



Universidade Federal do Espírito Santo
Centro Universitário Norte do Espírito Santo
Colegiado do Curso de Bacharelado em Matemática

Rodrigo Leppaus de Araujo

O problema da elaboração de grade de horários via
Programação Inteira

São Mateus

2014

Rodrigo Leppaus de Araujo

**O problema da elaboração de grade de horários via
Programação Inteira**

Trabalho submetido ao Colegiado do Curso de Bacharelado em Matemática da UFES (Campus São Mateus), como requisito parcial para a obtenção do grau de Bacharel em Matemática.

Orientador:

Prof. Leonardo Delarmelina Secchin

São Mateus

2014

Rodrigo Leppaus de Araujo

O problema da elaboração de grade de horários via Programação Inteira

Trabalho submetido ao Colegiado do Curso de Bacharelado em Matemática da UFES (Campus São Mateus), como requisito parcial para a obtenção do grau de Bacharel em Matemática.

Aprovada em 11 de Março de 2014.

Comissão Examinadora

Prof. Leonardo Delarmelina Secchin
Universidade Federal do Espírito Santo
Orientador

Prof. Lúcio Souza Fassarella
Universidade Federal do Espírito Santo

Prof. André Renato Sales Amaral
Universidade Federal do Espírito Santo

Agradecimentos

A DEUS. Ao seu infinito amor por mim, sempre iluminando meus caminhos.

A minha família. Aos meus pais Irineu e Maria. Todo exemplo de vida, meu berço de toda ternura e aconchego e acima de tudo que sempre acreditaram nesse sonho. A minha irmã Juliana que sempre me apoiou e incentivou, meu muito obrigado, e que a nossa união se perdue por toda vida. Ao meu caçula, meu guerreiro Elivelton, você é o meu herói, te amo demais. Minha vó Ilda que sempre perturbo.

Aos amigos da VSD, que a nove anos somos unidos com toda força de uma bela amizade, em especial a Scheppa, mulher que admiro e amo de ♡, Marcela mulher com um sorriso lindo e com alma bela, minha Jolie, Rômina mulher que admiro e que tenho profundo carinho, Gleuce mulher mais alegre do mundo. *Marcia*, que neste período esteve ao meu lado, nos bons e maus momentos, que sofreu com minhas ligações, meus choros, minhas brincadeiras, muito obrigado por tudo! Aos agregados que vieram juntamente dessa amizade, Murilo, Leide, Milson, Roberta, Charles, Serginho, Márcia Soares, Sérgio, Valmir, Nete, Clemilson, toda família Soares, família Scheppa e família Melotti, por sempre me acolherem em seus lares com muito amor e carinho. Aos amigos Murilo e Gabriela, muito obrigado pelo carinho e amizade sincera, que DEUS possa sempre edificar esse lar. Aos amigos Camila e Wesley, muito obrigado pelo carinho e amizade sincera, que DEUS possa sempre edificar esse lar. Isamara, minha Curica. Aos amigos formados nesse período de estudo, em especial a Fabrício, Josi e Ramoni. Muito obrigado por sempre me apoiarem, que nossa amizade aumente e fortaleça sempre.

Aos colegas de graduação em matemática, obrigado por me aturarem, sei que não sou uma pessoa de fácil convivência, mas sei que grandes sorrisos arranquei de vocês.

Aos professores do DMA/DCN, muito obrigado pelo conhecimento transmitido.

Ao meu orientador, Secchin, que muito contribuiu para essa jornada. Muito obrigado pela paciência e confiança.

*Tenho sorte de respirar, tenho sorte de sentir, sou feliz por acordar, sou feliz por estar
aqui.*

S.O.J.A

Resumo

Modelos de programação inteira consistem em minimizar uma função linear cx nas variáveis inteiras x_i , sujeitas à restrições de igualdade e/ou desigualdade que, em geral, podem ser escritas na forma $Ax \leq b$. Neste trabalho estudamos conceitos básicos desses modelos bem como dois métodos de resolução: o Método de Planos de Corte e o método enumerativo *Branch-and-Bound*. Modelamos o problema da geração de grade de horários escolares que, em particular, apresenta variáveis binárias. O uso de variáveis que assumem 0 ou 1 é bastante comum na literatura não só na modelagem de problemas de grade de horários, mas também em uma rica variedade de outros problemas. Testes computacionais foram realizados com dados reais do Departamento de Matemática Aplicada da UFES. Utilizamos o método enumerativo *Branch-and-Cut* implementado no software CPLEX, amplamente utilizado na literatura. Uma breve discussão desse método de resolução é feita neste trabalho.

Palavras-chave: Programação inteira, otimização, grade de horários.

Lista de Figuras

1	Figura para o Exemplo 1.1.1.	11
2	Diferentes formulações. A região do primeiro quadrante delimitada pelas linhas pretas representa P_1 ; A delimitada pelas vermelhas, P_2	15
3	Representação da região de viabilidade do Exemplo 2.1.3. O conjunto X é formado pelos pontos em destaque. Em vermelho $\text{conv}(X)$. Em azul desigualdades válidas.	21
4	Representação da região de viabilidade do Exemplo 2.2.2.	24
5	Representação do corte na região de viabilidade do Exemplo 2.2.2. A região em destaque foi descartada pelo corte.	25
6	Figura do Exemplo 3.1.1. Árvore de enumeração para $P = \{0, 1\}^3$	32
7	Ramificação sobre variáveis fracionárias.	34
8	Corte por otimalidade.	35
9	Corte por limitantes.	35
10	Nenhum corte é possível.	36
11	Representação da região de viabilidade do Exemplo 3.1.6.	37
12	$P_2 = \emptyset$: corte por inviabilidade 3.1.6.	38
13	Representação gráfica de P_1 no Exemplo 3.1.6.	39
14	Representação gráfica de P_{11} no Exemplo 3.1.6.	40
15	Representação gráfica de P_{12} no Exemplo 3.1.6.	40

Lista de Símbolos

$\lfloor x \rfloor$	maior número inteiro menor do que ou igual a x
$\lceil x \rceil$	menor número inteiro maior do que ou igual a x
\mathbb{R}_+	conjunto dos números reais não negativos
\mathbb{Z}_+	conjunto dos números inteiros não negativos
s.a.	abreviação para “sujeito a”
$Ax \leq b$	sistema das desigualdades $ax_i \leq b_i$ onde a_i é a i -ésima linha da matriz A
min	abreviação para “minimizar” ou “mínimo”, dependendo do contexto
max	abreviação para “maximizar” ou “máximo”, dependendo do contexto

Sumário

1	Programação Inteira	10
1.1	Introdução	10
1.2	Exemplos em Programação Linear	12
1.2.1	Problema de Alocação	12
1.2.2	Problema Knapsack	12
1.2.3	Problema do Caixeiro Viajante	13
1.3	Formulações e formulação ideal	14
2	Método de Planos de Corte	19
2.1	Desigualdades Válidas	19
2.2	O Método de Planos de Corte	21
2.2.1	Desigualdades válidas em Programação Inteira	22
2.2.2	Cortes via Quadro Simplex	26
3	Métodos Enumerativos	30
3.1	Branch and Bound	31
3.1.1	O método <i>Branch and Bound</i> - limitantes via relaxação linear	36
3.2	Branch and Cut	41
4	O problema da elaboração de grade de horários	42
4.1	Modelagem Matemática	43
4.2	Resultados	48
5	Conclusões	51

Referências	52
Apêndice A - Programação Linear	53
Apêndice B - O Dual de um problema de Programação Linear	65

1 Programação Inteira

1.1 Introdução

Um modelo de Programação Linear (PL), é um modelo da forma

$$\begin{aligned} \text{PL: } \min \quad & cx \\ \text{s.a. } & Ax \leq b \\ & x \in Y \end{aligned}$$

onde $Y = \mathbb{R}_+^n$, c um funcional linear, A é uma matriz $m \times n$ de posto $m < n$ e b um vetor de ordem m . Durante todo este trabalho, vamos supor que as matrizes A e b têm entradas racionais apenas. A função cx é chamada *função objetivo*. O termo “s.a.” é a abreviação de “sujeito a”. A notação $Ax \leq b$ indica as m desigualdades $a_i x \leq b_i$, onde a_i é a i -ésima linha de A e b_i o i -ésimo termo de b . Tais desigualdades são chamadas de *restrições*. A condição $x \in \mathbb{R}_+^n$, também denotada por $x \geq 0$, é referenciada como *restrições de não negatividade*. Ao conjunto $P = \{x; Ax \leq b, x \in Y\}$ damos o nome de *conjunto de viabilidade* ou *conjunto viável*. Conjuntos dessa forma, definidos por inequações/equações lineares, são chamados *poliedros*. Um ponto $x \in P$ é uma *solução* ou *ponto viável* e a solução que minimiza a função objetivo é dita ser uma *solução ótima* ou simplesmente *ótimo*. O valor da função objetivo em uma solução ótima é o *valor ótimo* do modelo.

Quando $Y = \mathbb{Z}_+^n$, dizemos que PL é um modelo de Programação Inteira (PI). Neste caso nos referimos à condição $x \in \mathbb{Z}^n$ como *restrições de integralidade*. Se $Y = \mathbb{Z}_+^k \times \mathbb{R}_+^l$, onde $k, l > 0$ e $k + l = n$, dizemos que PL é um modelo de Programação Inteira Mista (PIM). A seguir apresentamos um modelo de PI e a representação gráfica de seu conjunto viável.

Exemplo 1.1.1. Considere o modelo

$$\begin{aligned}
 \min \quad & -x_1 \quad -x_2 \\
 \text{s.a.} \quad & 3x_1 + 2x_2 \leq 15 \\
 & 2x_1 - 3x_2 \geq -6 \\
 & x_1, \quad x_2 \in \mathbb{Z}_+^n
 \end{aligned}$$

O conjunto $P = \{(x_1, x_2) \in \mathbb{R}_+^2; 3x_1 + 2x_2 \leq 15, 2x_1 - 3x_2 \geq -6\}$ obtido do conjunto de viabilidade descartando as restrições de integralidade, bem como o conjunto de viabilidade $P \cap \mathbb{Z}^2$ são representados na Figura 1. Nas restrições, as setas vermelhas indicam a região que cada restrição delimita. As curvas de nível da função objetivo são retas. As soluções ótimas dos modelos sobre P e $P \cap \mathbb{Z}^2$ são indicadas pelos círculos em azul, onde são representadas as curvas de nível e o sentido de minimização. \square

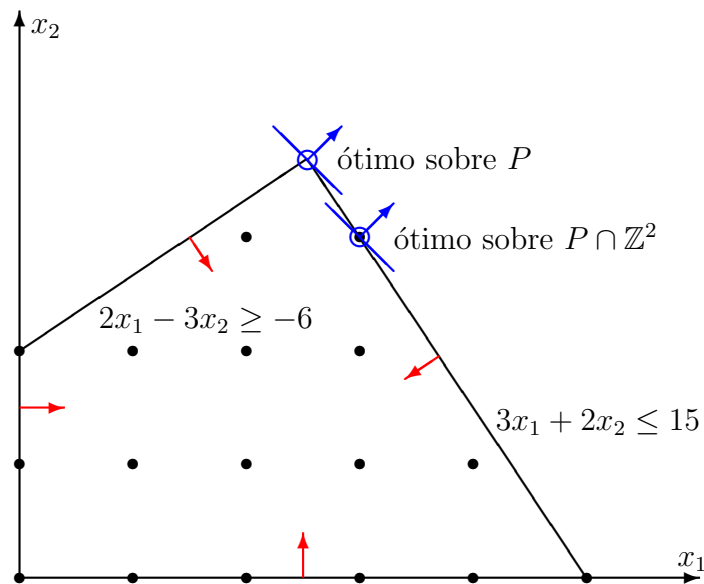


Figura 1: Figura para o Exemplo 1.1.1.

Quando em um problema temos que decidir se um evento ocorre ou não, é adequado o uso de *variáveis binárias*, definidas pondo

$$x = \begin{cases} 1, & \text{caso o evento ocorra} \\ 0, & \text{caso contrário} \end{cases} .$$

É frequente problemas de programação linear terem variáveis binárias, como veremos em alguns exemplos adiante. No Capítulo 4 apresentaremos um modelo com variáveis binárias para o problema da elaboração de grade de horários escolares.

1.2 Exemplos em Programação Linear

1.2.1 Problema de Alocação

O problema de alocação consiste em designar n pessoas a n trabalhos a serem realizados. Cada pessoa é alocada a um único trabalho, e cada trabalho deve ser realizado por uma única pessoa. O custo para a pessoa i realizar o trabalho j é dado por $c_{ij} > 0$. O objetivo é minimizar o custo total de alocação.

Definição das variáveis. Devemos decidir se a pessoa i realiza o trabalho j . Assim, definimos as variáveis binárias

$$x_{ij} = \begin{cases} 1, & \text{caso a pessoa } i \text{ realize o trabalho } j \\ 0, & \text{caso contrário} \end{cases}.$$

Definição das restrições. Cada pessoa realiza exatamente um trabalho, e assim exigimos

$$\sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n.$$

Como cada trabalho é realizado por exatamente uma única pessoa, temos

$$\sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n.$$

Assim, o modelo para o problema de alocação é dado por

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n \\ & \sum_{i=1}^n x_{ij} = 1, \quad j = 1, \dots, n \\ & x_{ij} \in \{0, 1\}, \quad \forall i, j. \end{aligned}$$

1.2.2 Problema Knapsack

Uma empresa dispõe de um certo orçamento b para investir em projetos durante o ano. Sejam a_j o investimento necessário para realização do projeto j e c_j o retorno financeiro esperado para o projeto j . A meta é um conjunto de projetos em que o orçamento não é

excedido e o retorno é o máximo possível.

Definição das variáveis.

$$x_j = \begin{cases} 1, & \text{se o projeto } j \text{ é selecionado} \\ 0, & \text{caso contrário} \end{cases}$$

Definição das restrições. Como o orçamento não pode ser excedido, inserimos a restrição

$$\sum_{j=1}^n a_j x_j \leq b.$$

Assim, o problema Knapsack é dado da seguinte forma:

$$\begin{aligned} \max \quad & \sum_{j=1}^n c_j x_j \\ \text{s.a.} \quad & \sum_{j=1}^n a_j x_j \leq b \\ & x_j \in \{0, 1\}, \quad \forall j \end{aligned}$$

1.2.3 Problema do Caixeiro Viajante

O problema do caixeiro viajante é um dos problemas mais clássicos em Pesquisa Operacional. Um vendedor (caixeiro) tem que visitar exatamente n cidades e voltar para a cidade inicial, passando uma única vez por cada cidade. A distância da cidade i para a cidade j é $c_{ij} > 0$. O objetivo é o de minimizar a distância total de viagem.

Definição das variáveis. Para $i \neq j$ definimos

$$x_{ij} = \begin{cases} 1, & \text{se o caixeiro sai da cidade } i \text{ para a cidade } j \\ 0, & \text{caso contrário} \end{cases}.$$

As variáveis x_{ii} não são definidas por motivos claros.

Definição das restrições. A fim de que o viajante passe uma única vez por cada cidade, podemos proceder como segue. Exigimos que em cada cidade, o caixeiro “entre” uma única vez e também “saia” uma única vez. Com as variáveis propostas, estabelecemos facilmente essas condições.

Pois bem, o caixeiro “sai” da cidade i uma única vez, e então

$$\sum_{j:j \neq i} x_{ij} = 1, \quad i = 1, \dots, n.$$

O caixeiro “entra” da cidade j uma única vez, e logo exigimos

$$\sum_{i:i \neq j} x_{ij} = 1, \quad j = 1, \dots, n.$$

Assim segue o modelo para o problema do caixeiro viajante:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{s.a.} \quad & \sum_{j:j \neq i} x_{ij} = 1, \quad i = 1, \dots, n \\ & \sum_{i:i \neq j} x_{ij} = 1, \quad j = 1, \dots, n \\ & x_j \in \{0, 1\}, \quad \forall j. \end{aligned}$$

1.3 Formulações e formulação ideal

Considere os conjuntos

$$P_1 = \{(x, y) \in \mathbb{R}_+^2; 6x + 14y \leq 35, 14x + 6y \leq 35\}$$

e

$$P_2 = \{(x, y) \in \mathbb{R}_+^2; x \leq 2, y \leq 2, x + y \leq 3\},$$

representados na Figura 2.

