Universidade Estadual de Campinas - UNICAMP
Instituto de Computação - IC
MO824 - Tópicos em Otimização Combinatória

Combinatória Poliédrica – Parte II
O Problema da Mochila

Cid Carvalho de Souza

2^o semestre de 2006

Planos de corte para mochila binária

- Conjunto de soluções: $S = \{x \in \mathbb{B}^n : \sum_{j \in N} a_j x_j \leq b\}$, onde $N = \{1, \ldots, n\}$ é o conjunto de itens.
- Notação e convenções:
  - $a_j \leq b$ para todo $j \in N$ e $\sum_{j \in N} a_j > b$.
  - $a_1 \geq a_2 \geq \ldots \geq a_n > 0$.
  - Se $R \subseteq N$, x^R denota o *vetor de incidência (ou característico)* de R. Ou seja, $x^R \in \mathbb{B}^n$ e, para todo $j \in N$, $x_j^R = 1$ se e somente se $j \in R$.
  - $\mathcal{P} = \texttt{conv}(S) \subset \mathbb{R}^n$: a envoltória convexa dos vetores de incidência das soluções viáveis da mochila (um poliedro).

Estudo facial do poliedro $\mathcal{P}$

$\dim(\mathcal{P}) = n$, ou seja, o politopo da mochila tem dimensão cheia.

A desigualdade trivial $x_j \geq 0$ define uma facet de $\mathcal{P}$ para todo $j \in N$.

Dado $j \in N$, a desigualdade trivial $x_j \leq 1$ define uma facet de $\mathcal{P}$ *se e somente se* $a_i + a_j \leq b$ para todo $i \in N - \{j\}$.

Vamos provar estes resultados ?

Estudo facial do poliedro $\mathcal{P}$

Definição:

Seja $R \in N$. R é um **conjunto independente** de S se $\sum_{j \in R} a_j \leq b$. Por outro lado, se $\sum_{j \in R} a_j > b$, R é dito ser um **conjunto dependente** ou uma **cobertura** (*cover*) de S.

Observação:

Todo vértice *não inteiro* da relaxação

$$\{x \in \mathbb{R}^n : \textstyle\sum_{j \in N} a_j x_j \leq b, 0 \leq x_j \leq 1 \text{ para } j \in N\}$$

é da forma:

$$\begin{cases} x_j^* = 1, & \text{para } j \in C - \{k\}, \\ x_j^* = 0, & \text{para } j \in N \setminus C, \\ x_k^* = (b - \sum_{j \in C - \{k\}} a_j x_j^*)/a_k, & \end{cases}$$

para algum *conjunto dependente* C.

Estudo facial do poliedro $\mathcal{P}$

Proposição: (cover inequality)

Seja C uma cobertura para S. Então a desigualdade

$$\sum_{j \in C} x_j \leq |C| - 1,$$

é válida para $\mathcal{P}$.

Observações:

- a prova desta proposição é imediata pela própria definição de cobertura.
- usaremos a notação $x(R)$ para denotar $\sum_{j \in R} x_j$. Assim, a desigualdade acima para uma cobertura C pode ser escrita como $x(C) \leq |C| - 1$.

Estudo facial do poliedro $\mathcal{P}$

Exemplo: desigualdade de cobertura

Seja $S = \{x \in \mathbb{B}^7 : 11x_1 + 8x_2 + 7x_3 + 6x_4 + 6x_5 + 6x_6 + x_7 \leq 19\}$.
Então, $x_3 + x_4 + x_5 + x_6 \leq 3$ é uma desigualdade de cobertura para S.

Observação:

Uma cobertura C de S é dita ser **minimal** se a remoção de qualquer item de C faz com que ele se torne um conjunto independente.
Se C não for uma cobertura minimal, a desigualdade $x(C) \leq |C| - 1$ não pode definir uma facet de $\mathcal{P}$! (Por quê ?)

Definição:

Se C é uma cobertura de S, a **extensão** de C é dada pelo conjunto $E(C) = C \cup \{j \in N \setminus C : a_j \geq a_k, \text{ para todo } k \in C\}$.

