Simulação e Otimização

Capítulo 1: Técnicas de resolução de problemas de otimização combinatória

Raquel Bernardino

rbernardino@iseg.ulisboa.pt
Gabinete 511 Quelhas 6

Sumário

Sumário

Motivação

- Todos os dias estamos perante problemas de Investigação Operacional, como por exemplo:
  - Determinar o caminho rápido para vir para o ISEG.
  - Decidir onde fazer as compras para a semana (distância versus preços).

- Para formular estes problemas muitas vezes precisamos de variáveis inteiras.

Exemplos de aplicações de Investigação Operacional

European Journal of Operational Research
Volume 240, Issue 3, 1 February 2015, Pages 718-733

Decision Support

An optimization framework for the development of efficient one-way car-sharing systems

Burak Boyacı [a] [b], Konstantinos G. Zografos [b], Nikolas Geroliminis [a]

(a) 2017

European Journal of Operational Research
Volume 257, Issue 3, 16 March 2017, Pages 992-1004

Innovative Applications of O.R.

Inventory rebalancing and vehicle routing in bike sharing systems

J. Schuijbroek [a], R.C. Hampshire [1] [b], W.-J. van Hoeve [c]

(b) 2019

European Journal of Operational Research
Volume 271, Issue 3, 16 December 2018, Pages 1085-1099

Innovative Applications of O.R.

Scheduling last-mile deliveries with truck-based autonomous robots

Nils Boysen [1] [a], Stefan Schwerdfeger [b], Felix Weidinger [1] [a]

(c) 2020

European Journal of Operational Research
Volume 292, Issue 1, 1 July 2021, Pages 250-275

Innovative Applications of O.R.

Building disaster preparedness and response capacity in humanitarian supply chains using the Social Vulnerability Index

Douglas Alem [a], Hector F. Bonilla-Londono [b], Ana Paula Barbosa-Povoa [c], Susana Relvas [c], Deisemara Ferreira [d], Alfredo Moreno [e]

(d) 2023

Figura 1: Prémio aplicações inovadoras da IO da revista científica *European Journal of Operations Research.*

Sumário

Introdução

Um problema de programação linear inteira mista (PLIM) é um problema de programação linear em que algumas variáveis têm obrigatoriamente valores inteiros. $\Longrightarrow$ Perda da propriedade da divisibilidade da PL.

- Num problema de programação linear inteira (PLI) todas as variáveis têm obrigatoriamente valores inteiros.
- Num problema de programação linear binária todas as variáveis têm obrigatoriamente valor binário, isto é, valor 0 ou 1.

Um problema de otimização combinatória (POC) é um PLIM cuja região admissível (RA) é um conjunto finito.

- A solução ótima é um subconjunto de um conjunto finito - a RA.

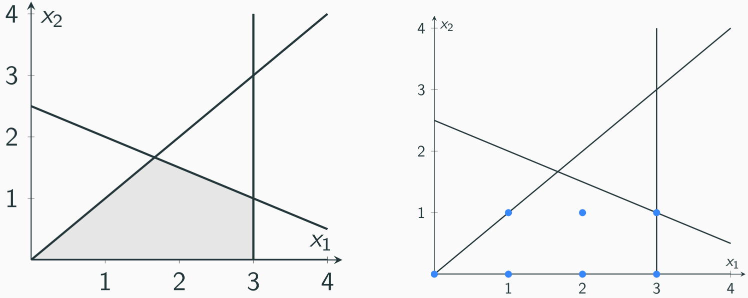


Figura 2: RA de um PL versus RA de um POC.

Introdução

Exemplos de problemas de otimização combinatória são:

- o problema da afetação;
- o problema dos transportes;
- o problema do saco-mochila;
- o problema do caixeiro viajante; ou
- o problema do roteamento de veículos.

■ O problema do caixeiro viajante e o problema do roteamento de veículos serão estudados no Capítulo 2.

Introdução

Como podemos resolver um problema de otimização combinatória?

- Como a RA é um conjunto finito, podemos calcular todas as soluções admissíveis (SA) e determinar a melhor. $\Longrightarrow$ Enumeração!

- A enumeração da RA de um POC pode ser um processo muito demorado.

Problema	#SA
Caixeiro viajante	$(n-1)!$
Problema do saco-mochila	2^n

- Uma instância do caixeiro viajante com 10 cidades tem 362880 SAs.
- Uma instância do saco-mochila com 10 itens tem 1024 SAs.
    - Estas são consideradas instâncias pequenas.

Introdução

- Como resolver um POC de forma eficiente?

Métodos de resolução

- Exatos
  - Branch-and-bound
  - Planos de corte de Gomory
  - Branch-and-cut
  - Branch-and-price
- Aproximados
  - Relaxações
  - Heurísticas

Nos métodos aproximados são calculados limites inferiores (minorantes) ou limites superiores (majorantes) para o valor ótimo.

- Num problema de minimização as relaxações dão minorantes e as heurísticas majorantes.
- Num problema de maximização as relaxações dão majorantes e as heurísticas minorantes.

Sumário

Relaxações

Ideia geral: Resolver uma versão "simplificada" do problema, que normalmente é obtida removendo restrições.

Definição: Relaxação

Um problema

$$(RP) \qquad z^R = \min\{f(x) : x \in T \subseteq \mathbb{R}^n\}$$

é uma *relaxação* de

$$(IP) \qquad z = \min\{c(x) : x \in X \subseteq \mathbb{R}^n\}$$

se:

(i) $X \subseteq T$; e

(ii) $f(x) \leq c(x), \forall x \in X$.

Proposição

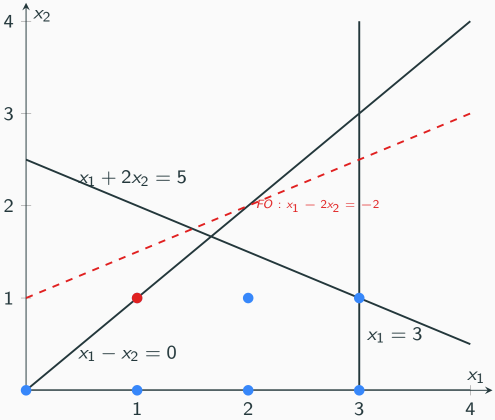
Se RP é uma relaxação de IP, então $z^R \leq z$.

Demonstração.

Se x^* é uma solução ótima de IP, então $x \in X$ e $z = c(x^*)$. Como RP é uma relaxação de IP sabemos que $X \subseteq T$ e $f(x^*) \leq c(x^*)$. Logo, $z^R \leq f(x^*) \leq c(x^*) \leq z$. $\square$

Exemplo

$$
\begin{aligned}
(IP) \equiv \min z = x_1 - 2x_2 \\
s.a: x_1 - x_2 \geq 0 \\
x_1 + 2x_2 \leq 5 \\
x_1 \leq 3 \\
x_1, x_2 \geq 0 \text{ e inteiros}
\end{aligned}
$$



$v(IP) = c(1,1) = 1 \times 1 - 2 \times 1 = -1$

Exemplo

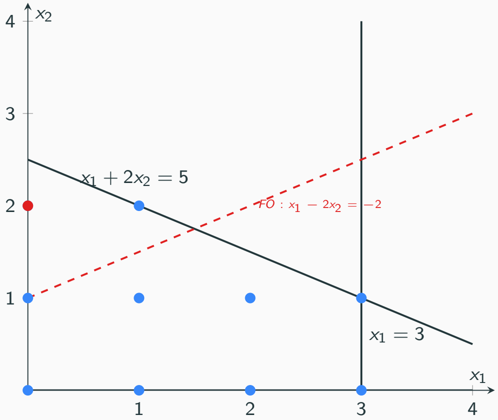
$(RL_1) \equiv \min z = x_1 - 2x_2$

$s.a : x_1 + 2x_2 \leq 5$

$x_1 \leq 3$

$x_1, x_2 \geq 0$ e inteiros

Removida a restrição $x_1 - x_2 \geq 0$.

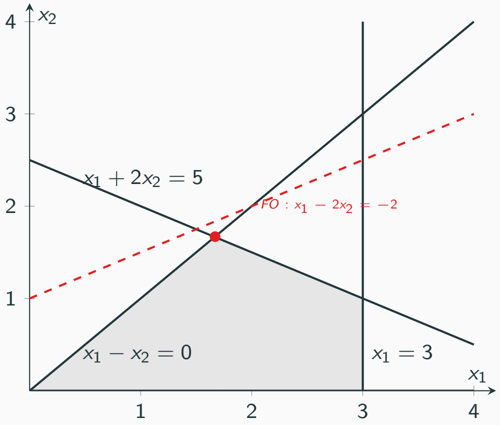


$v(RL_1) = c(0, 2) = 1 \times 0 - 2 \times 2 = -4$

Exemplo

$$
\begin{aligned}
(RL_2) \equiv \min z = x_1 - 2x_2 & \\
s.a: x_1 - x_2 &\geq 0 \\
x_1 + 2x_2 &\leq 5 \\
x_1 &\leq 3 \\
x_1, x_2 &\geq 0
\end{aligned}
$$

Removida a restrição de integralidade.

