Programação Inteira - lidar com a complexidade

Investigação Operacional

J.M. Valério de Carvalho
vc@dps.uminho.pt

Departamento de Produção e Sistemas
Escola de Engenharia, Universidade do Minho

28 de Dezembro de 2021

antes

- Vimos os métodos dos planos de corte e de partição e avaliação.

Guião

- O uso de procedimentos para encontrar soluções inteiras admissíveis de boa qualidade e de modelos mais fortes
- permite reduzir a carga computacional de resolução de um problema de PI no método de partição e avaliação.
- nota: a análise e os exemplos apresentados são de um problema de maximização. Para um problema de minimização, a análise é idêntica com os devidos ajustes.

A prática mostra que

- é possível resolver ou, pelo menos, obter soluções de boa qualidade para modelos de problemas reais complexos.

Lidar com a complexidade da Programação Inteira

- O problema de PI pertence à classe de problemas NP-completos: não é conhecido nenhum método de resolução 'eficiente'.
- Os métodos dos planos de corte e de partição e avaliação podem requerer, no pior caso, um esforço computacional exponencialmente grande em relação ao número de variáveis de decisão.
- Dada a complexidade do problema de PI, temos de aceitar que a solução óptima pode não ser encontrada em tempo razoável.

Vamos ver, no âmbito do método de partição e avaliação, métodos para

1. tentar reduzir o esforço computacional, e
2. avaliar a qualidade da melhor solução inteira admissível conhecida (quão próximo é o seu valor do valor da solução óptima).

Esforço computacional (problema de maximização)

- A redução do esforço computacional é tentada através do aumento da taxa de sucesso de abandono de nós da árvore de pesquisa.
- Um critério de abandono de um nó é não haver, nos seus descendentes, uma solução melhor do que a melhor solução inteira admissível conhecida:

Lembrete: na avaliação, o nó i pode ser abandonado se

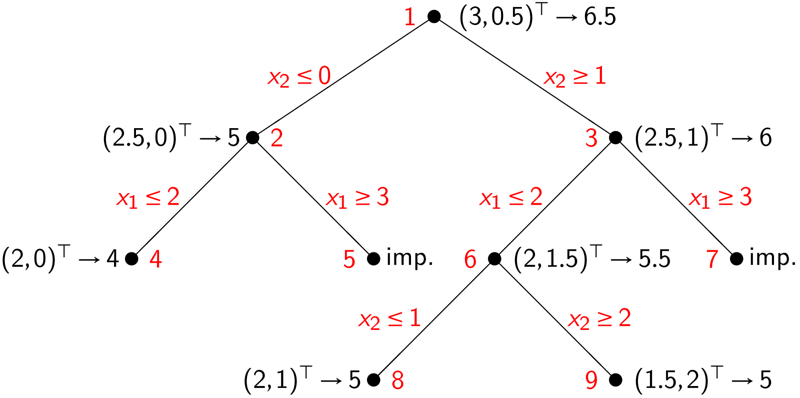
- o valor $z_{PL}(i)$ da solução óptima de PL do nó i, $\boldsymbol{x_{PL}}(i)$, for menor ou igual ao valor da melhor solução inteira admissível conhecida.