Graficamente, vemos que $P_2 \subsetneq P_1$ e $P_2 \cap \mathbb{Z}^2 = P_1 \cap \mathbb{Z}^2$. Ou seja, P_2 aproxima melhor os pontos de coordenadas inteiras pertencentes aos conjuntos. Dizemos neste caso que P_1 e P_2 são *formulações* para o conjunto

$$X = \{(x, y) \in \mathbb{Z}_+^2; 6x + 14y \leq 35, 14x + 6y \leq 35\}.$$

Mais geralmente,

Definição 1.3.1. *Um poliedro $P \subset \mathbb{R}^{n+p}$ é uma formulação para o conjunto X se $X = P \cap (\mathbb{Z}^n \times \mathbb{R}^p)$.*

Uma estratégia de resolução de modelos de programação inteira consiste em descartar

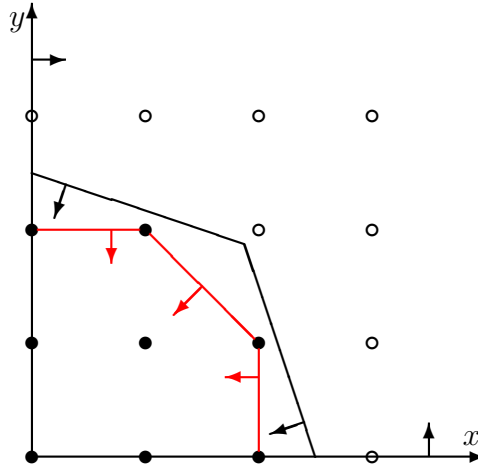


Figura 2: Diferentes formulações. A região do primeiro quadrante delimitada pelas linhas pretas representa P_1 ; A delimitada pelas vermelhas, P_2 .

as restrições de integralidade $x \in \mathbb{Z}^n$. O modelo assim obtido é chamado *relaxação linear*. É de interesse que o poliedro resultante desse descarte seja o mais próximo possível do conjunto de viabilidade original, com as restrições de integralidade. Introduzimos assim o seguinte conceito.

Definição 1.3.2. *Dado um conjunto $X \subset \mathbb{R}^n$, ao menor conjunto convexo que o contém damos o nome de envoltória convexa. Denotamos-o por $\text{conv}(X)$.*

No exemplo inicial, pela Figura 2 vemos que P_2 é a envoltória convexa de X .

Dadas duas formulações P_1 e P_2 de X , dizemos que P_2 é *melhor* que P_1 se $P_2 \subset P_1$. No exemplo, P_2 é melhor que P_1 (veja Figura 2). Pela definição de $\text{conv}(X)$, não há uma formulação de X melhor que $\text{conv}(X)$. Dizemos então que $\text{conv}(X)$ é a *formulação ideal* para X . Parece natural concluir que quanto mais uma formulação se aproxima da envoltória convexa, melhor ela é. Convém portanto estudar a estrutura da envoltória convexa de um conjunto.

O resultado a seguir fornece uma descrição adequada de $\text{conv}(X)$.

Teorema 1.3.3. *Seja $X \subseteq \mathbb{R}^n$ um conjunto. Temos*

$$\text{conv}(X) = \left\{ x; x = \sum_{i=1}^t \lambda_i x^i, \sum_{i=1}^t \lambda_i = 1, \lambda_i \geq 0, \forall i = 1, \dots, t, \right. \\ \left. \text{sobre todos subconjuntos finitos } \{x^1, \dots, x^t\} \text{ de } X \right\}$$

É interessante observar que quando formulamos o conjunto viável de um modelo por sua envoltória convexa, o modelo resultante não deixa de ser de programação linear, como

mostra o resultado a seguir.

Teorema 1.3.4. $\text{conv}(X)$ é um poliedro, isto é, um conjunto da forma $\{x \in \mathbb{R}^n; Ax \leq b\}$.

Definição 1.3.5. Todo ponto $p \in X$ tal que não existem $p_1, p_2 \in X$ distintos que satisfaça $p = \alpha p_1 + (1 - \alpha)p_2$ para todos $\alpha \in (0, 1)$ é denominado ponto extremo ou vértice de X .

Na Figura 2 vemos que os vértices de $P_2 = \text{conv}(X)$ são pontos com coordenadas inteiras, e logo pontos do conjunto X . Isso vale em geral, como mostra o próximo resultado.

Teorema 1.3.6. Um ponto extremo de $\text{conv}(X)$ é um ponto de X .

Essa característica torna $\text{conv}(X)$ interessante na resolução de modelos de PI. Mais adiante veremos o método de planos de corte que se utiliza deste fato.

Finalizamos este capítulo exibindo diferentes formulações para o *problema de localização de facilidades não capacitado*¹, descrito no exemplo a seguir.

Exemplo 1.3.7. Informalmente, dado um conjunto de possíveis fornecedores de um certo produto e clientes cujas localidades estão pré-definidas, o problema de localização de facilidades não capacitado consiste em estabelecer para cada fornecedor a ser instalado quais clientes ele atenderá, de forma a minimizar custos com logística de transporte. Precisamente, seja $N = \{1, \dots, n\}$ o conjunto de fornecedores e $M = \{1, \dots, m\}$ o conjunto de clientes. Suponha que o custo fixo de instalação f_j esteja associado ao fornecedor j , e que o transporte do cliente i para o depósito j tenha custo c_{ij} . Suponhamos que um cliente possa ser atendido por mais de um fornecedor, que sua demanda total seja de uma unidade, e que cada fornecedor instalado tenha capacidade de atender a todos os clientes. O problema recai em decidir qual a parcela do produto que cada fornecedor instalado enviará a cada cliente, com a menor soma do custo de transporte e custo fixo de instalação dos fornecedores.

Definição das variáveis. Para cada $j \in N$ definimos

$$y_j = \begin{cases} 1, & \text{se o fornecedor } j \text{ é selecionado} \\ 0, & \text{caso contrário} \end{cases}$$

e para cada $(i, j) \in M \times N$, denotamos a fração da demanda do cliente i enviada pelo fornecedor j por $x_{ij} \geq 0$.

¹Em inglês, *Uncapacitated Facility Location Problem*.

Definição das restrições. Cada cliente i deve ter sua demanda (unitária) satisfeita, e então exigimos

$$\sum_{j=1}^n x_{ij} = 1, \quad \forall i \in M.$$

Essa restrição, juntamente com o fato de $x_{ij} \geq 0$, garante que $x_{ij} \leq 1$. Com isso x_{ij} realmente representará uma fração da demanda unitária do cliente i .

Para representar a relação entre as variáveis x_{ij} e y_j , notemos que $\sum_{i \in M} x_{ij} \leq m$, e então

$$\sum_{i \in M} x_{ij} \leq m y_j, \quad \forall j \in N. \quad (1.1)$$

Essa restrição diz que se o fornecedor j não for instalado, isto é, se $y_j = 0$, então ele fornece o produto a nenhum cliente. Essa restrição é necessária para uma correta definição das variáveis: sem ela, poderia ocorrer uma solução onde um fornecedor não instalado enviasse produto a algum cliente.

Definição da função objetivo. O objetivo é minimizar o custo total de transporte e instalação

$$\min \sum_{j \in N} \sum_{i \in M} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j.$$

Observe que pela restrição anterior, se o fornecedor j não é instalado a soma $\sum_{i \in M} c_{ij} x_{ij}$ é zero, o que condiz com o fato de não haver custos de transporte para um fornecedor que não existe. Está correta então nossa função objetivo, e o modelo obtido é

$$\begin{aligned} & \min \sum_{j \in N} \sum_{i \in M} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j \\ & \text{s.a.} \quad \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in M \\ & \quad \sum_{i \in M} x_{ij} \leq m y_j, \quad \forall j \in N \\ & \quad y_j \in \{0, 1\}, \quad \forall j \\ & \quad x_{ij} \geq 0, \quad \forall i, j. \end{aligned}$$

Agora, seja P_1 a formulação do conjunto viável obtida do modelo acima quando descartamos restrições de integralidade, isto é, o poliedro da relaxação linear dado por

$$P_1 = \left\{ (x, y) \in \mathbb{R}^{mn+n}; \quad \sum_{j=1}^n x_{ij} = 1, \quad \sum_{i \in M} x_{ij} \leq m y_j, \quad 0 \leq y_j \leq 1, \quad x_{ij} \geq 0, \quad \forall i, j \right\}.$$

Vamos obter uma nova formulação para o conjunto viável $X = P_1 \cap (\mathbb{R}^{mn} \times \mathbb{Z}^n)$. A relação

entre as variáveis $x_{ij}, i \in M$, e y_j dada pela restrição (1.1) pode ser representada pelas m restrições

$$x_{ij} \leq y_j, \quad \forall i \in M.$$

De fato, garantimos que se o fornecedor j não é instalado ($y_j = 0$), nenhum cliente é atendido por ele. Mais ainda, notemos que essas restrições implicam (1.1). Portanto, se P_2 é a formulação obtida de P_1 trocando (1.1) pelas restrições acima (para todo $j \in N$), temos $P_2 \subset P_1$ (P_2 é formulação para X pois $X = P_2 \cap \mathbb{R}^{mn} \times \mathbb{Z}^n$). Mostremos que, em certos casos, $P_2 \subsetneq P_1$. Por simplicidade, suponha que $n < m = kn$ para algum $k \geq 2$, e considere o ponto de P_1 em que cada fornecedor $j \in N = \{1, \dots, n\}$ atenda exclusivamente os k clientes $k(j-1) + 1, \dots, k(j-1) + k$, e onde $y_j = k/m \in [0, 1]$. Assim, cada cliente é atendido por um único fornecedor, e para cada $j \in N$ temos

$$x_{ij} = \begin{cases} 1, & \text{se } i \in \{k(j-1) + 1, \dots, k(j-1) + k\} \\ 0, & \text{caso contrário} \end{cases}.$$

Assim para cada $j \in N$ temos $\sum_{i \in M} x_{ij} = k \leq m \left(\frac{k}{m}\right) = my_j$, mas $x_{ij} = 1 > k/m = y_j$ para um certo índice i , e logo este ponto pertence à $P_1 \setminus P_2$. Isso mostra que P_2 é melhor que P_1 , ou ainda que P_2 se “aproxima mais” de $\text{conv}(X)$ que P_1 . \square

A discussão desta seção coloca uma questão importante da modelagem, foco de pesquisas na área: o quanto a formulação obtida pelo descarte das restrições de integralidade se aproxima da envoltória convexa. No exemplo anterior, podemos esperar que o modelo

$$\begin{aligned} \min & \sum_{j \in N} \sum_{i \in M} c_{ij} x_{ij} + \sum_{j \in N} f_j y_j \\ \text{s.a.} & \sum_{j=1}^n x_{ij} = 1, \quad \forall i \in M \\ & x_{ij} \leq y_j, \quad \forall j \in N, \forall i \in M \\ & y_j \in \{0, 1\}, \quad \forall j \\ & x_{ij} \geq 0, \quad \forall i, j \end{aligned}$$

tenha melhores resultados que o anterior se utilizássemos métodos que trabalham sobre relaxações lineares, como o Método de Planos de Corte apresentado no próximo capítulo.

2 Método de Planos de Corte

2.1 Desigualdades Válidas

Tipicamente, para resolver o modelo

$$\begin{aligned} \min \quad & cx \\ \text{s.a.} \quad & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

utilizamos sua relaxação linear. Logo, é importante relacionar as soluções ótimas deste problema e sua relaxação linear. Denotaremos X como conjunto viável.

Teorema 2.1.1. (BAZARAA; JARVIS, 1977) *Sejam Q um modelo de PI e R sua relaxação linear. Se o conjunto de viabilidade de R for a envoltória convexa do conjunto viável X , então existe uma solução ótima de R com coordenadas inteiras, viável para Q .*

Ou seja, as restrições de integralidade em X podem ser “ignoradas” e uma solução ótima fornecida para esse problema relaxado ainda assim será uma solução inteira.

O resultado fornece, em tese, uma maneira de reescrever um problema de programação inteira em um problema com variáveis contínuas usando a envoltória convexa da região viável, tipicamente mais fácil de resolver (por exemplo via Simplex). Em geral, e particularmente para problemas de difícil resolução, infelizmente não há esperança de se encontrar uma descrição completa de $\text{conv}(X)$. Buscaremos no entanto uma maneira de aproximá-la. Tal aproximação será construída gradativamente inserindo novas restrições lineares que “cortam” a região viável sem descartar pontos de $\text{conv}(X)$. Essa estratégia de inserir novos cortes é possível devido ao Teorema 1.3.4, pois $\text{conv}(X)$ neste caso só possui restrições lineares. Definimos assim o que é uma desigualdade válida.

Definição 2.1.2. *Uma desigualdade $\pi^T x \leq \pi_0$ é válida para $Y \subseteq \mathbb{R}^n$ se $\pi^T x \leq \pi_0$ para todo $x \in Y$.*

A definição permite pensar o seguinte: podemos “cortar” o conjunto de viabilidade X inserindo uma desigualdade válida que não descarta pontos de $\text{conv}(X)$.

Exemplo 2.1.3. Seja $X = \{x \in \mathbb{Z}_+^2 : Ax \leq b\}$ onde

$$A = \begin{pmatrix} -1 & 2 \\ 5 & 1 \\ -2 & -2 \end{pmatrix} \quad \text{e} \quad b = \begin{pmatrix} 4 \\ 20 \\ -7 \end{pmatrix}.$$

Temos

$$X = \left\{ \begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 1 \end{pmatrix}, \begin{pmatrix} 3 \\ 2 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix} \right\} = \{x^1, x^2, \dots, x^6\}$$

(veja Figura 3). Assim,

$$\text{conv}(X) = \left\{ x : x = \sum_{i=1}^6 \lambda_i x^i, \quad \lambda_i \geq 0, \quad i = 1, \dots, 6 \quad \text{e} \quad \sum_{i=1}^6 \lambda_i = 1 \right\}.$$

Como

$$\begin{pmatrix} 3 \\ 2 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 3 \\ 1 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 3 \\ 3 \end{pmatrix}$$

e

$$\begin{pmatrix} 3 \\ 1 \end{pmatrix} = \frac{1}{2} \begin{pmatrix} 2 \\ 2 \end{pmatrix} + \frac{1}{2} \begin{pmatrix} 4 \\ 0 \end{pmatrix},$$

$\text{conv}(X)$ é um poliedro definido pelos quatro pontos extremos

$$\begin{pmatrix} 2 \\ 2 \end{pmatrix}, \begin{pmatrix} 2 \\ 3 \end{pmatrix}, \begin{pmatrix} 3 \\ 3 \end{pmatrix}, \begin{pmatrix} 4 \\ 0 \end{pmatrix}.$$

Neste caso é simples obter uma representação por desigualdades lineares de $\text{conv}(X)$ a partir de linhas definidas por pares de pontos extremos, conforme a Figura 3. Vemos que $\text{conv}(X)$ é definido pelas restrições $A'x \leq b'$, onde

$$A' = \begin{pmatrix} -1 & 0 \\ 0 & 1 \\ -1 & -1 \\ 3 & 1 \end{pmatrix}, \quad b' = \begin{pmatrix} -2 \\ 3 \\ -4 \\ 12 \end{pmatrix}.$$

pelo seguinte resultado,

Teorema 2.1.4. *Uma desigualdade é válida para $\text{conv}(X)$ se, e somente se, é válida para*

X

, uma vez que $\text{conv}(X)$ é o menor conjunto convexo contendo X . Na Figura 3 algumas desigualdades válidas são representadas. \square

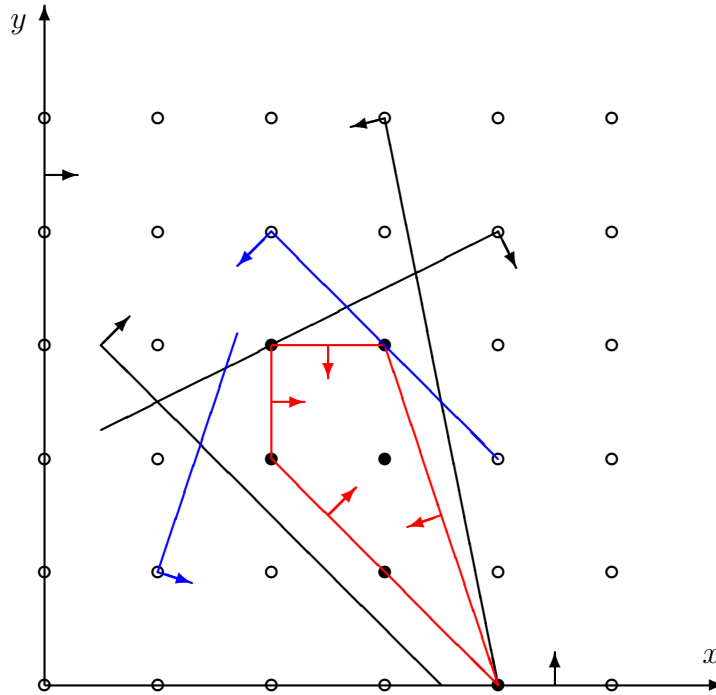


Figura 3: Representação da região de viabilidade do Exemplo 2.1.3. O conjunto X é formado pelos pontos em destaque. Em vermelho $\text{conv}(X)$. Em azul desigualdades válidas.