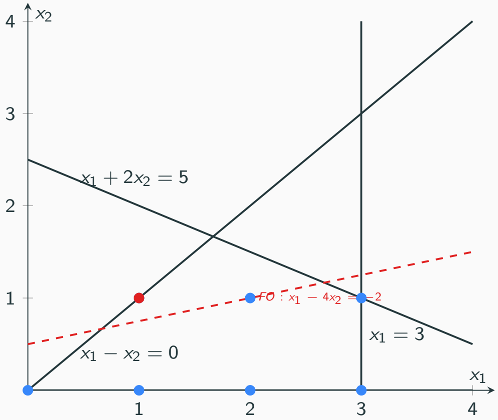


$v(RL_2) = c(\frac{5}{3}, \frac{5}{3}) = \frac{5}{3} - 2 \times \frac{5}{3} = -\frac{5}{3} \approx -1.67$

Exemplo

$$
\begin{aligned}
(RL_3) \equiv \min z = {} & x_1 - 4x_2 \\
s.a: {} & x_1 - x_2 \geq 0 \\
& x_1 + 2x_2 \leq 5 \\
& x_1 \leq 3 \\
& x_1, x_2 \geq 0 \text{ e inteiros}
\end{aligned}
$$

A FO é $x_1 - 4x_2$ em vez de $x_1 - 2x_2$.



$v(RL_3) = c(1,1) = 1 \times 1 - 4 \times 1 = -3$

Relaxações

Propriedade

(i) Se a relaxação RP é impossível, então o problema original IP é impossível.

(ii) Seja x^* a SO da relaxação RP e $f(x) = c(x), \forall x \in X$. Se $x^* \in X$, então x^* é a SO do problema IP.

Demonstração.

(i) Se RP é impossível, então $T = \emptyset$. Assim, como $X \subset T$ temos $T = \emptyset$ e podemos concluir que o problema original IP também é impossível.

(ii) Como x^* a SO da relaxação RP e $x^* \in X$, temos $z \leq c(x^*) f(x^*) = z^R$. Como RP é uma relaxação de IP temos $z^R \leq z$. Assim, $z = z^R$ e x^* é uma SO de IP, pois $z = c(x^*)$.

$\square$

Relaxação linear

A relaxação linear de um PLIM é obtida removendo as restrições de integralidade.

Definição: Relaxação linear

Dado o problema

$$(IP) \qquad z = \min\{c(x) : x \in X \subseteq \mathbb{Z}^n\},$$

a sua *relaxação linear* é

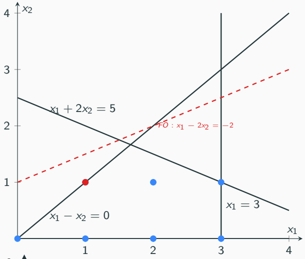
$$(PLR) \qquad z^{LR} = \min\{c(x) : x \in X \subseteq \mathbb{R}^n\}.$$

Para avaliar a qualidade do valor da relaxação linear podemos usar o gap.

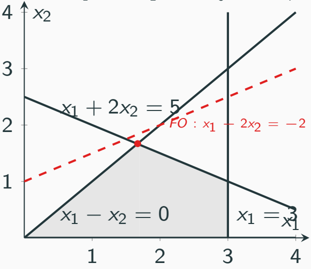
$$gap = \frac{z - z^{LR}}{z}.$$

Exemplo

$$(IP) \equiv z = \min x_1 - 2x_2$$
$$s.a: x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{ e inteiros}$$



$$(RL_2) \equiv z = \min x_1 - 2x_2$$
$$s.a: x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0$$



$$gap = \frac{-1-(-5/3)}{|-1|}[1] = \frac{2}{3} \implies 66.67\%.$$

[1] Usamos o módulo no denominador quando o valor ótimo é negativo.

Relembrando a dualidade

Definição

Os dois problemas

$$(IP) \qquad z = \max\{c(x) : x \in X\}$$

e

$$(D) \qquad w = \min\{w(u) : u \in U\}$$

formam um *par de problemas duais fraco* se $c(x) \leq w(u), \forall x \in X$ e $u \in U$. Quando $z = u$, temos um *par de problemas duais forte*.

- Num par de problemas duais, o valor de qualquer SA do problema de máximo é um minorante para o valor ótimo problema de mínimo.
  - O par de problemas duais apenas está definido para problemas de programação linear.

Relaxação Lagrangeana

- Considere-se o seguinte problema inteiro

$$(IP) \equiv z = \min\{c(x) : Dx \geq d \text{ e } x \in X\},$$

onde $Dx \geq d$ é o conjunto de restrições "complicadas".

Proposição

O problema

$$IP(u) \equiv z(u) = \min\{c(x) + u(d - Dx) : x \in X\}$$

é uma relaxação de IP para todo $u \geq 0$.

Demonstração.

Temos que $\{Dx \geq d \text{ e } x \in X\} \subseteq X$ e $c(x) + u(d - Dx) \leq c(x)$, logo pela definição de relaxação $IP(u)$ é uma relaxação de IP. $\square$

Relaxação Lagrangeana

Considere-se

$$(IP) \equiv z = \min\{c(x) : Dx \geq d \text{ e } x \in X\}$$

e

$$IP(u) \equiv z(u) = \min\{c(x) + u(d - Dx) : x \in X\},$$

com $u > 0$.

Definição

$IP(u)$ é a *relaxação Lagrangeana* de (IP) de parâmetro u.

Exemplo

$$
\begin{aligned}
(IP) \equiv z = \min\ & x_1 - 2x_2 \\
s.a:\ & x_1 - x_2 \geq 0 \implies \text{Restrições ``complicadas''} \\
& x_1 + 2x_2 \leq 5 \\
& x_1 \leq 3 \\
& x_1, x_2 \geq 0 \text{ e inteiros}
\end{aligned}
$$

- A relaxação lagrangeana é:

$$
\begin{aligned}
IP(u) \equiv z(u) = \min\ & x_1 - 2x_2 + u(0 - x_1 + x_2) = (1-u)x_1 + (-2+u)x_2 \\
s.a:\ & x_1 + 2x_2 \leq 5 \\
& x_1 \leq 3 \\
& x_1, x_2 \geq 0 \text{ e inteiros}
\end{aligned}
$$

Relaxação Lagrangeana

Em $IP(u)$ as restrições “complicadas” são adicionadas à função objetivo com uma penalidade u.

- O valor u é o multiplicador de Lagrange associado às restrições $Dx \geq d$.
- Na FO a penalidade é multiplicada pela pela “violação” da restrição relaxada.

O problema $IP(u)$ é a relaxação Lagrangeana de IP de parâmetro u.

- O valor ótimo de $IP(u)$ depende do valor de u.

Relaxação Lagrangeana

Como o $IP(u)$ é uma relaxação de IP, sabemos que $z(u) \leq z$. Contudo, $z(u)$ depende do valor de u.

- Para encontrar o melhor minorante possível para o valor ótimo de IP temos que otimizar o valor de u. $\Longrightarrow$ Resolver o *problema dual lagrangeano*.

Definição

O *problema dual Lagrangeano* define-se como

$$(LD) \equiv w_{DL} = \max\{z(u) : u \geq 0\}.$$

Exemplo

- O problema dual lagrangeano que queremos resolver é:

$$w_{DL} = \max_{u \geq 0} z(u) = \min_{(x_1,x_2) \in X} (1-u)x_1 + (-2+u)x_2$$

com $X = \{(x_1, x_2) \in \mathbb{Z}^2 : x_1 + 2x_2 \leq 5,\ x_1 \leq 3 \text{ e } x_1, x_2 \geq 0\}$.

- Queremos determinar a expressão de $z(u)$ para conseguirmos determinar o seu mínimo. Contudo, $z(u)$ depende da solução do PLI $\min\{(1-u)x_1 + (-2+u)x_2 : (x_1, x_2) \in X\}$.
    - Vamos determinar que pontos em X é que podem ser SOs do PLI referido.

