Fernanda Hong Chen

Atribuição de Encargos Didáticos em Instituições de Ensino

São Paulo

2016

Fernanda Hong Chen

Atribuição de Encargos Didáticos em Instituições de Ensino

Monografia apresentada a Universidade de São Paulo, como parte das exigências do curso de Graduação "Matemática Aplicada e Computacional com Habilitação em Saúde Animal", para obtenção do título de bacharel.

Universidade de São Paulo – USP

Instituto de Matemática e Estatística

Bacharelado em Matemática Aplicada e Computacional

Orientador: Gabriel Haeser

São Paulo

2016

Dedico este trabalho a todos que acreditaram que esse dia chegaria.

Agradecimentos

Agradeço primeiramente aos meus pais, Paulo e Daisy que jamais me deixaram desistir durante essa tumultuada caminhada.

À minha irmã Débora, que com seu jeito protetor acolheu-me nas horas mais difíceis e me deu forças para persistir.

Ao meu orientador Professor Gabriel Haeser por aceitar o desafio de me orientar mesmo com todos os desafios apresentados.

Ao Marco Costa, por ser extremamente compreensivo e companheiro nos momentos decisivos.

Ao Rafael Máximo, pelas horas que passamos quebrando a cabeça para elaborarmos o programa.

Por fim, a todos que contribuíram de alguma forma para que eu não me desviasse do objetivo.

"Obstáculo é aquilo que você enxerga
quando tira seus olhos do objetivo."
(Henry Ford)

Resumo

Este trabalho contempla um estudo sobre Programação Linear e Programação Linear Inteira a fim de propor uma solução para o problema de atribuição didática de instituições de ensino. Como método utiliza-se Otimização Binária buscando maximizar a satisfação geral de professores quanto a atribuição de disciplinas que irão ministrar em um dado semestre. O software utilizado é o Matlab.

Conclui-se que o método utilizado é eficiente, mas necessita de implementações para que se torne prático e usual.

Palavras-chaves: Otimização Binária. Atribuição Didática. Grade Horária. Matlab. Método Branch-and-Bound.

Abstract

This study has as goal propose a solution to the didactics atribution of educational institutes. As method we use Binary Optimization aiming to maximize general satisfaction of teachers concerning classes they will minister given a semester. For that we used Matlab as software.

We conclude that the method used is efficient, but needs implementation to be practical and useful.

Key-words: Binary Optimization. Didactic Attribution. Timetabling. Matlab.

Lista de ilustrações

Lista de abreviaturas e siglas

USP	Universidade de São Paulo
IME	Instituto de Matemática e Estatística
PL	Problema Linear
PLI	Problema Linear Inteiro
PLIP	Problema Linear Inteiro Puro
PLIM	Problema Linear Inteiro Misto
MATLAB	Matrix Laboratory

Sumário

1 Introdução

1.1 O Trabalho

As instituições de ensino enfrentam a cada período letivo o problema de atribuição de turmas, dias e horários de aulas que cada professor deve ministrar. Apesar de todo o avanço computacional, a maior parte das instituições ainda realiza essa tarefa de modo manual, o que torna este trabalho muito problemático e cansativo. Fatores como tempo de casa e renome comumente são levados em consideração no momento de efetuação da atribuição. Com isso, muitos professores acabam ficando insatisfeitos com dada atribuição por se considerarem injustiçados.

1.2 Objetivo do Trabalho

Este trabalho tem como objetivo desenvolver um modelo, utilizando um método matemático exato, que encontre a solução do problema automaticamente, satisfazendo as restrições de horário de cada professor e suas preferências. O modelo baseia-se no caso específico do Departamento de Matemática do Instituto de Matemática e Estatística da USP, mas pode ser aplicado a outros departamentos e instituições. Como método utilizou-se da elaboração da Modelagem Matemática do problema, abordando-o como um problema de Programação Linear Inteira Binária. Por consequência, o tema Programação Linear e Programação Linear Inteira são abordados neste trabalho.

Utilizou-se o software MATLAB por [3] para resolução do problema. Para armazenamento de dados e criação de tabelas utilizou-se o Excel 2013.

1.3 Cenário

Na USP a grande maioria das disciplinas são semestrais sendo as demais anuais. Para o Instituto de Matemática e Estatística (IME) o horário das disciplinas dos cursos é estabelecido a cada início de semestre e para obtenção dos dados dos professores são entregues formulários para que os mesmos preencham com seus dados pessoais e suas preferências por turmas. A partir dos formulários são criadas matrizes com dimensões apropriadas para serem lidas no programa para então a otimização ser realizada.

1.4 Estrutura

Começaremos com uma fundamentação de problemas lineares, explicando os métodos mais utilizados na atualidade. Seguimos então para um nicho mais específico de problema, os problemas inteiros, que também possuem métodos mais específicos de resolução. Com isso partimos para o modelo que o problema se encaixa, Programação Linear de Variáveis Inteiras Binárias. Comentamos também sobre o software utilizado para a resolução do problema e as dificuldades do problema. Ao final, concluímos o trabalho.

2 Fundamentos de Otimização Linear

um problema de Programação Linear consiste em determinar valores não negativos para as variáveis de decisão, de forma que satisfaçam as restrições impostas e que otimizem uma função linear dessas variáveis. Vamos ilustrar melhor através dos exemplos abaixo.

2.1 Exemplos de Problemas de Programação Linear

2.1.1 O problema de produção [1]

Suponha que uma empresa produz n diferentes produtos usando m diferentes matérias-primas. Seja b_i a quantidade disponível da i-ésima matéria-prima, a_{ij} a quantidade de matéria-prima i necessária para a produção do produto j, e c_j o lucro obtido com a venda do produto j. Se a variável x_j representa a quantidade de produto j a ser produzida, então a empresa terá o lucro máximo resolvendo o problema

$$\begin{aligned} \max \quad & c^T x \\ \text{s.a.} \quad & Ax \leq b \\ & x \geq 0. \end{aligned}$$

2.1.2 O problema do plantão [1]

Um hospital quer fazer a programação semanal dos plantões noturnos de seus enfermeiros. A cada dia da semana a demanda por enfermeiros de plantão é diferente, representada por um inteiro d_j, $j = 1, \ldots, 7$. Cada enfermeiro sempre trabalha 5 noites seguidas de plantão. O problema é encontrar o número mínimo de enfermeiros que o hospital precisa contratar. Se tentássemos criar uma variável x_j representando número de enfermeiros de plantão no dia j não seríamos capazes de escrever a restrição de que cada enfermeiro sempre trabalha 5 noites seguidas. Ao invés disso, representamos x_j o número de enfermeiros que *começa* a trabalhar no dia j, assim os enfermeiros que começarem a trabalhar no dia 5 trabalharão nos dias 5, 6, 7, 1 e 2. O problema pode então ser formulado como

$$
\begin{array}{lccccccccccccccccc}
\min & x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & & \\
\text{s.a.} & x_1 & & & & & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \geq & d_1 \\
& x_1 & + & x_2 & & & & & + & x_5 & + & x_6 & + & x_7 & \geq & d_2 \\
& x_1 & + & x_2 & + & x_3 & & & & & + & x_6 & + & x_7 & \geq & d_3 \\
& x_1 & + & x_2 & + & x_3 & + & x_4 & & & & & + & x_7 & \geq & d_4 \\
& x_1 & + & x_2 & + & x_3 & + & x_4 & + & x_5 & & & & & \geq & d_5 \\
& & & x_2 & + & x_3 & + & x_4 & + & x_5 & + & x_6 & & & \geq & d_6 \\
& & & & & x_3 & + & x_4 & + & x_5 & + & x_6 & + & x_7 & \geq & d_7 \\
& x_j & \geq & 0, & & & & & & & & & & & & \\
& x_j & \in & \mathbb{Z} & & & & & & & & & & & &
\end{array}
$$

Este é um problema de programação linear *inteira*.

2.2 Representação e Soluções

De forma geral, um PL pode ser escrito como:

$$
\begin{cases}
\min & c^T x \\
\text{s.a} & Ax \geq b.
\end{cases}
$$

Outra forma de escrever é na *forma padrão*:

$$
\begin{cases}
\min & c^T x \\
\text{s.a} & Ax = b \\
& x \geq 0.
\end{cases}
$$

Chamamos de *Função Objetivo* o que queremos otimizar, seja por maximização ou minimização. Na forma padrão ela está representada por $c^T x$. As restrições do problema estão representadas por $Ax = b$ onde $x \geq 0$, também na forma padrão. Deste modo podemos formular qualquer problema de otimização linear.

2.2.1 O Método Simplex

O Método Simplex é um algoritmo amplamente utilizado para solucionar problemas de PL. Para que possa ser aplicado, o problema precisa estar na forma padrão. Utilizaremos o problema descrito em [4] para ilustrar o método simplex:

$$
\begin{array}{lrcl}
\min & z = -x_1 - 2x_2 & & \\
\text{s.a} & -2x_1 + x_2 & \leq & 2 \\
& -x_1 + 2x_2 & \leq & 7 \\
& x_1 & \leq & 3 \\
& x_1, x_2 \geq 0. & &
\end{array}
$$

Como o problema não está na forma padrão, adicionamos variáveis de folga para transformá-lo em um.

$$
\begin{array}{lrcl}
\min & z = -x_1 - 2x_2 & & \\
\text{s.a} & -2x_1 + x_2 + x_3 & = & 2 \\
& -x_1 + 2x_2 + x_4 & = & 7 \\
& x_1 + x_5 & = & 3 \\
& x_1, x_2, x_3, x_4, x_5 & \geq & 0.
\end{array}
$$

Com isso temos

$$
c = \begin{pmatrix} -1 & -2 & 0 & 0 & 0 \end{pmatrix}^T, \quad A = \begin{pmatrix} -2 & 1 & 1 & 0 & 0 \\ -1 & 2 & 0 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix} \quad \text{e} \quad b = \begin{pmatrix} 2 \\ 7 \\ 3 \end{pmatrix}
$$

Dividimos então nossas variáveis entre *básicas* (dentro da base) e *não-básicas* (fora da base), representadas respectivamente por x_B e x_N. As variáveis que estiverem fora da base devem ter seus valores iguais a 0 ($x_N = 0$).

Neste caso, podemos escolher x_3, x_4 e x_5 para colocá-las dentro da base, transformando as restrições em:

$$
\begin{array}{rcl}
x_3 & = & 2 + 2x_1 - x_2 \\
x_4 & = & 7 + x_1 - 2x_2 \\
x_5 & = & 3 - x_1 \\
x_1, x_2, x_3, x_4, x_5 & \geq & 0.
\end{array}
$$

É fácil ver que

$$
\begin{pmatrix} x_1 & x_2 & x_3 & x_4 & x_5 \end{pmatrix}^T = \begin{pmatrix} 0 & 0 & 2 & 7 & 3 \end{pmatrix}^T \qquad \text{e} \qquad z = 0
$$

Essa solução é chamada de *solução básica*, mas não sabemos ainda se é a *solução ótima*. A solução ótima deve ser a que contém o menor valor de z possível no caso de minimização e o maior z possível no caso de maximização.

Como nosso objetivo é minimizar

$$z = -x_1 - 2x_2$$

parece razoável que se aumentarmos o valor de x_1 ou x_2 teremos um valor z menor, e portanto melhor. Seguindo a mesma lógica, x_2 parece diminuir o valor de z mais rapidamente que x_1, pois está multiplicado por 2. Sendo assim, escolheremos x_2 para entrar na base.