Estudo facial do poliedro $\mathcal{P}$

Exemplo: extensão de uma cobertura

Seja $S = \{x \in \mathbb{B}^5 : 79x_1 + 53x_2 + 53x_3 + 45x_4 + 45x_5 \leq 178\}$.
Abaixo são mostradas duas coberturas e suas respectivas extensões:

$$\begin{array}{rcl} C_1 = \{1,2,4,5\} & = & E(C_1) \\ C_2 = \{2,3,4,5\} & = & E(C_2) = \{1,2,3,4,5\} \end{array}$$

Proposição: (extended cover inequality)

Seja C uma cobertura para S. Então a desigualdade

$$\sum_{j \in E(C)} x_j \leq |C| - 1,$$

é válida para $\mathcal{P}$.

Quando esta desigualdade define uma facet para $\mathcal{P}$?

Estudo facial do poliedro $\mathcal{P}$: lifting

Definição:

Seja P um poliedro do $\mathbb{R}^n_+$ e $ax \leq b$ uma desigualdade válida para P. Um **lifting** desta desigualdade é uma operação que obtém uma nova desigualdade $a'x \leq b$ válida para P tal que $a \leq a'$ e, para algum $j \in N$, $a_j < a'_j$.

Observações:

Note que, se $a'x \leq b$ foi obtida de $ax \leq b$ por uma operação de *lifting* então $a'x \leq b$ <u>domina</u> $ax \leq b$.
Nos chamados *liftings* sequenciais, a e a' diferem em apenas uma componente. Porém várias operações de *lifting* consecutivas são executadas. Nos *liftings* simultâneos, uma única operação altera o valor de várias componentes de a ao mesmo tempo.
Também encontramos na literatura o termo *lifting* 0-1 para os casos onde $a'_j \neq a_j$ implica $a_j = 0$ e $a'_j = 1$.

Estudo facial do poliedro $\mathcal{P}$

Exemplo: *liftings*

Seja $S = \{x \in \mathbb{B}^7 : 11x_1 + 8x_2 + 7x_3 + 6x_4 + 6x_5 + 6x_6 + x_7 \leq 19\}$ e considere a desigualdade de cobertura dada por $x_3 + x_4 + x_5 + x_6 \leq 3\,(\star)$.
Sabemos que esta desigualdade não pode representar uma facet de $\mathcal{P}$ pois ela é dominada pela desigualdade da cobertura extendida de C:
$x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$.
Esta última desigualdade define uma facet ?
Seja $F = \{x \in \mathcal{P} : x_3 + x_4 + x_5 + x_6 = 3\}$. Outro modo de mostrar que F *não é uma facet de* $\mathcal{P}$ é notando que para qualquer $x \in F$ satisfaz $x_2 = 0$
(por quê isso equivale a dizer que F não é facet ?)
No *lifting* da variável x_2 procura-se pelo maior de α_2 de modo $\alpha_2 x_2 + x_3 + x_4 + x_5 + x_6 \leq 3\,(\dagger)$ seja válida para $\mathcal{P}$.
A desigualdade $(\dagger)$ é satisfeita para todo $x \in \mathcal{P}$ com $x_2 = 0$.
Mas, o que ocorre quando $x_2 = 1$?

Estudo facial do poliedro $\mathcal{P}$

Exemplo: *liftings* (cont.)

$$\begin{array}{ll}
\alpha_2 + x_3 + x_4 + x_5 + x_6 \leq 3 & \forall x \in S | x_2 = 1 \implies \\
\alpha_2 \leq 3 - (x_3 + x_4 + x_5 + x_6) & \forall x \in S | x_2 = 1 \implies \\
\alpha_2 \leq 3 - \max\{x_3 + x_4 + x_5 + x_6 : x \in S | x_2 = 1\} \implies & \\
\alpha_2 \leq 3 - 1 \implies \alpha_2 \leq 2. &
\end{array}$$

Temos então uma nova desigualdade válida para $\mathcal{P}$ dada por $2x_2 + x_3 + x_4 + x_5 + x_6 \leq 3\,(\ddagger)$ que define a face $F' = \{x \in \mathcal{P} : 2x_2 + x_3 + x_4 + x_5 + x_6 = 3\}$.
Note que $F' \supset F$ e $\dim(F') = \dim(F)$ (por quê ?).
Será que agora F' é uma facet de $\mathcal{P}$?
A resposta é não pois, para todo $x \in F'$, $x_1 = 0$.
Podemos então fazer um *lifting* de x_1.