Exemplo

- Suponhamos que $u \leq 1$

$$z(u) = \underbrace{(1-u)}_{\geq 0} x_1 + \underbrace{(-2+u)}_{<0} x_2$$

Como queremos minimizar, a SO é $(0,2)$ e

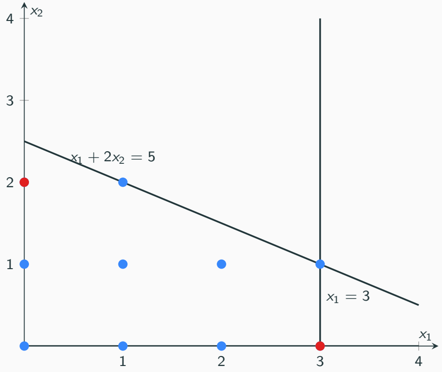
$$z(u) = (1-u) \times 0 + (-2+u) \times 2 = -4 + 2u.$$

- Suponhamos que $u \geq 2$

$$z(u) = \underbrace{(1-u)}_{<0} x_1 + \underbrace{(-2+u)}_{\geq 0} x_2$$

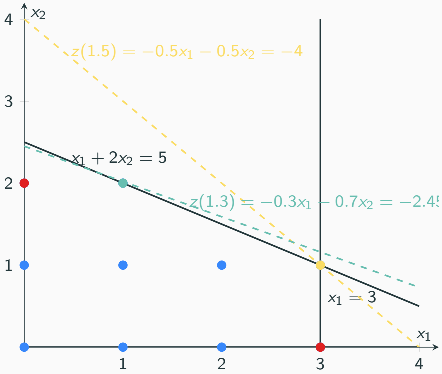
Como queremos minimizar, a SO é $(3,0)$ e

$$z(u) = (1-u) \times 3 + (-2+u) \times 0 = 3 - 3u.$$



  - O que acontece para $1 \leq u \leq 2$?
    - Que outros pontos em X podem ser soluções ótimas de $z(u)$?

Exemplo

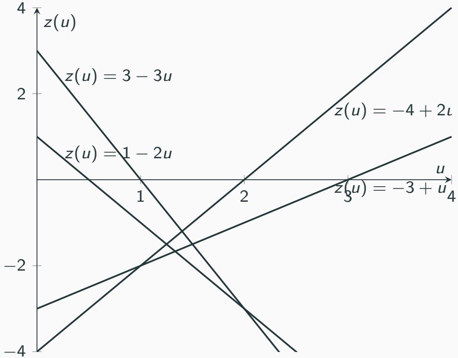


Para determinados valores de u, as soluções $(1, 2)$ e $(3, 1)$ também podem ser SO do problema.

Exemplo

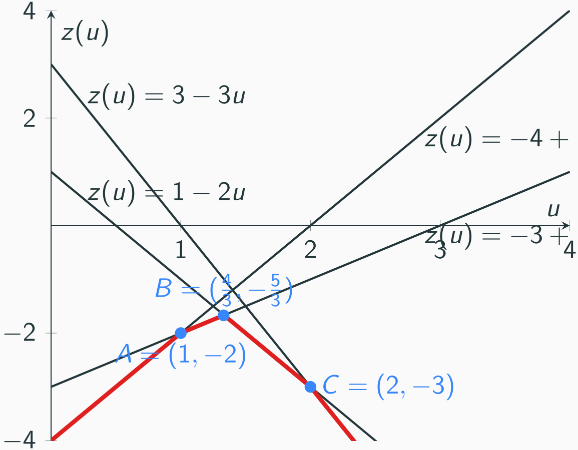
- Temos então as seguintes expressões para $z(u)$:
  - Ponto $(0,2) \implies z(u) = (1-u)\times 0 + (-2+u)\times 2 = 2u-4$
  - Ponto $(3,0) \implies z(u) = (1-u)\times 3 + (-2+u)\times 0 = -3u+3$
  - Ponto $(1,2) \implies z(u) = (1-u)\times 1 + (-2+u)\times 2 = u-3$
  - Ponto $(3,1) \implies z(u) = (1-u)\times 3 + (-2+u)\times 1 = -2u+1$

Resta-nos verificar qual é o mínimo de $z(u)$, isto é, para que valores de u são válidas.



Exemplo

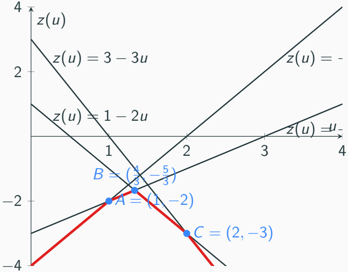
- Queremos determinar a reta que está a baixo (é o mínimo) de todas as outras.



Exemplo

Assim, a expressão da função que queremos maximizar é:

$$z(u) = \begin{cases} 2u - 4, & u \leq 1 \\ u - 3, & 1 \leq u \leq \frac{4}{3} \\ -2u + 1, & \frac{4}{3} \leq u \leq 2 \\ -3u + 3, & u \geq 2 \end{cases}$$



Qual o valor de u que maximiza $z(u)$? $\implies u = \frac{4}{3}$ (ponto B).

Para obtermos o melhor minorante possível agora temos que calcular $z(\frac{4}{3})$.

$$z\left(\frac{4}{3}\right) = \frac{4}{3} - 3 = -\frac{5}{3}$$

Relaxação Lagrangeana

Relembremos que $IP(u) \equiv z(u) = \min\{c(x) + u(d - Dx) : x \in X\}$.

Proposição

Se $u \geq 0$, e

(i) $x(u)$ é uma SO de $IP(u)$, e

(ii) $Dx(u) \geq d$, e

(iii) $(Dx(u))_i = d_i$ sempre que $u_i > 0$ (relações de complementaridade),

então $x(u)$ é SO de (IP).

Demonstração.

Por (i), como $x(u)$ é uma SO de $IP(u)$ temos $w_{DL} \geq z(u)$.

Além disso, $x(u)$ é admissível para o problema original (IP) pois $x \in X$ e satisfaz as restrições complicadas por (ii). Logo, $z(u) \geq z$.

Finalmente, por (iii), $z(u) = cx(u) + u(d - Dx(u)) = cx(u)$, pois quando $u > 0$ as restrições são satisfeitas na igualdade.

Como o problema dual lagrangeano é uma relaxação de (IP) temos $w_{DL} \leq z$. Logo, $w_{DL} = z$ e podemos concluir que $x(u)$ é uma SO de (IP). $\square$

- Caso as restrições a relaxar não estejam na forma $\geq$ as restrições de sinal da variável u alteram-se no problema dual lagrangeano.

Caso $\geq$

$(PLI) \equiv z = \min\{c(x) : Ax \geq b \text{ e } x \in X\}.$

$IP(u) \equiv z(u) = \min\{c(x) + u(b - Ax) : x \in X\}.$

$(DL) \equiv w_{DL} = \max\{z(u) : u \geq 0\}.$

Caso $\leq$

$(PLI) \equiv z = \min\{c(x) : Dx \leq d \text{ e } x \in X\}.$

$IP(v) \equiv z(v) = \min\{c(x) + v(d - Dx) : x \in X\}.$

$(DL) \equiv w_{DL} = \max\{z(v) : v \leq 0\}.$

Caso $=$

$(PLI) \equiv z = \min\{c(x) : Tx = t \text{ e } x \in X\}.$

$IP(y) \equiv z(y) = \min\{c(x) + y(t - Tx) : x \in X\}.$

$(DL) \equiv w_{DL} = \max\{z(y) : y \text{ livre } \}.$

Relaxação Lagrangeana

- Quão bom é o limite da relaxação lagrangeana?

Considere-se

$$
\begin{aligned}
(P) \equiv z = \min\ & cx \\
\text{s.a:}\ & Ax \geq b \implies \text{A relaxar} \\
& x \in X
\end{aligned}
$$

e

$$(DL) \equiv w^{DL} = \max_{u \geq 0} IP(u)$$

com

$$IP(u) \equiv z(u) = \min\{c(x) + u(b - Ax) : x \in X\}.$$

Proposição

$w^{DL} \geq z^{LR}$

- O valor da relaxação lagrangeana é melhor ou igual ao valor da relaxação linear. $\Longrightarrow$ A igualdade obtém-se quando os pontos extremos de X são inteiros.

Notas sobre a relaxação Lagrangeana

- Para obter o melhor valor possível para os multiplicadores de Lagrange é necessário resolver o problema dual Lagrangeano, que é não-linear.
  - Resolvido com métodos específicos. $\Longrightarrow$ Método do subgradiente.

