

Algoritmos de Planos de Corte e Branch-and-cut

Prof. Alexandre Salles da Cunha

Universidade Federal de Minas Gerais
Departamento de Ciência da Computação
Belo Horizonte, Brasil

`acunha@dcc.ufmg.br`

Outubro 2020



UNIVERSIDADE FEDERAL
DE MINAS GERAIS



Relativas à esta apresentação

- ① L. Wolsey. Integer Programming, Wiley, 1998. [Capítulo 8]
- ② D. Bertsimas e J. N. Tsitsiklis. Introduction to Linear Optimization, Athena Scientific, 1997. [Seção 11.1 e 11.2]

Definição

Dado um conjunto de pontos $X \subseteq \mathbb{R}^n$, uma desigualdade $\pi^T x \leq \pi_0$ é válida para X se $\pi^T x \leq \pi_0$ para todo $x \in X$.

Observação

Se $X = \{x \in \mathbb{Z}^n : Ax \leq b\}$ e $\text{conv}(X) = \{x \in \mathbb{R}^n : \bar{A}x \leq \bar{b}\}$ é a envoltória convexa dos pontos em X , então $a_i^T x \leq b_i$ e $\bar{a}_i^T x \leq \bar{b}_i$ são desigualdades válidas para X .

- ❶ Quais desigualdades são boas ou úteis ?
- ❷ Dado que conhecemos um conjunto ou família de desigualdades válidas para um problema de Otimização Combinatória ou Inteira, como podemos usá-las na tentativa de resolver uma instância particular do problema ?

Desigualdades para conjuntos 0 – 1 puros:

- Considere o conjunto da mochila

$$X = \{x \in \mathbb{B}^5 : 3x_1 - 4x_2 + 2x_3 - 3x_4 + x_5 \leq -2\}$$

- O que ocorre se $x_2 = x_4 = 0$?
- Estas duas variáveis não podem ser simultaneamente nulas. Logo, a desigualdade $x_2 + x_4 \geq 1$ é válida.

Desigualdades para conjuntos 0 – 1 mistos:

- Considere o conjunto

$$X = \{(x, y) : x \leq 9999y, \ 0 \leq x \leq 5, \ y \in \mathbb{B}\}$$

- É fácil verificar que $x \leq 5y$ é válido para o problema, uma vez que $X = \{(0, 0), (x, 1) \text{ com } 0 \leq x \leq 5\}$.
- Em particular, a adição de $x \leq 5y$ nos fornece a envoltória convexa de X .

M denota o conjunto de clientes e N denota o conjunto de localidades:

$$\sum_{i \in M} x_{ij} \leq b_j y_j, \quad \forall j \in N$$

$$\sum_{j \in N} x_{ij} = a_i, \quad \forall i \in M$$

$$x_{ij} \geq 0, \quad \forall i \in M, j \in N, y_j \in \{0, 1\} \quad \forall j \in N$$

Observe que:

- $x_{ij} \leq \sum_{i \in M} x_{ij} \leq b_j y_j$
- $x_{ij} \leq \sum_{j \in N} x_{ij} = a_i \rightarrow x_{ij} \leq a_i$.
- Assim sendo, como $y \in \{0, 1\}$, temos que $x_{ij} \leq \min\{a_i, b_j\} y_j$.

Considere os vetores de incidência associados aos emparelhamentos de um grafo $G = (V, E)$

$$\sum_{e \in \delta(i)} x_e \leq 1, \forall i \in V$$

$$x \in \mathbb{Z}_+^{|E|}$$

onde $\delta(i) = \{e \in E : e = \{i, j\} \text{ para algum } j \in V\}$.

Considere um conjunto de vértices $T \subset V$ (idealmente de cardinalidade ímpar). Não é difícil perceber que a família de desigualdades

$$\sum_{e \in E(T)} x_e \leq \left\lfloor \frac{|T|}{2} \right\rfloor$$

onde $E(T) = \{e = \{i, j\} \in E : i, j \in T\}$, é válida para o Problema do Emparelhamento Máximo.