Para que x_2 possa entrar na base, alguma outra variável deve sair. Podemos escolher x_3 ou x_4, pois ambos estão representados em função de x_2. Escolhemos arbitrariamente um deles, nesse caso x_3. Assim temos

$$x_2 = 2 + 2x_1 - x_3$$

Utilizamos essa equação para substituir o valor de x_2 nas demais restrições do problema e na função objetivo. Com isso temos um novo problema simplicado.

$$\min\ z = -4 - 5x_1 + 2x_3$$

sujeito às restrições

$$\begin{array}{rcl} x_2 & = & 2 + 2x_1 - x_3 \\ x_4 & = & 3 - 3x_1 + 2x_3 \\ x_5 & = & 3 - x_1 \end{array}$$

e com todas as variáveis não negativas ($x \geq 0$).

Como $x_N = \begin{pmatrix} x_1 \\ x_3 \end{pmatrix} = 0$, temos que $x_B = \begin{pmatrix} x_2 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 2 \\ 3 \\ 3 \end{pmatrix}$ e $z = -4$, que é menor que o z anterior. Com isso concluímos a primeira iteração do *Método Simplex*.

Assim podemos testar a otimalidade do problema, observando como a função objetivo pode ficar melhor ao aumentarmos os valores das variáveis não-básicas.

A solução básica viável encontrada não é ótima, pois podemos aumentar o valor de x_1 para melhorar a função objetivo, e assim sucessivamente. A cada iteração identificamos uma variável não-básica que possa melhorar nossa função objetivo. Essa variável escolhida é aumentada até que outra variável seja reduzida a 0. Isso nos dá uma nova solução básica, que deve ser testada por sua otimalidade até que a solução ótima seja encontrada.

Neste exemplo, a solução ótima é $x = (x_1, x_2, x_3, x_4, x_5)^T = (3, 5, 3, 0, 0)^T$.

O método apresentado é apenas uma das formas de representarmos o Método Simplex. Outras representações famosas são o *tableau* e a forma geral, que não serão abordados neste trabalho. Todos partem da mesma lógica para chegar à solução ótima.

É importante comentar que o problema pode conter mais de uma solução ótima. Entretanto isso não é um problema, pois o método Simplex garante encontrar uma dessas soluções ótimas.

2.2.2 Regra de Bland

Uma dificuldade que pode-se ter ao executar o método Simplex é a ciclagem. Isso ocorre quando escolhemos variáveis para entrar e sair da base sem um critério bem definido, fazendo com que o problema nunca seja resolvido apesar de ter resolução. Para evitar que isso ocorra, utiliza-se a *Regra de Bland*, que garante a melhor sequência para que as variáveis entrem e saiam da base sem que ocorra a ciclagem.

1. Encontre o menor índice j para o qual o custo reduzido $\overline{c}_j$ é negativo. Então a coluna A_j deve entrar na base
2. Escoha o menor índice i dentre as variáveis x_i possíveis para sair da base

2.3 Dualidade

Todo problema linear possui um problema correspondente chamado de Problema Linear "Dual", no qual os papéis de variáveis e restrições são invertidos. Ou seja, para cada variável do problema linear original ou "primal" existe uma restrição no problema dual e vice-versa.

Em uma aplicação, as variáveis do problema primal podem representar produtos, e a função objetivo pode representar o lucro associado à produção de tais produtos. Portanto a função obejtivo do problema primal indica como o aumento de produção afeta diretamente no lucro. As restrições do problema primal podem representar a disponibilidade de matéria-prima. Um aumento da disponibilidade de matéria-prima permite que haja um aumento de produção e por consequência de lucro, mas essa relação não é fácil de deduzir no problema primal. Um dos efeitos da dualidade é enxergar os efeitos que a mudança nas restrições causam no valor da função objetivo.

A dualidade também pode ser utilizada para desenvolver métodos eficientes de programação linear. Um outro método conhecido para resolução de problemas lineares é o *Métodos dos Pontos Interiores* que comumente utiliza uma combinação primal e dual para ser resolvido.

Apesar de ser possível definir um dual para qualquer problema linear, a simetria entre os dois é mais evidente quando postos na *forma canônica*. Um problema de minimização está na forma canônica quando todas as restrições são do tipo "$\geq$" e todas as variáveis são não negativas:

$$\begin{aligned} \min \quad & z = c^T x \\ \text{s.a} \quad & Ax \geq b \\ & x \geq 0. \end{aligned}$$

Esse problema será referido como Problema Linear *Primal*. O Problema Linar *Dual* correspondente será da forma

$$\begin{aligned} \max \quad & w = b^T y \\ \text{s.a} \quad & A^T y \leq c \\ & y \geq 0. \end{aligned}$$

Se o problema primal tiver n variáveis e m restrições, então o dual terá m variáveis (uma para cada restrição primal) e n restrições (uma para cada variável primal). Os coeficientes da função objetivo do problema primal serão os coeficientes do lado direito do problema dual, e vice-versa. A matriz de restrições no dual será a transposta da matriz do primal. O problema dual é um problema de maximização, onde todas as restrições são do tipo "$\leq$" e todas as variáveis são não negativas. Essa forma é considerada a *forma canônica* de um problema dual.

Abaixo um exemplo de Primal:

$$\begin{aligned} \min \quad & z = 6x_1 + 2x_2 - x_3 + 2x_4 \\ \text{s.a} \quad & 4x_1 + 3x_2 - 2x_3 + 2x_4 \geq 10 \\ & 8x_1 + x_2 - 2x_3 + 4x_4 \geq 18 \\ & x_1, x_2, x_3, x_4 \geq 0. \end{aligned}$$

E seu Dual:

$$\begin{aligned} \max \quad & w = 10y_1 + 18y_2 \\ \text{s.a} \quad & 4y_1 + 8y_2 \leq 6 \\ & 3y_1 + y_2 \leq 2 \\ & -2y_1 + 2y_2 \leq -1 \\ & 2y_1 + 4y_2 \leq 2 \\ & y_1, y_2 \geq 0. \end{aligned}$$

Nesse exemplo y_1 é a variável dual correpondente à primeira restrição do primal e y_2 é a variável dual correspondente à segunda restrição do primal. A primeira restrição do dual ($4y_1 + 8y_2 \leq 6$) corresponde à primeira variável do primal x_1; similarmente a segunda, terceira e quarta restrição para as variáveis primais x_2, x_3 e x_4 respectivamente.

Todo problema linear pode ser transformado em seu equivalente na forma canônica. Isso nos traz ao lema abaixo, descrito em [1].

Lema. O dual de um problema linear dual é o problema linear primal.

Apesar de ser possível determinar o dual de qualquer problema linear posto na forma canônica, algumas regras podem ser utilizadas para facilmente obter o dual diretamente do problema.

restrição primal/dual		**variável dual/primal**
consistente com a forma canônica	$\Longleftrightarrow$	variável ≥ 0
invertido com a forma canônica	$\Longleftrightarrow$	variável ≤ 0
igualdade	$\Longleftrightarrow$	variável irrestrita

Podemos visualizar melhor com o exemplo abaixo:

Primal:

$$\begin{aligned} \max \quad & z = 6x_1 + x_2 + x_3 \\ \text{s.a} \quad & 4x_1 + 3x_2 - 2x_3 = 1 \\ & 6x_1 - 2x_2 + 9x_3 \geq 9 \\ & 2x_1 + 3x_2 + 8x_3 \leq 5 \\ & x_1 \geq 0, x_2 \leq 0, x_3 \text{ irrestrito.} \end{aligned}$$

E seu dual:

$$\begin{aligned} \min \quad & w = y_1 + 9y_2 + 5y_3 \\ \text{s.a} \quad & 4y_1 + 6y_2 + 2y_3 \geq 6 \\ & 3y_1 - 2y_2 + 3y_3 \leq 1 \\ & -2y_1 + 9y_2 + 8y_3 = 1 \\ & y_1 \text{ irrestrito}, y_2 \leq 0, y_3 \geq 0. \end{aligned}$$

O problema primal é um problema de maximização. Sua primeira restrição é uma igualdade, e sua segunda e terceira restrição são, respectivamente, invertido e consistente com a forma canônica de um problema de maximização. Por esse motivo, y_1 é irrestrito, $y_2 \leq 0$, $y_3 \geq 0$. O problema dual é um problema de minimização. Já o problema dual é um problema de minimização. Como $x_1 \geq 0$ e $x_2 \leq 0$, as primeiras duas restrições serão, respectivamente, consistente e invertida com a forma canônica de um problema de mininimização. E como x_3 é irrestrito, a terceira restrição dual será uma igualdade.

Assim como o primal possui uma forma padrão o dual também possui forma padrão. Se o problema primal for

$$\begin{aligned} \min \quad & z = c^T x \\ \text{s.a} \quad & Ax = b \\ & x \geq 0, \end{aligned}$$

então seu dual

$$\begin{aligned} \max \quad & w = b^T y \\ \text{s.a} \quad & A^T y \leq c \\ & y \text{ irrestrito.} \end{aligned}$$

A formulação de problemas primais e duais são de extrema importância e resultam uma série de teoremas e corolários que validam os métodos de otimização existentes.

Teorema Fraco de Dualidade. Seja x um ponto viável do problema primal na forma padrão e seja y o ponto viável do problema dual. Então

$$z = c^T x \geq b^T y = w.$$

Corolário 1. Se o problema primal é ilimitado então seu dual é inviável. Se o dual é ilimitado então seu primal é inviável.

Corolário 2. Se x é uma solução viável do primal, então y é solução do dual, e $c^T x = b^T y$, então x e y são ótimos para seus respectivos problemas.

Teorema Forte de Dualidade. Considere um par de problemas primal e dual. Se um dos problemas possui solução ótima, então o outro também terá, e seus valores objetivos ótimos serão iguais.

Teorema das Folgas Complementares Considere um par de problemas primal e dual, sendo que o primal está na forma padrão. Se x é ótimo para o primal e y é ótimo para o dual, então $x^T(c - A^T y) = 0$. Se x é viável para o primal, y é viável para o dual e $x^T(c - A^T y) = 0$, então x e y são ótimos para seus respectivos problemas.

2.4 O Método Simplex Dual

Na versão apresentada do Método Simplex (Primal) começamos com uma solução básica viável e iteramos até que as condições primais de otimalidade sejam satisfeitas. O Método Simplex também pode ser aplicado no Dual, começando com uma solução viável dual e iterando até que as condições duais de otimalidade sejam satisfeitas.

As condições de otimalidade do primal correspondem à viabilidade das condições duais. O método simplex primal move-se por uma sequência de soluções viáveis primais mas inviáveis duais. A cada iteração procura-se diminuir a inviabilidade dual para que as condições do dual sejam satisfeitas.

O método simplex dual trabalha de forma análoga. Percorre-se uma sequência de soluções viáveis duais mas inviáveis primais e procura diminuir a inviabilidade primal a cada iteração até que as condições primais sejam satisfeitas.

Apesar de o método simplex dual parecer apenas uma versão do simplex primal aplicado ao problema dual, é possível implementar diretamente no método primal se houver alguma solução viável dual. E isso é fundamental quando não temos uma solução viável no Simples Primal.

Em primeiro lugar, realiza-se um teste de viabilidade. Precisamos de um $x_B = B^{-1}b \geq 0$ (onde $B(1) \ldots B(n)$ são os índices das variáveis básicas e $B = [A_{B(1)} \ldots A_{B(n)}]$ é a matriz básica), pois assim saberemos que a base tem uma solução. Caso contrário, trocamos a base para que a condição satisfaça.

Como a base é inviável para o primal, consideramos agora apenas o dual. Procuramos pelo, critério da razão, uma nova variável para entrar na base. Fazemos isso até que até obtermos uma base ótima e viável.