Estudo facial do poliedro $\mathcal{P}$

Exemplo: *liftings* (cont.)

Procura-se assim pelo maior de α_1 de modo $\alpha_1 x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 \leq 3\ (\diamond)$ seja válida para $\mathcal{P}$. A desigualdade $(\diamond)$ é satisfeita para todo $x \in \mathcal{P}$ com $x_1 = 0$.
Vejamos o que ocorre quando $x_1 = 1$?

$$\begin{aligned} &\alpha_1 \leq 3 - (2x_2 + x_3 + x_4 + x_5 + x_6), \forall x \in S|x_1 = 1 && \Longrightarrow \\ &\alpha_1 \leq 3 - \max\{2x_2 + x_3 + x_4 + x_5 + x_6 : x \in S|x_1 = 1\} && \Longrightarrow \\ &\alpha_1 \leq 3 - 2 \Longrightarrow \alpha_1 \leq 1. && \end{aligned}$$

Temos então uma nova desigualdade válida para $\mathcal{P}$ dada por $x_1 + 2x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ que define a **facet** F'' !

Estudo facial do poliedro $\mathcal{P}$

Observações:

1. Se tivéssemos feito o *lifting* de x_1 antes do *lifting* de x_2, chegaríamos a $2x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 3$ que *também* define facet em $\mathcal{P}$.
2. Note que a desigualdade da cobertura estendida para $C = \{3, 4, 5, 6\}$ é *dominada* por ambas as desigualdades obtidas por *lifting*.

Algoritmo de *lifting* para desigualdades de cobertura:

1. Seja $\{j_1, j_2, \ldots, j_r\}$ o conjunto ordenado dos índices em $N \setminus C$.
2. Para $t = 1$ até r faça
  2.1. seja a desigualdade corrente: $\sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \leq |C| - 1$.
  2.2. calcular:
$$\begin{aligned} \max \quad & \chi_t = \sum_{i=1}^{t-1} \alpha_{j_i} x_{j_i} + \sum_{j \in C} x_j \\ \text{s.a.} \quad & \sum_{i=1}^{t-1} a_{j_i} x_{j_i} + \sum_{j \in C} a_j x_j \leq b - a_{j_t}, x \in \mathbb{B}^n \end{aligned}$$
  2.3. fazer $\alpha_{j_t} = |C| - 1 - \chi_t$.

Observações:

1. As desigualdades obtidas pelo procedimento anterior *sempre definem facets* de $\mathcal{P}$ se C é uma cobertura *minimal* e $a_j \leq b$ para todo $j \in N$.
2. Para a_j inteiro para todo $j \in N$, os problemas da mochila a resolver no procedimento são **polinomiais** $(O(n^2|C|) \subseteq O(n^3))$.
3. Seja $k \in N \setminus C$. O *maior* coeficiente α_k^* de x_k em uma desigualdade obtida de $x(C) \leq |C| - 1$ através do procedimento de *lifting* anterior ocorre quando $k = j_1$, ou seja, quando k for a primeira variável a sofrer um *lifting*.

Pergunta:

Em que condições as desigualdades de cobertura estendidas podem definir facets de $\mathcal{P}$? (caso em que o procedimento de *lifting* é desnecessário)

Estudo facial do poliedro $\mathcal{P}$

Proposição:

Seja $C = \{j_1, j_2, \ldots, j_r\}$ uma cobertura minimal com $j_1 < j_2 < \ldots < j_r$. Se *(a)* ou *(b)* ou *(c)* ou *(d)* se verificarem, então a desigualdade $x(E(C)) \leq |C| - 1$ define uma facet de $\mathcal{P}$.