Considere a região $X = P \cap \mathbb{Z}^4$ onde

$$P = \{x \in \mathbb{R}_+^4 : 13x_1 + 20x_2 + 11x_3 + 6x_4 \geq 72\}.$$

Dividindo por 11 temos

$$\frac{13}{11}x_1 + \frac{20}{11}x_2 + \frac{11}{11}x_3 + \frac{6}{11}x_4 \geq \frac{72}{11}.$$

Observe que

$$\lceil \frac{13}{11} \rceil x_1 + \lceil \frac{20}{11} \rceil x_2 + \lceil \frac{11}{11} \rceil x_3 + \lceil \frac{6}{11} \rceil x_4 \geq \frac{13}{11}x_1 + \frac{20}{11}x_2 + \frac{11}{11}x_3 + \frac{6}{11}x_4 \geq \frac{72}{11}.$$

Como x é um vetor inteiro temos que:

$$2x_1 + 2x_2 + x_3 + x_4 \geq \lceil \frac{72}{11} \rceil = 7.$$

- Deseja-se associar caminhões de capacidade C_i para atender a demanda d_j de vários clientes. Existem n clientes, m tipos de caminhões. Para cada tipo i , existem a_i caminhões disponíveis. O custo de enviar o caminhão i ao cliente j é c_{ij} .
- Associamos uma variável de decisão inteira x_{ij} que denota o número de caminhões do tipo i destinados ao cliente j . Então:

$$\min \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

$$\sum_{i=1}^m C_i x_{ij} \geq d_j, \quad \forall j = 1, \dots, n \quad (2)$$

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad \forall i = 1, \dots, m \quad (3)$$

$$x_{ij} \in \mathbb{Z}_+^{nm} \quad (4)$$

Motivação: para gerar desigualdades válidas para Programas Inteiros, é conveniente compreender inicialmente a geração de desigualdades válidas para poliedros (ou para programas lineares).

Proposição

A desigualdade $\pi^T x \leq \pi_0$ (ou simplesmente (π, π_0)) é válida para o poliedro $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$ ($A \in \mathbb{Q}^{m \times n}$) se e somente se:

- 1 Existe $u \in \mathbb{R}_+^m$ e $v \in \mathbb{R}_+^n$ tal que $A^T u - v = \pi$ e $u^T b \leq \pi_0$ ou, alternativamente
- 2 Existe $u \in \mathbb{R}_+^m$ tal que $A^T u \geq \pi$ e $u^T b \leq \pi_0$.

Prova

Considere as condições de otimalidade da solução do Programa Linear $\max\{\pi^T x : Ax \leq b\}$ e do seu dual $\min\{u^T b : A^T u \geq \pi\}$. □

$$X = \{x \in \mathbb{Z}_+^n : Ax \leq b\}, \quad A \in \mathbb{Q}^{m \times n}$$

Vamos usar ideias utilizadas na demonstração do resultado anterior e na seguinte observação (trivial):

Proposição

Se $X = \{y \in \mathbb{Z}_+ : y \leq b\}$ então $y \leq \lfloor b \rfloor$ é válida para X .

- 1 Buscar combinações lineares com pesos não negativos das linhas de $Ax \leq b$.
- 2 Aplicar arredondamento inteiro ao resultado desta combinação linear.

Dedução algébrica das desigualdades Blossom:

- Passo 1: multiplique $\sum_{e \in \delta(i)} x_e \leq 1$ por $u_i = \frac{1}{2}$ para todo $i \in T$ e $u_i = 0$ para $i \in V \setminus T$, resultando

$$\sum_{e \in E(T)} x_e + \frac{1}{2} \sum_{e \in \delta(T)} x_e \leq \frac{|T|}{2}$$

- Passo 2: Como $x_e \geq 0$ temos:

$$\sum_{e \in E(T)} x_e \leq \sum_{e \in E(T)} x_e + \frac{1}{2} \sum_{e \in \delta(T)} x_e \leq \frac{|T|}{2} \text{ e então}$$

$$\sum_{e \in E(T)} x_e \leq \frac{|T|}{2}.$$

- Passo 3: Uma vez que $x \in \mathbb{Z}_+^{|E|}$, $\sum_{e \in E(T)} x_e$ deve ser inteiro e assim:

$$\sum_{e \in E(T)} x_e \leq \left\lfloor \frac{|T|}{2} \right\rfloor.$$

Exemplo - $X = P \cap \mathbb{Z}_+^2$ onde P é:

$$\begin{aligned} 7x_1 - 2x_2 &\leq 14 \\ x_2 &\leq 3 \\ 2x_1 - 2x_2 &\leq 3 \end{aligned} \tag{5}$$
$$x_1, x_2 \geq 0.$$