2.2 O Método de Planos de Corte

O Método de Planos de Corte¹, de acordo com Alves e Delgado, foi o primeiro método desenvolvido para programação inteira, datado de 1958 (ALVES; DELGADO, 1997). É um método exato que busca iterativamente aproximar-se da envoltória convexa da região de viabilidade. O método segue a seguinte estratégia: relaxamos o modelo de PI, obtendo um modelo de PL, de mais fácil resolução. Analisamos uma solução ótima do PL e caso a mesma não seja solução do problema original, inserimos cortes. O Teorema 2.1.1 indica que à medida em que nos aproximamos da envoltória convexa da região viável do modelo de PI, via inserção de cortes no PL, nos aproximamos de uma solução ótima do modelo de PI. O método pára quando uma solução inteira é encontrada, ou seja, uma solução ótima do modelo PI, ou quando não há mais desigualdades a inserir.

¹Em inglês, *cutting plane method*.

No que segue descreveremos o método. Suponha que $X = P \cap \mathbb{Z}^n$ onde $P = \{x; Ax \leq b, x \geq 0\}$ é um poliedro, e que o problema de interesse

$$\begin{aligned} \min \quad & cx \\ \text{s.a.} \quad & Ax \leq b \\ & x \in \mathbb{Z}_+^n \end{aligned}$$

admita solução ótima. Definimos \mathcal{F} como sendo a família das desigualdades válidas para X , ou seja, $\pi^T x \leq \pi_0, \forall x \in X$, para todas as desigualdades $(\pi, \pi_0) \in \mathcal{F}$. A família \mathcal{F} pode conter um número bastante expressivo de elementos (exponencial por exemplo), e logo não podemos introduzir todas as desigualdades na formulação do modelo. Em nosso algoritmo, vamos admitir que exatamente uma desigualdade é inserida por iteração. Evidentemente não estamos interessados em encontrar uma representação completa de $\text{conv}(X)$, até por ser esta uma tarefa difícil, mas apenas uma aproximação em torno de uma solução ótima. Assim, enunciamos o *Método de Planos de Corte*.

Algoritmo 1 Método de Planos de Corte

Inicialização: Defina $k = 0$ e $P^0 = \{x; Ax \leq b, x \geq 0\}$.

Passo principal: Resolva o problema linear

$$\begin{aligned} \min \quad & cx \\ \text{s.a.} \quad & x \in P^k \end{aligned}$$

Seja x^k uma solução ótima. Se $x^k \in \mathbb{Z}^n$, pare pois x^k é uma solução ótima para o modelo de PI. Se $x^k \notin \mathbb{Z}^n$, encontre uma desigualdade $(\pi, \pi_0) \in \mathcal{F}$ tal que $\pi^T x^k > \pi_0$. Se uma desigualdade (π, π_0) foi encontrada, então faça $P^{k+1} = P^k \cap \{x : \pi^T x \leq \pi_0\}$, incremente k e repita este passo. Caso contrário, pare.

2.2.1 Desigualdades válidas em Programação Inteira

Tendo em vista o Método de Planos de Corte, é de nosso interesse estabelecer uma maneira geral de gerar desigualdades válidas para um problema de programação inteira. O procedimento de Chvátal-Gomory (CG) que apresentaremos nesta seção caminha neste sentido. Nosso objetivo é, a partir das restrições do problema, obter desigualdades válidas que efetivamente “cortam” a região viável da relaxação linear sem descartar ótimos inteiros.

O resultado a seguir, cuja prova é trivial, será útil.

Proposição 2.2.1. A desigualdade $y \leq \lfloor b \rfloor$ é válida para $\{y \in \mathbb{Z} : y \leq b\}$.

Consideramos novamente o conjunto $X = P \cap \mathbb{Z}^n$ onde

- (i) $P = \{x \in \mathbb{R}^n : Ax \leq b, x \geq 0\} \neq \emptyset$ é um poliedro, e
- (ii) $A \in \mathbb{R}^{m \times n}$ é uma matriz com colunas $\{a_1, a_2, \dots, a_n\}$.

Reescrevemos as restrições $Ax \leq b$ na forma

$$\sum_{j=1}^n a_j x_j \leq b.$$

Dado $u \in \mathbb{R}_+^m$ qualquer, a desigualdade

$$\sum_{j=1}^n u^T a_j x_j \leq u^T b$$

é válida para X uma vez que $u \geq 0$. Como $x \geq 0$ temos $\lfloor u^T a_j \rfloor x_j \leq u^T a_j x_j$, e sabendo que $x \in \mathbb{Z}^n$, a Proposição 2.2.1 garante que a desigualdade

$$\sum_{j=1}^n \lfloor u^T a_j \rfloor x_j \leq \lfloor u^T b \rfloor \tag{2.1}$$

é válida para X .

A expressão (2.1) é chamada *corte de Chvátal-Gomory*. O processo visto acima de geração dessas desigualdades é conhecido como *procedimento de Chvátal-Gomory*. Sucessivas aplicações desse procedimento geram desigualdades que não são válidas para P em geral, como mostra o próximo exemplo.

Exemplo 2.2.2. Seja o modelo

$$\begin{aligned} \text{PI: } \min & -5x_1 + 2x_2 \\ \text{s.a. } & 7x_1 - 5x_2 \leq 13 \\ & 3x_1 + 2x_2 \leq 17 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{Z}^2 \end{aligned}$$

Relaxando o problema obtemos o modelo

$$\begin{aligned} \text{PL: } \min \quad & -5x_1 + 2x_2 \\ \text{s.a. } \quad & 7x_1 - 5x_2 \leq 13 \\ & 3x_1 + 2x_2 \leq 17 \\ & x_1, x_2 \geq 0 \\ & x_1, x_2 \in \mathbb{R}^2 \end{aligned}$$

cuja região de viabilidade é representada na Figura 4.

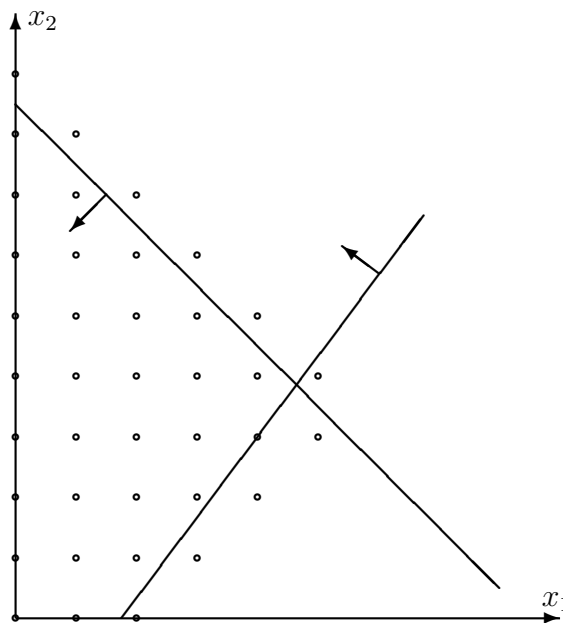


Figura 4: Representação da região de viabilidade do Exemplo 2.2.2.

Utilizamos o resultado anterior para encontrarmos uma desigualdade válida para o problema. Escolhendo $u = (\frac{3}{20}, 0)$ temos,

$$\begin{aligned} u^T Ax &= \begin{bmatrix} \frac{3}{20} & 0 \end{bmatrix} \begin{bmatrix} 7 & -5 \\ 3 & 2 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \leq \begin{bmatrix} \frac{3}{20} & 0 \end{bmatrix} \begin{bmatrix} 13 \\ 17 \end{bmatrix} = u^T b \\ \Rightarrow \left(\frac{21}{20} - 0\right) x_1 + \left(-\frac{15}{20} + 0\right) x_2 &\leq \frac{39}{20} \\ \Rightarrow \lfloor 21/20 \rfloor x_1 + \lfloor -15/20 \rfloor x_2 &\leq \lfloor 39/20 \rfloor. \end{aligned}$$

Obtemos assim seguinte desigualdade válida

$$x_1 - x_2 \leq 1. \quad (2.2)$$

Inserimos agora a desigualdade válida acima no modelo relaxado PL conforme Figura

5.

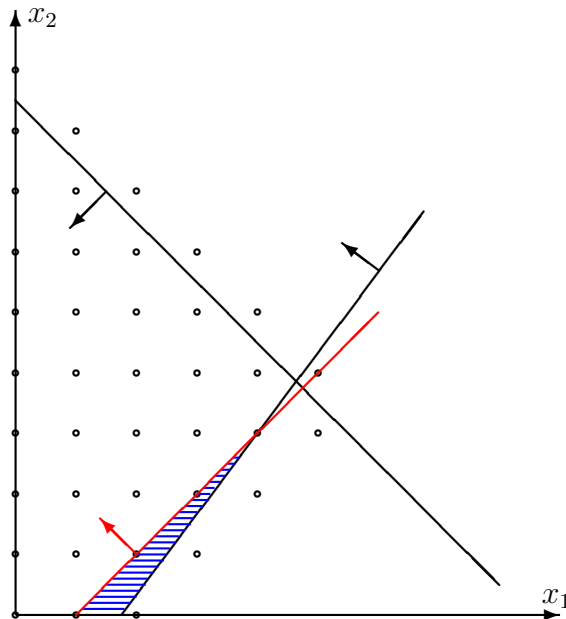


Figura 5: Representação do corte na região de viabilidade do Exemplo 2.2.2. A região em destaque foi descartada pelo corte.

Nota-se que inserindo a desigualdade válida (2.2) descartamos pontos que não pertencem ao conjunto de viabilidade de PI (região em destaque), mas que são válidos para o PL. \square

Isso significa que usando (2.1) para um problema de programação inteira com região viável X , podemos efetivamente cortar a região P de sua relaxação linear, sem descartar os pontos viáveis em X . Mais que isso, de maneira surpreendente tal procedimento é capaz de gerar todas as desigualdades válidas para um problema de PI, no sentido do resultado a seguir.

Teorema 2.2.3 (WOLSEY, 1998). *Toda desigualdade válida para X pode ser obtida por meio da aplicação de um número finito de vezes do procedimento de Chvátal-Gomory sobre as restrições em X e as próprias restrições geradas pelo procedimento.*²

Neste ponto podemos dizer, pelo menos teoricamente, que o método de planos de corte encontra uma solução ótima em finitos passos quando utilizamos o procedimento de Chvátal-Gomory e o método Simplex para resolução dos problemas lineares: primeiro, os Teoremas 1.3.4 (página 16) e 2.2.3 garantem que a envoltória convexa pode ser obtida

²O resultado só é garantido para poliedros racionais, isto é, aqueles onde as matrizes que o definem têm somente entradas racionais. Tal hipótese foi feita no início do Capítulo 1 e vale para todo o trabalho.

por finitas aplicações do procedimento de Chvátal-Gomory; segundo, os Teoremas 1.3.6 (página 16) e 2.1.1 (página 19) garantem que o vértice encontrado pelo método Simplex é uma solução ótima do modelo de programação inteira. Infelizmente, na prática pode ser necessário um número muito grande de cortes para tanto. É importante tentar estabelecer desigualdades mais fortes para um dado problema, que em geral dependem da estrutura específica do modelo tratado. De fato, são conhecidos cortes específicos para certos problemas, implementados nos *softwares* de programação linear de maior eficiência, como o CPLEX.

2.2.2 Cortes via Quadro Simplex

No Método de Planos de Corte, devemos resolver problemas de programação linear. Sempre que uma variável é não inteira, devemos ser capazes de gerar e adicionar cortes de Chvátal-Gomory no Quadro Simplex (QS). Apresentamos aqui uma maneira de fazer isso. Serão necessários conhecimentos básicos sobre o método Simplex, presentes no Apêndice A.

Supomos que a relaxação linear do modelo de PI foi resolvida. Portanto temos em mãos o quadro simplex ótimo. Como as variáveis não básicas (VNB) são zero, somente variáveis básicas (VB) podem ser não inteiras. Supomos então que a VB x_{B_r} não seja inteira. A linha de x_{B_r} no QS é

$$x_{B_r} + \sum_{j \in R} y_{rj} x_j = \bar{b}_r \quad (2.3)$$

onde R é o conjunto de índices das VNB's. Temos assim $\bar{b}_r \notin \mathbb{Z}$. Usaremos o conceito de *parte fracionária* de um escalar z , definida como $f_z = z - \lfloor z \rfloor$. Podemos notar facilmente que $0 \leq f_z < 1$. Pois bem, consideremos as partes fracionárias de y_{rj} e de b_r , respectivamente:

$$f_{rj} = y_{rj} - \lfloor y_{rj} \rfloor \quad \text{e} \quad f_r = \bar{b}_r - \lfloor \bar{b}_r \rfloor.$$

Da equação (2.3) segue portanto que

$$\begin{aligned} x_{B_r} + \sum_{j \in R} (f_{rj} + \lfloor y_{rj} \rfloor) x_j &= f_r + \lfloor \bar{b}_r \rfloor \\ \Rightarrow x_{B_r} + \sum_{j \in R} \lfloor y_{rj} \rfloor x_j - \lfloor \bar{b}_r \rfloor &= f_r - \sum_{j \in R} f_{rj} x_j. \end{aligned}$$

Nosso intuito é obter soluções inteiras. Observe que o membro esquerdo da última equação

é inteiro se a solução é inteira. Obrigamos-o então a ser inteiro, e portanto queremos que

$$f_r - \sum_{j \in R} f_{rj} x_j \in \mathbb{Z}.$$

Agora, afirmamos que $f_r - \sum_{j \in R} f_{rj} x_j < 1$. De fato, se fosse o contrário teríamos

$$f_r \geq 1 + \sum_{j \in R} f_{rj} x_j \geq 1$$

pois $f_{rj}, x_j \geq 0$ para todo $j \in R$, contrariando o fato de ser $f_r < 1$. Assim, juntamente com a imposição de integralidade que fizemos, devemos satisfazer a restrição

$$f_r - \sum_{j \in R} f_{rj} x_j \leq 0.$$

Essa restrição é conhecida como *corte de Gomory*, e é ela que utilizamos no método de Planos de Corte. A maneira com que a adicionamos no QS é simples: inserimos uma nova variável de folga $s \geq 0$, obtendo a equação

$$- \sum_{j \in R} f_{rj} x_j + s = -f_r.$$

Encaramos s como VB. Inserimos portanto uma nova coluna e uma nova linha no quadro simplex, relativas à s , obtendo o novo QS

	z	x_B	\cdots	x_j	\cdots	s	RHS
z	1	0	\cdots	\cdots	\cdots	0	\cdots
x_B	0	I	\cdots	\cdots	\cdots	0	$B^{-1}b$
s	0	0	\cdots	$-f_{rj}$	\cdots	1	$-f_r$

Esse novo QS não é primal viável pois, como $\bar{b}_r \notin \mathbb{Z}$, temos $f_r = \bar{b}_r - \lfloor \bar{b}_r \rfloor > 0$ e logo $-f_r < 0$. Mas é dual viável, e assim utilizamos o método dual Simplex para restabelecer viabilidade. Se após isso ainda existirem variáveis básicas não inteiras, o processo é repetido com a geração de um novo corte.