Para tentar aumentar a taxa de sucesso de abandono de nós, vamos

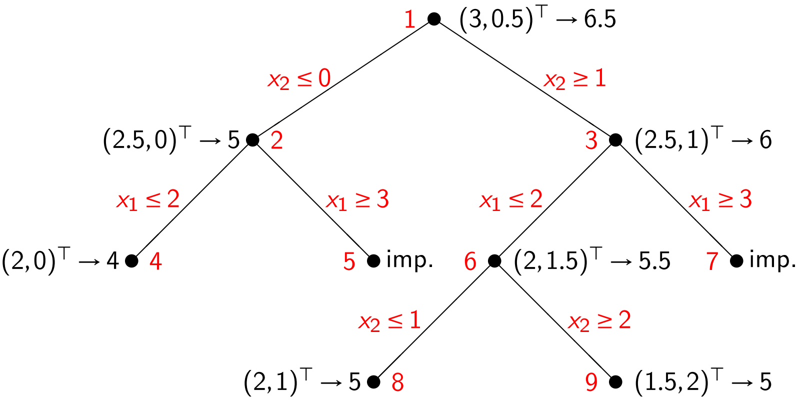
- procurar soluções inteiras admissíveis de boa qualidade,
- usar *modelos mais fortes* (com menor valor de $z_{PL}(i)$ do nó i).

Exemplo: avaliação do nó 9 (problema de maximização)

- Antes respondemos à questão: podemos abandonar o nó 9?



- Antes respondemos à questão: podemos abandonar o nó 9?



- Sim, no nó 8 foi encontrada uma solução inteira admissível de valor 5 e, nos descendentes do nó 9, não existe nenhuma solução melhor.

Soluções de boa qualidade (problema de maximização)

- Para além das soluções inteiras obtidas na resolução dos problemas de PL dos nós da árvore de pesquisa, podemos recorrer a:

- *Heurísticas* são procedimentos que geram soluções admissíveis $\boldsymbol{x}_{heur}$ para um modelo, mas que podem não ser óptimas.

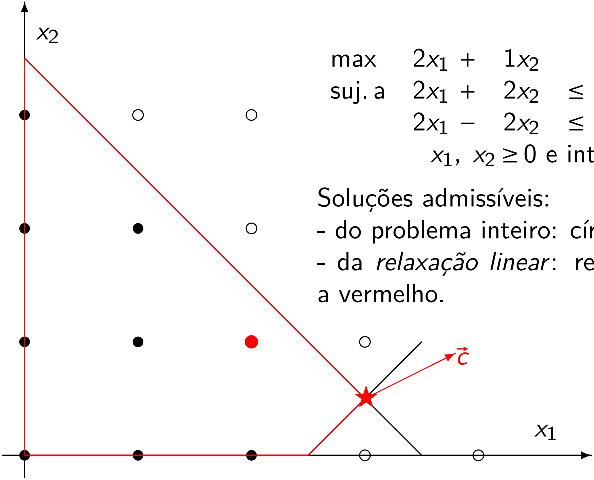
- Caso o valor de uma solução heurística seja melhor ($z_{heur} \geq z_{SI}$), essa solução heurística passa a ser a solução incumbente.

Limite Inferior (LI) para o valor do óptimo:

- O valor z_{SI} da solução incumbente $\boldsymbol{x_{SI}}$ é um *limite inferior* (LI) para o valor z_I^* da solução óptima inteira $\boldsymbol{x_I^*}$:

$$z_{SI} \leq z_I^*$$

- Há heurísticas de baixa complexidade (*e.g.*, linear ou quadrática em função da dimensão dos dados do problema) ou muito elaboradas.



$$
\begin{array}{llllll}
\max & 2x_1 & + & 1x_2 & & \\
\text{suj. a} & 2x_1 & + & 2x_2 & \leq & 7 \\
 & 2x_1 & - & 2x_2 & \leq & 5 \\
 & & & \multicolumn{3}{l}{x_1,\ x_2 \geq 0 \text{ e inteiros}}
\end{array}
$$

Soluções admissíveis:
- do problema inteiro: círculos cheios.
- da *relaxação linear*: região delimitada a vermelho.

Soluções óptimas:
- do problema inteiro: $\boldsymbol{x}_{\boldsymbol{I}}^* = (x_1, x_2)_{\boldsymbol{I}}^\top = (2,1)^\top \rightarrow z_{PI} = 5$ (●) ;
- da *relaxação linear*: $\boldsymbol{x}_{\boldsymbol{RL}} = (x_1, x_2)_{\boldsymbol{RL}}^\top = (3, 0.5)^\top \rightarrow z_{RL} = 6.5$ (★).