- Passo 1: Multiplicando a matriz de restrições por $u = (\frac{2}{7}, \frac{37}{63}, 0)$ obtemos $2x_1 + \frac{1}{63}x_2 \leq \frac{121}{21}$
- Passo 2: Reduzindo os coeficientes do lado esquerdo ao inteiro menor mais próximo (desigualdade $\leq$) temos a seguinte desigualdade válida para P temos $2x_1 + 0x_2 \leq \frac{121}{21}$
- Passo 3: Como o lado esquerdo deve ser inteiro temos $2x_1 \leq \lfloor \frac{121}{21} = 5 \rfloor$ e, pelos mesmos argumentos:

$$x_1 \leq \lfloor \frac{5}{2} \rfloor = 2.$$

Objetivo: gerar uma desigualdade válida para o conjunto $X = P \cap \mathbb{Z}_+^n$ onde $P = \{x \in \mathbb{R}_+^n : Ax \leq b\}$.

Assumimos que a matriz $A \in \mathbb{Q}^{m \times n}$ possui colunas $(A_1, A_2, \dots, A_n)$ e $u \in \mathbb{R}_+^m$. Passos principais:

- Passo 1: $\sum_{j=1}^n u^T A_j x_j \leq u^T b$ é válida para P .
- Passo 2: Já que $x \geq 0$, $\sum_{j=1}^n \lfloor u^T A_j \rfloor x_j \leq u^T b$ é válida para P ;
- Passo 3: Como x é inteiro, $\sum_{j=1}^n \lfloor u^T A_j \rfloor x_j$ é inteiro e $\sum_{j=1}^n \lfloor u^T A_j \rfloor x_j \leq \lfloor u^T b \rfloor$ é válida para P .

- Quais são os vetores u de interesse e como encontrá-los ?
- Qualquer desigualdade válida para X pode ser gerada através do procedimento de Chvátal-Gomory.

Teorema

Qualquer desigualdade válida para X pode ser obtida através da aplicação do procedimento de Chvátal-Gomory por um número finito de vezes.

- ① Adição inicial de desigualdades válidas.
- ② Reformulação automática ou algoritmo de planos de corte.

Dada uma formulação inicial $P = \{x : Ax \leq b, x \geq 0\}$ e o conjunto de soluções viáveis $X = P \cap \mathbb{Z}^n$:

- Caracterizamos um conjunto de desigualdades $Qx \leq q$, válidas para X .
- Acrescentamos estas desigualdades no modelo, obtendo uma nova formulação $P' = \{x : Ax \leq b, Qx \leq q, x \geq 0\}$ e então reescrevemos $X = P' \cap \mathbb{Z}^n$.
- Aplica-se um algoritmo desejado (por exemplo, Branch-and-bound) ao problema reformulado por meio de P' .

- Vantagens:
 - Pode-se usar um software de Branch-and-bound pré-implementado.
 - Se as desigualdades pré-adicionadas forem bem escolhidas (isto é, se P' for bem menor que P), os limites duais serão mais fortes e a árvore de enumeração será menor.
 - É mais provável que se encontre soluções inteiras de boa qualidade mais cedo, ao longo do curso do algoritmo BB.
- Desvantagens:
 - Muitas vezes o número de desigualdades candidatas a serem pré-adicionadas é enorme. Nestes casos, os Programas Lineares associados se tornam muito pesados, anulando o ganho trazido pela melhor aproximação de $\text{conv}(X)$.
- Como escolher um bom conjunto de restrições inicial ?