(a) $C = N$.

(b) $E(C) = N$ e **[i]** $(C \setminus \{j_1, j_2\}) \cup \{1\}$ é independente.

(c) $C = E(C)$ e **[ii]** $(C \setminus \{j_1\}) \cup \{p\}$ é independente para $p = \min\{j \in N \setminus E(C)\}$.

(d) $C \subset E(C) \subset N$ e **[i]** e **[ii]** se verificam.

Provas:

usar método indireto para *(b)*, *(c)* e *(d)* ...

Estudo facial do poliedro $\mathcal{P}$

O problema de separação para a desigualdade de cobertura:

Dada uma solução fracionária x^* de uma relaxação linear do problema da mochila, encontrar $C \subseteq N$ que seja uma cobertura minimal e tal que $x^*(C) = \sum_{j \in C} x_j^* > |C| - 1$.

Reescrevendo a desigualdade de cobertura $x(C) = \sum_{j \in C} x_j \leq |C| - 1$ temos que $|C| - \sum_{j \in C} x_j \geq 1$, ou ainda, $\sum_{j \in C}(1 - x_j) \geq 1$.
Logo o problema da separação reduz-se a *procurar uma cobertura C tal que $\sum_{j \in C}(1 - x_j^*) < 1$*. Isso pode ser feito calculando-se a cobertura que minimiza o valor de $\sum_{j \in C}(1 - x_j^*)$. Ou seja, resolvendo o PLI $(\circ)$ dado por:

$$\begin{array}{ll} \text{min} & \nu = \sum_{j \in N}(1 - x_j^*) z_j \\ \text{s.a} & \sum_{j \in N} a_j z_j \geq b + 1, \ z_j \in \{0, 1\}. \end{array}$$

Nota: o PLI $(\circ)$ é outro problema da mochila !

Estudo facial do poliedro $\mathcal{P}$

Proposição:

1. Se $\nu \geq 1$, x^* satisfaz todas desigualdades de cobertura.
2. Se $\nu < 1$ e $z^C \in \mathbb{B}^n$ é uma solução ótima do PLI $(\circ)$ então C é uma cobertura e $x(C) = \sum_{j \in C} x_j \leq |C| - 1$ está violada por x^*.

Algoritmos *Branch-and-Cut*:

INICIALIZAÇÃO:
$z = \max\{cx : x \in X\}$ com formulação P.
$\underline{z} = -\infty$, *incumbent* x^* vazio.
Pré-processe problema inicial e coloque-o em $\mathcal{L}$, a lista de nodos ativos.

NODO:
Se $\mathcal{L}$ está vazio, vá para SAIDA.
Se não escolha e remova nodo i de $\mathcal{L}$ e vá para RESTAURE.

RESTAURE:
Restaure (em memória) a formulação P^i do conjunto X^i.
Faça $k \leftarrow 1$ e $P^{i,1} \leftarrow P^i$.

RELAXAÇÃO:
Na iteração k, resolva $\overline{z}^{i,k} = \max\{cx; x \in P^{i,k}\}$
Se for `inviável`, pode o nodo corrente e vá para NODO.
Se não, faça $\overline{x}^{i,k}$ ser a solução ótima e vá para CORTE

CORTE:
Na iteração k, procure um corte que elimina a solução $\overline{x}^{i,k}$.
Se nenhum corte for encontrado, vá para PODA.
Se não adicione o(s) corte(s) a $P^{i,k}$ obtendo $P^{i,k+1}$.
Faça $k \leftarrow k+1$ e vá para RELAXAÇÃO.

PODA:
Se $\overline{x}^{i,k} \leq \underline{z}$, vá para NODO
Se $x \in X$, faça $\underline{z} = \overline{x}^{i,k}$, atualiza o *incumbent* para $x^* \leftarrow x^{i,k}$ e vá para NODO.
Se não vá para RAMIFICAÇÃO.

RAMIFICA:
Crie dois ou mais novos problemas X^i_t com formulações P^i_t.
Adicione-os à lista $\mathcal{L}$.