Vamos ilustrar melhor com o exemplo abaixo descrito em [4]:

$$\begin{aligned} \min \quad & z = 2x_1 + 3x_2 \\ \text{s.a} \quad & 3x_1 - 2x_2 \geq 4 \\ & x_1 + 2x_2 \geq 3 \\ & x_1, x_2 \geq 0. \end{aligned}$$

Adicionamos as variáveis de folga.

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

Considerando como base inicial $x_B = \begin{pmatrix} x_3 \\ x_4 \end{pmatrix}$, podemos multiplicar as restrições por -1. Assim obteremos A=$\begin{pmatrix} -3 & 2 & 1 & 0 \\ -1 & -2 & 0 & 1 \end{pmatrix}$.

O problema dual será

$$\begin{aligned} \max \quad & w = 4y_1 + 3y_2 \\ \text{s.a} \quad & 3y_1 + y_2 \leq 2 \\ & -2y_1 + 2y_2 \leq 3 \\ & y_1, y_2 \geq 0. \end{aligned}$$

Apesar de não ser necessário para a resolução do problema, vamos acompanhar os valores das soluções encontradas para melhor visualizarmos que o método simplex dual percorre uma sequência de soluções viáveis duais. A solução é dada por $y^T = c_B^T B^{-1}$. Para essa base temos que $y_1 = y_2 = 0$ e $w = 0$. Esse ponto é viável no dual mas não ótimo. Mantendo-se a *folga complementar* teremos que o valor objetivo será igual para primal e dual.

Voltando ao exemplo, temos que a base não é viável (no primal). Sendo assim retiramos a variável x_3 da base e inserimos x_1 no lugar dela. Lembrando que as regras de decisão são as mesmas do primal.

$$
\begin{array}{ll}
\min & z = \frac{13}{3}x_2 + \frac{2}{3}x_3 \\
\text{s.a} & x_1 - \frac{2}{3}x_2 - \frac{1}{3}x_3 \geq \frac{4}{3} \\
& -\frac{8}{3}x_2 - \frac{1}{3}x_3 + x_4 \geq -\frac{5}{3} \\
& x_1, x_2, x_3, x_4 \geq 0.
\end{array}
$$

Para essa base temos $y_1 = \frac{2}{3}$ e $y_2 = 0$.

Em seguida retiramos x_4 da base e colocamos x_2.

$$
\begin{array}{ll}
\min & z = \frac{1}{8}x_3 + \frac{13}{8}x_4 \\
\text{s.a} & x_1 - \frac{1}{4}x_3 - \frac{1}{4}x_4 \geq \frac{7}{4} \\
& x_2 + \frac{1}{8}x_3 - \frac{3}{8}x_4 \geq \frac{5}{8} \\
& x_1, x_2, x_3, x_4 \geq 0.
\end{array}
$$

Essa base é ótima e viável. A solução do dual é $y_1 = \frac{1}{8}$ e $y_2 = \frac{13}{8}$ e o valor ótimo é $\frac{43}{8}$ tanto para o primal quanto para o dual.

2.5 Sensibilidade

A Análise de Sensibilidade serve essencialmente para sabermos o que poderia mudar no resultado se as variáveis iniciais do problema fossem diferentes. Esse tipo de prática é bastante utilizada porque raramente sabemos os dados do problema com tanta precisão.

Por exemplo, se estamos estudando um problema de economia e uma das variáveis é a inflação em 6 meses, podemos apenas "adivinhar" o comportamento dessa variável. Entretanto, o objetivo pode ser satisfeito por apenas entender o efeito que o valor objetivo sofre ao alterar-se o lado direito de uma das restrições ou até mesmo os efeitos causados pela entrada de uma nova variável. Outra situação comum é quando uma das entradas do problema está dentro de um intervalo e queremos saber a solução do problema para todos os valores possíveis desse parâmetro.

A análise de sensibilidade procura obter as respostas para a situações descritas sem precisar refazer o problema. A ideia é partir das informações fornecidas pela base ótima para encontrar uma resposta.

Existem diversas situações em que a análise de sensibilidade faz-se necessária. Atualmente, grande parte dos softwares de programação linear mostram além do resultado a análise de sensbilidade.

Em geral, a análise de sensibilidade é utilizada nas seguintes situações:

- Mudança no lado direito
- Mudança em um coeficiente do objetivo (de uma variável não-básica)
- Novo coeficiente de restrição (de uma variável não-básica)
- Nova variável
- Nova restrição

3 Programação Linear Inteira

Muitas vezes, ao solucionarmos problemas lineares alcançamos soluções que não fazem sentido. Por exemplo, para maximizar lucros em um dado problema pode ser necessária a construção de 2.5[1] computadores. Parece pouco provável que alguém queira comprar metade de um computador. Quando soluções fracionárias acontecem, na prática, costumamos arredondar o número para que a solução torne-se mais aceitável. Em algumas situações isso pode ser feito, mas existem situações que arredondar deixa-nos muito longe do que seria realmente a solução ótima.

Os problemas que envolvem soluções inteiras podem ser divididos de 2 formas:

- Problemas em que todas a variáveis são Inteiras (Programa Linear Inteiro Puro - PLIP)

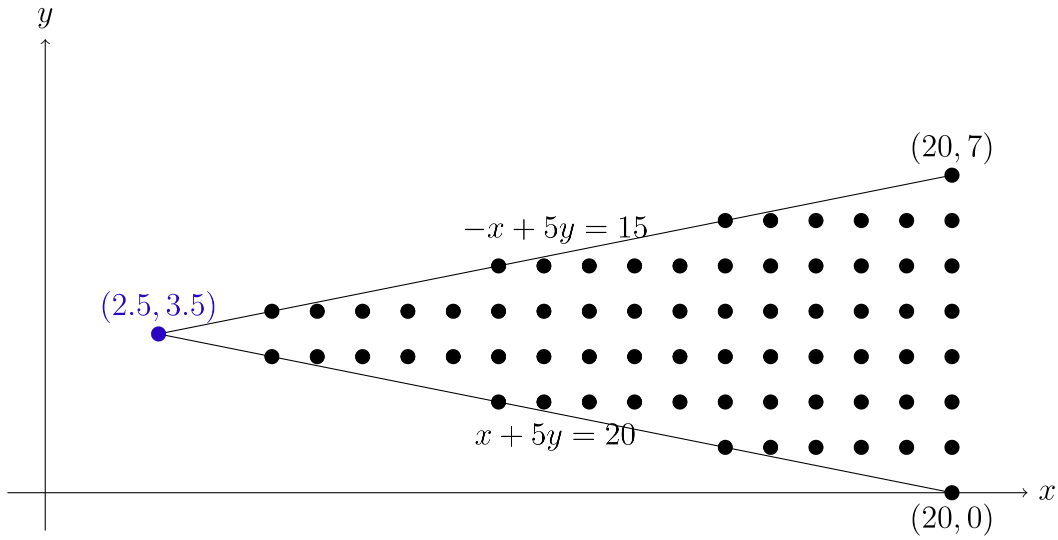
- Problemas em que pelo menos uma variável é Inteira e pelo menos uma é Não-Inteira (Programa Linear Inteiro Misto - PLIM)

Problemas Lineares Inteiros Puros e Mistos podem às vezes ser solucionados através do Método Simplex, mas geralmente necessitam de um método diferente para serem resolvidos. Vamos ilustrar pelo problema abaixo encontrado em [5]:

$$\begin{aligned} \text{min} \quad & z = 4x_1 + 4x_2 \\ \text{s.a} \quad & x_1 + 5x_2 \geq 20 \\ & -x_1 + 5x_2 \leq 15 \\ & x_1, x_2 \geq 0 \text{ e inteiros.} \end{aligned}$$

A figura abaixo representa graficamente o problema. Os pontos pretos representam pontos de coordenadas inteiras.

[1] Utilizaremos . como separado de decimal de um número



Se ignorarmos a restrição de que x_1 e x_2 devem ser inteiros temos o chamado *PL Relaxado* do problema original, que pode ser resolvido através do método simplex. Se o problema for resolvido pelo método simplex encontramos o valor de $z = 24$ onde $x_1 = 2.5$ e $x_2 = 3.5$. Vamos arredondar esse ponto para $(3, 3)$. Esse ponto não é viável e não satisfaz a primeira restrição do problema original. Isso mostra claramente um problema que podemos ter com arredondamentos. Claro que poderíamos ter arredondado o par de soluções de outra forma. Poderíamos ter $(2, 3)$, $(2, 4)$, $(3, 3)$ ou $(3, 4)$. Uma checagem rápida mostra que nenhum desses pares é viável. Com isso percebemos que a solução ótima pode não ser facilmente encontrada através de arredondamento da solução relaxada. Neste exemplo temos que o par ótimo é $(5, 3)$ e que $z = 32$, consideravelmente maior do que o valor dado pela relaxação. Ainda nesse exemplo, percebemos que a solução ótima encontra-se razoavelmente perto da solução relaxada.

Caso utilizássemos o problema

$$\begin{array}{rl} \min & z = 58x_1 + 200x_2 \\ \text{s.a} & 12x_1 + 40x_2 \leq 87 \\ & x_1, x_2 \geq 0 \text{ e inteiros.} \end{array}$$

a solução do problema relaxado é $(0, 2.175)$ onde $z = 435$. Entretanto, a verdadeira solução ótima encontra-se em $(7, 0)$, que está bem longe da solução relaxada. Com isso percebemos que a solução arredondada do PL Relaxado pode ser viável mas não ótima; a solução ótima não precisa estar nas redondenzas da solução relaxada e; mesmo que alguns dos valores ótimos relaxados sejam inteiros, não precisam estar necessariamente próximos dos valores ótimos do problema original.

3.1 Exemplos de Problemas de Programação Linear Inteira

Nesta seção vamos apresentar alguns problemas comuns de Programação Linear Inteira. Mais especificamente iremos olhar os casos em que as variáveis são binárias, tomando apenas os valores 0 ou 1. Essa denotação é bastante utilizada quando precisamos tomar uma decisão. Por exemplo, sair cedo de casa ou não, comprar determinado produto ou não, levar o guarda-chuva ou não, onde 1 costuma representar o "sim" e 0 o "não".

3.1.1 O Problema da Mochila [2]

Um alpinista dispõe de uma mochila e diversos objetos que podem ser carregados na mesma. Cada objeto possui um peso (p_i) e um valor de utilidade (v_i). Sabendo a capacidade máxima da mochila em termos de peso máximo, quais objetos dentre os n existentes devem ser escolhidos para que possamos ter o maior proveito?

As variáveis de decisão são:

$$x_i = \begin{cases} 1, \text{ se o objeto for escolhido} \\ 0, \text{ se o objeto não for escolhido} \end{cases} \quad \text{, onde } i = 1, \ldots, n$$

Então, teremos o modelo de PLI (Problema Linear Inteiro):

$$\begin{array}{rl} \max & z = \sum_{i=1}^{n} v_i x_i \\ \text{s.a} & \sum_{i=1}^{n} p_i x_i \leq P_{max} \\ & x_i \in \{0, 1\} \end{array}$$

3.1.2 O Problema de Alocação [2]

O problema de alocação é uma classe de problemas onde é necessário designar pessoas a horários, lugares, tarefas, zonas de trabalho, máquinas, etc. Aparece muitas vezes como se tratasse de um PL mas suas variáveis de decisão são binárias.

Suponha que precisamos alocar n indivíduos a n tarefas, sabendo que a medida de eficiência de alocar o indivíduo i à tarefa j é c_{ij} (que tanto pode representar um lucro quanto um custo). Precisamos determinar a alocação dos indivíduos para que haja otimização da eficiência total.