Exemplo 2.2.4. Vamos ilustrar a aplicação do método de planos de corte no modelo de

programação inteira

$$\begin{aligned}
 \text{PI: min } & 3x_1 + 4x_2 \\
 \text{s.a. } & -3x_1 - x_2 \leq -4 \\
 & -x_1 - 2x_2 \leq -4 \\
 & x_1, x_2 \geq 0 \\
 & x_1, x_2 \in \mathbb{Z}
 \end{aligned}$$

Relaxando o problema obtemos o modelo

$$\begin{aligned}
 \text{PL: min } & 3x_1 + 4x_2 \\
 \text{s.a. } & -3x_1 - x_2 \leq -4 \\
 & -x_1 - 2x_2 \leq -4 \\
 & x_1, x_2 \geq 0.
 \end{aligned}$$

Resolvendo a relaxação PL via simplex obtemos o seguinte quadro (QS) ótimo:

	z	x_1	x_2	x_3	x_4	RHS
z	1	0	0	$-2/5$	$-9/5$	$44/5$
x_1	0	1	0	$-2/5$	$1/5$	$4/5$
x_2	0	0	1	$1/5$	$-3/5$	$8/5$

Nota-se que a solução ótima $(x_1, x_2) = (4/5, 8/5)$ não é inteira. Assim, devemos gerar um corte de Gomory. Como $x_2 = \bar{b}_2 = 8/5 \notin \mathbb{Z}$, escolhemos-o para gerar a restrição. A equação da VB x_2 é

$$x_2 + 1/5x_3 - 3/5x_4 = 8/5.$$

Temos $R = \{3, 4\}$ e

$$\begin{aligned}
 f_{23} &= y_{23} - \lfloor y_{23} \rfloor = 1/5 - \lfloor 1/5 \rfloor = 1/5 - 0 = 1/5, \\
 f_{24} &= y_{24} - \lfloor y_{24} \rfloor = -3/5 - \lfloor -3/5 \rfloor = -3/5 - (-1) = 2/5, \\
 f_2 &= \bar{b}_2 - \lfloor \bar{b}_2 \rfloor = 8/5 - \lfloor 8/5 \rfloor = 8/5 - 1 = 3/5.
 \end{aligned}$$

Logo o corte de Gomory gerado é

$$3/5 - 1/5x_3 - 2/5x_4 \leq 0.$$

Inserimos uma nova variável básica s obtendo a equação

$$-1/5x_3 - 2/5x_4 + s = -3/5.$$

O quadro com a essa nova equação é

	z	x_1	x_2	x_3	x_4	s	RHS
z	1	0	0	$-2/5$	$-9/5$	0	$44/5$
x_1	0	1	0	$-2/5$	$1/5$	0	$4/5$
x_2	0	0	1	$1/5$	$-3/5$	0	$8/5$
s	0	0	0	$-1/5$	$-2/5$	1	$3/5$

Percebe-se que o QS é dual viável, mas não primal viável. Aplicando o dual simplex sobre o pivô y_{53} obtemos o QS ótimo do problema relaxado PL com o corte de Gomory gerado.

	z	x_1	x_2	x_3	x_4	s	RHS
z	1	0	0	0	-1	-2	10
x_1	0	1	0	0	1	-2	2
x_2	0	0	1	0	-1	1	1
x_3	0	0	0	1	2	-5	3

Como a solução encontrada é $(x_1, x_2) = (2, 1)$ inteira, ela é também uma solução ótima do problema original PI. □

3 Métodos Enumerativos

Considere o modelo de programação inteira

$$\min f(x_1, x_2) = -2x_1 - x_2 \tag{3.1}$$

$$\text{s.a. } x_1 + x_2 \leq 5 \tag{3.2}$$

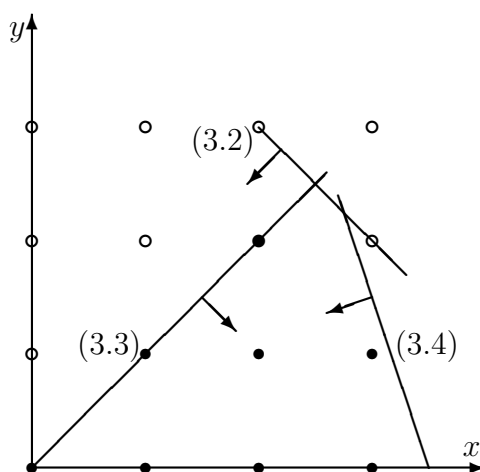
$$-x_1 + x_2 \leq 0 \tag{3.3}$$

$$6x_1 + 2x_2 \leq 21 \tag{3.4}$$

$$0 \leq x_1, x_2 \tag{3.5}$$

$$x_1, x_2 \in \mathbb{Z} \tag{3.6}$$

cuja representação da região viável é dada na figura abaixo.



Nosso objetivo é estudar as soluções desse problema enumerando-as. Vejamos duas possíveis formas de se fazer isso:

1ª Alternativa: Primeiramente,

$$(3.2) \text{ e } (3.5) \Rightarrow 0 \leq x_1, x_2 \leq 5, \tag{3.7}$$

$$(3.7) \text{ e } (3.4) \Rightarrow x_1 \leq 3, \tag{3.8}$$

$$(3.8) \text{ e } (3.3) \Rightarrow x_2 \leq 3. \tag{3.9}$$

Segue portanto que $0 \leq x_1, x_2 \leq 3$. Assim, temos 16 possibilidades para $(x_1, x_2) \in \mathbb{Z}^2$, e destas, enumeramos as oito viáveis: $(0, 0)$, $(1, 0)$, $(1, 1)$, $(2, 0)$, $(2, 1)$, $(2, 2)$, $(3, 0)$ e $(3, 1)$.

Analisando f em cada ponto viável, obtemos o ótimo $(3, 1)^*$.

2ª Alternativa: Temos

$$(3.2), (3.3) \text{ e } (3.6) \Rightarrow x_2 \leq \frac{5}{2}, \quad \text{e} \quad x_2 \in \mathbb{Z} \Rightarrow x_2 \leq \left\lfloor \frac{5}{2} \right\rfloor = 2; \quad (3.10)$$

$$f(3, 0) = -6 \Rightarrow f(x_1, x_2) = -2x_1 - x_2 \leq -6 \Rightarrow 6 \leq 2x_1 + x_2; \quad (3.11)$$

$$(3.10) \text{ e } (3.11) \Rightarrow 2x_1 \geq 6 - x_2 \geq 4 \Rightarrow x_1 \geq 2. \quad (3.12)$$

Reduzimos os pontos da Alternativa 1 para apenas 3, a saber, $(2, 2)$, $(3, 0)$ e $(3, 1)$. Por exaustão, obtemos o ótimo $(3, 1)^*$.

Concluimos do exemplo o seguinte:

- Resolvemos o problema de PI enumerando seus pontos viáveis em ambas as alternativas adotadas;
- Na alternativa 2, não testamos f em todos pontos viáveis da alternativa 1. No entanto garantimos que $(3, 1)$ é o ótimo estabelecendo condições que descartavam *a priori* pontos viáveis desinteressantes. Dizemos que tais pontos foram *enumerados implicitamente*;
- Com isso, a alternativa 2 é mais atrativa pois simplifica a busca pelo ótimo.

Essa ideia de enumeração implícita é central nos métodos aqui apresentados. Neste capítulo apresentamos dois métodos que enumeram implicitamente boa parte dos pontos viáveis de um problema de PI.

3.1 Branch and Bound

O método *Branch and Bound* é um método enumerativo que busca a solução de um modelo de PI enumerando os pontos inteiros de sua região viável, desde que haja uma quantidade finita deles. Deste modo, uma solução ótima é encontrada caso exista. Naturalmente, vários desses pontos são enumerados implicitamente.

É comum representarmos a aplicação do método sob uma árvore. Cada nó representa o subconjunto da região viável obtido pela imposição de certas condições sobre uma ou mais variáveis (ramificação). O exemplo a seguir ilustra a enumeração dos pontos do conjunto

das ternas de entradas binárias, cujas condições impostas são fixações das variáveis em 0 ou 1.

Exemplo 3.1.1. Seja $P = \{0, 1\}^3$.

As seguintes subdivisões são realizadas:

- P em $P_0 = \{x \in P; x_1 = 0\}$ e $P_1 = \{x \in P; x_1 = 1\}$
- P_0 em $P_{00} = \{x \in P; x_1 = x_2 = 0\}$ e $P_{01} = \{x \in P_0; x_2 = 1\}$
- Prosseguimos dividindo P_1 , P_{00} e P_{01} até que nos nós terminais da árvore todas as variáveis estejam fixadas.

Representamos a árvore de enumeração na Figura 6. □

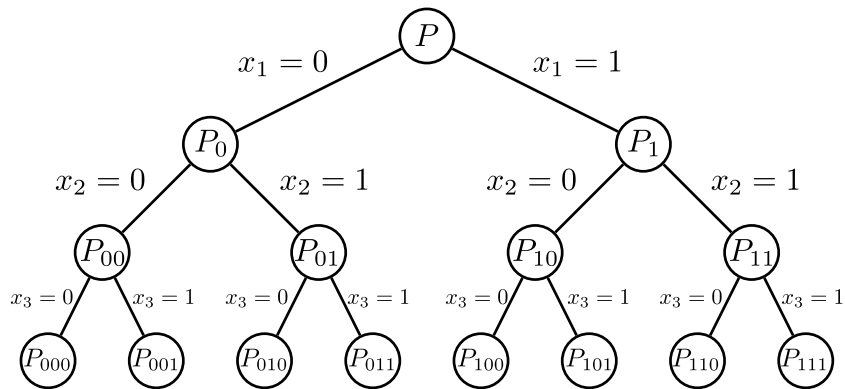


Figura 6: Figura do Exemplo 3.1.1. Árvore de enumeração para $P = \{0, 1\}^3$.

No exemplo acima, vemos que $P = \bigcup_{i,j,k \in \{0,1\}} P_{ijk}$, onde $P_{ijk} = \{i, j, k\}$. Podemos notar que o número de subdivisões finais, os nós terminais, cresce exponencialmente. Aí reside a utilidade da enumeração implícita: sem ela, deveríamos enumerar explicitamente todas as possibilidades, o que seria impraticável. Para ilustrar, o conjunto $\{0, 1\}^{30}$ têm $2^{30} = 1.073.741.824$ elementos.

Em geral, particionamos o poliedro

$$P = \{x \in \mathbb{Z}_+^n; Ax \leq b\}$$

em

$$P = P_1 \cup P_2 \cup \dots \cup P_m.$$

No método apresentado, como o próprio nome diz, limitantes na função objetivo serão utilizados para reduzir a enumeração, ou seja, enumerar pontos implicitamente. Observe que essa ideia foi utilizada na alternativa 2 no início do capítulo.

O seguinte resultado é fundamental para o estabelecimento do método *Branch and Bound*.

Proposição 3.1.2. *Seja P um poliedro e $P = P_1 \cup P_2 \cup \dots \cup P_k$ uma decomposição sua. Para cada k , sejam $z^k = \min\{cx; x \in P_k\}$, \bar{z}^k e \underline{z}^k limitantes superior e inferior para z^k , respectivamente. Então $\bar{z} = \min_k \bar{z}^k$ e $\underline{z} = \min_k \underline{z}^k$ são limitantes superior e inferior para $z = \min\{cx; x \in P\}$, respectivamente.*

Demonstração. Para cada k , como $P_k \subset P$, temos $z \leq z^k$. Logo $z \leq \min_k z^k$. Vale trivialmente a igualdade, já que se existe um ótimo, algum encontra-se em um dos conjuntos que compõem a partição de P . Desta forma,

$$\underline{z}^k \leq z^k \leq \bar{z}^k, \forall k \quad \Rightarrow \quad \underline{z} = \min_k \underline{z}^k \leq \min_k z^k = z \leq \min_k \bar{z}^k = \bar{z}.$$

□

Observe que no Exemplo 3.1.1 inserimos uma condição $x_i = 0$ ou $x_i = 1$ na medida em que ramificamos. A estratégia do método *branch and bound* procura, a partir de um nó da árvore, ramificar inserindo uma desigualdade válida para o problema. Existem várias formas de ramificação, ou seja, da escolha dessas desigualdades. Discutiremos neste trabalho a forma mais comum: ramificação sobre variáveis fracionárias.

Pois bem, estamos interessados em resolver o modelo

$$\begin{aligned} PI : \min \quad & cx \\ \text{s.a.} \quad & x \in X \end{aligned}$$

onde $X = P \cap \mathbb{Z}^n$, $P = \{x \in \mathbb{R}_+^n; Ax \leq b\}$. Para maior clareza, supomos que sua relaxação linear admita solução ótima.

Começemos pelo primeiro nó da árvore, a *raiz*. Neste nó associamos a relaxação linear de PI. Resolvendo-a, obtemos uma solução x^0 . Se $x^0 \in \mathbb{Z}^n$ o modelo PI está resolvido. Caso contrário, se algum $x_j^0 \notin \mathbb{Z}$, o ponto x^0 não é solução ótima de PI, e devemos ramificar. Como queremos $x_j \in \mathbb{Z}$, pelo menos uma das desigualdades seguintes são válidas para P :

$$x_j \leq \lfloor x_j^0 \rfloor \quad \text{ou} \quad x_j \geq \lceil x_j^0 \rceil.$$

Ramificamos então o nó raiz gerando dois novos subproblemas, com regiões viáveis

$$P_1 = P \cap \{x; x_j \leq \lfloor x_j^0 \rfloor\} \quad \text{e} \quad P_2 = P \cap \{x; x_j \geq \lceil x_j^0 \rceil\}.$$

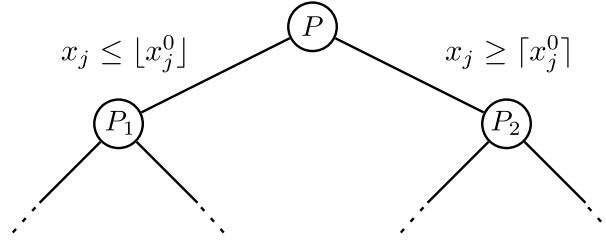


Figura 7: Ramificação sobre variáveis fracionárias.

Criamos dois novos nós P_1 e P_2 na árvore de enumeração, como ilustrado na Figura 7. Com isso, analisamos novamente a solução do problema relativo ao poliedro de cada nó, e verificamos se todas as coordenadas do ponto encontrado são inteiras. Caso não forem, fazemos novamente o processo de ramificação sobre esses nós, até que se encontre ponto ótimo do problema, ou até concluir que o ramo da árvore não produza pontos viáveis (podemos recair em um conjunto vazio). Se o problema original for limitado, há garantia de que esta estratégia gerará uma árvore finita.

Para uma enumeração eficiente, podemos indicar pelo menos três razões que nos permitem podar ramos da árvore de enumeração, e assim enumerar um grande número de soluções implicitamente. São elas:

- *Poda por otimalidade:* Se $\underline{z}^k = \bar{z}^k$ o problema $z^k = \min\{cx; x \in P_k\}$ foi resolvido e uma solução inteira foi obtida. Assim, nenhuma solução melhor será encontrada no ramo e portanto não há motivo para ramificar;
- *Poda por limitante:* Se $\underline{z}^k \geq \bar{z}$ então os subproblemas dos nós filhos não gerarão soluções melhores que a corrente. Assim, não há motivo para ramificar;
- *Poda por inviabilidade:* Se $P_k = \emptyset$, qualquer subproblema do nó será inviável. Evidentemente, não precisamos ramificar.