- É possível relaxar simultaneamente vários conjuntos de restrições.
  - Temos tantos multiplicadores de Lagrange quantas restrições a relaxar. Exemplo:

$$
\begin{aligned}
&(P) \equiv \min\{c(x) : Ax \geq b, Dx \geq d, x \in X\} \\
&z(u_1, u_2) = \min\{c(x) + u_1(b - Ax) + u_2(d - Dx) : x \in X\} \\
&w^{DL} = \max\{z(u_1, u_2) : u_1, u_2 \geq 0\}
\end{aligned}
$$

Sumário

Seja

$$(PLI) \equiv z = \min\{c(x) : x \in S\}.$$

Ideia geral: decompor S em conjuntos "mais pequenos" e resolver o PLI nesses conjuntos.

Proposição

Seja $S = S_1 \cup S_2 \cup \ldots \cup S_{|K|}$ uma decomposição de S e seja $z^k = \min\{c(x) : x \in S_k\}, k \in K$. Então,

$$z = \min_{k \in K}\{z^k\}.$$

Algoritmos de enumeração

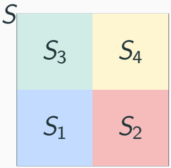
- Caso S_k não seja de resolução mais fácil pode ser decomposto, isto é,

$$S_k = S_{k_1} \cup S_{k_2} \cup \ldots .$$

- Este processo pode ser repetido até todos os problemas serem resolvidos.

$\Longrightarrow$ Pesquisa em árvore.

- Como decompor S em subconjuntos $S_1, \ldots, S_k$?
  - Os conjuntos $S_1, \ldots, S_k$ devem ser uma partição de S.



  - A forma mais simples é particionar S em dois subconjuntos S_1 e S_2.

Algoritmos de enumeração

Variáveis binárias

Seja x_i uma variável binária, isto é, $x_i \in \{0, 1\}$.

O conjunto S pode ser particionado em:

- $S_1 = \{x \in S : x_i = 0\}$
- $S_2 = \{x \in S : x_i = 1\}$

Variáveis inteiras

Seja x_i uma variável com valor fracionário, isto é, $x_i = \alpha \notin \mathbb{Z}$.

O conjunto S pode ser particionado em:

- $S_1 = \{x \in S : x_i \leq \lfloor \alpha \rfloor\}$
- $S_2 = \{x \in S : x_i \geq \lceil \alpha \rceil\}$

Podemos ir decompondo os conjuntos sucessivamente. $\Longrightarrow$ Árvore de enumeração.

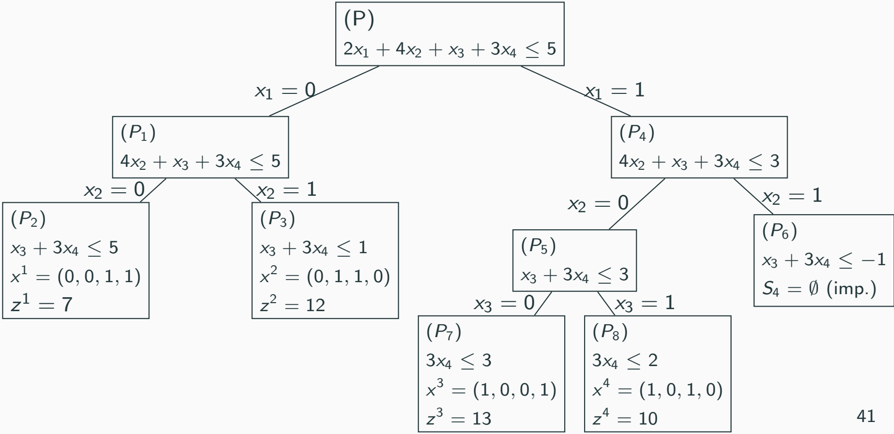
Exemplo

$$(P) \equiv \max\ 8x_1 + 10x_2 + 2x_3 + 5x_4$$

Problema saco-mochila

$$\text{s.a:}\ \ 2x_1 + 4x_2 + x_3 + 3x_4 \leq 5$$

$$x_1, x_2, x_3, x_4 \in \{0, 1\}$$



Algoritmos de enumeração

- A cada nodo k está associado um problema P_k e um conjunto S_k, que é a RA de P_k.
  - Se P_k é obtido de P_j através de uma ramificação diz-se que k é filho de j.
  - Se P_k é obtido de P_j através de uma sequência de ramificações diz-se que k é descendente de j.
  - Temos $v(P_j) \leq v(P_k)$, para todo o descendente k de j.
    - Relação válida para problemas de minimização.
  - Temos $v(P_j) \geq v(P_k)$, para todo o descendente k de j.
    - Relação válida para problemas de maximização.

Nota: Cada vez que ramificamos estamos a adicionar restrições e, consequentemente, a piorar o valor da solução ótima dos subproblemas.

- Como se relaciona z com os limites nos valores de z^k?

Proposição

Seja:

- $S = S_1 \cup S_2 \cup \ldots \cup S_K$ uma decomposição de S;
- $z^k = \min\{c(x) : x \in S_k\}, k = 1, \ldots, K$;
- $\overline{z}^k$ um limite superior para z^k; e
- $\underline{z}^k$ um limite inferior para z^k.

Então, $\overline{z} = \min_{k=1,\ldots,K}\{\overline{z}^k\}$ é um limite superior para z e $\underline{z} = \min_{k=1,\ldots,K}\{\underline{z}^k\}$ é um limite inferior para z.

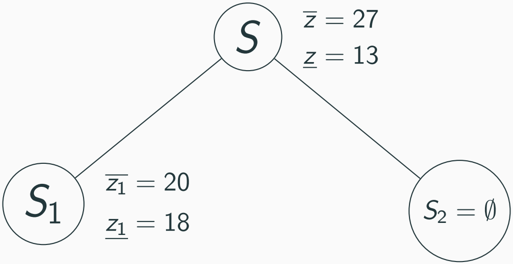
Algoritmos de enumeração

- Podemos usar a proposição anterior para reduzir o tamanho da árvore de enumeração.

- Um nodo k pode ser cancelado (não ramificado) em três situações:
  1. o problema é impossível, $S_k = \emptyset$;
  2. a solução obtida é admissível para o problema original P; ou
  3. é possível provar que P_k não contém a solução ótima (ou soluções melhores que a melhor solução conhecida para P).

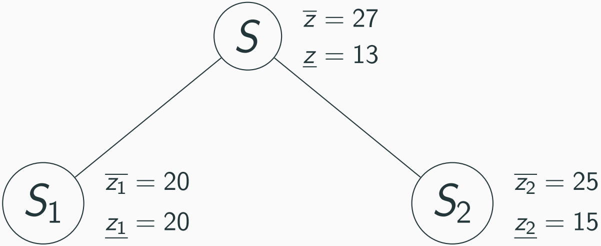
Algoritmos de enumeração

Problema de minimização



O problema P_2 é impossível, não é necessário ramificá-lo. $\Longrightarrow$ Corte por impossibilidade.

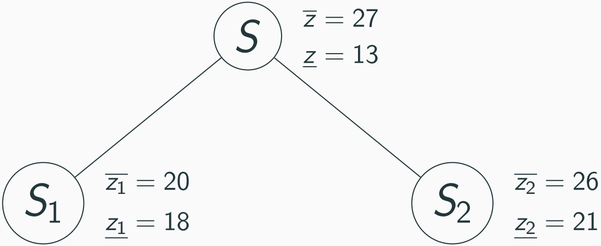
Problema de minimização



Como o limite superior e inferior de P_1 são iguais ($\overline{z}_1 = \underline{z}_1 = 20$) obtivemos o seu valor ótimo e por isso não é necessário ramificá-lo. $\Longrightarrow$ Corte por otimalidade.

Algoritmo de branch-and-bound

Problema de minimização



O valor ótimo de P_2 é pelo menos 21, enquanto o valor ótimo de S_1 é no máximo 20. Assim, como a partir de P_2 vamos sempre obter soluções piores que as de P_1 podemos não explorá-lo mais. $\Longrightarrow$ Corte por limite.