As variáveis de decisão são:

$$x_{ij} = \begin{cases} 1, \text{ se o indivíduo } i \text{ for alocado à tarefa } j \\ 0, \text{ se o indivíduo } i \text{ } \underline{\text{não}} \text{ for alocado à tarefa } j \end{cases} \quad \text{, onde } i = 1, \ldots, n \text{ e } j = 1, \ldots, n$$

Então, teremos o modelo de PLI (Programação Linear Inteira):

$$
\begin{array}{rll}
\text{min ou max} & z = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}x_{ij} & \\
\text{s.a} & \sum\limits_{j=1}^{n} x_{ij} = 1, & \forall_{i:\ i=1,\dots,n} \\
 & \sum\limits_{i=1}^{n} x_{ij} = 1, & \forall_{j:\ j=1,\dots,n} \\
 & x_{ij} \in \{0,1\} &
\end{array}
$$

A primeira restrição diz que cada indivíduo pode ser alocado para apenas uma tarefa. Já a segunda restrição diz que cada tarefa só pode ser feita por apenas um indivíduo.

Trata-se de um modelo bastante simples cujo sistema de restrições tem uma estrutura particular com certas propriedades. Situações em que o número de indivíduos é diferente do número de tarefas também pode ser representada por esse modelo.

3.1.3 O Problema do Caixeiro Viajante [2]

O problema do caixeiro viajante é um típico PLI onde o caixeiro precisa visitar cada uma de n cidades apenas uma vez e retornar ao seu ponto de partida. Basicamente o caixeiro precisa encontrar um circuito que liga todas as cidades, passando por todas apenas uma vez e terminando sua jornada no vértice de partida, onde as cidades correspondem à vértices ou nós de uma rede e os arcos representam as ligações entre as cidades. O tempo necessário para ir da cidade i a cidade j é dado por c_{ij}.

Podemos definir as variáveis de decisão por:

$$
x_{ij} = \begin{cases} 1, \text{ se o caixeiro visita a cidade } j \text{ imediatamente após a cidade } i \\ 0, \text{ c.c} \end{cases}
$$

onde $i = 1, \dots, n$ e $j = 1, \dots, n$

Então, teremos o modelo do PLI:

$$
\begin{array}{rll}
\text{min} & z = \sum\limits_{i=1}^{n}\sum\limits_{j=1}^{n} c_{ij}x_{ij} & \\
\text{s.a} & \sum\limits_{j=1}^{n} x_{ij} = 1, & \forall_{i:\ i=1,\dots,n} \\
 & \sum\limits_{i=1}^{n} x_{ij} = 1, & \forall_{j:\ j=1,\dots,n} \\
 & \sum\limits_{i\in S}\sum\limits_{j\notin S} x_{ij} \geq 1, & \forall_{S\ \subset\ \text{conjunto total de cidades a visitar}} \\
 & x_{ij} \in \{0,1\} &
\end{array}
$$

As primeiras duas restrições garantem que as cidades são visitadas uma e apenas uma vez (Veja Figura 1). Já a terceira restrição garante que o circuito a ser percorrido ligue todas as cidades (Veja Figura 2).

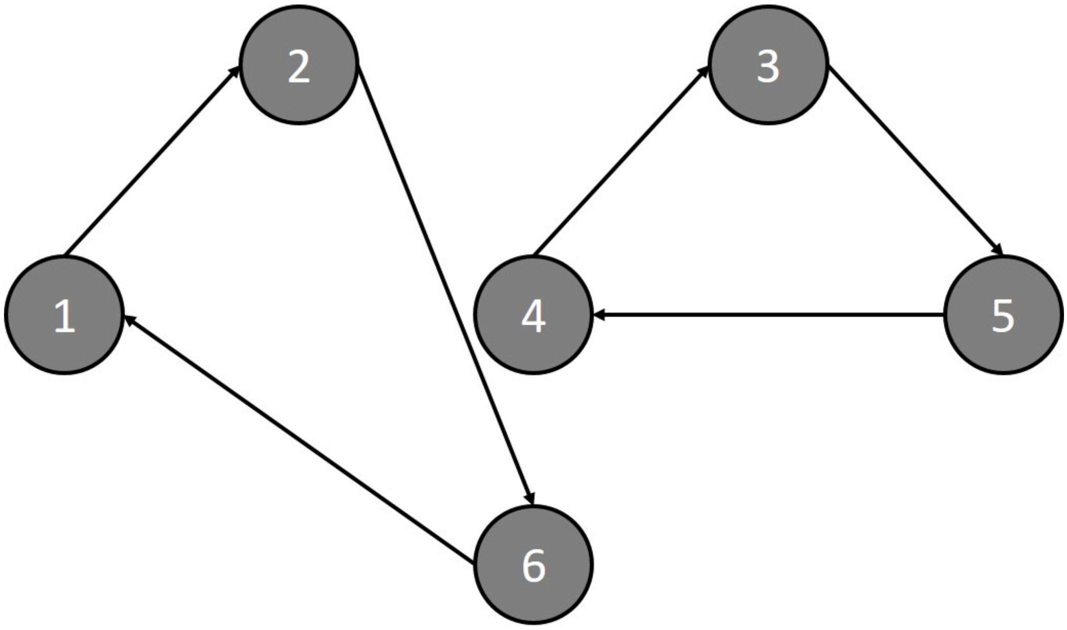


Figura 1: Caso em que apenas as duas primeiras restrições são satisfeitas

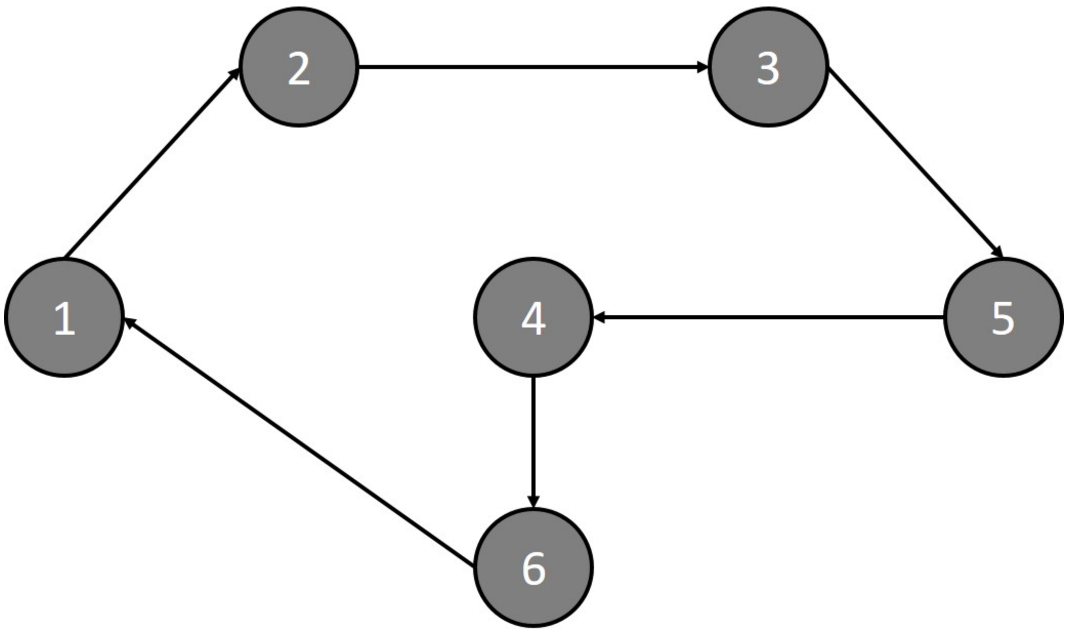


Figura 2: Caso em que todas as restrições são satisfeitas

3.2 Métodos de Resolução de um PLI

Existem 2 grandes categorias de algoritmos para resolução de um PLI:

1. **Algoritmos Exatos:** Garantem que a solução ótima seja encontrada, mas podem executar iterações exponencialmente

2. **Algoritmos Heurísticos:** Trazem soluções quase ótimas ou ótimas com um corte de garantia na qualidade da solução

Para este trabalho estaremos interessados apenas em algoritmos exatos, mais especificamente no Método de Partição e Avaliações Sucessivas (*Branch and Bound*). Falaremos agora rapidamente sobre outro método exato, o Método de Planos de Corte.

3.2.1 Método de Planos de Corte

Esse método consiste em introduzir sucessivamente novas restrições na relaxação linear do PLI. Essas restrições cortam o conjunto possível de soluções eliminando algumas delas e a própria solução ótima do PL, mas sem eliminar qualquer solução inteira.

De modo geral, consideramos o problema inteiro

$$\begin{array}{rl} \min & z = c^T x \\ \text{s.a} & Ax = b \\ & x \geq 0 \\ & x \text{ inteiro} \end{array}$$

e sua forma relaxada

$$\begin{array}{rl} \min & z = c^T x \\ \text{s.a} & Ax = b \\ & x \geq 0 \end{array}$$

De maneira geral temos o algoritmo para aplicação do método:

1. Resolver o problema relaxado, encontrando x^* como sua solução ótima.

2. Se x^* for inteiro, então parar.

3. Se não, adicione uma restrição linear de desigualdade no problema relaxado que satisfaça todas as restrições do PLI mas que x^* não satisfaça. Volte ao primeiro passo.

3.2.2 Método de Partição e Avaliação Sucessiva

O método "*Branch and Bound*" (literalmente, método de ramificação e limitação) consiste na partição (ramificação) sucessiva de soluções possíveis do PLI em subconjuntos e na limitação (avaliação) do valor ótimo da função objetivo (limite inferior caso seja uma maximização e limite superior para minimização), de modo a excluir os subconjuntos que não contenham a solução ótima.

Se tomarmos como verdade que *se, na solução ótima da relaxação linear do PLI as variáveis tomarem valor inteiros, então essa solução será a solução ótima do PLI*, podemos começar a resolver o problema pela relaxação linear do PLI inicial.

Com isso, resolvemos o problema relaxado pelo método Simplex Dual, descrito anteriormente. Se as soluções forem inteiras, então encontramos a solução ótima do PLI. Caso contrário, dividimos esse PL em dois. Cada uma das partes é uma cópia do PL original mas que contempla uma restrição adicional, que é responsável por particionar o conjunto de soluções possíveis para cada um dos problemas. Então, sempre aplicando o método Simplex Dual, resolvem-se os sucessivos problemas do PL, estabelecendo-se novos limites para o valor ótimo da função objetivo e, assim, eliminando diversos subconjuntos até alcançar a solução ótima do PLI.

De modo geral, o método é resolvido por:

Seja o PLI: max $z = c^T x$ sujeito a: $\{Ax = b$, $x \geq 0$, x_j inteiro para $j \in I$, com $I =$ conjunto das variáveis inteiras$\}$.

1. Resolva por Simplex Dual o PLI sem as restrições de integralidade (resolva o problema relaxado). Denote PL_1 com solução ótima z_1.
   Se todo x_j for inteiro para $j \in i \rightarrow$ *solução ótima* para o PLI então parar o algoritmo.
   Caso contrário: há x_j fracionário para $j \in i$ e z_1 é um limitante superior para o valor ótimo da função objetivo do PLI.

2. Particionar a região viável do PL_1 escolhendo algum x_j fracionário para $j \in I$, procurando achar soluções inteiras que serão limitantes inferiores para o z do PLI.
   Suponha que x_j seja selecionado para realizar a partição e que a_j é seu valor fracionário. Criamos dois novos problemas PL_2 e PL_3 e introduzimos as restrições $x_j \leq \lfloor a_j \rfloor$ e $x_j \geq \lceil a_j \rceil$, respectivamente. Onde $\lfloor a_j \rfloor$ é o *maior inteiro menor* que a_j e $\lceil a_j \rceil$ é o *menor inteiro maior* que a_j.