Heurística miópica (problema de maximização)

- uma a uma, sem atender às outras variáveis, atribuir à variável de decisão o valor inteiro (permitido pelos recursos ainda disponíveis) que optimiza a contribuição para a função objectivo.

Algoritmo 1 Heurística miópica (problema de maximização)

Input: $\boldsymbol{A}, \boldsymbol{c} \geq \boldsymbol{0}, \boldsymbol{b}$ ▷ modelo
  for $j \leftarrow 1, n$ **do**
    if $\exists x_j$ inteiro que respeite as restrições **then**
      $x_j^h \leftarrow$ valor x_j inteiro que optimiza contribuição para o f. obj.
    else
      return Heurística falhou
    end if
    actualizar modelo fazendo a atribuição: $x_j \leftarrow x_j^h$
  end for
  return solução inteira admissível $\boldsymbol{x}_{heur} = (x_1^h, \ldots, x_n^h)^\top$

Assume-se $\boldsymbol{c} \geq \boldsymbol{0}$ para simplificar a apresentação da heurística.

Exemplo: aplicação da heurística miópica

$$\begin{array}{llcll} \max & 2x_1 & + & 1x_2 & \\ \text{suj. a} & 2x_1 & + & 2x_2 & \leq & 7 \\ & 2x_1 & - & 2x_2 & \leq & 5 \\ & & & x_1,\ x_2 \geq 0 \text{ e inteiros} \end{array}$$

- $j \leftarrow 1$: atribuir a x_1 o valor 2 (o valor 3 violaria a segunda restrição); quando se fixa $x_1 = 2$, as restrições ficam $2x_2 \leq 3$ e $-2x_2 \leq 1$ (*i.e.*, $2x_2 \geq -1$), respectivamente.
- $j \leftarrow 2$: atribuir a x_2 o valor 1.

- Solução heurística: $\boldsymbol{x}_{heur} = (2,1)^{\top}$, com valor $z_{heur} = 5$.
- Limite inferior para o valor do óptimo: $z_{heur} = 5 \leq z_I^*$.

Heurística de arredondamento (problema de maximização)

- arredondar (para cima/para baixo) os valores da solução óptima fraccionária, para construir uma solução inteira.

Algoritmo 2 Heurística de arredondamento (problema de maximização)

Input: $\boldsymbol{x}^f = (x_1^f, \ldots, x_n^f)^\top; \boldsymbol{A}, \boldsymbol{c} \geq \boldsymbol{0}, \boldsymbol{b}$ ▷ sol. óptima fraccionária; modelo

  for $j \leftarrow 1, n$ **do**

    if $x_j = \lceil x_j^f \rceil$ for admissível **then**

      $x_j^h \leftarrow \lceil x_j^f \rceil$

    else if $x_j = \lfloor x_j^f \rfloor$ for admissível **then**

      $x_j^h \leftarrow \lfloor x_j^f \rfloor$

    else

      return Heurística falhou

    end if

    actualizar modelo fazendo a aribuição: $x_j \leftarrow x_j^h$

  end for

  return solução inteira admissível $\boldsymbol{x}_{heur} = (x_1^h, \ldots, x_n^h)^\top$

Assume-se $\boldsymbol{c} \geq \boldsymbol{0}$ para simplificar a apresentação da heurística