Algoritmo de branch-and-bound

- É um algoritmo de enumeração!
- Como resolver os subproblemas?
  - É resolvida a sua relaxação linear.
- Notação:

$(P) \equiv z = \min\{c(x) : x \in X\}$

$(P_k) \equiv z_k = \min\{c(x) : x \in X_k\}$

L Lista de problemas por resolver

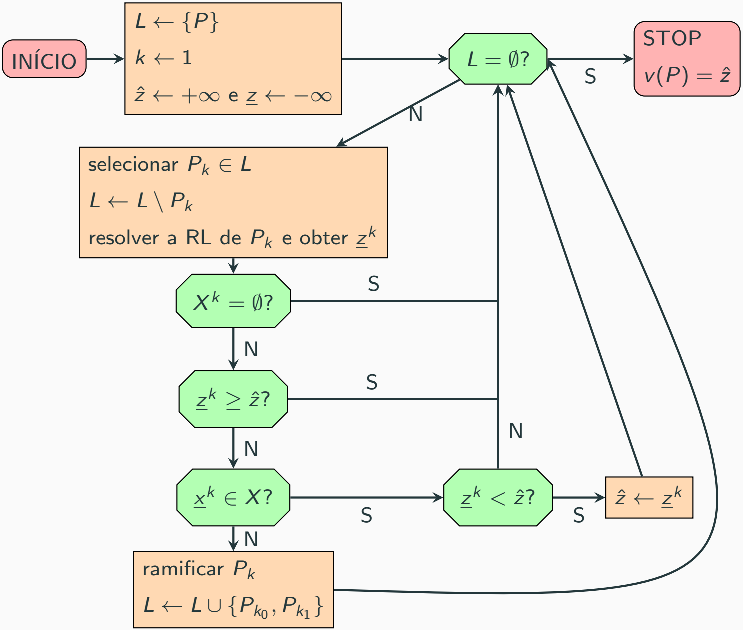
$\hat{z}$ valor da melhor SA conhecida - incumbente.

Caso não tenha sido determinada nenhuma SA, $\hat{z} = +\infty$.

$\underline{z}^k$ minorante de z^k.

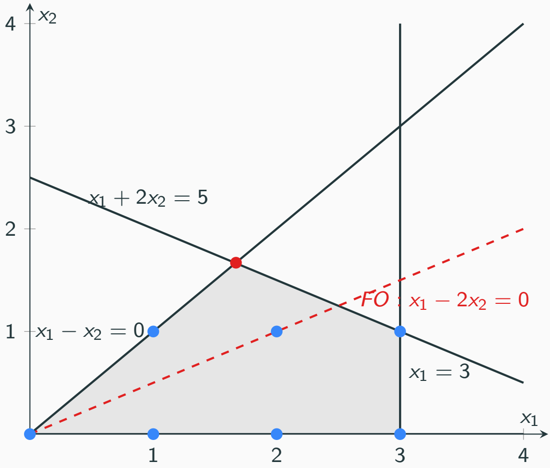
Temos $\underline{z}^k = -\infty$ se P_k impossível e $\underline{z}^k = z^k$ caso contrário.

Algoritmo de branch-and-bound



Exemplo

$$(P) \equiv \min z = x_1 - 2x_2$$
$$s.a : x_1 - x_2 \geq 0$$
$$x_1 + 2x_2 \leq 5$$
$$x_1 \leq 3$$
$$x_1, x_2 \geq 0 \text{ e inteiros}$$



Resolvendo a relaxação linear:

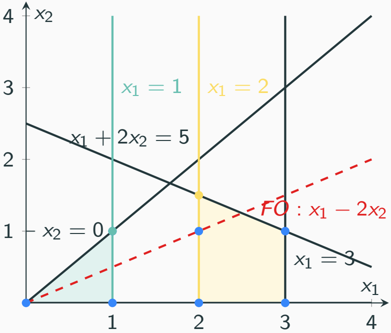
$x_{RL} = (\frac{5}{3}, \frac{5}{3})$ e $\underline{z} = \frac{5}{3} - 2 \times \frac{5}{3} = -\frac{5}{3}$

(P)
$\underline{z} = -\frac{5}{3}$
$\hat{z} = +\infty$ $= -1$

$x_1 \leq \lfloor \frac{5}{3} \rfloor = 1$ — $x_1 \geq \lceil \frac{5}{3} \rceil = 2$

(P_1)
$\underline{z_1} = -1$
$\hat{z}_1 = -1$

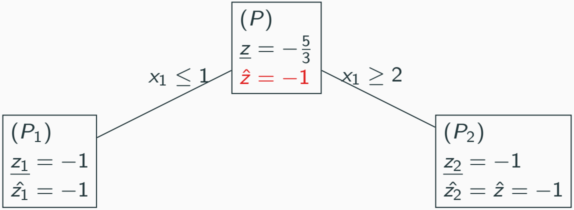
(P_2)
$\underline{z_2} = -1$
$\hat{z}_2 = \hat{z} = -1$



- $(P_1) : x_1^* = (1, 1)$ e $\underline{z_1} = \overline{z_1} = -1$
- $(P_2) : x_2^* = (2, \frac{3}{2})$ e $\underline{z_2} = -1$

■ Não é preciso ramificar mais a partir de P_1 pois já encontramos uma SA. Podemos atualizar o majorante.

Exemplo



Como o minorante de P_2 é igual a $\hat{z}$ podemos cancelar o nodo pois não vai conter a SO. Todas as soluções de descendentes de P_2 vão ter valor pior ou igual que -1.

Algoritmo de branch-and-bound

- Estratégias de ramificação e pesquisa
  - Como ramificar?
    - Escolher a variável fracionária com valor α mais próximo de $\lfloor \alpha \rfloor + \frac{1}{2}$ na relaxação linear (variável "mais fracionária").
    - Escolher a variável fracionária com o índice mais baixo.
  - Como selecionar o subproblema?
    - Pesquisa em profundidade.
      Escolher um dos últimos problemas criados. $\Longrightarrow$ Descer rapidamente na árvore de pesquisa de modo a encontrar rapidamente uma solução admissível.
    - Seleção do melhor nodo.
      Escolher o problema em aberto com maior valor de $\underline{z}^k$.

Algoritmo de branch-and-bound

- Estratégias de redução do número de nodos da árvore
  - Utilizar heurísticas para determinar soluções admissíveis (majorantes).
  - Se os coeficientes da função objetivo são inteiros, z só toma valores inteiros para soluções admissíveis. Neste caso os minorantes $\underline{z}^k$ podem ser substituídos por $\lceil \underline{z}^k \rceil$.
  - Tentar melhorar a qualidade dos minorantes.

■ O algoritmo de branch-and-bound pode ser utilizado como uma heurística.

$\Longrightarrow$ Por exemplo, parando o algoritmo quando uma SA for encontrada.

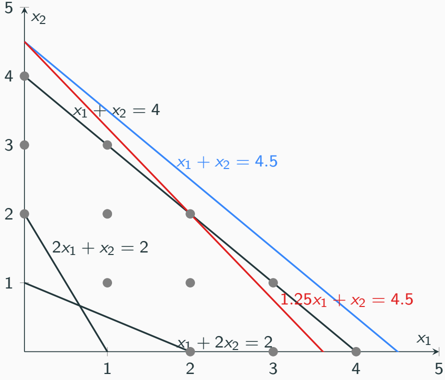
Algoritmo de planos de corte de Gomory

Definição

Seja $X \subseteq \mathbb{R}^m$. Uma *desigualdade válida* para X é uma desigualdade $\pi_1 x_1 + \ldots + \pi_n x_n \leq \pi_0$ ($\pi x \leq \pi_0$) que é satisfeita por todos os pontos de X.

Exemplo

Seja $X = \{(x_1, x_2) \in \mathbb{Z}^2 : x_1 + x_2 \leq 4, x_1 + 2x_2 \geq 2, 2x_1 + x_2 \geq 2 \text{ e } x_1, x_2 \geq 0\}$. $\Longrightarrow$ Pontos cinzentos no gráfico.



A desigualdade $x_1 + x_2 \leq 4.5$ é uma desigualdade válida para X.

A desigualdade $1.25x_1 + x_2 \leq 4.5$ não é uma desigualdade válida para X. $\Longrightarrow$ Os pontos $(3, 1)$ e $(4, 0)$ pertencem a X e não satisfazem a desigualdade.

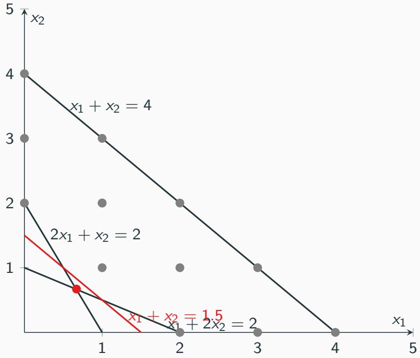
Algoritmo de planos de corte de Gomory

Definição

Sejam $X \subseteq \mathbb{R}^m$ e $x^* \notin X$. Uma desigualdade válida para X não satisfeita por x^*, diz-se um *plano de corte*.

Exemplo

Seja $X = \{(x_1, x_2) \in \mathbb{Z}^2 : x_1 + x_2 \leq 4, x_1 + 2x_2 \geq 2, 2x_1 + x_2 \geq 2 \text{ e } x_1, x_2 \geq 0\}$. $\Longrightarrow$ Pontos cinzentos no gráfico.