SAIDA:
Retorne o *incumbent* x^* como solução ótima e $\underline{z}$ como sendo o valor ótimo.

Resolução de IPs 0–1 usado desigualdades da mochila

- Até o início dos anos 80, mesmo a resolução de instâncias *"pequenas"* para os padrões de hoje era algo bastante difícil.
- Surgiu então um método alternativo para resolver IPs binários (puros) não estruturados baseado nas desigualdades conhecidas para o politopo associado ao problema da mochila.
- A idéia era tratar cada restrição do problema original como uma mochila independente. Assim, usando um algoritmo *branch-and-cut*, a cada solução de uma relaxação linear, verifica-se cada uma destas mochilas em busca de desigualdades válidas violadas (p.ex., *cover inequalities* e os seus *liftings*).
- Esta proposta foi feita por Crowder, Johnson e Padberg (*Solving large–scale zero–one linear programming problems, Operations Research, 31:803–834, 1983*), vencedor do *Lanchester Prize*.
- Eles resolveram um IP binário com 2756 variáveis, que não podia ser computado por nenhum resolvedor disponível na época.

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 1
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
Bounds
0 <= x1 <= 1     x1 0.709921
0 <= x2 <= 1     x3 0.341858
0 <= x3 <= 1     x4 1.000000
0 <= x4 <= 1     x5 1.000000
0 <= x5 <= 1     x7 1.000000
0 <= x6 <= 1     x8 1.000000
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (iteração 1)
Minimize
obj: 0.290079 x1 + x2 + 0.658142 x3 + x6 + x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1     x1 1.000000
0 <= x2 <= 1     x7 1.000000
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (iteração 1)
Minimize
obj: 0.290079 x1 + x2 + 0.658142 x3 + x6 + x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1     x3 1.000000
0 <= x2 <= 1     x7 1.000000
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 2
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
Bounds
0 <= x1 <= 1     x2 1.000000
0 <= x2 <= 1     x4 1.000000
0 <= x3 <= 1     x5 1.000000
0 <= x4 <= 1     x7 1.000000
0 <= x5 <= 1     x8 1.000000
0 <= x6 <= 1     x9 0.612739
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (iteração 2)
Minimize
obj: x1 + x3 + x6 + 0.387261 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1      x7 = 1.000000
0 <= x2 <= 1      x9 = 1.000000
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (iteração 2)
Minimize
obj: x1 + x3 + x6 + 0.387261 x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1      Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 3
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
Bounds
0 <= x1 <= 1     x1 0.631241
0 <= x2 <= 1     x2 1.000000
0 <= x3 <= 1     x3 0.602287
0 <= x4 <= 1     x4 1.000000
0 <= x5 <= 1     x5 1.000000
0 <= x6 <= 1     x7 0.368759
0 <= x7 <= 1     x8 1.000000
0 <= x8 <= 1     x9 0.631241
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (iteração 3)
Minimize
obj: 0.368759 x1 + 0.397713 x3 + x6 + 0.631241 x7 + 0.368759 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1     x1 1.000000
0 <= x2 <= 1     x2 1.000000
0 <= x3 <= 1     x4 1.000000
0 <= x4 <= 1     x5 1.000000
0 <= x5 <= 1     x7 1.000000
0 <= x6 <= 1     x8 1.000000
0 <= x7 <= 1     x9 1.000000
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (iteração 3)
Minimize
obj: 0.368759 x1 + 0.397713 x3 + x6 + 0.631241 x7 + 0.368759 x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1     x2 1.000000
0 <= x2 <= 1     x3 1.000000
0 <= x3 <= 1     x4 1.000000
0 <= x4 <= 1     x5 1.000000
0 <= x5 <= 1     x7 1.000000
0 <= x6 <= 1     x8 1.000000
0 <= x7 <= 1     x9 1.000000
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 4
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
Bounds
0 <= x1 <= 1     x2 1.000000
0 <= x2 <= 1     x4 1.000000
0 <= x3 <= 1     x5 1.000000
0 <= x4 <= 1     x6 0.632895
0 <= x5 <= 1     x7 1.000000
0 <= x6 <= 1     x8 1.000000
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (iteração 4)
Minimize
obj: x1 + x3 + 0.367105 x6 + x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1      x2 1.000000
0 <= x2 <= 1      x4 1.000000
0 <= x3 <= 1      x5 1.000000
0 <= x4 <= 1      x6 1.000000
0 <= x5 <= 1      x7 1.000000
0 <= x6 <= 1      x8 1.000000
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (iteração 4)
Minimize