3. Resolver o PL_2 e o PL_3 e encontrar suas soluções ótimas. Se uma dessas soluções forem inteiras, adotar o valor ótimo como novo limitante inferior para o valor máximo de z do PLI original.

4. Desenvolver novas ramificações para PL_2 e PL_3, se ncessário.

5. Repetir até que todos os PLs tenham sido explorados. O PL com maior valor de z corresponderá à solução ótima do PLI.

O exemplo abaixo, descrito em [2], mostra de forma mais gráfica o funcionamento do método de partição e avaliação sucessiva:

Suponha que um carpinteiro produza 2 tipos de brinquedo: cavalos-de-pau e trenzinhos de madeira. O lucro de cada um dos itens é de R$2400 e R$1500. Cada cavalo requer 1 hora de trabalho e 9m^2 de madeira, enquanto cada trenzinho requer 1 hora de trabalho e 5m^2 de madeira. Supondo que estão disponíveis 6 horas de trabalho e 45m^2 de madeira, que quantidades fabricar de forma a maximizar o lucro?

Adotamos x_1 como o número de cavalos fabricados e x_2 o número de trenzinhos fabricados. O PLI será:

$$
\begin{array}{lrcrcrcll}
\max & z & = & 2400x_1 & + & 1500x_2 & & & \\
\text{s.a} & & & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
 & & & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
 & & & & & x & \geq & 0 \text{ e inteiros} &
\end{array}
$$

A solução ótima do problema relaxado é $x = (x_1, x_2) = (3.75, 2.25)$ e $z = 12375$.

Sabemos então que o valor ótimo não pode ultrapassar 12375. Como a solução ótima encontrada não é inteira, precisamos efetuar a partição, dando origem a dois novos problemas PL_2 e PL_3. Assim, introduzimos novas restrições que eliminam algumas das soluções não-inteiras: $x_1 \leq 3$ e $x_1 \geq 4$. (Poderíamos ter escolhido a variável x_2, mas vamos prosseguir com a variável x_1.)

$$
\begin{array}{llrcrcrcll}
\mathbf{PL_2:} & \max & z & = & 2400x_1 & + & 1500x_2 & & & \\
 & \text{s.a} & & & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
 & & & & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
 & & & & x_1 & & & \leq & 3 & \text{(nova restrição)} \\
 & & & & & & x & \geq & 0 &
\end{array}
$$

$$
\begin{array}{llrcrcrcll}
\mathbf{PL_3:} & \max & z & = & 2400x_1 & + & 1500x_2 & & & \\
 & \text{s.a} & & & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
 & & & & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
 & & & & x_1 & & & \geq & 4 & \text{(nova restrição)} \\
 & & & & & & x & \geq & 0 &
\end{array}
$$

A solução do subproblema PL_2 é inteira ($x_1 = 3$ e $x_2 = 3$), o que significa que encontrou-se uma solução inteira cujo valor da função objetivo é 11700. O valor ótimo da função objetivo estará compreendido entre 11700 e 12375.

A solução do subproblema PL_3 é (4, 1.8) e não é inteira. Entretanto, o valor da função objetivo é 12300 (>11700), este subproblema pode conter uma solução inteira melhor que a do subproblema PL_2. Logo, é necessário realizar a sua partição, dando origem aos subproblemas PL_{31} e PL_{32}, pela introdução das restrições $x_2 \leq 1$ e $x_2 \geq 2$, respectivamente.

Os novos subproblemas são da forma:

$$\begin{array}{llrcrcll}
\mathbf{PL_{31}:} & \max\ z = & 2400x_1 & + & 1500x_2 & & & \\
& \text{s.a} & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
& & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
& & x_1 & & & \geq & 4 & \\
& & & & x_2 & \leq & 1 & \text{(nova restrição)} \\
& & & & x & \geq & 0 &
\end{array}$$

$$\begin{array}{llrcrcll}
\mathbf{PL_{32}:} & \max\ z = & 2400x_1 & + & 1500x_2 & & & \\
& \text{s.a} & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
& & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
& & x_1 & & & \geq & 4 & \\
& & & & x_2 & \geq & 2 & \text{(nova restrição)} \\
& & & & x & \geq & 0 &
\end{array}$$

O subproblema PL_{32} não tem soluções possíveis, sendo portanto excluído.

O subproblema PL_{31} possui solução $x = (x_1, x_2) = (4.4, 1)$ e $z = 12167$. Como $12167 > 11700$, dividimos em 2 subproblemas, adicionando as restrições $x_1 \leq 4$ e $x_1 \geq 5$. Então temos:

$$\begin{array}{llrcrcll}
\mathbf{PL_{311}:} & \max\ z = & 2400x_1 & + & 1500x_2 & & & \\
& \text{s.a} & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
& & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
& & x_1 & & & \geq & 4 & \\
& & & & x_2 & \leq & 1 & \\
& & x_1 & & & \leq & 4 & \text{(nova restrição)} \\
& & & & x & \geq & 0 &
\end{array}$$

Essa nova restrição em conjunto com a restrição $x_1 \geq 4$ resulta em $x_1 = 4$.

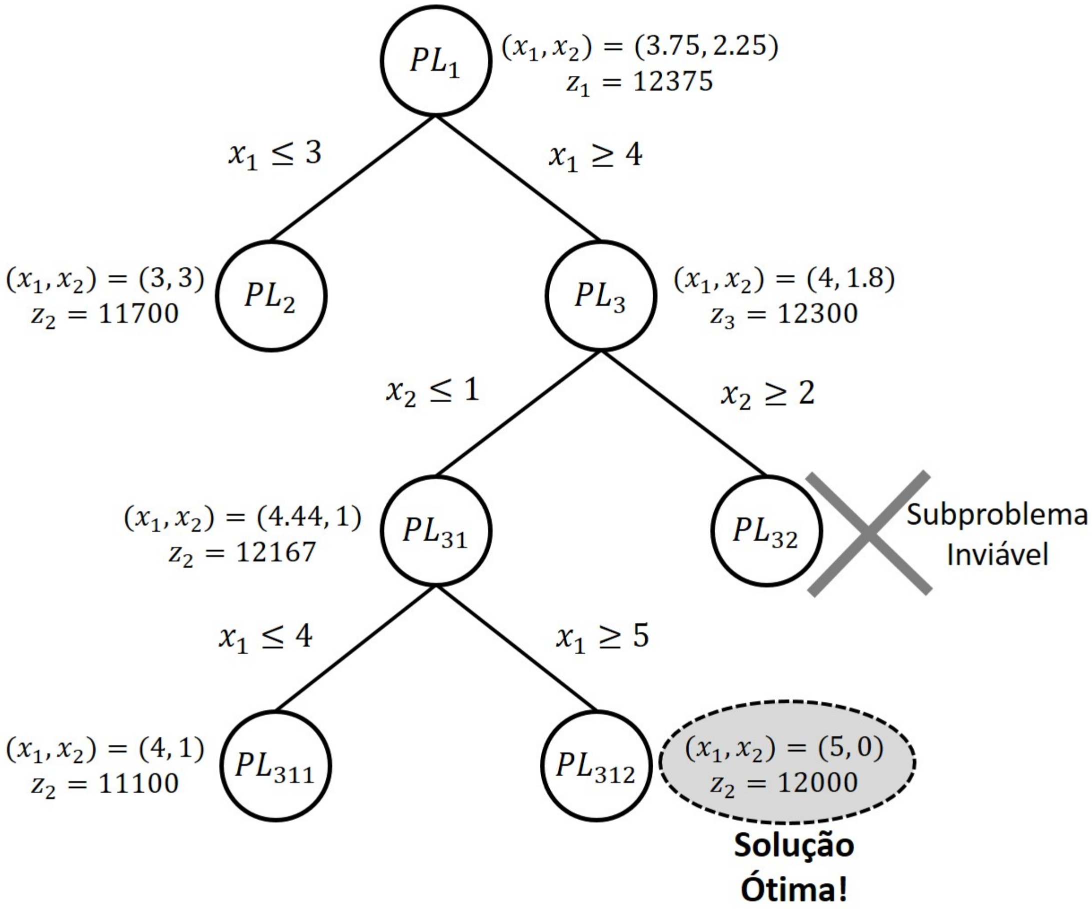


Figura 3: Diagrama que representa a resolução por partições do problema apresentado

$$
\begin{array}{llrcrcll}
\mathbf{PL_{312}:} & \max\ z = & 2400x_1 & + & 1500x_2 & & & \\
 & \text{s.a} & x_1 & + & x_2 & \leq & 6 & \text{(horas de trabalho)} \\
 & & 9x_1 & + & 5x_2 & \leq & 45 & \text{(madeira)} \\
 & & x_1 & & & \geq & 4 & \\
 & & & & x_2 & \geq & 2 & \\
 & & x_1 & & & \geq & 5 & \text{(nova restrição)} \\
 & & & & x & \geq & 0 &
\end{array}
$$

Essa nova restrição sobrepõe a restrição $x_1 \geq 4$ deixando apenas a restrição nova ($x_1 \geq 5$).

Os subproblemas PL_{311} e PL_{312} possuem soluções inteiras. Entretanto, o valor ótimo do PL_{311} é $11100 < 11700$, ou seja, pior do que a solução que já tínhamos. Já o valor ótimo do subproblema PL_{312} é $12000 > 11700$. Assim, nossos limites superior e inferior serão iguais a 12000, tornando esta a solução ótima do PLI.

Na Figura 3 podemos ver a diagramação do problema.

À medida que se vai "descendo" na árvore vão se atualizando os limites inferior e superior do valor ótimo da função objetivo (F^*) do PLI. No nó incial (raiz da árvore), $11700 < F^* < 12167$. Por fim, no quarto e último nível, $12000 < F^* < 12000$. Concluindo então que $x = (x_1, x_2) = (5, 0)$, $F^* = 1200$ é a solução ótima, não precisando realizar nenhuma outra partição.

Com isso, vemos que:

1. A partição de um subproblema é feita quando a solução ótima do subproblema possuir pelo menos uma variável não-inteira e seu subproblema puder conter uma solução inteira melhor que a já existente.

2. Eliminam-se os subproblemas que não tenha soluções possíveis (problema inviável) ou que não possam conter uma solução melhor do que a já existente.

4 O problema de atribuição de encargos didáticos

Como citado na Introdução, o problema de atribuição didática é solucionado através de Modelagem Matemática que tem como proposta maximizar a satisfação dos professores. O modelo escolhido recai em um problema de programação linear binária. Em [6] utiliza diversos métodos (exato, heurístico e misto) e os compara, não chegando a conclusão de um método predileto. Utilizaremos como base o trabalho realizado em [7].

Os dados de entrada são baseados nas informações que a comissão utiliza quando realiza a atribuição manualmente.

4.1 Conjuntos

Para modelagem matemática, cada professor recebe uma identificação. Essa identificação é em ordem sequencial conforme ordem alfabética. Analogamente temos a identificação das turmas, mas a ordem é realizada por código da disciplina. Sendo assim, temos os seguintes conjuntos no problema:

$$\mathcal{P} = \{1, 2, \ldots, P\};$$

$$\mathcal{T} = \{1, 2, \ldots, T\};$$

$$\mathcal{H} = \{1, 2, \ldots, H\};$$

$$\mathcal{D} = \{1, 2, \ldots, D\};$$

Sendo,

- $P = 74$ a quantidade total de professores,
- $T = 144$ a quantidade total de turmas,
- $H = 93$ a quantidade total de intervalos de 10 minutos entre 07:30 e 22:50,
- $D = 5$ a quantidade total de dias letivos de uma semana.