Se o conjunto de soluções viáveis do problema consiste em na interseção de dois conjuntos com estrutura $X = X_1 \cap X_2$, podemos:

- Empregar decomposição:
 - Se a otimização sobre o conjunto $X_2 = P_2 \cap \mathbb{Z}^n$ é fácil, talvez possamos obter a descrição da envoltória convexa de $P'_2 = \text{conv}(P_2 \cap \mathbb{Z}^n)$ e substituir a formulação inicial $P_1 \cap P_2$ por uma formulação fortalecida $P_1 \cap P'_2$.
 - Ainda que a otimização sobre X_2 não seja *fácil*, podemos tentar encontrar desigualdades válidas para X_2 e trabalhar com uma formulação fortalecida.
- Concentrar em cada um dos conjuntos com estrutura de uma vez.

Formulação fraca das décadas de 1950-60:

$$\sum_{j=1}^n x_{ij} = 1, \forall i = 1, \dots, m$$

$$\sum_{i=1}^m x_{ij} \leq m y_j, \forall j = 1, \dots, n$$

$$x_{ij} \geq 0, \forall i = 1, \dots, m, \forall j = 1, \dots, n$$

$$0 \leq y_j \leq 1, \forall j = 1, \dots, n.$$

Seja X_j o conjunto de pontos no poliedro P_j dado por:

$$\sum_{i=1}^m x_{ij} \leq my_j$$

$$x_{ij} \geq 0, \forall i = 1, \dots, m$$

$$0 \leq y_j \leq 1$$

com y_j inteiro. Observe que o conjunto P'_j dado por:

$$x_{ij} \leq y_j, \forall i = 1, \dots, m$$

$$x_{ij} \geq 0, \forall i = 1, \dots, m$$

$$0 \leq y_j \leq 1$$

é inteiro (a matriz associada é TUM). Logo P'_j define $\text{conv}(X_j)$.

Substituindo os poliedros P_j por P'_j obtemos a formulação fortalecida:

$$\sum_{j=1}^n x_{ij} = 1, \forall i = 1, \dots, m$$

$$x_{ij} \leq y_j, \forall i = 1, \dots, m, j = 1, \dots, n$$

$$x_{ij} \geq 0, \forall i = 1, \dots, m, j = 1, \dots, n$$

$$0 \leq y_j \leq 1, j = 1, \dots, n.$$

Usando a notação empregada anteriormente, uma formulação básica para o problema é dada pela região:

$$s_{t-1} + x_t = d_t + s_t, \quad \forall t = 1, \dots, n,$$

$$x_t \leq Cy_t, \quad \forall t = 1, \dots, n,$$

$$s_0 = 0, s_t, x_t \geq 0, y_t \in \{0, 1\}, \quad \forall t = 1, \dots, n$$

$$s \in \mathbb{R}^n, x \in \mathbb{Z}^n, y \in \mathbb{B}^n.$$

- Uma vez que $s_{t-1} \geq 0$ e que $y_t \in \{0, 1\}$ temos que

$$x_t \leq d_t + s_t \leq d_t y_t + s_t$$

.

- Novamente, uma vez que $s_t \geq 0$ obtemos $\sum_{i=1}^t x_i \geq \sum_{i=1}^t d_i$.
Usando $x_t \leq C y_t$ temos que $C \sum_{i=1}^t y_i \geq \sum_{i=1}^t d_i$ ou
equivalentemente $\sum_{i=1}^t y_i \geq \frac{\sum_{i=1}^t d_i}{C}$. Como o lado esquerdo deve
ser inteiro temos:

$$\sum_{i=1}^t y_i \geq \lceil \frac{\sum_{i=1}^t d_i}{C} \rceil.$$

- Adicionando estes conjuntos de $2n$ desigualdades fortalecemos a formulação do problema.