Exemplo: aplicação de heurística de arredondamento

$$\begin{array}{llcllcl} \max & 2x_1 & + & 1x_2 & & \\ \text{suj. a} & 2x_1 & + & 2x_2 & \leq & 7 \\ & 2x_1 & - & 2x_2 & \leq & 5 \\ & & x_1, & x_2 \geq 0 & \text{e inteiros} & \end{array}$$

	x_1	x_2	s_1	s_2	
x_1	1	0	1/4	1/4	3
x_2	0	1	1/4	-1/4	1/2
	0	0	3/4	1/4	6 1/2

- $j \leftarrow 1$: a variável x_1 é inteira;

  quando se fixa $x_1 = 3$, as restrições ficam $2x_2 \leq 1$ e $-2x_2 \leq -1$ (*i.e.*, $2x_2 \geq 1$), respectivamente.
- $j \leftarrow 2$: A variável x_2 não pode ser arredondada para cima, nem para baixo; não há nenhum valor inteiro que obedeça às duas restrições.
- A heurística falha, não produzindo uma solução admissível.

- Ambas as heurísticas são simples, de complexidade linear.
- Heurísticas deste tipo são por vezes usadas em todos os nós da árvore de pesquisa.

Limite superior (problema de maximização)

Seja *LS* um *Limite Superior* para o valor do óptimo:

- Um *LS* é um valor para o qual é válida a seguinte relação:

$$z_I^* \leq LS.$$

- Não está associado a nenhuma solução inteira admissível.

Exemplos:

- o valor do óptimo da relaxação linear z_{RL} é um *limite superior* para o valor do óptimo inteiro z_I^*, dado que todas as soluções inteiras pertencem à relaxação linear do modelo.
- o mesmo para o valor da solução após adicionar planos de corte.

Qualidade da solução (problema de maximização)

- O método de partição e avaliação fornece também limites superiores.

O valor da solução óptima inteira (z_I^*) está contido no intervalo

$$z_I^* \in [LI, LS],$$

sendo LI (Limite Inferior) e LS (Limite Superior).

A *diferença (gap) de integralidade* (em %), dada por $(LS - LI)/LI$,

- é uma medida da qualidade da melhor solução inteira conhecida.

- Há a garantia de que a melhor solução inteira conhecida tem um valor que se aproxima do valor do óptimo a menos de uma percentagem igual ao *gap*.

Majorante e minorante são designações alternativas de LS e LI, respectivamente.

Exemplo: limite superior (raiz da árvore)

$$\begin{array}{llllll} \max & 2x_1 & + & 1x_2 & & \\ \text{suj. a} & 2x_1 & + & 2x_2 & \leq & 7 \\ & 2x_1 & - & 2x_2 & \leq & 5 \\ & & & x_1,\ x_2 \geq 0 \text{ e inteiros} & & \end{array}$$

	x_1	x_2	s_1	s_2	
x_1	1	0	1/4	1/4	3
x_2	0	1	1/4	-1/4	1/2
	0	0	3/4	1/4	6 1/2

- O valor do óptimo inteiro $z_I^* \leq z_{RL} = 6\ \frac{1}{2}$.
- Neste exemplo, é possível fixar um limite superior mais forte, $z_I^* \leq 6 = \lfloor 6\ \frac{1}{2} \rfloor$, dado que z_I^* é necessariamente um valor inteiro, porque os coeficientes da função objectivo são inteiros e as variáveis de decisão têm de ser inteiras.

- As heurísticas e o LS podem ser obtidos após a resolução da relaxação linear do problema, permitindo obter a seguinte informação.

Exemplo

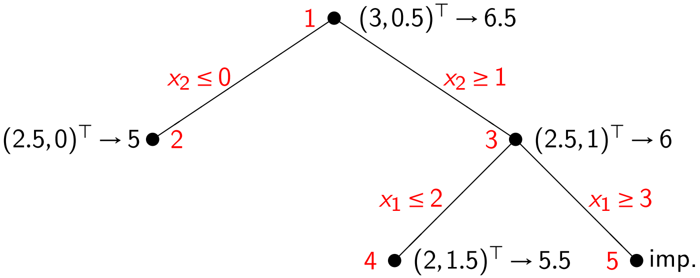
na raiz da árvore de pesquisa,

- LS=6 e a solução heurística (inteira e admissível) tem valor 5 = LI.
- O valor do óptimo inteiro $z_I^* \in [5, 6]$.
- O *gap* = (6 - 5) / 5 = 20%.
- Há a garantia de que a solução inteira conhecida está a menos de 20% do óptimo (ou, em valor absoluto, a 1 unidade do óptimo).