4.2 Variável de Decisão

Como estamos procurando atribuir professores às turmas de modo que a satisfação geral seja maximizada, é natural que a variável de decisão seja representada por uma relação professor-turma. Se definirmos essa variável como binária, teremos:

$$x(p,t) = \begin{cases} 1, & \text{se o professor } p \text{ assume a turma } t, \\ 0, & \text{c.c.} \end{cases} \tag{4.1}$$

Considera-se $X = \{x(p,t) \mid p \in \mathcal{P} \text{ e } t \in \mathcal{T}\}$ o conjunto com todas as variáveis definidas em (4.1).

4.3 Formulação do problema

Nesta seção vamos definir o problema, identificando a função objetivo, as restrições, as variáveis e por fim o modelo geral desse problema.

4.3.1 Função Objetivo

Como descrito anteriormente, a função objetivo deste trabalho procura maximizar satisfação de professores. Sendo e uma matrix $P \times T$, cujos elementos representem pesos de 1 a 5 associados a satisfação[1] dos professores e X o conjunto de variáveis descrito em (4.1), a função objetivo pode ser escrita dessa forma,

$$S(x) = \sum_{p \in \mathcal{P}} \sum_{t \in \mathcal{T}} e(p,t)x(p,t) \tag{4.2}$$

4.3.2 Restrições

Restrições em otimização são mais do que comuns, e aqui discutem-se algumas restrições encontradas para este problema. As restrições aqui apresentadas não são regra, e podem variar de instituição para instituição.

4.3.2.1 Indisponibilidade de Horários

Para este tipo de problema, não há como esperar que os professores não tenham restrições em seus horários. Seja por motivos pessoais ou agendas profissionais pré-definidas, deve-se levar em conta este fator para cada professor.

O método utilizado para transpor o problema foi "forçar" que o modelo não atribuísse as turmas restritivas ao professor. Para tal multiplicou-se negativamente um M

[1] $e(p,t) = 5$ é a maior satisfação do professor p para ministrar a turma t e $e(p,t) = 1$ é a menor satisfação do professor p para ministrar a turma t

suficientemente grande na função objetivo. Deste modo, o algoritmo evita atribuir a disciplina ao professor para não causar uma diminuição expressiva no valor ótimo.

4.3.2.2 Carga Horária

Cada instituição possui normas quanto a quantidade de horas que seus professores devem trabalhar. Para efeito deste trabalho, considerou-se que um professor não pode lecionar menos que 2 créditos, ou seja, 100 minutos e não deve ultrapassar 18 créditos, ou seja, 15 horas.

Como criamos $\mathcal{H}$ com intervalos de 10 minutos, temos $H_m = 10$ a quantidade mínima de horários que cada professor deve ministrar. Analogamente, temos $H_M = 90$ como a quantidade máxima.

4.3.2.3 Turmas

Em algumas instiuições, é permitido que mais de um professor lecione uma mesma turma. Para este trabalho isso não será considerado e cada turma deve conter 1, e somente 1, professor atribuído à mesma.

4.4 Modelo Geral

Tendo em mãos a função objetivo e todas as restrições descritas acima, pode-se formular o modelo geral a ser utilizado neste trabalho. Considerando

$$HT(t,h,d) = \begin{cases} 1, & \text{se a turma } t \text{ tem aula no horário } h \text{ do dia } d, \\ 0, & \text{c.c.} \end{cases} \tag{4.3}$$

então temos a seguinte formulação:

$$\text{maximizar} \quad \sum_{p=1}^{P}\sum_{t=1}^{T} e(p,t)x(p,t) \tag{4.4}$$

sujeito a

$$\sum_{t=1}^{T} HT(t,h,d)x(p,t) \leq 1, \qquad \text{para todo } p,\ d,\ h, \tag{4.5}$$

$$\sum_{p=1}^{P} x(p,t) = 1, \qquad \text{para todo } t, \tag{4.6}$$

$$\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{d=1}^{D} HT(t,h,d)x(p,t) \geq H_m(p), \qquad \text{para todo } p, \tag{4.7}$$

$$\sum_{t=1}^{T}\sum_{h=1}^{H}\sum_{d=1}^{D} HT(t,h,d)x(p,t) \leq H_M(p), \qquad \text{para todo } p, \tag{4.8}$$

$$x(p,t) = \{0,1\}, \qquad \text{para todo } p, t. \tag{4.9}$$

As restrições (4.5) exigem que cada professor não pode assumir mais do que uma turma num mesmo horário de um determinado dia. As restrições (4.6) garantem que toda turma deve ter 1 e, apenas 1, professor. As restrições (4.7) e (4.8) impõem as cargas mínimas e máximas de cada professor, respectivamente. Finalmente, em (4.9), definimos as variáveis como binárias.

5 Estudo de Caso

5.1 Coleta de Dados

Apesar deste trabalho ter como intuito facilitar o trabalho dos responsáveis por atribuição de disciplinas, por questões éticas não foi possível obter os dados mais recentes existentes. Sendo assim, foram utilizados os formulários preenchidos pelos professores em 2011 para a competência do ano 2012. A grade horária utilizada também foi a de 2012.

Neste capítulo será discutido o caso específico do Departamento de Matemática do IME-USP, onde as turmas a serem designadas são de cursos diversos.

O período letivo nessa instituição é em grande parte semestral e em sua minoria anual. Para este estudo consideraremos apenas as disciplinas semestrais das graduações.

Os dados utilizados para esse trabalho são referentes ao ano de 2012, que apesar de não atual, adequa-se ao problema genericamente. Como ponto de partida, apenas as disciplinas da sigla MAT do primeiro semestre de 2012 são consideradas.

5.2 Matlab R2014a

O Matlab é um software interativo de alta performance cujo elemento básico de informação é uma matriz que não requer dimensionamento. Esse sistema permite a resolução de muitos problemas numéricos em apenas uma fração do tempo que se gastaria para escrever um programa semelhante em linguagem Fortran, Basic ou C. Além disso, as soluções dos problemas são expressas quase exatamente como elas são escritas matematicamente.

Escolheu-se utilizar a versão R2014a por conter a função *intlinprog* que abrange a função *bintprog* presente em versões anteriores. Essa nova função permite otimizar problemas inteiros como um todo, e não apenas os binários. Apesar desse estudo não utilizar variáveis inteiras não binárias, foi considerada uma função melhor e mais eficiente que sua predecessora.

5.3 O Departamento de Matemática – IME-USP

No Departamento de Matemática do IME-USP a atribuição dos encargos didáticos é realizada ao final do semestre anterior.

O Departamento envia, no final do ano, um formulário aos professores para que preencham quais disciplinas desejam ministrar e em quais prioridades. Respondidos os formulários, enviam ao Departamento que entrega ao professor responsável por realizar a atribuição.

No momento de preenchimento dos formulários os horários das disciplinas ainda não se encontram definidos. Os professores tomam como base o ano anterior, pois poucas mudanças são realizadas a cada ano.

5.4 Turmas ofertadas no semestre

Como o estudo foi realizado com dados de 2012, a atribuição real de turmas e disciplinas pôde ser utilizada na elaboração do problema. As disciplinas do Departamento de Matemática totalizaram 244 turmas, distribuídas em diversas instituições e não apenas no IME-USP.

As cargas das disciplinas variam entre 2, 4 e 6 créditos, onde cada crédito equivale a 50 minutos. As turmas podem ser nos períodos da manhã, tarde e noite, não havendo padronização entre os institutos.

5.5 Professores

Em 2012 houve um total de 74 professores no Departamento de Matemática do IME-USP. Muitos ajustes quanto à preferência dos professores tiveram que ser realizados por conta de mau preenchimento ou até mesmo não preenchimento dos formulários.

Para aqueles que não preencheram o formulário, foi considerado que o professor tinha prioridade máxima em todas as turmas ofertadas. Para os demais casos, a turma que não estivesse na prioridade do professor entraria com prioridade mínima.

Quanto à carga horária dos professores, considerou-se que nenhum poderia ministrar menos de 2 créditos no semestre e também não poderia ministrar mais que 18 créditos nesse período.

5.6 Considerações

Devido ao tamanho do processo e das matrizes de resultado do problema utilizaremos um problema fictício mais simples para uma análise profunda dos resultados finais, validando assim a formulação e resolução do problema original.

6 O Outro Estudo de Caso

Ainda no universo das grades horárias, vamos supor que uma instituição de ensino menor esteja com dificuldades em atribuir disciplinas a seus professores.

Para este exemplo-caso, utilizaremos uma escola que dispõe apenas de 7 professores, 15 turmas, 6 aulas por dia e 5 dias letivos. Então temos:

$$\begin{aligned}
\mathcal{P} &= \{1,2,3,4,5,6,7\};\\
\mathcal{T} &= \{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15\};\\
\mathcal{H} &= \{1,2,3,4,5,6\};\\
\mathcal{D} &= \{1,2,3,4,5\};
\end{aligned}$$

Continuaremos com

$$x(p,t) = \begin{cases} 1, & \text{se o professor } p \text{ assume a turma } t,\\ 0, & \text{c.c.} \end{cases}$$

como nossa variável de decisão.

E teremos a mesma função objetivo

$$S(x) = \sum_{p\in\mathcal{P}} \sum_{t\in\mathcal{T}} e(p,t)x(p,t)$$

O problema segue pelos mesmos padrões que o problema real, exceto pela representação crédito-aula. Nesse exemplo não utilizaremos minutos como unidade de medida, e sim quantidade de aulas. Cada aula representará 2 créditos. Também para adequação ao exemplo, teremos novos valores para as quantidades máxima e mínima de créditos exigidos.

O modelo geral será exatamente o mesmo do problema original.

Para resolução do problema, o mesmo código de MATLAB utilizado foi adaptado para simplificar as matrizes de entrada.

6.1 Visualizações do problema

Vamos considerar a grade horária da Figura 4 para o problema. Segue um exemplo de como lê-se a tabela: Turma 1 possui aulas às segundas no primeiro e segundo horários e às sextas também no primeiro e segundo horários.

<table>
<tr><th></th><th></th><th colspan="2">SEG (D1)</th><th colspan="2">TER (D2)</th><th colspan="2">QUA (D3)</th><th colspan="2">QUI (D4)</th><th colspan="2">SEX (D5)</th></tr>
<tr><td rowspan="2">Manhã</td><td>H1</td><td>1</td><td>2</td><td>5</td><td>3</td><td>7</td><td>6</td><td colspan="2">4</td><td>2</td><td>1</td></tr>
<tr><td>H2</td><td>1</td><td>7</td><td>5</td><td>4</td><td>2</td><td>6</td><td colspan="2">3</td><td>7</td><td>1</td></tr>
<tr><td rowspan="2">Tarde</td><td>H3</td><td colspan="2">10</td><td>5</td><td>8</td><td colspan="2">9</td><td colspan="2">11</td><td colspan="2"></td></tr>
<tr><td>H4</td><td colspan="2">11</td><td>5</td><td>8</td><td colspan="2">10</td><td colspan="2"></td><td colspan="2">11</td></tr>
<tr><td rowspan="2">Noite</td><td>H5</td><td colspan="2">12</td><td>5</td><td>14</td><td colspan="2">12</td><td colspan="2">15</td><td colspan="2">13</td></tr>
<tr><td>H6</td><td colspan="2">13</td><td colspan="2">5</td><td colspan="2">13</td><td>15</td><td>14</td><td colspan="2">12</td></tr>
</table>

Figura 4: Grade Horária das Turmas da escola

	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
P1	4	3	5	5	1	5	4	1	1	1	1	1	1	1	1
P2	1	1	1	1	2	1	1	3	3	3	3	5	5	4	4
P3	3	3	2	2	4	2	3	2	4	3	4	4	4	4	4
P4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
P5	1	3	4	4	3	4	3	4	5	3	2	2	2	4	4
P6	3	2	5	3	1	2	5	3	2	5	3	2	4	5	5
P7	5	1	1	1	5	1	1	1	1	1	1	1	1	1	1

Figura 5: Preferência de cada professor para lecionar cada turma

As preferências dos professores estão descritas na Figura 5. Para cada professor temos 15 números de 1 a 5 que representam o desejo do professor de lecionar a turma correspondente. Por exemplo, o professor 1 deseja fortemente (nota atribuída 5) as turmas 3, 4 e 6, enquanto não deseja lecionar as turmas 5, 8, 9, 10, 11, 12, 13, 14 e 15 (nota atribuída 1). Lembrando que o não desejo de lecionar determinada turma não o impede de ser designado a uma delas no resultado final.