Observe que os dois primeiros motivos utilizam limitantes, fazendo jus ao nome do método. Os limites superiores de cada nó \bar{z}^k são fornecidos por pontos viáveis (inteiros) x^k pois se x^* é ótimo então $cx^* \leq cx^k$. Neste caso, o limitante superior global $\bar{z} = \min_k \bar{z}^k$ é atualizado. Já os limitantes inferiores \underline{z}^k são fornecidos por relaxação ou pelo problema dual (veja Teorema B.0.10), e também o limitante inferior global é atualizado.

O leitor poderia perguntar se a condição $\bar{z}^k \leq \underline{z}$ não seria critério para podar por limitante. De fato, neste caso não há motivos para ramificar. O fato é que se isso ocorre então $\underline{z} \leq \underline{z}^k \leq \bar{z}^k \leq \underline{z}$, e o ramo é podado por otimalidade.

Para melhor ilustrar a forma de corte da árvore de enumeração, exibimos exemplos com árvores típicas.

Exemplo 3.1.3. Corte por otimalidade

A figura abaixo mostra um conjunto P e subconjuntos P_1 e P_2 de P . Limitantes superiores e inferiores são mostrados em cada nó da árvore. Em particular, os limitantes do nó raiz são os globais.

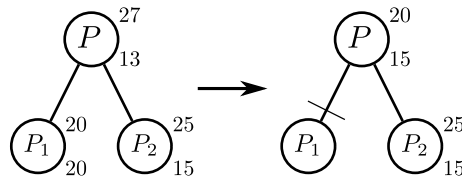


Figura 8: Corte por otimalidade.

Observamos que os limitantes superior e inferior em P_1 são iguais a $z_1 = 20$, e logo podemos o ramo. Os limitantes globais são atualizados para

$$\bar{z} = \min_k \bar{z}^k = \min\{20, 25\} = 20$$

$$\text{e } \underline{z} = \min_k \underline{z}^k = \min\{20, 15\} = 15.$$

□

Exemplo 3.1.4. Corte por limitante

Na Figura 9 são indicados o conjunto $P = P_1 \cup P_2$ e os limitantes em cada nó.

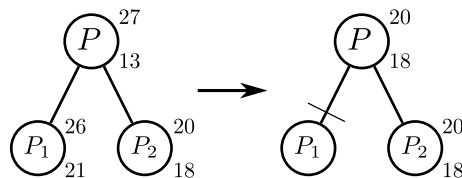


Figura 9: Corte por limitantes.

Notemos que $\bar{z} = \min_k \bar{z}^k = \min\{20, 26\} = 20$ e $\underline{z} = \min_k \underline{z}^k = \min\{18, 21\} = 18$. O ramo do nó P_1 é cortado, visto que $\underline{z}^1 = 21 \geq 20 = \bar{z}$. Isto significa que as possíveis soluções geradas no ramo de P_1 têm valor fora do intervalo $[18, 20]$ onde encontra-se o valor ótimo.

□

Exemplo 3.1.5. Na Figura 10 o limitante inferior do nó P_2 não foi calculado.

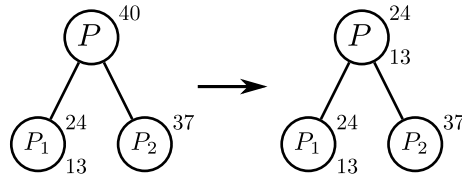


Figura 10: Nenhum corte é possível.

Notemos que $\bar{z} = \min_k \bar{z}^k = \min\{24, 37\} = 24$ e $\underline{z} = \min_k \underline{z}^k = \min\{13\} = 13$. Nenhum corte é possível, já que há possibilidade, em cada nó, de encontrar ótimos. \square

3.1.1 O método *Branch and Bound* - limitantes via relaxação linear

É comum na resolução de problemas de (PI) via *Branch-and-Bound* que os limitantes inferiores sejam obtidos via relaxação linear, como já comentamos. Essas relaxações são problemas de programação linear e comumente utilizamos o método Simplex para resolvê-los. Devemos ser capazes, portanto, de interpretar a inserção das restrições $x_i \leq \lfloor x_i^* \rfloor$ e $x_i \geq \lceil x_i^* \rceil$, típicas do método *Branch and Bound*, no Quadro Simplex (QS) da relaxação.

Como as variáveis não básicas (VNB) são zero, a ramificação ocorre sempre sobre variáveis básicas (VB). Supomos então que estamos de posse de uma solução x^* cuja VB $x_{B_r}^*$ não seja inteira. A linha de x_{B_r} no QS é

$$x_{B_r} + \sum_{j \in R} y_{rj} x_j = \bar{b}_r. \quad (3.13)$$

Assim, as restrições geradoras dos ramos serão

$$\bar{b}_r - \sum_{j \in R} y_{rj} x_j \leq \lfloor x_{B_r}^* \rfloor \quad \text{e} \quad \bar{b}_r - \sum_{j \in R} y_{rj} x_j \geq \lceil x_{B_r}^* \rceil.$$

Para adicionar tais restrições no QS, inserimos a variável de folga $s \geq 0$ e a tratamos como VB, de maneira inteiramente análoga à subseção 2.2.2.

Para ilustrar, vejamos o exemplo a seguir.

Exemplo 3.1.6.

$$\begin{aligned} \min z &= -4x_1 + x_2 \\ \text{s.a. } 7x_1 - 2x_2 &\leq 14 \\ x_2 &\leq 3 \\ 2x_1 - 2x_2 &\leq 3 \\ x &\in \mathbb{Z}_+^2. \end{aligned}$$

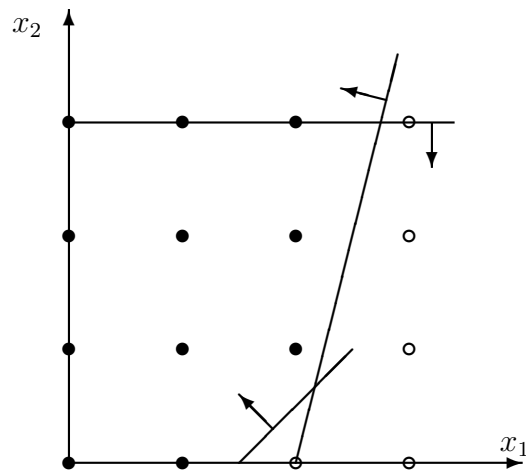


Figura 11: Representação da região de viabilidade do Exemplo 3.1.6.

Adequamos o problema para uso do Simplex adicionando variáveis de folga x_3, x_4 e x_5 . Além disso, descartamos restrições de integralidade, resultando no PL

$$\begin{aligned} \min z &= -4x_1 + x_2 \\ \text{s.a. } 7x_1 - 2x_2 + x_3 &= 14 \\ x_2 + x_4 &= 3 \\ 2x_1 - 2x_2 + x_5 &= 3 \\ x &\geq 0. \end{aligned}$$

Abaixo segue a resolução do deste problema via método Simplex.

	z	x_1	x_2	x_3	x_4	x_5	RHS
z	1	4	-1	0	0	0	0
x_3	0	7	-2	1	0	0	14
x_4	0	0	1	0	1	0	3
x_5	0	2	-2	0	0	1	3

	z	x_1	x_2	x_3	x_4	x_5	RHS
z	1	0	3	0	0	-2	-6
x_3	0	0	5	1	0	-7/2	7/2
x_4	0	0	1	0	1	0	3
x_1	0	1	-1	0	0	1/2	3/2

	z	x_1	x_2	x_3	x_4	x_5	RHS
z	1	0	0	$-3/5$	0	$1/10$	$-81/10$
x_2	0	0	5	$1/5$	0	$-7/10$	$7/10$
x_4	0	0	0	$-1/5$	1	$7/10$	$23/10$
x_1	0	1	0	$1/5$	0	$-1/5$	$11/5$

	z	x_1	x_2	x_3	x_4	x_5	RHS
z	1	0	0	$-4/7$	$-1/7$	0	$-59/7$
x_2	0	0	1	0	1	0	3
x_5	0	0	0	$-2/7$	$10/7$	1	$23/7$
x_1	0	1	0	$1/7$	$2/7$	0	$20/7$

O último QS fornece o limitante inferior $\underline{z} = -59/7$, e o ponto não inteiro $(x_1, x_2) = (20/7, 3)$. Como nenhuma solução viável inteira foi encontrada, tomamos $\underline{z} = -\infty$.

Na solução do PL atual, $x_1 = 20/7 \notin \mathbb{Z}$. Assim, ramificamos gerando os ramos

$$P_1 = P \cap \{x; x_1 \leq \lfloor 20/7 \rfloor = 2\} \quad \text{e} \quad P_2 = P \cap \{x; x_1 \geq \lceil 20/7 \rceil = 3\}.$$

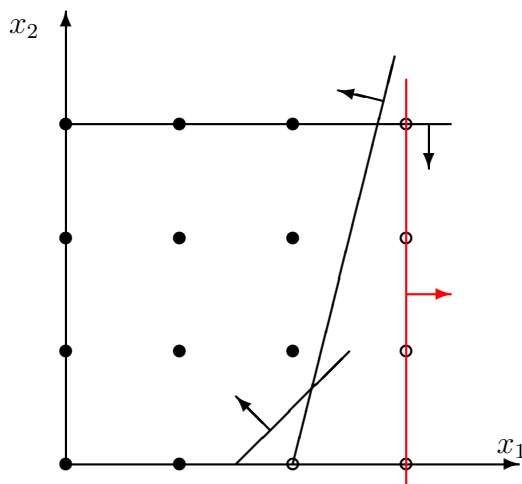


Figura 12: $P_2 = \emptyset$: corte por inviabilidade 3.1.6.

Graficamente vemos que $P_2 = \emptyset$, e logo cortamos o ramo de P_2 por inviabilidade (veja Figura 12). Resta avaliarmos P_1 , com a restrição adicional $x_1 \leq 2$ (veja Figura 13). A linha da VB x_1 no QS é

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

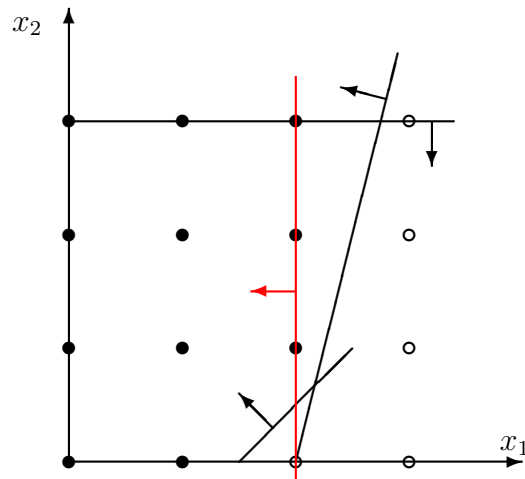


Figura 13: Representação gráfica de P_1 no Exemplo 3.1.6.

O corte $x_1 \leq 2$ é reescrito como

$$\frac{20}{7} - \frac{1}{7}x_3 - \frac{2}{7}x_4 \leq 2.$$

Inserindo a variável de folga $s \geq 0$, devemos adicionar a restrição

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

no QS, tratando s como VB. Obtemos assim o quadro

	z	x_1	x_2	x_3	x_4	x_5	s	RHS
z	1	0	0	$-4/7$	$-1/7$	0	0	$-59/7$
x_2	0	0	1	0	1	0	0	3
x_5	0	0	0	$-2/7$	$10/7$	1	0	$23/7$
x_1	0	1	0	$1/7$	$2/7$	0	0	$20/7$
s	0	0	0	$-1/7$	$-2/7$	0	1	$-6/7$

Observamos que este QS não é primal viável pois $s < 0$. Aplicamos o método Simplex Dual para obter o QS ótimo

	z	x_1	x_2	x_3	x_4	x_5	s	RHS
z	1	0	0	0	0	$-1/2$	-3	$-15/2$
x_1	0	1	0	0	0	0	1	2
x_2	0	0	1	0	0	$-1/2$	1	$1/2$
x_3	0	0	0	1	0	-1	-5	1
x_4	0	0	0	0	1	$1/2$	6	$5/2$

Atualizamos o limitante inferior para $z = \min\{-15/2\} = -15/2$. A solução encontrada $(x_1, x_2) = (2, 1/2)$ ainda é não inteira. Devemos ramificar o nó P_1 em

$$P_{11} = P_1 \cap \{x; x_2 \leq \lfloor 1/2 \rfloor = 0\},$$

$$P_{12} = P_1 \cap \{x; x_2 \geq \lceil 1/2 \rceil = 1\}.$$

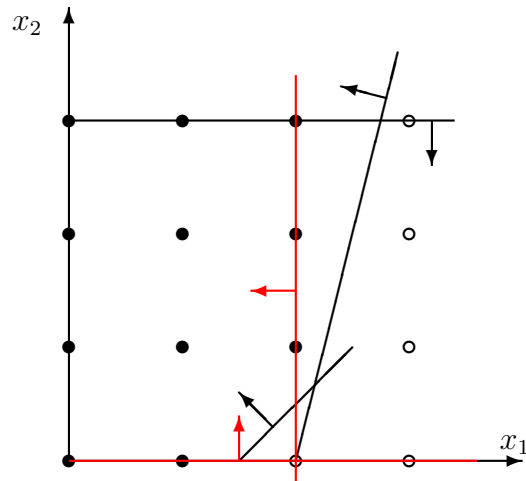


Figura 14: Representação gráfica de P_{11} no Exemplo 3.1.6.

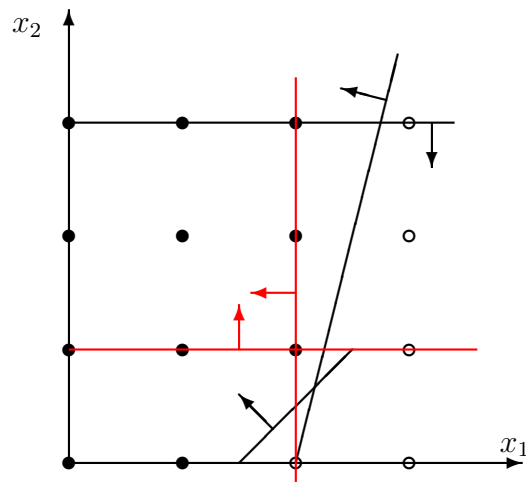


Figura 15: Representação gráfica de P_{12} no Exemplo 3.1.6.

Resolvendo novamente usando o Método Simplex, encontramos

$$x_{12}^* = (2, 1) \in \mathbb{Z}^2$$

com valor -7 , que é uma solução ótima do problema original. □

3.2 Branch and Cut

Sabemos que aproximar a região viável da envoltória convexa é interessante, e foi usada no método de Planos de Corte. Ainda que, em geral, não consigamos encontrar todas as desigualdades que definem a envoltória convexa em tempo polinomial, é uma estratégia muito útil na prática.

No método *Branch and Bound* podemos, em cada nó, adicionar cortes afim de obter soluções fracionárias mais próximas de uma solução inteira do problema. Ou seja, aproximar a região viável da envoltória convexa tende a melhorar o valor da relaxação linear nos nós, que por sua vez tende a antecipar os cortes por limitantes. Assim, é mais provável que cortes por limitantes sejam realizados em níveis menos profundos da árvore, diminuindo-a. Nisso consiste o método conhecido como *Branch and Cut*.

Em resumo, o método denominado *Branch and Cut* é a junção do *Branch and Bound* com o método de Planos de Cortes. No próximo capítulo resolvemos o problema da elaboração de grade de horários escolares via *Branch and Cut* implementado no pacote comercial CPLEX.

4 O problema da elaboração de grade de horários

O problema da elaboração de horários escolares consiste no escalonamento de um conjunto de disciplinas entre professores e salas de aula, durante um período de tempo pré definido. Restrições inerentes à própria divisão, como o de um professor não ter aula em turmas distintas num mesmo horário, devem ser satisfeitas. Adicionalmente, restrições de preferências devem ser atendidas sempre que possível. Tais restrições incluem preferências de um professor por certas disciplinas, período do dia em que uma dada disciplina seja dada, boa distribuição das aulas de uma turma durante a semana, etc.