- Quando $gap = 0$, o LS (num problema de maximização) prova que a solução incumbente (com o valor LI) é a solução óptima.
- Podemos querer interromper a pesquisa quando o valor do *gap* é suficientemente pequeno.

- O uso de limites inferiores e superiores pode reduzir substancialmente o esforço computacional para encontrar uma solução óptima inteira.

Exemplo: resolução com LI e LS (prob. de maximização)



1. LS=6, LI=5
2. solução fraccionária, com valor 5; abandonar nó: nos descendentes, não há solução melhor do que $\boldsymbol{x}_{heur} = (2,1)^\top$ com valor $z_{heur} = 5$.
3.
4. solução fraccionária, com valor 5.5; $\lfloor 5.5 \rfloor = 5$; abandonar nó.
5. Impossível; ListaNós vazia; solução dada pela heurística é óptima.

Evolução do LS (problema de maximização)

O LS vai diminuindo à medida que se pesquisa a árvore:

- Seja A_k o conjunto de *nós em aberto* (nós que ainda falta avaliar) quando se avalia o nó k.
- O LS para o valor do óptimo inteiro é dado pelo maior valor de um nó em aberto quando se avalia o nó k:

$$LS = \max_{i \in A_k} \{z_{PL}(i)\},$$

- dado que potencialmente ainda podemos vir a encontrar uma solução inteira admissível com esse valor num descendente do nó que determina o valor de LS.

- Menores valores de LS aumentam a taxa de sucesso de abandono dos nós.

Modelos mais fortes (problemas de maximização)

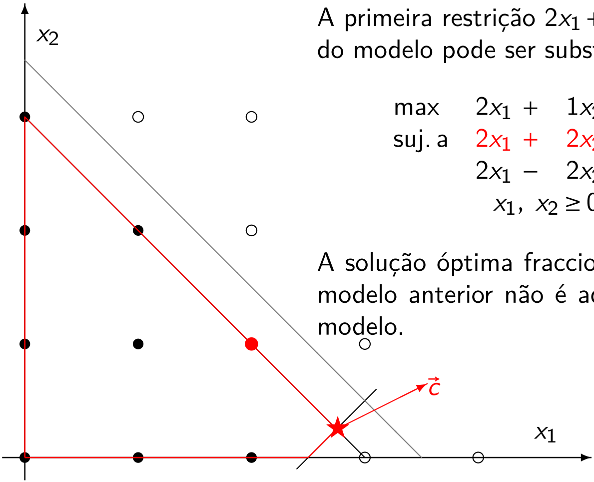
Ideia a explorar:

- O LS é melhor se tivermos um modelo de PI cuja relaxação linear seja 'mais próxima' do conjunto de soluções inteiras admissíveis.

- Restrições equivalentes para o problema inteiro (com o mesmo domínio inteiro) podem ter domínios diferentes quando se relaxam as restrições de integralidade das variáveis.

Exemplo

- A restrição $2x_1 + 2x_2 \leq 6$ define o mesmo domínio inteiro (tem o mesmo conjunto de soluções inteiras admissíveis) que $2x_1 + 2x_2 \leq 7$.
- Os domínios das respectivas relaxações lineares são diferentes.



A primeira restrição $2x_1 + 2x_2 \leq 7$ do modelo pode ser substituída por:

$$\begin{array}{llcll} \max & 2x_1 & + & 1x_2 & \\ \text{suj. a} & \color{red}{2x_1} & \color{red}{+} & \color{red}{2x_2} & \color{red}{\leq} \; \color{red}{6} \\ & 2x_1 & - & 2x_2 & \leq \; 5 \\ & & & x_1, \; x_2 \geq 0 \text{ e inteiros} & \end{array}$$

A solução óptima fraccionária $(3, 0.5)^\top$ do modelo anterior não é admissível no novo modelo.

