# Alocação de Funcionários:

## Um Problema de Programação Linear Aplicada em Emparelhamento de Grafos

**Cristina Zaniol**

Orientador: Carlos Hoppen

## XXIII SALÃO DE INICIAÇÃO CIENTÍFICA

### *Introdução*

O presente trabalho visa aplicar métodos de resolução de problemas de programação linear para emparelhamento de grafos bipartidos, utilizando como motivação a alocação ideal de funcionários em uma empresa. Otimizar os recursos de uma empresa é de suma importância, principalmente com a valorização das aptidões dos funcionários em determinada tarefa. Para propiciar a alocação ideal segundo as aptidões dos funcionários, por exemplo, é possível utilizar ferramentas matemáticas. O problema pode ser descrito através de grafos bipartidos, enquanto que as restrições para o emparelhamento constroem um problema de programação linear.

### *Grafos e Emparelhamentos*

Grafos são pares $G = (V, E)$, onde $V$ é um conjunto de vértices e $E$, um conjunto de arestas, em que $E \subseteq V^2$ ([1], [2]). Grafos simples são aqueles que não possuem arestas paralelas nem laços.

Quando é possível dividir o grafo em dois conjuntos de vértices não-adjacentes, o grafo é denominado *grafo bipartido* ([1]). Um emparelhamento em um grafo é um conjunto de arestas duas a duas não adjacentes ([2]). Normalmente, o objetivo é encontrar o emparelhamento que satura determinado conjunto de vértices através de caminhos aumentadores. É dito *emparelhamento maximal* quando não é subconjunto próprio de qualquer outro emparelhamento; enquanto, *emparelhamentos máximos* são aqueles que saturam o maior número de vértices possível, não havendo emparelhamento maior que estes. Ainda, *emparelhamento perfeito*, é quando um vértice do conjunto A incide em um vértice do conjunto B, não restando vértices não-saturados.

### *Programação Linear*

Um mecanismo natural para formular e resolver uma gama de problemas é a *programação linear*, caracterizada por funções lineares cujo objetivo é encontrar uma *solução viável* que minimiza ou maximiza a *função objetivo* designando uma solução viável, segundo as restrições impostas ([3],[4]).

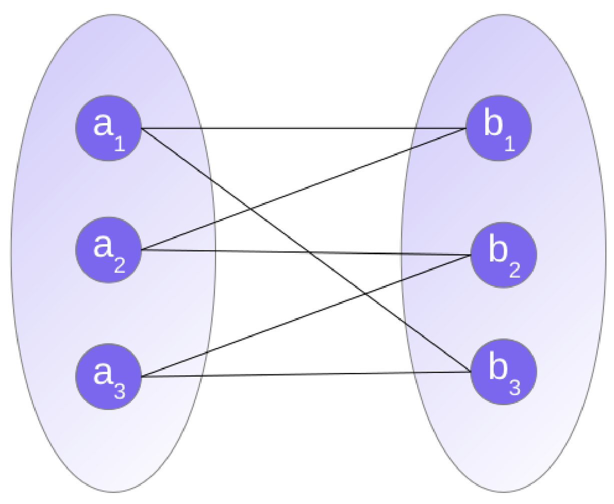
O processo de busca é realizado através do *algoritmo simplex*, onde, a partir de um vetor inicial $x_0$, de soluções viáveis básicas, encontra-se um resultado melhor descrito por $x_1$. O processo é realizado continuamente, até encontrar um vetor $x_n$ que não apresenta melhora potencial no resultado de $x_{n-1}$. O processo é realizado através do *tableau*, apresentado abaixo.

