Escolher uma restrição na forma ≤ com apenas coeficientes não negativos
- Encontrar um grupo de *N* variáveis (cobertura mínima) tais que:
  - a restrição seja violada se as variáveis do grupo tomarem o valor 1 e todas as outras o valor 0
  - mas a restrição fica satisfeita se o valor de qualquer das variáveis do grupo de cobertura mínima mudar de 1 para 0.
- O plano de corte tem a forma

  Soma das variáveis (de cobertura mínima) ≤ N -1

81

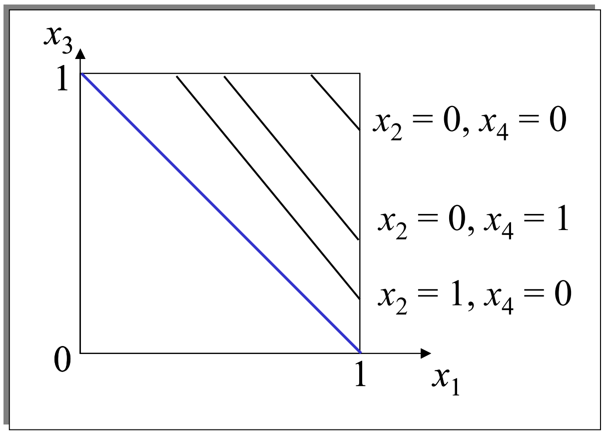
Um procedimento para a geração de planos de corte caso binário (exemplo)

- Restrição: $6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$
- Cobertura mínima
  - $\{x_1, x_2, x_4\}$ pois $6(1) + 3(1) + 5(0) + 2(1) = 11 > 10$, mas se qualquer destas variáveis toma o valor 0, a restrição é satisfeita.
  - $\{x_1, x_3\}$ pois $6(1) + 3(0) + 5(1) + 2(0) = 11 > 10$, mas se qualquer destas variáveis toma o valor 0, a restrição é satisfeita.
  - $\{x_1, x_2, x_3\}$ ou $\{x_1, x_3, x_4\}$ não são coberturas mínimas, pois quando x_2 (ou x_4) passam de 1 para 0 a restrição continua a não ser satisfeita.

82

Um procedimento para a geração de planos de corte caso binário (exemplo - cont)

- Restrição: $\boxed{6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10}$
- Planos de corte
  - $\{x_1, x_2, x_4\}$
    $N = 3$
    $x_1 + x_2 + x_4 \leq 2$
  - $\{x_1, x_3\}$
    $N = 2$
    $\boxed{x_1 + x_3 \leq 1}$



O que se ganha com os planos de corte?

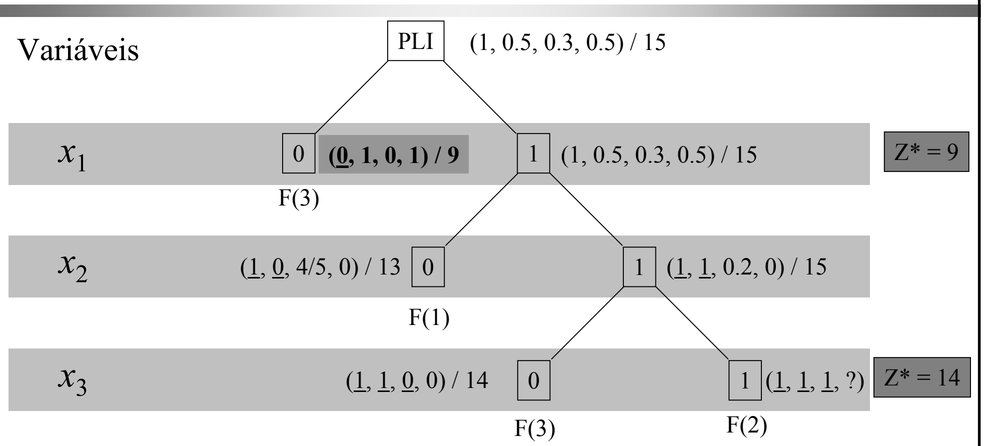
Maximizar $Z = 9x_1 + 5x_2 + 6x_3 + 4x_4$
$6x_1 + 3x_2 + 5x_3 + 2x_4 \leq 10$
$x_3 + x_4 \leq 1$
$-x_1 + x_3 \leq 0$
$-x_2 + x_4 \leq 0$
$x_1 + x_2 + x_4 \leq 2$
$Z \leq 15$
(1, 0.5, 0.3, 0.5), $Z = 15.3$

$X_1 = 1$ / $X_1 = 0$

Subproblema 1 — $Z \leq 15$
Maximizar $Z = 9 + 5x_2 + 6x_3 + 4x_4$
$3x_2 + 5x_3 + 2x_4 \leq 4$
$x_3 + x_4 \leq 1$
$x_3 \leq 1$
$-x_2 + x_4 \leq 0$
$x_2 + x_4 \leq 1$
(**1**, 0.5, 0.1, 0.5), $Z = 15.3$

Subproblema 2 — $Z \leq 9$
Maximizar $Z = 5x_2 + 6x_3 + 4x_4$
$3x_2 + 5x_3 + 2x_4 \leq 10$
$x_3 + x_4 \leq 1$
$x_3 \leq 0$
$-x_2 + x_4 \leq 0$
$x_2 + x_4 \leq 2$
(**0**, 1, 0, 1), $Z = 9$

O que se ganha com os planos de corte?



As estimativas são melhores!

Menos uma iteração!

Compara com árvore original

O que se ganha com os planos de corte?

- Um grande aumento da eficiência do branch & bound
  - Obtenção de melhores estimativas
  - Mais cedo se obtêm soluções inteiras
  - Mais problemas podem ser eliminados
  - Menor árvore de pesquisa
- Os planos de corte podem ser aplicados em cada nó
- Existem outras técnicas de corte

Algumas referências de produtos

- Optimization Software Library (OSL) - IBM
- CPLEX- Large-Scale Mathematical Programming Software for Optimization (http://www.cplex.com/)
- MINTO, a Mixed INTeger Optimizer (http://akula.isye.gatech.edu/~mwps/projects/minto.html)

Exercícios para resolver

- Para familiarizar:
  - 12.1-1, 12.1-2
- Interpretação
  - 12.2-2, 12-2-3
- Entender o modelo PLI e PLIB
  - 12.2-4, 12.2-6, 12-3-4
- Outros
  - 12.3-8, 12.3-10, 12.3-11
- Branch and Bound
  - 12.4-1, 12.4-1, 12.5-1, 12.5-2 12.6-1, 12.6-2