obj: x1 + x3 + 0.367105 x6 + x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1      Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 5
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
covrow1_4: x2 + x4 + x5 + x6 + x7 + x8 <= 5
Bounds
0 <= x1 <= 1     x1 0.217943
0 <= x2 <= 1     x2 0.782057
0 <= x3 <= 1     x3 0.217943
0 <= x4 <= 1     x4 1.000000
0 <= x5 <= 1     x5 1.000000
0 <= x6 <= 1     x6 0.435886
0 <= x7 <= 1     x7 0.782057
0 <= x8 <= 1     x8 1.000000
0 <= x9 <= 1     x9 0.217943
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (iteração 5)
Minimize
obj: 0.782057 x1 + 0.217943 x2 + 0.782057 x3 + 0.435886 x6 + 0.217943 x7 + 0.782057 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1     x4 1.000000
0 <= x2 <= 1     x5 1.000000
0 <= x3 <= 1     x6 1.000000
0 <= x4 <= 1     x7 1.000000
0 <= x5 <= 1     x8 1.000000
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (iteração 5)
Minimize
obj: 0.782057 x1 + 0.217943 x2 + 0.782057 x3 + 0.435886 x6 + 0.217943 x7 + 0.782057 x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1     Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 6
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
covrow1_4: x2 + x4 + x5 + x6 + x7 + x8 <= 5
covrow1_5: x4 + x5 + x6 + x7 + x8 <= 4
Bounds
0 <= x1 <= 1     x1 0.332412
0 <= x2 <= 1     x2 0.667588
0 <= x3 <= 1     x3 0.332412
0 <= x4 <= 1     x4 1.000000
0 <= x5 <= 1     x5 1.000000
0 <= x6 <= 1     x6 0.332412
0 <= x7 <= 1     x7 0.667588
0 <= x8 <= 1     x8 1.000000
0 <= x9 <= 1     x9 0.332412
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (iteração 6)
Minimize
obj: 0.667588 x1 + 0.332412 x2 + 0.667588 x3 + 0.667588 x6 + 0.332412 x7 + 0.667588 x8
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1      Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (iteração 6)
Minimize
obj: 0.667588 x1 + 0.332412 x2 + 0.667588 x3 + 0.667588 x6 + 0.332412 x7 + 0.667588 x8
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1      Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 1, branch x7=0
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
covrow1_4: x2 + x4 + x5 + x6 + x7 + x8 <= 5
covrow1_5: x4 + x5 + x6 + x7 + x8 <= 4
Bounds
0 <= x1 <= 1      x1 1.000000
0 <= x2 <= 1      x2 0.288109
0 <= x3 <= 1      x3 0.921346
0 <= x4 <= 1      x4 1.000000
0 <= x5 <= 1      x5 1.000000
0 <= x6 <= 1      x8 1.000000
x7 = 0            x9 0.711891
0 <= x8 <= 1
0 <= x9 <= 1      Limitante dual = -1.7495154972e+02
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 1, branch x7=1
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
covrow1_4: x2 + x4 + x5 + x6 + x7 + x8 <= 5
covrow1_5: x4 + x5 + x6 + x7 + x8 <= 4
Bounds
0 <= x1 <= 1      x2 1.000000
0 <= x2 <= 1      x4 0.330084
0 <= x3 <= 1      x5 1.000000
0 <= x4 <= 1      x6 0.669916
0 <= x5 <= 1      x7 1.000000
0 <= x6 <= 1      x8 1.000000
x7 = 1
0 <= x8 <= 1      Limitante dual = -1.7600000000e+02
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (branching x7=1, iteracao 1)
Minimize
obj: x1 + x3 + 0.669916 x4 + 0.330084 x6 + x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1     x2 1.000000
0 <= x2 <= 1     x5 1.000000
0 <= x3 <= 1     x6 1.000000
0 <= x4 <= 1     x7 1.000000
0 <= x5 <= 1     x8 1.000000
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (branching x7=1, iteracao 1)
Minimize
obj: x1 + x3 + 0.669916 x4 + 0.330084 x6 + x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1     Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 2, branch x7=1
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
covrow1_4: x2 + x4 + x5 + x6 + x7 + x8 <= 5
covrow1_5: x4 + x5 + x6 + x7 + x8 <= 4
covrow_1_6 : x2 + x5 + x6 + x7 + x8 <= 4
Bounds
0 <= x1 <= 1     x2 0.250779
0 <= x2 <= 1     x4 0.250779
0 <= x3 <= 1     x5 1.000000
0 <= x4 <= 1     x6 0.749221
0 <= x5 <= 1     x7 1.000000
0 <= x6 <= 1     x8 1.000000
x7 = 1
0 <= x8 <= 1
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row1 (branching x7=1, iteracao 2)
Minimize
obj: x1 + 0.749221 x2 + x3 + 0.749221 x4 + 0.250779 x6 + x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3+ 42 x4+ 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 >= 1501
Bounds
0 <= x1 <= 1     x5 = 1.000000
0 <= x2 <= 1     x6 = 1.000000
0 <= x3 <= 1     x7 = 1.000000
0 <= x4 <= 1     x8 = 1.000000
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```

Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: cover inequality para a restricao row2 (branching x7=1, iteracao 2)
Minimize
obj: x1 + 0.749221 x2 + x3 + 0.749221 x4 + 0.250779 x6 + x9
Subject To
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 >= 1501
Bounds
0 <= x1 <= 1      Não encontrou desigualdade violada !
0 <= x2 <= 1
0 <= x3 <= 1
0 <= x4 <= 1
0 <= x5 <= 1
0 <= x6 <= 1
0 <= x7 <= 1
0 <= x8 <= 1
0 <= x9 <= 1
Integers
x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```


Exemplo de resolução de um IP 0–1 por *branch-and-cut*

```
# Problem name: iteration 3, branch x7=1
Minimize
obj: - 77 x1 - 6 x2 - 3 x3 - 6x4 - 33 x5 - 13 x6 -110 x7 -21 x8 - 47 x9
Subject To
row1 : 774 x1 + 76 x2 + 22 x3 + 42 x4 + 21 x5 + 760 x6 + 818 x7 + 62 x8 + 785 x9 <= 1500
row2 : 67 x1 + 27 x2 + 794 x3 + 53 x4 + 234 x5 + 32 x6 + 797 x7 + 97 x8 + 435 x9 <= 1500
covrow1_1 : x1 + x7 <= 1
covrow2_1 : x3 + x7 <= 1
covrow1_2 : x7 + x9 <= 1
covrow1_3: x1 + x2 + x4 + x5 + x7 + x8 + x9 <= 5
covrow2_3: x2 + x3 + x4 + x5 + x7 + x8 + x9 <= 5
covrow1_4: x2 + x4 + x5 + x6 + x7 + x8 <= 5
covrow1_5: x4 + x5 + x6 + x7 + x8 <= 4
covrow_1_6 : x2 + x5 + x6 + x7 + x8 <= 4
covrow_1_7 : x5 + x6 + x7 + x8 <= 3
Bounds
0 <= x1 <= 1      x2 1.000000
0 <= x2 <= 1      x4 1.000000
0 <= x3 <= 1      x5 1.000000
0 <= x4 <= 1      x7 1.000000
0 <= x5 <= 1      x8 1.000000
0 <= x6 <= 1
x7 = 1
0 <= x8 <= 1
0 <= x9 <= 1
# Integers
# x1 x2 x3 x4 x5 x6 x7 x8 x9
End
```