A desigualdade $x_1 + x_2 \geq 1.5$ é um plano de corte para $x^* = (\frac{1}{3}, \frac{1}{3})$, que é a SO da relaxação linear do problema

$$\min\{3x_1 + 2x_2 : (x_1, x_2) \in X\}.$$

- Esta é a ideia geral do algoritmo de planos de corte de Gomory.

Algoritmo de planos de corte de Gomory

Seja

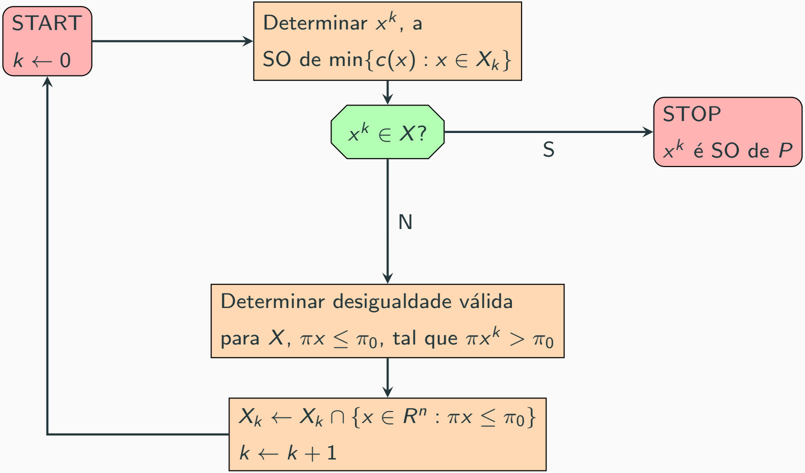
$$(P) \equiv \min\{c(x) : x \in X\},$$

com $X = \{x \in R^n : Ax \geq b$ e x_j inteiro, para $j \in J\}$ sendo J um subconjunto de $\{1, ..., n\}$.

Ideia geral: resolver uma sucessão de relaxações lineares até à solução obtida estar em X. Cada relaxação linear é obtida da anterior acrescentando-lhe planos de corte.

Algoritmo de planos de corte de Gomory

Seja $\overline{X}_0 = \{x \in R^n : Ax \geq b\}$ (X sem as restrições de integralidade).



Algoritmo de planos de corte de Gomory

Como determinar planos de corte para a SO da relaxação linear?

- Procedimento de geração de cortes de Gomory, que é válido para qualquer PLIM

Procedimento de geração de cortes de Gomory

Seja

$$(P) \equiv z = \min\{c(x) : Ax \geq b,\ x \geq 0 \text{ e inteiros}\}.$$

Passo 1: Resolver a relaxação linear de P.

Passo 2: Escrever o sistema $Ax \geq b$ na forma aumentada.

Passo 3: Rescrever o sistema em função das VB da SO, isto é, $x_i + \sum_{j \in VNB} \bar{a}_{ij} x_j = \bar{b}_i$ para toda a VB x_i.

Passo 4: O corte de Gomory é obtido através do seguinte procedimento:

$$x_i + \sum_{j \in VNB} \bar{a}_{ij} x_j = \bar{b}_i \implies x_i + \sum_{j \in VNB} \lfloor \bar{a}_{ij} \rfloor x_j \leq \bar{b}_i \implies x_i + \sum_{j \in VNB} \lfloor \bar{a}_{ij} \rfloor x_j \leq \lfloor \bar{b}_i \rfloor$$

$$\implies \sum_{j \in VNB} (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \geq \bar{b}_i - \lfloor \bar{b}_i \rfloor$$

Algoritmo de planos de corte de Gomory

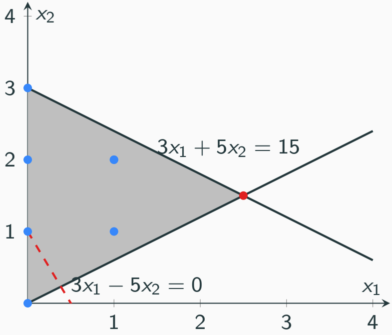
Definição

O *corte de Gomory* é $\sum_{j \in VNB} (\bar{a}_{ij} - \lfloor \bar{a}_{ij} \rfloor) x_j \geq \bar{b}_i - \lfloor \bar{b}_i \rfloor$.

Exemplo

$$
\begin{aligned}
(P) \equiv \max z &= 2x_1 + x_2 \\
\text{s.a: } & 3x_1 - 5x_2 \leq 0 \\
& 3x_1 + 5x_2 \leq 15 \\
& x_1, x_2 \geq 0 \text{ e inteiros}
\end{aligned}
$$

- Resolvendo a relaxação linear de P obtém-se $x^0 = (\frac{5}{2}, \frac{3}{2})$.



Exemplo

- Escrevendo o sistema na forma aumentada obtemos:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \end{cases}$$

- A solução na forma aumentada é $x^0 = (\frac{2}{3}, \frac{2}{3}, 0, 0)$ e tem como VBs x_1 e x_2.

Exemplo

- Reescrevendo o sistema em função das VB obtemos:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \end{cases} \iff \begin{cases} x_1 = \frac{5}{3}x_2 - \frac{1}{3}x_3 \\ 3(\frac{5}{3}x_2 - \frac{1}{3}x_3) + 5x_2 + x_4 = 15 \end{cases}$$

$$\iff \begin{cases} - \\ 5x_2 - x_3 + 5x_2 + x_4 = 15 \end{cases} \iff \begin{cases} x_1 = \frac{5}{3}(\frac{15}{10} + \frac{1}{10}x_3 - \frac{1}{10}x_4) - \frac{1}{3}x_3 \\ x_2 - \frac{1}{10}x_3 + \frac{1}{10}x_4 = \frac{15}{10} \end{cases}$$

$$\iff \begin{cases} x_1 = \frac{5}{2} - \frac{1}{6}x_3 - \frac{1}{6}x_4 \\ x_2 - \frac{1}{10}x_3 + \frac{1}{10}x_4 = \frac{15}{10} \end{cases} \iff \begin{cases} x_1 + \frac{1}{6}x_3 + \frac{1}{6}x_4 = \frac{5}{2} \\ x_2 - \frac{1}{10}x_3 + \frac{1}{10}x_4 = \frac{3}{2} \end{cases}$$

Como ambos os $\bar{b}_i$ são fracionários podemos escolher qualquer restrição para gerar um corte de Gomory. $\Longrightarrow$ Vamos escolher a linha do x_1.

$$(\frac{1}{6} - 0)x_3 + (\frac{1}{6} - 0)x_4 \geq (\frac{5}{2} - 2) \iff \frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2}$$

Exemplo

- Temos que escrever o corte $\frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2}$ em função de x_1 e x_2 para o conseguirmos representar no gráfico.
- Da forma aumentada de P tiramos que:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \iff x_3 = -3x_1 + 5x_2 \\ 3x_1 + 5x_2 + x_4 = 15 \iff x_4 = 15 - 3x_1 - 5x_2 \end{cases}$$

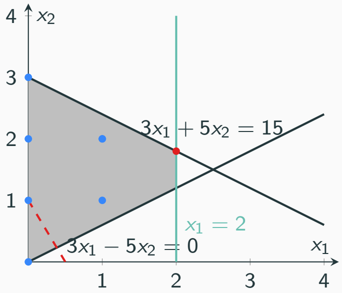
- Logo,

$$\frac{1}{6}x_3 + \frac{1}{6}x_4 \geq \frac{1}{2} \iff \frac{1}{6}(-3x_1 + 5x_2) + \frac{1}{6}(15 - 3x_1 - 5x_2) \geq \frac{1}{2}$$
$$\iff -\frac{3}{6}x_1 + \frac{5}{6}x_2 + \frac{15}{6} - \frac{3}{6}x_1 - \frac{5}{6}x_2 \geq \frac{1}{2} \iff -x_1 \geq -\frac{12}{6} \iff x_1 \leq 2$$

Exemplo

$$
\begin{aligned}
(P^1) \equiv \max z &= 2x_1 + x_2 \\
\text{s.a: } & 3x_1 - 5x_2 \leq 0 \\
& 3x_1 + 5x_2 \leq 15 \\
& x1 \leq 2 \\
& x_1, x_2 \geq 0 \text{ e inteiros}
\end{aligned}
$$

- Resolvendo a relaxação linear de P^1 obtém-se $x^1 = (2, \frac{9}{5})$.



Exemplo

- Escrevendo o sistema na forma aumentada obtemos:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \end{cases}$$

- A solução na forma aumentada é $x^1 = (2, \frac{9}{5}, 3, 0, 0)$ e tem como VBs x_1, x_2 e x_3.