Dado isso, podemos iniciar o processo de otimização no MATLAB.

6.2 Implementação

Para resolução de um problema inteiro, no Matlab utiliza-se a função *intlinprog*. Esta função é utilizada pelo comando:

$$\text{intlinprog}(f, \text{intcon}, A, b, Aeq, beq, lb, ub)$$

Onde,

f é o vetor dos coeficientes da função objetivo

intcon é o vetor que indica quais variáveis são inteiras. No caso aplicado, todas as variáveis são inteiras, logo o intcon[i] = i onde i = $\{1, \ldots, n\}$ e n é a quantidade de variáveis do problema.

A é a matriz que possui os coeficientes do lado esquerdo das inequações do problema

b é o vetor que possui os coeficientes do lado direito das inequações do problema

Aeq é a matriz que possui os coeficientes do lado esquerdo das equações do problema

beq é o vetor que possui os coeficientes do lado direito das equações do problema

lb é o vetor que representa o valor mínimo que cada variável deve assumir. Neste caso, como o problema é binário, o valor mínimo é representado por um vetor preenchido de zeros.

ub é o vetor que representa o valor máximo que cada variável deve assumir. Neste caso, como o problema é binário, o valor máximo é representado por um vetor preenchido de uns.

6.2.1 Adequação para *input*

Apesar de o problema ser matematicamente formulado como em (4.4), (4.5), (4.6), (4.7), (4.8) e (4.9), precisamos fazer adequações nas matrizes e vetores para *input* na função *intlinprog*. Por exemplo, **b** pode ser lido apenas na forma de vetor. Para tal, precisamos que a matriz de $x(p,t)$ representada por

$$x(p,t) = \begin{pmatrix} x(1,1) & x(1,2) & \ldots & x(1,15) \\ x(2,1) & x(2,2) & \ldots & x(2,15) \\ \vdots & \vdots & \ddots & \vdots \\ x(7,1) & x(7,2) & \ldots & x(7,15) \end{pmatrix}$$

deva se transformar em um vetor da forma

$$\mathbf{x} = \begin{pmatrix} x(1,1) \\ x(1,2) \\ \vdots \\ x(1,15) \\ x(2,1) \\ x(2,2) \\ \vdots \\ x(7,1) \\ \vdots \\ x(7,15) \end{pmatrix}$$

Com isso, a matriz de inequações **A** também precisa ser adequada. Como $HT(t,h,d)$ não pode ser tridimensional, precisamos transformá-la em uma matriz bidimensional. Sendo assim, aplicamos um método análogo ao de $x(p,t)$. A matriz **HT** é então definida por:

$$\mathbf{HT}(d,h,t) = \begin{pmatrix} \mathrm{HT}(1,1,1) & \mathrm{HT}(1,1,2) & \dots & \mathrm{HT}(1,1,15) \\ \mathrm{HT}(1,2,1) & \mathrm{HT}(1,2,2) & \dots & \mathrm{HT}(1,2,15) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{HT}(1,6,1) & \mathrm{HT}(1,6,2) & \dots & \mathrm{HT}(1,6,15) \\ \mathrm{HT}(2,1,1) & \mathrm{HT}(2,1,2) & \dots & \mathrm{HT}(2,1,15) \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{HT}(5,5,1) & \mathrm{HT}(5,5,2) & \dots & \mathrm{HT}(5,5,15) \\ \mathrm{HT}(5,6,1) & \mathrm{HT}(5,6,2) & \dots & \mathrm{HT}(5,6,15) \end{pmatrix}$$

onde

$$\mathbf{HT}(d,h,t) = \begin{cases} 1, & \text{se a turma } t \text{ está alocada no dia } d \text{ no horário } t \\ 0, & \text{c.c.} \end{cases}$$

Na figura 4 vemos que para a Turma 1 teremos

$$\begin{aligned}
\mathrm{HT}(1,1,1) &= 1\\
\mathrm{HT}(1,2,1) &= 1\\
\mathrm{HT}(1,3,1) &= 0\\
&\vdots\\
\mathrm{HT}(5,1,1) &= 1\\
\mathrm{HT}(5,2,1) &= 1\\
&\vdots\\
\mathrm{HT}(5,6,1) &= 0
\end{aligned}$$

Para as restrições (4.7) e (4.8), criamos um vetor transposto com o equivalente de créditos que cada turma representa. Como cada horário desginado representa 2 créditos, a turma 1 será representada por 8 créditos, pois ocupa 4 horários da grade horária. Sendo assim, o primeiro valor do vetor será 8. No caso da restrição de carga mínima do professor, teremos o vetor análigo mas com sinal negativo, pois precisamos deixar todas as inequações com "$\leq$". Assim teremos um $\mathbf{Hmax} = -\mathbf{Hmin}$.

O lado direito das inequações precisa se transformado em vetor. Como a primeira inquação (4.5) possui lado direito ≤ 1 necessitamos apenas de um vetor preenchido de 1's. Para os valores mínimos e máximos de carga horária, preenchemos com um vetor com valor máximo e um vetor negativo com valor mínimo (novamente porque a restrição precisa ser "$\leq$").

Restam agora apenas as equações. O lado direito é preenchido por 1's, e o lado esquerdo é preenchidos por matrizes de identidade para que a equação seja satisfeita.

Ao final, para que as matrizes e vetores correspondam ao problema, teremos a seguinte formulação.

$$\mathbf{A} = \underbrace{\left(\begin{array}{ccccc}
\mathbf{HT}(d,h,t) & 0 & 0 & \dots & 0 \\
0 & \mathbf{HT}(d,h,t) & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \dots & \mathbf{HT}(d,h,t) \\
\hline
\mathbf{Hmax} & 0 & 0 & \dots & 0 \\
0 & \mathbf{Hmax} & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \dots & \mathbf{Hmax} \\
\hline
-\mathbf{Hmin} & 0 & 0 & \dots & 0 \\
0 & -\mathbf{Hmin} & 0 & \dots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & 0 & \dots & -\mathbf{Hmin}
\end{array}\right)}_{p*t \text{ colunas}}$$

$$\mathbf{b} = \left.\left(\begin{array}{c}
1 \\ \vdots \\ 1 \\ \hline \mathbf{Hmin} \\ \vdots \\ \mathbf{Hmin} \\ \hline -\mathbf{Hmax} \\ \vdots \\ -\mathbf{Hmax}
\end{array}\right)\right\} p \times d \times h + 2 \times p \text{ linhas}$$

Assim temos $\mathbf{Ax} \leq \mathbf{b}$ devidamente formulado.

E

$$\mathbf{Aeq} = \underbrace{\left(\begin{array}{ccc} \mathbb{I} & \dots & \mathbb{I} \end{array}\right)}_{p*t \text{ colunas}}$$

Onde $\mathbb{I}$ é a matriz identidade $t \times t$, nesse caso 15×15.

$$\mathbf{beq} = \left.\left(\begin{array}{c} 1 \\ \vdots \\ 1 \end{array}\right)\right\} t \text{ linhas}$$

Formulando assim $\mathbf{Aeq}\ \mathbf{x} = \mathbf{beq}$.

Finalmente, temos a matriz que representa $e(p,t)$ para leitura no MATLAB.

Utilizando a Figura 5 e seguindo o mesmo padrão das matrizes e vetores anteriores precisamos de um vetor que multiplicado a $\mathbf{x}$ resulte em um escalar, que será o nosso valor ótimo. Sendo assim, precisamos de uma matriz 1×105 $(t \times p)$. Assim, colocamos as preferências dos professores lado a lado por ordem de professor. Feito isso, multiplicamos a matriz por -1, pois a função do MATLAB faz, por padrão, minizações. Multiplicando por -1 transformamos o problema de maximização em minimização.

Agora estamos prontos para rodar o problema no MATLAB.

6.2.2 Código Fonte

O código escrito pode ser encontrado nos Anexos. Aqui comentamos brevemente como a construção do código é feita.

Como todo código, necessitam-se de inicializações de alguns parâmetros. Criam-se as principais variáveis que são utilizadas com suas devidas dimensões. Também na inicialização, é informado o H_m (mínimo de créditos por professor) e H_M (máximo de créditos por professor) a serem utilizados.

No segundo bloco temos a leitura da matriz de grade horária vista na Figura 6 e as preferências dos professores visto na Figura 7:

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
1															
2	1		2ª	3ª	4ª	5ª	6ª			2ª	3ª	4ª	5ª	6ª	2
3		H1	1	5	6		1		H1	2	3	7	4	2	
4		H2	1	5	6		1		H2	7	4	2	3	7	
5		H3	10	5	9	11			H3		8				
6		H4	11	5	10		11		H4		8				
7		H5	12	5	12	15	13		H5		14				
8		H6	13	5	13	15	12		H6				14		

Figura 6: Grade horária ajustada para o programa

No terceiro bloco transformamos a matriz de preferência dos professores em um vetor transposto.

No quarto bloco cria-se a matriz **Aeq** e o vetor **beq**.

No quinto bloco cria-se a primeira parte da matriz **A**, as segunda e terceira partes da matriz **A** são criadas no sexto bloco.

No sétimo bloco criam-se as partes do vetor **b** e consolidam-se as as partes de **A** e **b**.

No oitavo bloco temos finalmente a aplicação da função de otimização *intlinprog*.

	Turma														
Professor	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4	3	5	5	1	5	4	1	1	1	1	1	1	1	1
2	1	1	1	1	2	1	1	3	3	3	3	5	5	4	4
3	3	3	2	2	4	2	3	2	4	3	4	4	4	4	4
4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
5	1	3	4	4	3	4	3	4	5	3	2	2	2	4	4
6	3	2	5	3	1	2	5	3	2	5	3	2	4	5	5
7	5	1	1	1	5	1	1	1	1	1	1	1	1	1	1

Figura 7: Preferência dos professores em forma de matriz ajustada ao problem

Por fim, no nono bloco, uma transformação do vetor solução em matriz para melhor visualização da solução encontrada.

7 Resultados

Vamos mostrar os resultados dos problemas fictícios para então finalizar com os resultados do problema original.

7.1 Problemas Fictícios

Foram simuladas 3 situações diferentes para restrições diferentes de H_m e H_M e uma quarta situação para mostrar o problema de restrição forte de um professor que não pode lecionar determinada turma de qualquer jeito..

7.1.1 Primeira situação: $H_m = 0$ e $H_M = 100$

Nessa simulação, utilizam-se restrições "infinitas" apenas para visualizar um resultado caso não houvessem restrições de cargas máximas e mínimas.