O estudo da elaboração de horários escolares é assunto já abordado por diversos autores. Dentre eles MirHassani (MIRHASSANI, 2006) relata o problema na Faculdade de Matemática da Universidade de Tecnologia de Sharood, no Irã, onde um modelo de programação inteira é proposto. Tais modelos são frequentes na descrição do problema (outro exemplo interessante é o trabalho de Daskalaki e Birbas (DASKALAKI; BIRBAS, 2005)). MirHassani propõe uma formulação simplificada e eficiente do ponto de vista computacional, já que algumas formulações de programação inteira requerem muito esforço computacional em sua resolução, por se tratar de um problema NP-difícil (MIRHASSANI, 2006).

Várias técnicas foram utilizadas na resolução do problema de horários escolares. Dentre elas podemos citar *Simulating Annealing*, Busca tabu, relaxação Lagrangeana e o método de Geração de Colunas. Para uma referenciação adequada, consulte (MIRHASSANI, 2006; BURKE; PETROVIC, 2002). Como já comentamos, utilizamos a clássica técnica de *Branch and Cut* para programação inteira.

4.1 Modelagem Matemática

Este trabalho foca no estudo do problema presente no Departamento de Matemática Aplicada (DMA/CEUNES/UFES) (ARAUJO, 2012). Com base no artigo de MirHassani (MIRHASSANI, 2006), modelamos o problema de horários semanais deste Departamento. Para uma maior adequação à realidade estudada, restrições e variáveis foram adicionadas, conforme mencionaremos ao longo deste capítulo.

A Tabela 1 descreve os índices utilizados no modelo do problema.

Descrição	Índice	Descrição
Disciplina	c	Todas as disciplinas ofertadas pelo DMA
Professor	p	Todos os professores alocados no DMA
Dia	d	Os dias da semana (Segunda a Sexta)
Horário	h	Os horários disponíveis, no caso janelas
Sala	r	Tipo de sala (normal/laboratório)

Tabela 1: Índices das variáveis.

Para descrever o modelo estipulamos as seguintes variáveis:

$$x_{pcdh} = \begin{cases} 1, & \text{se o professor } p \text{ ministra a disciplina } c \text{ no dia } d \text{ no horário } h \\ 0, & \text{caso contrário} \end{cases},$$

$$y_{pc} = \begin{cases} 1, & \text{se o professor } p \text{ ministra a disciplina } c \\ 0, & \text{caso contrário} \end{cases}.$$

De modo a representar adequadamente o modelo, estipulamos os seguintes parâmetros:

$$CHTurno(c, h) = \begin{cases} 1, & \text{se a disciplina } c \text{ puder ser alocada no horário } h \\ 0, & \text{caso contrário} \end{cases},$$

$\mathbb{N} \ni CHSmax(p)$: carga horária máxima semanal do professor p em janelas de 2 horas.

$NumTS(r)$: quantidade de salas do tipo r .

$\mathbb{N} \ni PC(p, c)$: preferência do professor p para ministrar a disciplina c .

Convencionamos que se $PC(p, c) = 0$, o professor p é impedido de ministrar a disciplina c . Quanto menor o valor, maior a preferência do professor p em ministrar a disciplina c . Assim, $PC(p, c) = 1$ significa prioridade máxima. Convencionamos ainda que se nada for dito a respeito da preferência do professor p na disciplina c , $PC(p, c) = 10$.

$CS(c)$: quantidade de janelas que a disciplina c necessita.

Seguindo MirHassani, consideramos as seguintes simplificações no modelo: levamos em consideração que uma janela de tempo equivale a 2 horas de aula, e que as disciplinas trabalhadas no modelo são de cargas horárias 30, 60 e 90 horas, ou seja, de 1, 2 e 3 janelas semanais.

Os dias e horários utilizados são descritos nas Tabelas 2 e 3. Observamos que são associados aos horários os turnos “diurno” e “noturno”. Cada disciplina tem seu turno definido, cujos horários são controlados pelo parâmetro $CHturno$.

Dia da Semana	índice
Segunda-Feira	1
Terça-Feira	2
Quarta-Feira	3
Quinta-Feira	4
Sexta-Feira	5

Tabela 2: Dias da Semana.

Horário	Turno
7:00-8:40	Diurno
8:40-10:20	Diurno
10:40-12:20	Diurno
13:00-14:40	Diurno
14:40-16:20	Diurno
16:40-18:20	Diurno
18:50-20:30	Noturno
20:50-22:30	Noturno

Tabela 3: Horários de Aula.

Finalmente, estabelecemos o modelo como segue:

$$\min \sum_{c,d} R * v_{cd} + \sum_p S * s_p - 300 * eq + \sum_{p,c} P * PC(p, c) * y_{pc} \quad (4.1)$$

sujeito às seguintes restrições:

$$\sum_p y_{pc} = 1, \quad \forall c \quad (4.2)$$

A restrição (4.2) garante que cada disciplina é ministrada por um único professor.

$$\sum_{p,d} x_{pcd(7:00-8:40)} + x_{pcd(8:40-10:20)} + x_{pcd(10:40-12:20)} = CS(c) * w_c, \quad \forall c \text{ diurno} \quad (4.3)$$

$$\sum_{p,d} x_{pcd(13:00-14:40)} + x_{pcd(14:40-16:20)} + x_{pcd(16:40-18:20)} = CS(c) * (1 - w_c), \quad \forall c \text{ diurno} \quad (4.4)$$

As restrições (4.3) e (4.4) garantem que disciplinas sejam ofertadas por turno (diurno/noturno). Com isso impede que disciplinas pertinentes a um turno específico sejam ofertadas em outro turno. Fixam ainda a carga horária de cada disciplina c em $CS(c)$ janelas de tempo.

$$\sum_{p,h} x_{pc(d+1)h} - v_{cd} \leq 1, \quad \forall c, d \text{ tal que } d \neq 5 \quad (4.5)$$

A restrição (4.5) exprime o desejo que cada disciplina tenha pelos menos um dia livre entre as aulas. No caso, v_{cd} é a medida de violação da restrição com penalidade R na função objetivo. Em nossos testes, utilizamos $R = 1$. Note que poderão ocorrer disciplinas com aulas em dias seguidos. Observemos que é suficiente que $v_{cd} \in \{0, 1\}$.

$$\sum_{p,h} x_{pc1h} + \sum_{p,h} x_{pc5h} \leq 1, \quad \forall c \text{ tal que } CS(c) = 2 \quad (4.6)$$

A restrição (4.6) garante que disciplinas com 2 janelas devem ter no máximo 2 dias livres entre as aulas.

$$\sum_{p,c} x_{pcdh} \leq NumTS(r), \quad \forall d, h, r \quad (4.7)$$

A restrição (4.7) reflete as limitações de espaço físico.

$$\sum_{p,d} x_{pcdh} = 0, \quad \forall c, h \text{ tal que } CHturno(c, h) = 0 \quad (4.8)$$

A restrição (4.8) garante que turmas diurnas não tenham aulas no período noturno e vice-versa. É claro que as variáveis x_{pcdh} envolvidas serão zero, e podem ser eliminadas do modelo.

$$\sum_{p,c;Gdisc(c)\ni g} x_{pcdh} \leq 1, \quad \forall g, d, h \quad (4.9)$$

A restrição (4.9) garante que disciplinas de um mesmo grupo (turma) não tenham conflitos de horários. Aqui $Gdisc(c)$ é o conjunto dos grupos de alunos que assistem a disciplina c .

$$\sum_c x_{pcdh} \leq 1, \quad \forall p, d, h \quad (4.10)$$

A restrição (4.10) exprime que cada professor só pode ministrar uma única disciplina num mesmo dia e horário.

$$\sum_{d,h} x_{pcdh} = CS(c) * y_{pc}, \quad \forall p, c \quad (4.11)$$

A restrição (4.11) diz que todos os horários de uma certa disciplina devem ser dados por um único professor.

$$\sum_c CS(c) * y_{pc} \leq CHSmax(p), \quad \forall p \text{ efetivo não chefe} \quad (4.12)$$

A restrição (4.12) garante que cada professor do quadro efetivo que não seja coordenador ou chefe tenha carga horária até o seu limite.

$$\sum_c CS(c) * y_{pc} = 2, \quad \forall p \text{ chefe ou coordenadores} \quad (4.13)$$

A restrição (4.13) diz que os coordenadores ou chefes tenham exatamente duas janelas de carga horária, conforme descrito por leis internas da instituição.

$$\sum_c CS(c) * y_{pc} - s_p = CHSmax(p), \quad \forall p \text{ substituto} \quad (4.14)$$

A restrição (4.14) garante que os professores substitutos tenham suas cargas horárias maximizadas. No caso, $s_p \in \mathbb{N}$ é a medida de violação da restrição, com penalidade S na função objetivo. Utilizamos $S = 10$.

$$\sum_c CS(c) * y_{pc} \geq eq, \quad \forall p \text{ efetivo não chefe e não coordenador} \quad (4.15)$$

$$eq \geq 4. \quad (4.16)$$

A restrição (4.15) faz com que a carga horária dos professores do quadro efetivo que não sejam chefe ou coordenador tenham um mínimo comum eq . Portanto ao maximizar eq , a carga horária dos professores são equilibradas. Por sua vez, a restrição (4.16) garante que os professores em questão dêem no mínimo 8 horas semanais de aula, ou seja, 4 janelas.

$$\sum_c CS(c) * y_{p_1c} - \sum_c CS(c) * y_{p_2c} \leq 2, \quad \forall p_1, p_2 \text{ efetivos distintos não chefe e não coordenadores} \quad (4.17)$$

A restrição (4.17) garante que as cargas horárias entre os professores do quadro efetivo que não sejam chefe ou coordenador difiram em no máximo 2 janelas de tempo (4 horas).

$$\sum_c x_{pcdh_1} + \sum_c x_{pcdh_2} \leq 1, \quad \forall p, d, h_1, h_2 \quad (4.18)$$

$$\sum_c x_{pcdh_1} + \sum_c x_{pc(d+1)h_2} \leq 1, \quad \forall p, d, h_1, h_2 \quad \text{tal que } d \neq 5 \quad (4.19)$$

As restrições (4.18) e (4.19) impossibilitam que os professores ministrem disciplinas em horários conflitantes no mesmo dia e em dias consecutivos, respectivamente, de acordo com a Tabela 4 abaixo. Tais restrições, além de impedir horários indesejados de trabalho docente em sala de aula, contribuem para diminuir o tempo vago entre aulas de um professor (condensa os horários).

Horário (h_1)	Horário (h_2)	Mesmo dia	Dias consecutivos
20:50-22:30	7:00-8:40	X	X
20:50-22:30	8:40-10:20	X	X
20:50-22:30	10:40-12:20	X	X
18:50-20:30	7:00-8:40	X	X
18:50-20:30	8:40-10:20	X	X
18:50-20:30	10:40-12:20	X	
13:00-14:40	10:40-12:20	X	

Tabela 4: Horários conflitantes.

$$x_{pcdh} \in \{0, 1\}, \quad \forall p, c, d, h \quad (4.20)$$

$$y_{pc} \in \{0, 1\}, \quad \forall p, c \quad (4.21)$$

$$s_p \in \mathbb{N}, \quad \forall p \quad (4.22)$$

$$v_{cd} \in \{0, 1\}, \quad \forall c, d \quad (4.23)$$

As restrições (4.20) a (4.23) referem-se à integralidade das variáveis.

4.2 Resultados

Iniciamos a modelagem do problema, baseando-se no artigo de MirHassani (MIRHASSANI, 2006).

A necessidade de interação com os professores foi necessária para poder desenvolver um modelo próximo da realidade. Com isso, foi possível ter a noção exata de como se estabelece as rotinas na formulação do horário semestral. A partir dessa experiência, foi possível obter dados suficientes para os testes computacionais. Para a modelagem computacional, utilizamos o *software* livre ZIMPL (KOCH, 2011). Na resolução do modelo de programação inteira utilizamos o *software* IBM CPLEX 12.4. Trata-se de uma implementação comercial de vários métodos (dentre eles o *Branch and Cut*) amplamente utilizada na literatura mundial, e considerada como a de maior credibilidade no meio acadêmico.

À medida em que os testes eram realizados, a cada etapa fazíamos acertos para adequação do modelo à realidade. Nesse sentido, analisamos se a solução do modelo respondia satisfatoriamente as necessidades do Departamento.

Na fase inicial, o modelo tinha alguns problemas, como a distribuição de carga horária, e a condensação de horário dos professores. Com a inserção das restrições (4.15) e (4.16) o problema da distribuição da carga horária entre os professores foi contornado. Sendo assim, todos os professores do quadro efetivo tiveram suas cargas horárias semanais equilibradas. O problema da condensação foi amenizado, ou seja, aquele relacionado aos longos períodos vagos entre aulas, pelas restrições (4.18) e (4.19).

Os dados colhidos no DMA, e utilizados na simulação, têm as seguintes características, referentes ao semestre letivo 2012/1:

Das 20 salas de aula, 16 são salas de aula normal, 2 laboratórios de informática,

Descrição	Quantidade
Disciplinas	46
Salas	20
Professor Efetivo em exercício	16
Professor Substituto	3

Tabela 5: Dados utilizados no modelo.

1 laboratório de matemática e 1 laboratório de ensino. No departamento existem 19 professores efetivos sendo 3 em afastamento. Assim, 16 professores do quadro efetivo estavam disponíveis para aulas, chefia do departamento e duas coordenações de cursos. Quase a totalidade dos professores disponíveis podiam ministrar aulas em qualquer horário da semana, a menos dos horários da reunião departamental.

O computador utilizado nas simulações possui as seguintes configurações: processador Intel Core 2 Quad 2.33 GHz, 2Gb de memória RAM e sistema operacional GNU/Linux 64 bits. Realizamos testes trocando algumas restrições por outras equivalentes, de modo a avaliar o desempenho computacional de cada uma. A seguir listamos tais equivalências.

A restrição (4.6) é equivalente à restrição

$$\sum_{p,d,h;d \in \{2,\dots,4\}} x_{pcdh} \leq 1, \quad \forall c \text{ tal que } CS(c) = 2. \quad (4.24)$$

As restrições (4.12), (4.13) e (4.14) são equivalentes às restrições

$$\sum_{c,d,h} x_{pcdh} \leq CHSmax(p), \quad \forall p \text{ efetivo não chefe} \quad (4.25)$$

$$\sum_{c,d,h} x_{pcdh} = CHSmax(p), \quad \forall p \text{ coordenador e chefe} \quad (4.26)$$

$$\sum_{c,d,h} x_{pcdh} - s_p = CHSmax(p), \quad \forall p \text{ substituto} \quad (4.27)$$

respectivamente. A Tabela 6 resume os resultados obtidos.

Descrição	Teste 1	Teste 2	Teste 3	Teste 4
Número de variáveis	34.089	34.089	34.089	34.089
Número de restrições	4.464	4.464	4.464	4.464
Elementos não nulos	343.291	375.371	371.123	339.043
Números de nós	12	15	21	7
Tempo da resolução	152.06sec	161.95sec	175.56sec	78.20sec

Tabela 6: Testes computacionais.

As trocas de restrições por suas equivalentes foram feitas como segue. Dois grupos foram estabelecidos: um formado somente pelas restrições (4.6) e sua equivalente (4.24);

e outro formado pelas restrições (4.12), (4.13), (4.14) e suas equivalentes (4.25), (4.26), (4.27). A junção de restrições ocorrida no segundo grupo levou em consideração o fato de que elas tratam de carga horária. Daí trocamos, dentro de um mesmo grupo, as restrições envolvendo variáveis x_{pcdh} com as envolvendo variáveis y_{pc} . Sendo assim, no Teste 1 foram utilizadas as restrições (4.24), (4.12), (4.13) e (4.14). No Teste 2, as restrições (4.24), (4.25), (4.26) e (4.27). Já no Teste 3 foram utilizadas as restrições (4.6), (4.25), (4.26) e (4.27), e no Teste 4, (4.6), (4.12), (4.13) e (4.14).

Observamos que o número de restrições e variáveis em todos os testes são os mesmos (por motivos óbvios). No entanto, o número de elementos não nulos na matriz das restrições do problema varia. Nota-se que o Teste 4 tem melhores resultados, pois é o de menor número de elementos não nulos.