Soluções óptimas:

- do problema inteiro: $\boldsymbol{x}_I^* = (x_1, x_2)_I^\top = (2, 1)^\top \rightarrow z_{PI} = 5$ ($\color{red}{\bullet}$) ;
- da *relaxação linear*: $\boldsymbol{x}_{RL} = (x_1, x_2)_{RL}^\top = (2.75, 0.25)^\top \rightarrow z_{RL} = 5.75$ ($\color{red}{\bigstar}$).

Exemplo de motivação

na raiz da árvore de pesquisa,

- O valor do óptimo inteiro $z_I^* \leq z_{RL} = 5.75$.
- Neste exemplo, é possível fixar um limite superior mais forte, $z_I^* \leq 5 = \lfloor 5.75 \rfloor$ (como vimos, z_I^* é necessariamente um valor inteiro).
- LS=5 e a solução heurística (inteira e admissível) tem valor 5 = LI.
- O $gap = 0$: o LS comprova que a solução heurística é óptima.
- O problema fica resolvido na raiz da árvore de pesquisa.

Relaxação linear de um modelo de PI

- Um modelo de programação inteira (PI) têm a seguinte forma:

$$\begin{aligned} \max z_{PI} &= \boldsymbol{cx} \\ \text{suj. a} \quad & \boldsymbol{Ax} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \text{ e inteiro} \end{aligned}$$

- O domínio (conjunto de soluções admissíveis) do modelo PI é:

$$X_{PI} = \{\boldsymbol{x} \in \mathbb{Z}_+^n : \boldsymbol{Ax} \leq \boldsymbol{b}\}.$$

- O modelo de Programação Linear (PL) que resulta do relaxação das *restrições de integralidade* tem variáveis de decisão $\boldsymbol{x} \in \mathbb{R}_+^n$:

$$\begin{aligned} \max z_{RL} &= \boldsymbol{cx} \\ \text{suj. a} \quad & \boldsymbol{Ax} \leq \boldsymbol{b} \\ & \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

- O domínio (conjunto de soluções admissíveis) da relaxação linear do modelo é:

$$X_{RL} = \{\boldsymbol{x} \in \mathbb{R}_+^n : \boldsymbol{Ax} \leq \boldsymbol{b}\}.$$

Relaxações lineares de diferentes modelos de PI

- Muitas vezes, existem diferentes formulações de um problema de PI.
- Qualquer modelo de PI é válido se o seu conjunto de soluções inteiras admissíveis for o X_{PI} desejado.
- Mas pode haver diferenças nas respectivas relaxações lineares.
- Sejam $X_{RL_1} \subseteq \mathbb{R}^n$ e $X_{RL_2} \subseteq \mathbb{R}^n$ os domínios das relaxações lineares de dois modelos de PI válidos.
- Ambas as relaxações lineares contêm o mesmo conjunto de soluções inteiras admissíveis:

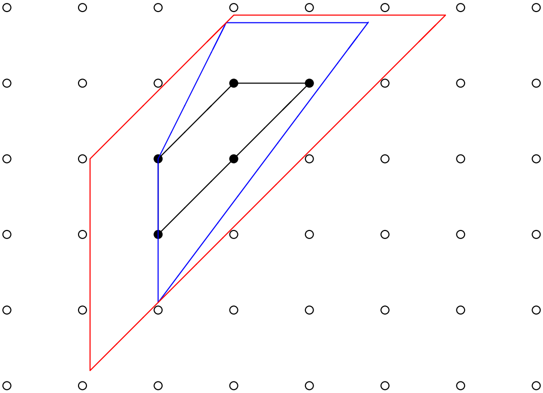
$$X_{PI} \quad = \quad X_{RL_1} \cap \mathbb{Z}^n_+ \quad = \quad X_{RL_2} \cap \mathbb{Z}^n_+.$$