Exemplo

- Reescrevendo o sistema em função das VB obtemos:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \end{cases} \iff \begin{cases} - \\ x_2 = 3 - \frac{3}{5}x_1 - \frac{1}{5}x_4 \\ x_1 = 2 - x_5 \end{cases} \iff \begin{cases} - \\ x_2 = \frac{9}{5} + \frac{3}{5}x_5 - \frac{1}{5}x_4 \\ - \end{cases}$$

$$\iff \begin{cases} 3(2 - x_5) - 5(\frac{9}{5} + \frac{3}{5}x_5 - \frac{1}{5}x_4) + x_3 = 0 \\ - \\ - \end{cases} \iff \begin{cases} x_3 + x_4 - 6x_5 = 3 \\ x_2 + \frac{1}{5}x_4 - \frac{3}{5}x_5 = \frac{9}{5} \\ x_1 + x_5 = 2 \end{cases}$$

Vamos escolher a restrição associada a x_2 para gerar um corte de Gomory porque é a única com um $\bar{b}_i$ fracionário.

$$(\frac{1}{5} - 0)x_4 + (-\frac{3}{5} - (-1))x_5 \geq (\frac{9}{5} - 1) \iff \frac{1}{5}x_4 + \frac{2}{5}x_5 \geq \frac{4}{5}$$

Exemplo

- Temos que escrever o corte $\frac{1}{5}x_4 + \frac{2}{5}x_5 \geq \frac{4}{5}$ em função de x_1 e x_2 para o conseguirmos representar no gráfico.
- Da forma aumentada de P^1 tiramos que:

$$\begin{cases} x_1 + 5x_2 + x_4 = 15 \iff x_4 = 15 - 3x_1 - 5x_2 \\ x_1 + x_5 = 2 \iff x_5 = 2 - x_1 \end{cases}$$

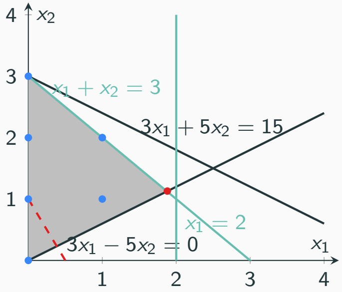
- Logo,

$$\frac{1}{5}x_4 + \frac{2}{5}x_5 \geq \frac{4}{5} \iff \frac{1}{5}(15 - 3x_1 - 5x_2) + \frac{2}{5}(2 - x_1) \geq \frac{4}{5}$$
$$\iff x_1 + x_2 \leq 3$$

Exemplo

$$\begin{aligned}(P^2) \equiv \max z &= 2x_1 + x_2 \\ \text{s.a: } & 3x_1 - 5x_2 \leq 0 \\ & 3x_1 + 5x_2 \leq 15 \\ & x1 \leq 2 \\ & {\color{red} x_1 + x_2 \leq 3} \\ & x_1, x_2 \geq 0 \text{ e inteiros}\end{aligned}$$

- Resolvendo a relaxação linear de P^1 obtém-se $x^2 = (\frac{15}{8}, \frac{9}{8})$.



Exemplo

- Escrevendo o sistema na forma aumentada obtemos:

$$
\begin{cases}
3x_1 - 5x_2 + x_3 = 0 \\
3x_1 + 5x_2 + x_4 = 15 \\
x_1 + x_5 = 2 \\
x_1 + x_2 + x_6 = 3
\end{cases}
$$

- A solução na forma aumentada é $x^2 = (\frac{15}{8}, \frac{9}{8}, 0, \frac{30}{8}, \frac{1}{8}, 0)$ e tem como VBs x_1, x_2, x_4 e x_5.

Exemplo

- Reescrevendo o sistema em função das VB obtemos:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \\ 3x_1 + 5x_2 + x_4 = 15 \\ x_1 + x_5 = 2 \\ x_1 + x_2 + x_6 = 3 \end{cases} \iff \dots \iff \begin{cases} x_1 + \frac{1}{8}x_3 + \frac{5}{8}x_6 = \frac{15}{8} \\ \frac{2}{8}x_3 + x_4 - \frac{30}{8}x_6 = \frac{30}{8} \\ -\frac{1}{8}x_3 + x_5 - \frac{5}{8}x_6 = \frac{1}{8} \\ x_2 - \frac{1}{8}x_3 + \frac{3}{8}x_6 = \frac{9}{8} \end{cases}$$

Como todos os $\bar{b}_i$ são fracionários podemos escolher qualquer restrição para gerar um corte de Gomory. $\implies$ Vamos escolher a linha do x_1.

$$(\frac{1}{8} - 0)x_3 + (\frac{5}{8} - 0)x_6 \geq (\frac{15}{8} - 1) \iff \frac{1}{8}x_3 + \frac{5}{8}x_6 \geq \frac{7}{8}$$

Exemplo

- Temos que escrever o corte $\frac{1}{8}x_3 + \frac{5}{8}x_6 \geq \frac{7}{8}$ em função de x_1 e x_2 para o conseguirmos representar no gráfico.
- Da forma aumentada de P^2 tiramos que:

$$\begin{cases} 3x_1 - 5x_2 + x_3 = 0 \iff x_3 = -3x_1 + 5x_2 \\ x_1 + +x_2 + x_6 = 3 \iff x_6 = 3 - x_1 - x_2 \end{cases}$$

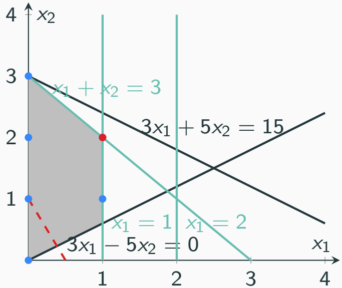
- Logo,

$$\frac{1}{8}x_3 + \frac{5}{8}x_6 \geq \frac{7}{8} \iff \frac{1}{8}(-3x_1 + 5x_2) + \frac{5}{8}(3 - x_1 - x_2) \geq \frac{7}{8}$$
$$\iff x_1 \leq 1$$

Exemplo

$$(P^3) \equiv \max z = 2x_1 + x_2$$
$$\text{s.a: } 3x_1 - 5x_2 \leq 0$$
$$3x_1 + 5x_2 \leq 15$$
$$x1 \leq 2,\ x_1 + x_2 \leq 3$$
$$x_1 \leq 1$$
$$x_1, x_2 \geq 0 \text{ e inteiros}$$

- Resolvendo a relaxação linear de P^3 obtém-se $x^3 = (1, 2)$. Como x^3 é inteira é a SO de P. Logo, $z = 2 \times 1 + 2 = 4$. $\Longrightarrow$ *FIM.*



Técnicas de melhoria

- Para melhorar o desempenho dos algoritmos apresentados existem algumas técnicas que podemos adotar.
  - Pré-processamento.
    - Fixar variáveis e/ou melhorar limites superiores e inferiores de variáveis.
    - Remover restrições e/ou variáveis redundantes.
    - Melhorar coeficientes/termos independentes de restrições.
  - Adição de desigualdades válidas.
    - Juntar restrições que são redundantes para P mas que melhoram o valor da relaxação linear.

Pré-processamento

- Ideia geral: simplificar a formulação inicial.

- Algumas das técnicas utilizadas são:
  - Fixação de variáveis.
  - Eliminação de restrições redundantes.
  - "Fortelecer" restrições de forma a reduzir a RA do PLR.

- As técnicas utilizadas não podem cortar soluções admissíveis para o problema.

- Os softwares que resolvem problemas de PLIM aplicam algumas destas técnicas.

Pré-processamento

- Fixação de variáveis
  - É possível fixar o valor de algumas variáveis observando apenas as restrições do modelo.
- Exemplos:

Considere-se x_1, x_2 e x_3 variáveis binárias. $\implies x_i \in \{0,1\}, i = 1,2,3.$

1. $3x_1 + x_2 + x_3 \leq 2 \implies x_1 = 0$
2. $x_1 + x_2 - 2x_3 \geq 2 \implies x_3 = 0$
3. $5x_1 + x_2 - 2x_3 \leq 2 \implies x_1 = 0$
4. $x_1 + 3x_2 - x_3 \geq 2 \implies x_2 = 1$
5. $3x_1 + x_2 - 3x_3 \geq 2 \implies x_3 = 0$ e $x_1 = 1$

- Ao fixarmos o valor de uma variável devemos substituí-la na formulação.

Pré-processamento

- Remoção de restrições redundantes

Definição

Considere-se $\pi x \leq \pi_0$ e $\mu x \leq \mu_0$ duas desigualdades válidas para X. Dizemos que $\pi x \leq \pi_0$ *domina* $\mu x \leq \mu_0$ se existe uma constante $u \in \mathbb{R}$ tal que $\pi \geq u\mu$, $\pi_0 \leq u\mu_0$ e $(\pi, \pi_0) \neq (u\mu, u\mu_0)$.