Ressaltamos que as variáveis y_{pc} e restrições envolvendo-as foram propostas do presente estudo, e contribuíram para acelerar a resolução do modelo. É importante destacar ainda que as restrições (4.2) e (4.11), no artigo de MirHassani (MIRHASSANI, 2006), são descritas equivalentemente como

$$\sum_{p,d,h} x_{pcdh} = CS(c), \quad \forall c. \quad (4.28)$$

Testes foram realizados com essa restrição e todos foram piores que os reportados aqui. Isso se deve em certa medida ao processo de ramificação utilizado no método *Branch and Cut*: é sabido que a ramificação sobre a restrição (4.2) é eficiente (veja o tópico *Generalized Upper Bound* em (WOLSEY, 1998)).

5 Conclusões

No capítulo 1, fizemos uma breve explanação da teoria da Programação Inteira. No capítulo 2, introduzimos o conceito de desigualdades válidas, que servem de base para o método de planos de corte. No capítulo 3 é discutido o método de enumeração *Branch and Bound*. Comentários acerca do método *Branch and Cut* foram feitos.

No capítulo 4, aplicamos a teoria no problema da elaboração de grade de horários escolares. O modelo desenvolvido satisfaz os objetivos traçados, ou seja, soluciona o problema presente do Departamento de Matemática Aplicada (DMA/CEUNES/UFES). Conforme relatado anteriormente, levamos em consideração somente disciplinas de cargas horárias 30, 60 e 90 horas. No entanto, existem no Departamento disciplinas de 45 e 75 horas.

A condensação dos horários para os professores foi atingida com sucesso. O modelo não contempla alocação de salas de aula. O artigo proposto por Daskalaki e Birbas (DASKALAKI; BIRBAS, 2005) resolve o problema com a alocação de salas, propondo-o ser resolvido em duas fases. A primeira é similar ao apresentado aqui, ou seja, aloca disciplinas para os professores e estabelece os horários das mesmas, levando em consideração somente a capacidade de espaço físico. A segunda fase cuida da alocação de salas, onde tenta-se minimizar, dentre outras coisas, a variedade de salas de aula que uma mesma turma usa. É relatado pelos autores que esta segunda fase é de fácil resolução frente à primeira.

O pacote utilizado, o CPLEX, possui vários recursos que possibilitam resolver problemas de grande porte, tais como cortes especializados, recursos de pré-processamento e heurísticas que aceleram a resolução. Esses inúmeros recursos não são abordados no presente trabalho, mas foram utilizados nos testes.

Referências

- ALVES, R.; DELGADO, C. *Programacao Linear Inteira*. [S.l.]: Faculdade de Economia - Universidade de Porto, 1997.
- ARAÚJO, L. D. S. e Rodrigo Leppaus de. *Anais do Congresso CMAC - Congresso de Matemática Aplicada e Computacional, Natal-RN*, 2012.
- BAZARAA, M. S.; JARVIS, J. J. *Linear programming and network flows*. [S.l.]: John Wiley & Sons, 1977.
- BURKE, E. K.; PETROVIC, S. Recent research directions in automated timetabling. *European Journal of Operational Research*, v. 140, p. 266–280, 2002.
- DASKALAKI, S.; BIRBAS, T. Efficient solutions for a university timetabling problem through integer programming. *European Journal of Operational Research*, v. 160, p. 106–120, 2005.
- KOCH, T. *ZIMPL User Guide 3.2.0*. [S.l.], 2011.
- MIRHASSANI, S. A. A computational approach to enhancing course timetabling with integer programming. *Applied Mathematics and Computation*, v. 175, p. 814–822, 2006.
- SECCHIN, L. D. *Programacao Linear - Notas de aula*. [S.l.]: UFES, 2012.
- WOLSEY, L. A. *Integer Programming*. [S.l.]: John Wiley & Sons, 1998. (Series in Discrete Mathematics and Optimization).

APÊNDICE A – Programação Linear

Este apêndice traz uma breve revisão de conceitos de programação linear. Uma discussão mais abrangente do assunto pode ser encontrada em (BAZARAA; JARVIS, 1977; SECCHIN, 2012).

Qualquer modelo de programação linear pode ser escrito na *forma padrão*

$$\begin{aligned} \text{PL: } \min \quad & cx \\ \text{s.a. } & Ax = b \\ & x \geq 0 \end{aligned}$$

onde $b \geq 0$ e A é uma matriz $m \times n$ de posto $m < n$. Por exemplo, o modelo do Capítulo 1

$$\begin{aligned} \min \quad & cx \\ \text{s.a. } & Ax \leq b \\ & x \geq 0 \end{aligned}$$

pode ser reescrito como

$$\begin{aligned} \min \quad & cx + 0x_f \\ \text{s.a. } & A'x + I'x_f = b' \\ & x, x_f \geq 0 \end{aligned}$$

onde I' é a matriz identidade de ordem m cujas linhas j são multiplicadas por -1 sempre que $b_j < 0$ (assim $b' \geq 0$). As variáveis extras x_f são chamadas *variáveis de folga*.

Antes de entrarmos diretamente na resolução, apresentaremos definições e resultados.

Definição A.0.1. *Dado um conjunto convexo P , $x \in P$ é dito ponto extremo ou vértice de P se não pode ser representado por uma combinação convexa de dois pontos distintos de x em P , isto é, não existem $x_1, x_2 \in P \setminus \{x\}$ e $\lambda \in (0, 1)$ tais que $x = x_1 + (1 - \lambda)x_2$.*

Definição A.0.2. Dado um conjunto convexo P , um vetor não nulo p é uma direção de P se para cada $x \in P$, a semi-reta $x + \lambda d, \lambda > 0$ está contida em P . Notemos que se P for limitado, não admite direções. Uma direção x é extrema se não pode ser escrita como uma combinação positiva de duas direções distintas em P .

Teorema A.0.3. Considere o modelo

$$\begin{aligned} PL: \min \quad & cx \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

onde o conjunto viável é não vazio com pontos extremos x_1, x_2, \dots, x_k e direções extremas d_1, \dots, d_l . Então

1. PL admite solução ótima se, e somente se, $cd_j \geq 0$ para todo $j = 1, \dots, l$;

2. Se PL admite solução ótima, uma é um dos pontos extremos x_1, \dots, x_k .

Agora, buscaremos descrever analiticamente os vértices do conjunto viável, na busca de soluções ótimas.

Definição A.0.4. Considere o sistema $Ax = b, x \geq 0$, onde A é $m \times n$ ($m \leq n$). Suponha que o posto de A seja m . Rearranjando se necessário as colunas de A , escrevemos $A = \begin{bmatrix} B & N \end{bmatrix}$ onde B é matriz de ordem m inversível e N é $m \times (n - m)$. Uma solução

$$x = \begin{bmatrix} X_B \\ X_N \end{bmatrix} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix}$$

de $Ax = b$ é dita ser uma solução básica do sistema (note que ao arranjar as colunas de A , o mesmo ocorre com as linhas de x). Se além disso tivermos $x_B \geq 0$ (e assim $x \geq 0$), então dizemos que x é solução básica viável do sistema. Neste caso, B é chamada matriz básica ou simplesmente base, e a matriz N é chamada matriz não básica. As componentes de x_B são chamadas variáveis básicas e as componentes de x_N variáveis não básicas.

O resultado a seguir caracteriza os vértices do conjunto viável.

Teorema A.0.5. Seja $P = \{x; Ax = b; x \geq 0\}$ um conjunto não vazio. Então x é vértice de P se, e somente se, x é solução básica viável de $Ax = b; x \geq 0$. Mais ainda, x é vértice não degenerado se, e somente se, é solução básica viável não degenerada.

Teorema A.0.6. 1. Se $P = \{x; Ax = b; x \geq 0\}$ é não vazio, admite solução básica viável.

2. Se o (PL)

$$\begin{aligned} PL: \min \quad & cx \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

tem conjunto viável não vazio e admite solução ótima, então existe uma solução básica viável ótima.

3. O conjunto viável do PL

$$\begin{aligned} PL: \min \quad & cx \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

admite um número finito de soluções básicas viáveis.

Com os resultados apresentados até o momento, podemos resolver

$$\begin{aligned} PL: \min \quad & cx \\ \text{s.a.} \quad & Ax = b \\ & x \geq 0 \end{aligned}$$

enumerando as soluções básicas viáveis de seu conjunto viável, se houverem. O *Método Simplex* que apresentaremos é baseado nessa ideia. Sem a necessidade de enumerar todas as soluções básicas, que seria impraticável, o Simplex é capaz de reconhecer a otimalidade de uma solução básica. Também é capaz de reconhecer problemas ilimitados.

Tome uma solução básica viável de PL

$$x = \begin{bmatrix} x_B \\ x_N \end{bmatrix} = \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix}$$

cujo valor da função objetivo (FO) é

$$z_0 = c \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} = \begin{bmatrix} c_B & c_N \end{bmatrix} \begin{bmatrix} B^{-1}b \\ 0 \end{bmatrix} = c_B B^{-1}b.$$

Temos $Ax = b$ se, e somente se,

$$x_B = B^{-1}b - B^{-1}Nx^N \quad (\text{A.1})$$

$$= B^{-1}b - \sum_{j \in R} B^{-1}a_j x_j \quad (\text{A.2})$$

$$= \bar{b} - \sum_{j \in R} y_j x_j \quad (\text{A.3})$$

onde R é o conjunto dos índices das variáveis não básicas (VNB's), a_j é a coluna j de A , $\bar{b} = B^{-1}b$ e $y_j = B^{-1}a_j$. O valor da FO em x é

$$z = cx = c_B x_B + c_N x_N \quad (\text{A.4})$$

$$= c_B (B^{-1}b - \sum_{j \in R} B^{-1}a_j x_j) + \sum_{j \in R} c_j x_j \quad (\text{A.5})$$

$$= z_0 - \sum_{j \in R} (z_j - c_j) x_j \quad (\text{A.6})$$

onde $z_j = c_B B^{-1}a_j$. Com isso, PL pode ser reescrito como

$$\text{PL: } \min z_0 - \sum_{j \in R} (z_j - c_j) x_j$$

$$\text{s.a. } \sum_{j \in R} y_j x_j + x_B = \bar{b}$$

$$x_j \geq 0, \forall j \in R$$

$$x_B \geq 0.$$

onde x_B são variáveis de folga, podendo ser reescrito ainda como

$$\text{PL: } \min z_0 - \sum_{j \in R} (z_j - c_j) x_j$$

$$\text{s.a. } \sum_{j \in R} y_j x_j \leq \bar{b}$$

$$x_j \geq 0, \forall j \in R$$

Como $b \geq 0$, do último modelo temos o seguinte critério de otimalidade:

Teorema A.0.7. *Seja x uma solução básica viável de PL e R seu conjunto dos índices das VNB's.*

1. *Se $z_j - c_j \leq 0$ para todo $j \in R$, então x é solução ótima de PL.*

Como relatado até o momento, o método Simplex busca por soluções básicas viáveis ótimas, ou seja, devemos obter uma solução básica viável melhor que uma corrente. Como

cada solução básica viável corresponde a uma base B , devemos obter uma outra base B' melhor que B . Temos que as matrizes básicas são formadas por colunas de A . Deste modo passaremos de uma base B para outra B' trocando uma de suas colunas por uma coluna não básica, de N . Devemos estabelecer então critérios para entrada e saída de variáveis da base.

Inicialmente, suponha que temos uma solução básica $\begin{bmatrix} B^{-1}b & 0 \end{bmatrix}^t \geq 0$ e consideremos o modelo PL anterior, escrito nas VNB's. Como sabemos, a FO só pode ser melhorada se existir j tal que $z_j - c_j > 0$. A fim de diminuir ao máximo a FO, escolhemos o índice k com maior $z_k - c_k$. Esse será o critério para entrada de uma variável na base, ou seja, escolhemos a variável não básica x_k tal que

$$z_k - c_k = \max_{j \in R} \{z_j - c_j\} \quad (\text{A.7})$$

e a aumentamos. As VNB's restantes continuam iguais zero, e então o novo valor da FO será

$$z = z_0 - (z_j - c_j)x_k \quad (\text{A.8})$$

Escolhida a VNB que entrará na base, escolhemos a VB que sairá, que esse critério de saída é dado pela expressão

$$\frac{\bar{b}_r}{y_{rk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}}; y_{ik} > 0 \right\} \quad (\text{A.9})$$

Com os critérios de entrada e saída, é possível demonstrar a cara dos novos valores da FO e das novas VNB's que são dados pelas expressões

$$x_k = \frac{\bar{b}_r}{y_{rk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}}; y_{ik} > 0 \right\} \quad (\text{A.10})$$

e

$$x_{Bi} = \bar{b}_i - \frac{y_{ik}}{y_{rk}} \bar{b}_r \quad \forall i = 1, \dots, m \quad (\text{A.11})$$

respectivamente.

Assim, após verificado os critérios de entrada e saída e o valor da nova FO, apresentaremos o método Simplex em formato de Quadro.

Suponha que tenhamos uma solução básica viável x com base B . Pelo que vimos,

podemos escrever o (PL) como

$$\begin{aligned}
 \text{PL min } z \\
 \text{s.a. } z - c_B x_B - c_N x_N &= 0 \\
 Bx_B + Bx_N &= b \\
 x_B, x_N &\geq 0
 \end{aligned}$$

Assim,

$$x_B + B^{-1}N x_N = B^{-1}b \tag{A.12}$$

Multiplicando a equação acima por c_B e somando a primeira restrição do PL, obtemos

$$z + 0x_B + (c_B B^{-1}N - c_N)x_N = c_B B^{-1}b \tag{A.13}$$

Representamos as equações (A.12) e (A.13) em um quadro, como segue. Colocamos a equação (A.13) relativa à FO z na primeira linha do quadro (linha 0), onde os coeficientes de z , x_B e x_N são organizados em colunas. O lado direito da igualdade, $c_B B^{-1}b$ é colocado na última coluna, denominada RHS (em inglês, *Right-Hand-Side*). As m outras linhas (linhas 1 à m) contidas na equação (A.12) vêm logo abaixo, seguindo a mesma separação em colunas. O Quadro Simplex é então organizado da seguinte forma:

	z	x_B	x_N	RHS
z	1	0	$c_B B^{-1}N - c_N$	$c_B B^{-1}b$
x_B	0	I	$B^{-1}N$	$B^{-1}b$

Observamos que o quadro acima contém todas as informações para execução do Simplex: na linha 0, a coluna j de x_N é

$$(c_B B^{-1}N - c_N)e_j = c_B B^{-1}a_j - c_j = z_j - c_j \tag{A.14}$$

e RHS consiste no valor corrente da FO, $z_0 = c_B B^{-1}b$; o bloco $B^{-1}N$ é exatamente formado pelas colunas $y_k = B^{-1}a_k$. Portanto, através do quadro conseguimos dizer se a solução básica viável corrente é ótima ($c_B B^{-1}N - c_N \leq 0$), ou se o problema é ilimitado ($y_k \leq 0$ quando x_k entra na base). Agora, vamos analisar o processo de entrada e saída de variáveis da base pelo quadro. Lembrando que $b = B^{-1}b$, reescrevemos o quadro como segue

	z	x_{B_1}	\dots	x_{B_r}	\dots	x_{B_m}	\dots	x_j	\dots	x_k	\dots	RHS
z	1	0	\dots	0	\dots	0	\dots	$z_j - c_j$	\dots	$z_k - c_k$	\dots	$c_B \bar{b}$
x_{B_1}	0	1	\dots	0	\dots	0	\dots	y_{1j}	\dots	y_{1k}	\dots	\bar{b}_1
\vdots	\vdots	\vdots		\vdots		\vdots		\vdots		\vdots		\vdots
x_{B_r}	0	0	\dots	1	\dots	0	\dots	y_{rj}	\dots	y_{rk}	\dots	\bar{b}_r
\vdots	\vdots	\vdots		\vdots		\vdots		\vdots		\vdots		\vdots
x_{B_m}	0	0	\dots	0	\dots	1	\dots	y_{mj}	\dots	y_{mk}	\dots	\bar{b}_m