Um modelo de PI é mais forte

- se o domínio da sua relaxação linear X_{RL_1} estiver contido no domínio da relaxação linear do outro modelo, com um domínio X_{RL_2} :

$$X_{RL_1} \subset X_{RL_2}.$$

Exemplo



- O modelo mais forte de todos é o que tem uma relaxação linear cujos vértices são soluções inteiras admissíveis;
- o domínio dessa relaxação linear designa-se por *envolvente convexa* do domínio inteiro (um conjunto discreto de pontos).

A envolvente convexa de um conjunto de pontos é o menor conjunto convexo que contém todos os pontos do conjunto.

Algumas formas de tornar os modelos mais fortes

- aumento de coeficientes em restrições do tipo $\leq$.
- redução de coeficientes em restrições do tipo $\geq$.
- desagregação de restrições
- restrições de clique e de ciclo

- Muitos dos exemplos que se seguem usam variáveis binárias.
- na relaxação linear, substitui-se x_j binário por $0 \leq x_j \leq 1$.

Aumento de Coeficientes (*coefficient lifting*)

- Se o coeficiente de uma variável puder ser aumentado de uma quantidade que não altera o conjunto de soluções inteiras admissíveis, a restrição resultante é mais forte.

Exemplo

- Restrição com variáveis binárias $7x_1 + 8x_2 + 9x_3 \leq 12$.
- Todos os coeficientes das variáveis podem ser aumentados para 12, porque, se qualquer uma das variáveis tomar o valor 1, as outras variáveis devem ser nulas.
- A restrição é equivalente a $x_1 + x_2 + x_3 \leq 1$.
- O conjunto de soluções 0-1 das duas restrições é o mesmo.

Redução de coeficientes

- Se uma restrição se tornar redundante quando uma das variáveis binárias é fixada num dos seus limites,
- o coeficiente dessa variável pode ser reduzido de uma quantidade igual ao valor pelo qual a restrição se tornou redundante.

Exemplo

- Restrição com variáveis binárias $3x_1 + x_2 + x_3 \geq 2$.
- Se a variável x_1 for fixada em 1, a restrição torna-se redundante.
- O coeficiente de x_1 pode ser reduzido para o valor 2, passando a restrição a ser $2x_1 + x_2 + x_3 \geq 2$.
- O conjunto de soluções 0-1 das duas restrições é igual.

Desagregação de restrições (*constraint disaggregation*)

- Restrições agregadas originam modelos mais fracos.
- Do ponto de vista computacional, é preferível ter um número de restrições mais elevado, para obter uma relaxação linear mais forte.

Exemplo

- Restrições com variáveis binárias:

$$\begin{aligned} x_{1A} &\leq y_A \\ x_{2A} &\leq y_A \end{aligned}$$

- A restrição agregada $x_{1A} + x_{2A} \leq 2y_A$ é válida, mas mais fraca que as duas restrições em conjunto (porquê?)

- Este tipo de restrições ocorre, *e.g.*, em problemas de localização.
- A primeira restrição traduz uma implicação: atribuir o cliente 1 ao servidor A implica a implantação do servidor A; idem para a segunda.
- A restrição agregada também: atribuir, pelo menos, um dos clientes ao servidor A implica a implantação do servidor A.

Pré-processamento

- É largamente reconhecido que o uso de modelos fortes se traduz em melhorias substanciais dos tempos de computação.
- Os *solvers* de PI fazem um pré-processamento dos modelos que lhes são submetidos, que inclui, entre outras, as operações apresentadas.
- O objectivo é estabelecer limites mais apertados para os valores das variáveis, ou mesmo a fixação do seu valor, obtendo modelos mais fortes e com menos variáveis.