- Supomos conhecer uma família $\mathcal{F}$ de desigualdades válidas $\pi^T x \leq \pi_0$ para $X = P \cap \mathbb{Z}^n$, isto é, $(\pi, \pi_0) \in \mathcal{F}$.
- Muitas vezes o número de desigualdades em $\mathcal{F}$ é enorme, impedindo sua adição automática na formulação.
- Além disto, dada uma função objetivo específica, o nosso **real objetivo é aproximar a envoltória convexa de X nas vizinhanças do ponto ótimo**.
- A ideia dos algoritmos de planos de corte é identificar desigualdades válidas para X dinamicamente, na medida em que são necessárias, por exemplo quando são violadas pela solução de uma relaxação linear da formulação do problema.

```
1:  $k \leftarrow 0$ 
2:  $P_0 \leftarrow P$ 
3: FLAG = VERDADEIRO
4: while FLAG do
5:   Resolva  $\bar{w}_k = \min \{ \sum_i c_i x_i : x \in P_k \}$  obtendo  $\bar{x}_k$ 
6:    $D_k \leftarrow \text{ResolveSeparacao}(\bar{x}_k, \mathcal{F})$ 
7:   if  $D_k = \emptyset$  then
8:     FLAG = FALSO
9:   else
10:     $P_{k+1} \leftarrow P_k \cap D_k$ 
11:     $k \leftarrow k + 1$ 
12:   end if
13: end while
```

A cada iteração

Dado $\bar{x}$, solução de

$$\bar{w}_k = \min \left\{ \sum_i c_i x_i : x \in P_k \right\},$$

e uma classe $\mathcal{F}$ de desigualdades válidas para X :

- encontrar $(\pi, \pi_0) \in \mathcal{F}$ tal que $\pi \bar{x} > \pi_0$, ou
- assegurar que não existe desigualdade desta classe violada por $\bar{x}$.

Ao fim do algoritmo

- Se $\bar{x} \in \mathbb{Z}^n$, $\bar{x}$ resolve o problema de Otimização Inteira.
- Caso contrário, uma formulação fortalecida P_k foi obtida. Esta formulação pode ser empregada em um algoritmo Branch-and-bound.

Considere o problema

$$\max\{c^T x : Ax \leq b, x \geq 0, x \text{ inteiro}\}.$$

A idéia do algoritmo consiste em:

- Resolver uma relaxação linear do problema, obtendo uma base ótima.
- Escolher uma variável básica que assume valor fracionário na relaxação.
- Gerar uma desigualdade de Chvátal-Gomory a partir da linha do dicionário à qual se associa esta variável básica.

$$\max \bar{a}_{00} + \sum_{j \in NB} \bar{a}_{0j} x_j$$

$$x_{B_u} + \sum_{j \in NB} \bar{a}_{uj} x_j = \bar{a}_{u0}, \quad \forall u = 1, \dots, m$$

$$x \geq 0, x \text{ inteiro}$$

satisfazendo as condições sobre custos reduzidos:

- $\bar{a}_{0j} \leq 0, \forall j \in NB$
- $\bar{a}_{u0} \geq 0, \forall u \in 1, \dots, m,$

onde NB denota o conjunto de variáveis não básicas.

- Então existe alguma linha u tal que $\bar{a}_{u0} \notin \mathbb{Z}$.
- Vamos aplicar um corte de Chvátal-Gomory para a linha u :

$$x_{B_u} + \sum_{j \in NB} \lfloor \bar{a}_{uj} \rfloor x_j \leq \lfloor \bar{a}_{u0} \rfloor$$

- Como $x_{B_u} = \bar{a}_{u0} - \sum_{j \in NB} \bar{a}_{uj} x_j$, reescrevemos a desigualdade em termos das variáveis não básicas:

$$\sum_{j \in NB} (\bar{a}_{uj} - \lfloor \bar{a}_{uj} \rfloor) x_j \geq \bar{a}_{u0} - \lfloor \bar{a}_{u0} \rfloor$$

- ou, definindo $f_{uj} = \bar{a}_{uj} - \lfloor \bar{a}_{uj} \rfloor$ para $j \in NB$ e $f_{u0} = \bar{a}_{u0} - \lfloor \bar{a}_{u0} \rfloor$

$$\sum_{j \in NB} f_{uj} x_{uj} \geq f_{u0}$$

- Pela definição de f_{uj} e pela escolha da linha u temos que $0 \leq f_{uj} < 1$ e $0 < f_{u0} < 1$;
- Observe que para a solução ótima x^* associada à base B temos que $x_j^* = 0, \forall j \in NB$. Assim sendo, a desigualdade

$\sum_{j \in NB} f_{uj} x_{uj} \geq f_{u0} \ (0 \not\geq f_{uj} > 0)$ é claramente violada e **corta** a solução corrente x^* .