Com isso temos:

Valor ótimo: 70

Turma	Professor
1	7
2	3
3	6
4	1
5	7
6	1
7	6
8	5
9	1
10	6
11	3
12	2
13	2
14	2
15	6

Professor	1	2	3	4	5	6	7
Créditos Totais	8	16	12	0	6	18	20

Perceba que o professor 4 está livre e não possui nenhuma turma alocada.

7.1.2 Segunda situação: $H_m = 2$ e $H_M = 100$

Agora que sabemos que o problema é viável, adicionamos a restrição de mínimo de 2 créditos por professor, impossibilitando que qualquer um dos professores fique sem turmas alocadas.

Com isso temos:

Valor ótimo: 70

Turma	Professor
1	7
2	4
3	1
4	1
5	7
6	1
7	6
8	5
9	5
10	6
11	3
12	2
13	2
14	2
15	6

Professor	1	2	3	4	5	6	7
Créditos Totais	12	16	6	6	6	14	20

Vemos agora que todos os professores estão alocados a pelo menos uma turma, apesar do valor ótimo não ter mudado. Isso indica que esse resultado também se aplica ao primeiro método dado que todas as restrições dessa situação são satisfeitas na primeiro situação.

7.1.3 Terceira situação: $H_m = 2$ e $H_M = 12$

Adicionamos agora uma restrição de máximo significativa. Vemos que os resultados das situações anteriores não satisfazem essa nova situação pois muitos professores tem créditos totais acima de 12.

Com isso temos:

Valor ótimo: 66

Turma	Professor
1	1
2	4
3	6
4	1
5	7
6	5
7	4
8	5
9	5
10	1
11	3
12	2
13	2
14	6
15	3

Professor	1	2	3	4	5	6	7
Créditos Totais	12	12	10	12	10	12	12

Esse resultado atende todas as restrições e possui valor ótimo bastante próximo das situações anteriores. Todos os professores estão com cargas de crédito bastante equilibradas, mas ainda com a garantia de maior satisfação geral.

7.1.4 Quarta situação: $H_m = 2$ e $H_M = 12$ e Restrição Forte

Vamos supor agora que o professor 3, apesar de ter preferência 4 para ministrar a turma 15, não possa de jeito algum estar alocado a essa turma por alguma restrição pessoal. Para tal, alteramos a preferência dele para um M negativo suficientemente grande. Sendo assim teremos a nova matriz de preferência na Figura 8.

Com isso temos:

Valor ótimo: 66

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
1		Turma														
2	Professor	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
3	1	4	3	5	5	1	5	4	1	1	1	1	1	1	1	1
4	2	1	1	1	1	2	1	1	3	3	3	3	5	5	4	4
5	3	3	3	2	2	4	2	3	2	4	3	4	4	4	4	-999
6	4	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
7	5	1	3	4	4	3	4	3	4	5	3	2	2	2	4	4
8	6	3	2	5	3	1	2	5	3	2	5	3	2	4	5	5
9	7	5	1	1	1	5	1	1	1	1	1	1	1	1	1	1

Figura 8: Preferência dos professores em forma de matriz ajustada ao problem

Turma	**Professor**
1	3
2	4
3	1
4	1
5	7
6	1
7	6
8	5
9	5
10	6
11	4
12	2
13	2
14	3
15	5

Professor	1	2	3	4	5	6	7
Créditos Totais	12	12	12	12	10	10	12

Nessa última simulação, temos o mesmo valor ótimo da situação anterior, Entretanto, vemos que o Professor 3 não está mais atribuído à turma 15. O número suficiente negativo impediu que o algoritmo atribuísse essa turma a esse professor para não prejudicar o valor ótimo.

7.2 Problema Original

A solução ótima encontrada pelo Matlab nos traz um valor ótimo de **570**, que representa uma média de **4,4** de índice de satisfação. Já a atribuição real de 2012 nos traz um valor ótimo de **393**, que representa um índice médio de **3** de satisfação.

Para a atribuição real, reparou-se que alguns professores não constavam nos formulários preenchidos. Sendo assim, não havia um vetor de satisfação para os mesmos. Para esses casos, considerou-se um índice 5 de satisfação.

Conclusão

Neste trabalho estuda-se um método de automatização da atribuição de professores e turmas nas instituições de ensino. Temos um modelo bastante adequável às peculiaridades de cada instituição de ensino.

Os testes realizados com problemas fictícios mostraram-se bastante satisfatórios e válidos. O software MATLAB demonstrou-se capaz de resolver de problemas lineares inteiros.

No caso específico do Departamento de Matemática do IME-USP, muitos ajustes tiveram que ser realizados. Ao final, obtivemos um resultado notavelmente superior ao método manual aplicado em 2012, mesmo com todos os ajustes.

Conclui-se então que o método aplicado é eficaz e é de extrema importância que os dados de entrada do problema sejam o mais assertivos possíveis para melhores resultados.

Considerações para trabalhos futuros: É sugerida a criação de um formulário virtual no qual os professores possam preencher suas restrições e preferências. Tal formulário deve facilitar a criação da matriz de entrada do problema. Também sugere-se uma máquina poderosa para realizar a criação da matriz bidimensional a partir da matriz tridimensional. Esse processo é o mais demorado do programa.

Também para trabalhos futuros, pode-se estudar a possibilidade de inserção de novas restrições como licença prêmio, grades para 1 ano inteiro e não apenas um semestre, e assim por diante.

Referências

[1] Dimitris Bertsimas and John N Tsitsiklis. *Introduction to linear optimization*, volume 6. Athena Scientific Belmont, MA, 1997. Citado 3 vezes nas páginas 15, 19 e 24.

[2] Rui Alves and Catarina Delgado. Programação linear inteira. *Porto, Faculdade de Economia–Universidade do Porto. Apostila*, 1997. Citado 4 vezes nas páginas 15, 33, 34 e 38.

[3] MathWorks. Matlab r2014a. Acesso em 10/01/2016. Citado na página 17.

[4] Igor Griva, Stephen G Nash, and Ariela Sofer. *Linear and nonlinear optimization*. Siam, 2009. Citado 2 vezes nas páginas 20 e 27.

[5] Alan Sultan. *Linear programming: an introduction with applications*. Elsevier, 2014. Citado na página 31.

[6] Anderson Roges Teixeira Góes. Otimização na distribuição da carga horária de professores: método exato, método heurístico, método misto e interface. *Universidade Federal do Paraná, Curitiba, Paraná*, 2005. Citado na página 43.

[7] Priscila Savulski Ferreira, Elizabeth Wegner Karas, FL Polucoski, Ademir Alves Ribeiro, and Arinei L Silva. Aplicação de programação inteira na distribuição de encargos didáticos em instituições de ensino doi: 10.5540/tema. 2011.012. 02.0135. *Trends in Applied and Computational Mathematics*, 2011. Citado na página 43.

Anexos

ANEXO A – Código do Matlab

```
clear all
close all
clc

tic

%% 1) Parâmetros de entrada

% limite de créditos de trabalho
Hmin = 2;
Hmax = 12;

N_prof = []; % número de professores
N_turmas = []; % número de turmas
N_horarios = []; % número de horários
N_dias = []; % número de dias

arq = 'C:\Users\Marco\Desktop\NAO_MEXA\novo_problema.xlsx';

```

Figura 9: Primeiro Bloco: Inicialização do programa

```
%% 2) Leitura de matrizes

% Inicializações
col_i = 3;
lin_i = 3;
cont = 1;
m = 1;

hd_ind = xlsread(arq,'HD');
[N_horarios, N_dias] = size(hd_ind);

% Matriz hd_all, para fazer a matriz tridimensional hd
hd_all = xlsread(arq,'GRADE');

[aux1, aux2] = size(hd_all);

n=1;
n1=1;
n2=1;
cont=1;
hd=[];
```

Figura 10: Segundo Bloco: Primeira Parte

```
while hd_all(1,n) ~= 2
    for n1 = 1:N_dias
        for n2 = 1:N_horarios
            if isnan(hd_all(n2+1,n+n1+1)) == 1
                hd(n2,n1,cont) = 0;
            else
                hd(n2,n1,cont) = hd_all(n2+1,n+n1+1);
            end
        end
    end
    n = n+7;
    cont = cont +1;
end

% número de horas para cada par h-d
hd_carga=hd;
hd_carga(:,:,:)=2;

% preferência dos professores
pref=xlsread(arq,'PREF','B3:P9');


[N_prof, N_turmas] = size(pref);

```

Figura 11: Segundo Bloco: Segunda Parte

```
%% 3) Cria o vetor linear de f
f = [];
for ii=1:N_prof
    f = [f pref(ii,:)];
end

```

Figura 12: Terceiro Bloco: Vetor de f

```
%% 4) Cria a matriz Aeq e do vetor beq
% restrição 2
Aeq = [];
for ii=1:N_prof
    Aeq = [Aeq eye(N_turmas)];
end
beq = ones(1,N_turmas);

```

Figura 13: Quarto Bloco: Aeq e beq

```matlab
%% 5) Cria a primeira parte da matriz A

%Fazer a matriz inicial e depois criar os espelhos

%matriz inicial
A01 = zeros(N_dias*N_horarios,N_turmas);

for kk=1:cont-1
    for jj=1:N_dias
        for ii=1:N_horarios
            if hd(ii,jj,kk) ~= 0
                ind = hd(ii,jj,kk);
                A01(((jj-1)*N_horarios)+ii,ind) = 1;
            end
        end
    end
end

%criando matriz espelho a partir da inicial
A02 = zeros(N_dias*N_horarios,N_turmas);
A03 = [];

```

Figura 14: Quinto Bloco: Primeira Parte

```matlab
tic

for pp=1:N_prof
    linhaA03=[];
    for ppp = 1:N_prof
        if ppp == pp
            linhaA03=[linhaA03 A01];
        else
        linhaA03=[linhaA03 A02];
        end
    end
    A03 = [A03; linhaA03];
end

toc

```

Figura 15: Quinto Bloco: Segunda Parte

```matlab
%% 6) Cria a segunda e a terceira partes da matriz A

% restrições 4 e 3
A04 = [];
A05 = [];

for ii=1:N_prof
    linhaA04 = zeros(1,N_turmas*N_prof); % cria linha de zeros
    for jj=1:N_turmas
        soma = sum(hd_carga(hd==jj));
        linhaA04(1,jj+(ii-1)*N_turmas) = soma; % atribui as somas nas posicoes
    end
    A04 = [A04; linhaA04];
    A05 = [A05; -linhaA04];
end

```

Figura 16: Sexto Bloco: Segunda e Terceira Parte de A

```
%% 7) Cria as partes b

b01 = ones(1,N_dias*N_horarios*N_prof);
b02 = [];
b03 = [];
for ii=1:N_prof
    b02 = [b02 Hmax];
    b03 = [b03 -Hmin];
end

A = [A03; A04; A05];
b = [b01 b02 b03];
```

Figura 17: Sétimo Bloco: Cria b

```
%% 8) Final: Otimização

    l_x = length(f);
    intlinprog(-f,1:l_x,A,b',Aeq,beq',zeros(l_x,1),ones(l_x,1));
```

Figura 18: Oitavo Bloco: Otimização

```
%% 9) Adequação dos resultados

x_matriz = [];
for ii=1:N_prof
    x_linha=zeros(1,N_turmas);
    for jj=1:N_turmas
        x_linha(1,jj) = ans(N_turmas*(ii-1)+jj);
    end
    x_matriz=[x_matriz;x_linha];
end
```

Figura 19: Nono Bloco: Matriz Ótima