Definição

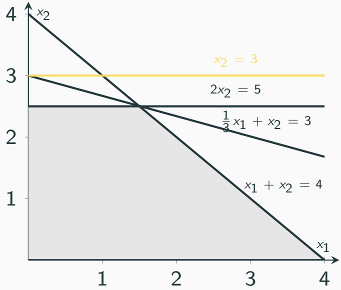
Uma desigualdade válida $\pi x \leq \pi_0$ é *redundante* para X se existem $k \geq 2$ restrições de P da forma $\mu^i x \leq \mu_0^i$ e pesos $u_i \geq 0$ com $i = 0, \ldots, k$ tais que $(\sum_{i=1}^k u_i \mu^i)x \leq \sum_{i=1}^k u_i \mu_0^i$ domina $\pi x \leq \pi_0$.

- Uma restrição redundante não é necessária para o modelo, podendo ser removida.

Exemplo

Seja

$$X = \{(x_1, x_2) \in \mathbb{R}^2 : 2x_1 \leq 5,\ \frac{1}{3}x1 + x_2 \leq 3,\ x_1 + x_2 \leq 4 \text{ e } x_1, x_2 \geq 0\}.$$



- A desigualdade $x_2 \leq 3$ é dominada por $2x_2 \leq 5 \iff x_2 \leq \frac{5}{2}$, pois o lado esquerdo é igual e $\frac{5}{2} < 3$.
- A desigualdade $(1/3)x1 + x_2 \leq 3$ é redundante para X pois pode ser obtida como uma combinação linear de $2x_1 \leq 5$ e $x_1 + x_2 \leq 4$, isto é,

$$(1/3)(2x_1 \leq 5) + (1/3)(x_1 + x_2 \leq 4) \iff (1/3)x_1 + x_2 \leq 3$$

  - O conjunto X é o mesmo sem a restrição $(1/3)x_1 + x_2 \leq 3$.

Pré-processamento

- Fortalecimento de restrições
  - Em certos casos, é possível apenas por inspeção tornar as restrições mais fortes.
  - Existem algoritmos para tornar mais fortes restrições com variáveis binárias.
- Exemplos (por inspeção):

Sejam x_1 e x_2 variáveis inteiras não negativas e y_1 e y_2 variáveis binárias.

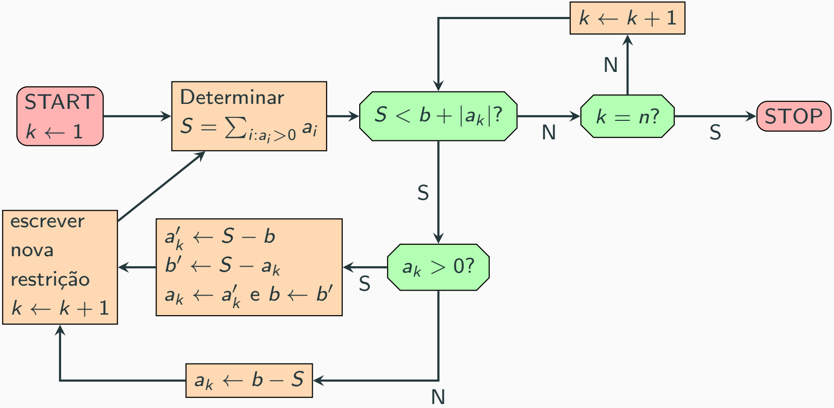
1. $x_1 + x_2 \leq 2.5 \Longrightarrow x_1 + x_2 \leq 2$
2. $4y_1 + 3y_2 \leq 4 \Longrightarrow y_1 + y_2 \leq 1$
3. $x_1 \leq 3$ e $x_1 \leq My_1$, com M suficientemente grande $\Longrightarrow$ $x_1 \leq 3$ e $x_1 \leq 3y_1$

Pré-processamento

- Fortalecimento de restrições
- Algoritmo para fortalecer restrições com variáveis binárias

Considere-se uma restrição de $\leq$ com variáveis binárias, isto é,

$$a_1x_1 + a_2x_2 + \ldots + a_nx_n \leq b,\ x_i \in \{0,1\}, i = 1 \ldots n.$$



Exemplo

- Fortalecer a restrição $4x_1 - 3x_2 + x_3 + 2x_4 \leq 5$, com $x_i \in \{0, 1\},\ i = 1, 2, 3, 4$.
- k = 1
  - S = 4 + 1 + 2 = 7
  - $S = 7 < b + |a_1| = 5 + 5 = 9$? Sim.
    - $a_1 = 4 > 0$? Sim.
    - $a_1' \leftarrow 7 - 5 = 2, b' \leftarrow 7 - 4 = 3$
    - Nova restrição: $2x_1 - 3x_2 + x_3 + 2x_4 \leq 3$
- k = 2
  - S = 2 + 1 + 2 = 5
  - $S = 5 < b + |a_2| = 3 + 3 = 6$? Sim.
    - $a_2 = -3 > 0$? Não.
    - $a_2 \leftarrow 3 - 5 = -2$
    - Nova restrição: $2x_1 - 2x_2 + x_3 + 2x_4 \leq 3$
- k = 3
  - S = 2 + 1 + 2 = 5
  - $S = 5 < b + |a_3| = 3 + 1 = 4$? Não.
  - 3 = 4? Não.
- k = 4
  - S = 2 + 1 + 2 = 5
  - $S = 5 < b + |a_4| = 3 + 2 = 5$? Não.
  - 4 = 4? Sim. FIM.

- A restrição fortalecida é $2x_1 - 2x_2 + x_3 + 2x_5 \leq 3$.

Adição de desigualdades válidas

- Uma forma de tornar os algoritmos apresentados mais eficientes é "aproximando" a RA do PLR com o invólucro convexo do problema original. $\Longrightarrow$ Feito através da adição de desigualdades válidas, que cortem soluções fracionárias.

- Existem várias formas de obter desigualdades válidas para um problema.
  - Algoritmos de geração de desigualdades válidas. $\Longrightarrow$ Algoritmo de Chavátal-Gomory.
  - Desigualdades válidas baseadas nas especificidades do problema. $\Longrightarrow$ Famílias de desigualdades válidas.
    - Por exemplo, desigualdades de cobertura para o problema do saco-mochila.

Adição de desigualdades válidas

Considere-se o caso geral do problema do saco-mochila

$$\max\{c(x) : x \in X\},$$

com

$$X = \{x \in \{0,1\}^n : \sum_{i=1}^{n} a_i x_i \leq b\}.$$

Definição

Uma *cobertura* é um subconjunto de artigos $S \subseteq N$ que não cabem todos na mochila, isto é,

$$\sum_{i \in S} a_i > b.$$

Definição

Sendo S uma cobertura, uma desigualdade válida - *desigualdade cobertura* - para o problema do saco-mochila é

$$\sum_{i \in S} x_i \leq |S| - 1.$$

Exemplo

$$\max z = 32x_1 + 20x_2 + 19x_3 + 13x_4 + 11x_5 + 8x_6$$
$$\text{s.a: } 12x_1 + 9x_2 + 7x_3 + 5x_4 + 5x_5 + 3x_6 \leq 14$$
$$x_i \in \{0, 1\}, \; i = 1, \ldots, 6$$

- Resolvendo (Solver) a relaxação linear do problema apresentado obtemos $z^{PLR} = 37.(6)$. $\Longrightarrow$ Quando temos variáveis binárias é necessário adicionar as restrições $x_i \leq 1$ à relaxação linear.

- Desigualdades de cobertura para a instância apresentada são, por exemplo:

1. $x_1 + x_2 \leq 1$
2. $x_1 + x_3 \leq 1$
3. $x_1 + x_6 \leq 1$
4. $x_2 + x_3 + x_4 \leq 2$
5. $x_2 + x_3 \leq 1$ $\Longrightarrow$ Domina a desigualdade 4.

Adicionando estas desigualdades válidas à relaxação linear obtemos $z^{PLR^+} = 37.4$.

Notas sobre técnicas de melhoria

- Apesar de alguns softwares aplicarem técnicas de pré-processamento, devemos tentar sempre dar ao software uma formulação o mais "simplicada" possível. $\Longrightarrow$ Temos conhecimento sobre o nosso problema que o solver não tem.

- Uma técnica não abordada neste capítulo são as restrições de quebra de simetria.

Referências bibliográficas

Hillier, F. S. & Lieberman, G. J. (2021). *Introduction to Operations Research* Mc Graw-Hill, New York, USA, 11th edition.

Wolsey, L. A. (1998). *Integer programming*. John Wiley & Sons