- Quando x^* é inteira, a diferença entre o lado esquerdo e o lado direito também é inteiro (verique isto fazendo

$s + x_{B_u} + \sum_{j \in NB} [\bar{a}_{uj} x_j] = [\bar{a}_{u0}] = \bar{a}_{u0}$). Então:

$$s = -f_{u0} + \sum_{j \in NB} f_{uj} x_{uj},$$

onde s denota uma variável de folga inteira não negativa.

PI

$$\begin{aligned}
 z &= \max & 4x_1 &- x_2 \\
 & & 7x_1 &- 2x_2 \leq 14 \\
 & & & x_2 \leq 3 \\
 & & 2x_1 &- 2x_2 \leq 3 \\
 & & x_1, & x_2 \in \mathbb{Z}_+
 \end{aligned}$$

Adicionando variáveis de folga ($x_3, x_4, x_5 \in \mathbb{Z}_+$) e resolvendo a relaxação linear do PI temos:

Base ótima $B^* = \{1, 2, 5\}$

$$\begin{aligned}
 z &= \max & \frac{59}{7} & & -\frac{4}{7}x_3 & -\frac{1}{7}x_4 & & \\
 & & x_1 & & +\frac{1}{7}x_3 & +\frac{2}{7}x_4 & & = \frac{20}{7} \\
 & & & x_2 & & +x_4 & & = 3 \\
 & & & & -\frac{2}{7}x_3 & +\frac{10}{7}x_4 & +x_5 & = \frac{23}{7} \\
 & & x_1, & x_2, & x_3, & x_4, & x_5 & \in \mathbb{Z}_+
 \end{aligned}$$

Aplicando um corte de Chvátal-Gomory para a primeira linha do Tableau Simplex, obtemos a desigualdade válida violada:

$$\frac{1}{7}x_3 + \frac{2}{7}x_4 \geq \frac{6}{7} \text{ ou}$$

$$s = -\frac{6}{7} + \frac{1}{7}x_3 + \frac{2}{7}x_4,$$

onde $s, x_3, x_4 \in \mathbb{Z}_+$.

Inserindo este corte e reotimizando, temos $B^* = \{1, 2, 3, 4\}$

z	$=$	$\max$	$\frac{15}{2}$					$-\frac{1}{2}x_5$	$-3s$		
			x_1						$+s$	$=$	2
				x_2				$-\frac{1}{2}x_5$	$+s$	$=$	$\frac{1}{2}$
					x_3			$-x_5$	$-5s$	$=$	1
						x_4		$+\frac{1}{2}x_5$	$+6s$	$=$	$\frac{5}{2}$
			$x_1,$	$x_2,$	$x_3,$	$x_4,$	x_5	s	$\in$	$\mathbb{Z}_+$	

Aplicando um corte de Chvátal-Gomory para a segunda linha do Tableau Simplex anterior, obtemos

$$-\frac{1}{2}x_5 + t = -\frac{1}{2}$$

onde $t \in \mathbb{Z}_+$

Inserindo este corte e reotimizando, temos $B^* = \{1, 2, 3, 4, 5\}$

z	$=$	$\max$	7						$-3s$	$-t$		
			x_1						$+s$		$=$	2
				x_2					$+s$	$-t$	$=$	1
					x_3				$-5s$	$-2t$	$=$	1
						x_4			$+6s$	$+t$	$=$	2
							x_5			$-t$	$=$	1
		$x_1,$	$x_2,$	$x_3,$	$x_4,$	x_5	s	t	$\in$	$\mathbb{Z}_+$		

Proposição

Seja β a linha u de B^{-1} e $q_i = \beta_i - \lfloor \beta_i \rfloor$, $i = 1, \dots, m$. O corte de Gomory $\sum_{j \in NB} f_{uj} x_j \geq f_{u0}$ quando escrito em termos das variáveis originais do problema é o corte de Chátal-Gomory

$$\sum_{j=1}^n \lfloor q^T a_j \rfloor x_j \leq \lfloor q^T b \rfloor$$

No caso do primeiro corte introduzido no exemplo, β pode ser lido na primeira linha do dicionário, $\beta = (\frac{1}{7} \ \frac{2}{7} \ 0)$ corresponde aos coeficientes das variáveis de folga na linha $u = 1$.