Outras técnicas de obtenção de formulações mais fortes

- Há formulações que são mais fortes, mas que podem ter um número exponencial de variáveis.
- Há técnicas para resolver estes modelos sem haver necessidade de gerar todas as variáveis possíveis.
- Este tipo de modelos é usado, por exemplo, em encaminhamento de veículos (*vehicle routing*) e corte e empacotamento (*cutting and packing*).

Conclusão

- Nas últimas décadas, desenvolvimentos teóricos (modelos mais fortes) e o aumento da capacidade de computação têm permitido resolver instâncias de dimensão cada vez maior.
- A programação inteira é hoje em dia usada rotineiramente para encontrar soluções para problemas reais complexos em muitas indústrias e serviços.
- Geralmente, é possível obter, em tempos razoáveis, soluções inteiras óptimas, ou então soluções com diferenças *(gaps)* de integralidade inferiores a 5%, muitas vezes de 1% ou 2%.
- O valor de *gap* é uma garantia da qualidade das soluções!
- Formulação mais forte produz um melhor valor do limite superior (inferior) para o valor do óptimo do problema de programação inteira de maximização (minimização).
- Qualidade da formulação é mais importante do que a dimensão da formulação (número de variáveis ou restrições).

Apêndice

- Inequações válidas para o PI

Inequações válidas para o problema inteiro

- Há também *inequações válidas para o problema inteiro* que podem ser obtidas a partir de um conjunto de restrições do modelo original.
- Adicioná-las ao modelo, tornam-no mais forte.
- Exemplo: restrições que representam um conflito entre os vértices i e j, de um grafo $G = (V, A)$.

Exemplo 1: inequações de clique:

- Uma clique K_p é um subgrafo completo com p vértices.

$$\sum_{i \in K_p} x_i \leq 1$$

Exemplo 2: inequações de ciclos ímpares *(odd holes)* :

- Um ciclo C_p com um número de vértices ímpar p.

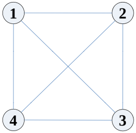
$$\sum_{i \in C_p} x_i \leq \lfloor p/2 \rfloor$$

Problema de empacotamento *(set packing problem)*

$$\max\{\boldsymbol{cx} : \boldsymbol{Ax} \leq \mathbf{1}, \boldsymbol{x} \in \{0,1\}^n\}, \text{ sendo } \boldsymbol{A} \in \{0,1\}^{m \times n}.$$

- Exemplo:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
1	1	1	1				1	≤ 1
2	1			1	1			≤ 1
3		1		1		1	1	≤ 1
4			1		1	1	1	≤ 1
solução:	0.33	0.33	0.33	0.33	0.33	0.33		



Inequação para a clique K_4:

- $x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \leq 1$

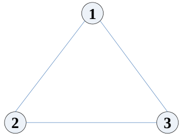
Inequações válidas para o problema de empacotamento

Problema de empacotamento *Set packing problem*

$$\max\{\boldsymbol{cx} : \boldsymbol{Ax} \le \mathbf{1}, \boldsymbol{x} \in \{0,1\}^n\}, \text{ sendo } \boldsymbol{A} \in \{0,1\}^{m \times n}.$$

- Exemplo:

	x_1	x_2	x_3	x_4	x_5	x_6	x_7	
1	1	1		1			1	≤ 1
2		1	1		1		1	≤ 1
3	1		1			1	1	≤ 1
solução:	0.5	0.5	0.5					



- A inequação válida é mais forte, porque a soma das restrições:

$$\begin{array}{cccccc} 2x_1 & +2x_2 & +2x_3 & \le & 3 \\ x_1 & +\ x_2 & +\ x_3 & \le & 3/2 \\ x_1 & +\ x_2 & +\ x_3 & \le & 1 \end{array}$$

Inequação para o ciclo C_3:

- $x_1 + x_2 + x_3 \le 1$

Fim