| $x$ | $x_1$ | $x_2$ | $\dots$ | $x_n$ | $x_{n+1}$ | $x_{n+2}$ | $\dots$ | $x_{n+p}$ | $b$ |
|---|---|---|---|---|---|---|---|---|---|
| $x_{n+1}$ | $a_{11}$ | $a_{12}$ | $\dots$ | $a_{1n}$ | $1$ | $0$ | $\dots$ | $0$ | $b_1$ |
| $\vdots$ | $\vdots$ | $\ddots$ | $\ddots$ | $\vdots$ | $0$ | $\dots$ | $\ddots$ | $0$ | $b_i$ |
| $x_{n+p}$ | $a_{p1}$ | $a_{p2}$ | $\dots$ | $a_{pn}$ | $0$ | $0$ | $\dots$ | $1$ | $b_p$ |
| F. O. | $c_1$ | $c_2$ | $\dots$ | $c_n$ | $0$ | $0$ | $\dots$ | $0$ | $Q(x)$ |

Note, ainda, que o problema, muitas vezes, possui desigualdades. Para isso, criam-se variáveis de folga descritas por $x_{n+1}, \dots, x_{n+p}$. Quando a linha da função objetivo do *tableau* contiver zeros para as variáveis básicas e nenhum valor negativo para as variáveis de folga, então a solução é ótima e o vetor que otimiza a função objetivo é representada pela última coluna, pelo vetor $b$. Se o conjunto de soluções viáveis for limitado, existe pelo menos uma solução ótima; do contrário, pode-se encontrar soluções viáveis, mas não ótima.

### *O Problema de Alocação de Funcionários*

Seja $F$ o conjunto de funcionários e $T$, o de tarefas. Supondo que cada funcionário desempenhará no máximo uma das tarefa a que está habilitado e que a realização de cada tarefa requer um único funcionário, deseja-se maximizar a eficiência dos funcionários em relação às tarefas, respeitando as afinidades.




Para cada aresta é atribuído um valor $w_{i,j} \in [0, 1]$, indicando a afinidade o funcionário $i$ em relação a tarefa $j$. Desta forma, a afinidade do funcionário em relação a tarefa é dada por $x_{i,j}$, onde $\{i, j\}$ representa uma aresta $e \in E$. Se o funcionário $i$ é escalado para a tarefa $j$, $x_{i,j} = 1$, ou seja, $x_{i,j}$ pertence ao emparelhamento. Caso contrário, o valor será $0$. Note que este é um problema de programação inteira. Se houver solução viável para problemas de programação linear, haverá pelo menos uma solução inteira. Então, admite-se que $x_{i,j} \in [0, 1]$. Então, para cada $v \in V$ e $0 \leq x_{i,j} \leq 1$ para cada $\{i, j\} \in E$, tem-se:

$$\text{max} \sum_{i,j \in E} w_{i,j} x_{i,j}$$

$$\text{Sujeito a} \sum_{\substack{\{i,j\} \in E \\ v \in \{i,j\}}} x_{i,j} = 1.$$

Este problema de programação linear torna possível, não somente um conjunto de soluções que permite designar uma tarefa para cada funcionário, como também encontrar a solução ótima que maximiza a eficiência.

### *Considerações Finais*

Encontrar emparelhamentos pequenos em grafos bipartidos é relativamente fácil. Entretanto, ao acrescer pesos nas arestas, é necessário buscar outras ferramentas que auxiliem na busca da solução ótima. Além de definir um emparelhamento, escrever um grafo como um problema de programação linear permite encontrar a solução ótima para um dado problema com um número de iterações finitas.

### *Referências*

[1] DIESTEL, Reinhard. *Graph Theory.* Springer: New York, 2005. [VERSÃO ELETRÔNICA]

[2] FEOFILOFF, Paulo; KOHAYAKAWA, Yoshiharu; WAKABAYASHI, Yoshiko. *Uma Introdução Sucinta à Teoria dos Grafos.* <http://www.ime.usp.br/ pf/teoriadosgrafos/texto/TeoriaDosGrafos. pdf>, 2011.

[3] LUENBERGER, David; YE, KYinyu. *Linear and Nonlinear Programming.* 3. ed. Springer: New York, 2008.

[4] MATOUSEK, Jiri; GARTNER, Bernd. *Understanding and Using Linear Programming.* Springer: New York, 2006.

criszaniol@gmail.com