Interpretação

Ou seja, o corte de Chvátal-Gomory nada mais é que a aplicação do procedimento de arredondamento inteiro à uma combinação linear não negativa de linhas da formulação do problema.

Primeiro corte de CG gerado a partir de:

$$x_1 + \frac{1}{7}x_3 + \frac{2}{7}x_4 + 0x_5 = \frac{20}{7}$$

Então:

- x_3 associada à linha 1: $(7x_1 - 2x_2 \leq 14) \times \frac{1}{7}$
- x_4 associada à linha 2: $(x_2 \leq 3) \times \frac{2}{7}$
- x_5 associada à linha 3: $(2x_1 - 2x_2 \leq 3) \times 0$
- Somando estas desigualdades válidas temos: $x_1 \leq \frac{20}{7} \rightarrow x_1 \leq 2$.

Poderíamos ter gerado um corte CG a partir da terceira linha do primeiro dicionário:

$$-\frac{2}{7}x_3 + \frac{10}{7}x_4 + x_5 = \frac{23}{7}$$

Então:

Restrições iniciais

$$\begin{aligned} z &= \max & 4x_1 &- x_2 \\ & & 7x_1 &- 2x_2 \leq 14 \\ & & & x_2 \leq 3 \\ & & 2x_1 &- 2x_2 \leq 3 \end{aligned}$$

Cortes gerados: $x_1 \leq 2$ e $x_1 - x_2 \leq 1$

- Após a aplicação de *muitas* rodadas de cortes de Chvátal-Gomory, os coeficientes dos cortes passam a ser muito grandes.
- Matrizes de coeficientes passam a ser malcondicionadas.
- Problemas numéricos para se inverter tais matrizes.
- Ao longo das décadas de 1960 e 1970, houve uma certa dose de ceticismo quanto à utilidade prática do uso dos algoritmos de planos de corte para resolver problemas reais de otimização combinatória
- Utilização de cortes de CG de baixo rank é seguro do ponto de vista numérico.

- 1 São algoritmos Branch-and-bound que utilizam Relaxações Lineares para se obter limites duais.
- 2 Antes de se realizar Branching, são separadas classes de desigualdades válidas visando tentar fortalecer os limites duais.
- 3 Se nenhuma desigualdade válida violada pela solução da Relaxação Linear for caracterizada pelos algoritmos de separação (seja por não existir ou por não termos sido capazes de identificá-la), fazemos branching. Se forem encontradas, inserimos uma ou mais desigualdades na relaxação linear e reotimizamos.
- 4 Ou seja, há um algoritmo de planos de corte sendo executado durante a resolução dos subproblemas encontrados na árvore de enumeração Branch-and-bound.

Formulação não compacta

$$z = \min \left\{ \sum_{(i,j) \in A} x_{ij} c_{ij} : x \in P \cap \mathbb{B}^{|A|} \right\}$$

onde P é dada por:

$$\sum_{(j,i) \in \delta^-(i)} x_{ji} = 1, i \in N$$

$$\sum_{(i,j) \in \delta^+(i)} x_{ij} = 1, i \in N$$

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq 1, \forall S \subset N, S \neq \emptyset$$

$$x_{ij} \geq 0, (i,j) \in A$$

Problema de separação

- Dado o ponto $\bar{x} \in [0, 1]^m$, separar as desigualdades de eliminação de subcircuitos direcionada

$$\sum_{(i,j) \in \delta^+(S)} x_{ij} \geq 1, \forall S \subset N, S \neq \emptyset$$

equivale a resolver um problema de fluxo máximo em uma rede dada pelo grafo de definição do TSP, com capacidades dos arcos dados por $\{\bar{x}_{ij} : (i, j) \in A\}$.

- Este problema de separação é polinomialmente solúvel.

Outras classes de desigualdades válidas para o TSP podem ser separadas no algoritmo.