Supomos que x_k entra na base e que x_{B_r} sai da base. Vamos explicitar expressões para z e para as novas VB's tal como em (A.12) e (A.13). Como x_k passa a ser VB devemos explicitá-la em função das novas VNB's, o que inclui x_{B_r} . Do quadro atual, temos

$$\begin{aligned}
 x_{B_r} + \sum_{j \in R} y_{rj} x_j &= \bar{b}_r \\
 \Leftrightarrow x_{B_r} + y_{rk} x_k + \sum_{j \in R} y_{rj} x_j &= \bar{b}_r \\
 \Leftrightarrow x_k + \frac{1}{y_{rk}} x_{B_r} + \sum_{j \in R \setminus \{k\}} \frac{y_{rj}}{y_{rk}} x_j &= \frac{\bar{b}_r}{y_{rk}}
 \end{aligned} \tag{A.15}$$

onde lembramos que $y_{rk} > 0$ e que o novo conjunto das VNB's é $(R \setminus \{k\}) \cup \{B_r\}$. Para as outras VB's, ou seja, para $i \neq r$, temos

$$x_{B_i} + \sum_{j \in R} y_{ij} x_j \Leftrightarrow x_{B_i} + y_{ik} x_k + \sum_{j \in R \setminus \{k\}} y_{ij} x_j = \bar{b}_i$$

Por (A.15), substituímos x_k para obter

$$\begin{aligned}
 x_{B_i} + y_{ik} \left(\frac{\bar{b}_r}{y_{rk}} - \frac{1}{y_{rk}} x_{B_r} - \sum_{j \in R \setminus \{k\}} \frac{y_{rj}}{y_{rk}} x_j \right) + \sum_{j \in R \setminus \{k\}} y_{ij} x_j &= \bar{b}_i \\
 \Leftrightarrow x_{B_i} + \frac{y_{ik} \bar{b}_r}{y_{rk}} - \frac{y_{ik}}{y_{rk}} x_{B_r} - \sum_{j \in R \setminus \{k\}} \frac{y_{rj}}{y_{rk}} y_{ik} x_j + \sum_{j \in R \setminus \{k\}} y_{ij} x_j &= \bar{b}_i \\
 \Leftrightarrow x_{B_i} + \left(-\frac{y_{ik}}{y_{rk}} \right) x_{B_r} + \sum_{j \in R \setminus \{k\}} \left(y_{ij} - \frac{y_{rj}}{y_{rk}} y_{ik} \right) x_j &= \left(\bar{b}_i - \frac{y_{ik} \bar{b}_r}{y_{rk}} \right)
 \end{aligned} \tag{A.16}$$

As equações (A.15) e (A.16) fornecem a linha 1 a m do quadro após a atualização da base. Quanto a linha zero (A.13),

$$z + 0x_B + \sum_{j \in R} (z_j - c_j) x_j = c_B \bar{b}$$

substituindo x_k por (A.15) para obter

$$\begin{aligned}
 z + \sum_{j \in R} (z_j - c_j)x_j &= c_B \bar{b} \\
 \Leftrightarrow z + (z_k - c_k)x_k + \sum_{j \in R \setminus \{k\}} (z_j - c_j)x_j &= c_B \bar{b} \\
 \Leftrightarrow z + (z_k - c_k) \left(\frac{\bar{b}_r}{y_{rk}} - \frac{1}{y_{rk}} x_{B_r} - \sum_{j \in R \setminus \{k\}} \frac{y_{rj}}{y_{rk}} x_j \right) + \sum_{j \in R \setminus \{k\}} (z_j - c_j)x_j &= c_B \bar{b} \\
 \Leftrightarrow z + (z_k - c_k) \frac{\bar{b}_r}{y_{rk}} - \frac{z_k - c_k}{y_{rk}} x_{B_r} - \sum_{j \in R \setminus \{k\}} (z_j - c_j)x_j &= c_B \bar{b} \\
 \Leftrightarrow z + \left(-\frac{(z_k - c_k)}{y_{rk}} \right) x_{B_r} + \sum_{j \in R \setminus \{k\}} \left((z_j - c_j) - (z_k - c_k) \frac{y_{rj}}{y_{rk}} \right) x_j &= \left(c_B \bar{b} - (z_k - c_k) \frac{\bar{b}_r}{y_{rk}} \right) \quad (\text{A.17})
 \end{aligned}$$

Definidas as variáveis x_k de entrada e x_{B_r} de saída, o termo y_{rk} é chamado *pivô*. Ao processo de mudança de base, chamamos de pivoteamento. Nos referimos ao Quadro Simplex abreviadamente por QS. As equações (A.15), (A.16) e (A.17) fornecem o QS após pivoteamento:

	z	x_{B_1}	...	x_{B_r}	...	x_{B_m}	...	x_j	...	x_k	...	RHS
z	1	0	...	$-\left(\frac{z_k - c_k}{y_{rk}}\right)$...	0	...	$(z_j - c_j) - (z_k - c_k)\frac{y_{rj}}{y_{rk}}$...	0	...	$c_B\bar{b} - (z_k - c_k)\frac{\bar{b}_r}{y_{rk}}$
x_{B_1}	0	1	...	$-\frac{y_{1k}}{y_{rk}}$...	0	...	$y_{1j} - \frac{y_{rj}}{y_{rk}}y_{1k}$...	0	...	$\bar{b}_1 - \frac{y_{1k}\bar{b}_r}{y_{rk}}$
\vdots	\vdots	\vdots		\vdots		\vdots		\vdots		\vdots		\vdots
x_k	0	0	...	$\frac{1}{y_{rk}}$...	0	...	$\frac{y_{rj}}{y_{rk}}$...	1	...	$\frac{\bar{b}_r}{y_{rk}}$
\vdots	\vdots	\vdots		\vdots		\vdots		\vdots		\vdots		\vdots
x_{B_m}	0	0	...	$-\frac{y_{mk}}{y_{rk}}$...	1	...	$y_{mj} - \frac{y_{rj}}{y_{rk}}y_{mk}$...	0	...	$\bar{b}_m - \frac{y_{mk}\bar{b}_r}{y_{rk}}$

Podemos então estabelecer o Simplex como segue:

1.(INICIALIZAÇÃO) Encontre uma solução básica viável com base B . Forme o QS inicial.

2.(PASSO PRINCIPAL)

(i)Seja $z_k - c_k = \max_{j \in R} \{z_j - c_j\}$ (já calculado na linha 0 do QS). Se $z_k - c_k \leq 0$, então pare com a solução ótima, associada com a base B corrente (os valores das VB's estão na coluna RHS e das VNB's são zero). Caso contrário, vá para o próximo passo.

(ii)Se $y_k \leq 0$ (já calculado no QS) pare com a conclusão de que o problema é ilimitado. Caso contrário, vá para o próximo passo.

(iii) x_k entra na base. x_{B_r} sai da base sendo r de modo que

$$\frac{\bar{b}_r}{y_{rk}} = \min_{1 \leq i \leq m} \left\{ \frac{\bar{b}_i}{y_{ik}}; y_{ik} > 0 \right\}$$

Atualize o QS realizando pivoteamento no pivô y_{rk} (Quadro 7.1). Atualize o conjunto R dos índices das VNB's, e repita o passo 2.

Antes de fazermos exemplos, vamos discutir uma maneira interessante e fácil de realizar pivoteamentos no QS. Suponha que o pivô seja y_{rk} . Observando os Quadros 7.1 e 7.2, vemos que o processo de pivoteamento consiste nos passos (volte aos quadros e entenda os passos):

1.Atualizar cada elemento da posição (i, j) do quadro, inclusive da linha zero, que não esteja na linha ou na coluna do pivô y_{rk} , imaginando um retângulo com vértices em (i, j) e y_{rk} , e fazendo a operação correspondente. O esquema abaixo ilustra a operação genérica de atualização do elemento \bigcirc

$$\bigcirc - \left(\frac{\Delta}{y_{rk}} \right) \square$$

	z	x_1	\dots	x_j	\dots	x_k	\dots	x_n	RHS
z	1	•	\dots	•	\dots	•	\dots	•	•
x_{B_1}	0	•	\dots	•	\dots	•	\dots	•	•
\vdots	\vdots	\vdots		\vdots		\vdots		\vdots	\vdots
x_{B_t}	0	•	\dots	○	\dots	□	\dots	•	•
\vdots	\vdots	\vdots		\downarrow^-		\uparrow_\times		\vdots	\vdots
x_{B_r}	0	•	\dots	△	\rightarrow	y_{rk}	\dots	•	•
\vdots	\vdots	\vdots		\vdots	\div			\vdots	\vdots
x_{B_1}	0	•	\dots	•	\dots	•	\dots	•	•

2. Dividir a linha de x_{B_r} por y_{rk} ;

3. Fazer todos os elementos da coluna de x_k iguais a zero, exceto o elemento da linha de x_{B_r} ;

4. Atualizar as VB's, trocando x_{B_r} por x_k .

Para melhor entendimento, aplicatemos o método Simplex no seguinte exemplo.

Exemplo A.0.8. Resolver pelo simplex (via quadros) o PL

$$\begin{aligned}
 \min \quad & x_1 + 1x_2 - 4x_3 \\
 \text{s.a.} \quad & x_1 + 1x_2 + 2x_3 \leq 9 \\
 & x_1 + 1x_2 - 1x_3 \leq 2 \\
 & -1x_1 + 1x_2 + 1x_3 \leq 4 \\
 & x_1, \quad x_2, \quad x_3 \geq 0
 \end{aligned}$$

Na forma padrão, temos o PL

$$\begin{aligned}
 \min \quad & x_1 + 1x_2 - 4x_3 + 0x_4 + 0x_5 + 0x_6 \\
 \text{s.a.} \quad & x_1 + x_2 + 2x_3 + x_4 = 9 \\
 & x_1 + 1x_2 - 1x_3 + 0x_4 + 1x_5 + 0x_6 = 2 \\
 & -1x_1 + x_2 + x_3 + 0x_4 + 0x_5 + x_6 = 4 \\
 & x_1, \quad x_2, \quad x_3, \quad x_4, \quad x_5, \quad x_6 \geq 0
 \end{aligned}$$

Como $b \geq 0$, iniciamos com a base

$$B = \begin{bmatrix} a_4 & a_5 & a_6 \end{bmatrix}$$

pois assim, $x_B = B^{-1}b = b \geq 0$, e a solução básica associada é viável. Temos $c_B = 0$ e

logo

$$z_j - c_j = c_B B^{-1} a_j - c_j = -c_j; \quad \forall j \in R = \{1, 2, 3\}$$

Também, $B^{-1}N = N$, $b = B^{-1}b = b$ e $z_0 = c_B b = 0$. Assim o QS inicial (QS 1)

	z	x_1	x_2	x_3	x_4	x_5	x_6	RHS
z	1	-1	-1	4	0	0	0	0
x_4	0	1	1	2	1	0	0	9
x_5	0	1	1	-1	0	1	0	2
x_6	0	-1	1	1	0	0	1	4

Da linha zero, vemos que $z_3 - c_3 = \max_{j \in R} z_j - c_j > 0$, e logo x_3 entra. Da coluna de x_3 , vemos que somente $y_{13}, y_{33} > 0$. Agora, com a coluna RHS, vemos que

$$\frac{\bar{b}_1}{y_{13}} = \frac{9}{2} > \frac{1}{4} = \frac{\bar{b}_3}{y_{33}}$$

e logo $x_{B_3} = x_6$ sai. O pivô é portanto o elemento destacado y_{33} , e após pivoteamento, obtemos o QS 2

	z	x_1	x_2	x_3	x_4	x_5	x_6	RHS
z	1	3	-5	0	0	0	-4	-16
x_4	0	3	-1	0	1	0	-2	1
x_5	0	0	2	0	0	1	1	6
x_3	0	-1	1	1	0	0	1	4

QS 2 não é ótimo (pois $z_1 - c_1 > 0$) e temos y_{11} como pivô. Pivoteando, obtemos o QS 3

	z	x_1	x_2	x_3	x_4	x_5	x_6	RHS
z	1	0	-4	0	-1	0	-2	-17
x_1	0	1	-1/3	0	1/3	0	-2/3	1/3
x_5	0	0	2	0	0	1	1	6
x_3	0	0	2/3	1	1/3	0	1/3	13/3

□

APÊNDICE B – O Dual de um problema de Programação Linear

A cada problema de PL, associamos um outro problema, denominado *dual*, que fornece informações sobre o problema original, como limitantes e valor ótimo. O método *Simplex Dual*, uma variação do Simplex, utiliza o problema dual para resolver problemas de programação linear. Nesta apêndice apresentaremos de forma muito resumida dois resultados sobre dualidade utilizados no trabalho.

Considere o problema de PL na forma canônica

$$\begin{aligned} \text{P: } & \min cx \\ & \text{s.a. } Ax \geq b \\ & x \geq 0 \end{aligned}$$

onde A é uma matriz $m \times n$. Definimos o seu *dual* como o PL

$$\begin{aligned} \text{D: } & \max wb \\ & \text{s.a. } wA \leq c \\ & w \geq 0 \end{aligned}$$

Neste contexto, dizemos que P é o problema *primal*. Observe que o problema primal tem n variáveis e m restrições, enquanto seu dual m variáveis e n restrições. Existe uma relação entre variáveis primais (duais) e restrições duais (primais). Além disso, os objetivos (minimização e maximização) são distintos no primal e dual. Exploraremos esses fatos para introduzir o dual de qualquer PL.

Todo PL admite dual, mesmo aqueles que não estão na forma canônica, pois qualquer problema pode ser posto nessa forma (BAZARAA; JARVIS, 1977). Para obter o dual de um dado PL, na prática procedemos da seguinte forma: cada restrição primal tem uma única variável dual associada, e cada variável primal tem uma única restrição dual associada.

Além disso, como já dito, os objetivos são distintos. A relação entre variáveis/restrições primais e restrições/variáveis duais é dada na Tabela 7.

	Minimização		Maximização	
Variáveis	≥ 0	\leftrightarrow	\leq	Restrições
	≤ 0	\leftrightarrow	\geq	
	irrestrito	\leftrightarrow	$=$	
Restrições	\geq	\leftrightarrow	≤ 0	Variáveis
	\leq	\leftrightarrow	≥ 0	
	$=$	\leftrightarrow	irrestrito	

Tabela 7: Dual de um PL genérico.

Exemplo B.0.9. Considere o PL primal

$$\begin{aligned}
 \text{P: } & \max 8x_1 + 3x_2 - 2x_3 \\
 \text{s.a. } & x_1 - 6x_2 + x_3 \geq 2 \\
 & 5x_1 + 7x_2 - 2x_3 = -4 \\
 & x_1 \leq 0 \\
 & x_2 \geq 0 \\
 & x_3 \in \mathbb{R} \\
 & x \geq 0
 \end{aligned}$$

Utilizando a Tabela 7, seu dual é o PL

$$\begin{aligned}
 \text{D: } & \min 2w_1 - 4w_2 \\
 \text{s.a. } & w_1 + 5w_2 \leq 8 \\
 & -6w_1 + 7w_2 \geq 3 \\
 & w_1 - 2w_2 = -2 \\
 & w_1 \leq 0 \\
 & w_2 \in \mathbb{R}
 \end{aligned}$$

□

Os dois teoremas a seguir relacionam os problemas primal e seu dual.

Teorema B.0.10 (Dualidade Fraca). *Seja dado um problema primal na forma canônica P_1 e seu dual D_1 . Se x_0 e w_0 são pontos viáveis de P_1 e D_1 , respectivamente, então*

$$cx_0 \geq w_0b.$$

Ou seja, a FO do dual fornece um limitante inferior para a FO do problema primal de minimização na forma canônica.

Pelo Teorema da Dualidade Fraca, o dual fornece limitantes inferiores para o problema primal, no caso de minimização. Esse fato pode ser utilizado pelo método *Branch and Bound* no cálculo de limitantes da relaxação linear em um nó.

Teorema B.0.11 (Teorema Fundamental da Dualidade). *Sejam P um problema primal e D seu dual. Exatamente uma das afirmações ocorre:*

- (i) *Ambos possuem solução ótima x^* e w^* com $cx^* = w^*b$.*
- (ii) *Um dos problemas é ilimitado. Neste caso o outro é inviável.*
- (iii) *Ambos os problemas são inviáveis.*