Aplicação de programação linear no problema de alocação de salas, através do método húngaro em linguagem R

William da Silva Queiroz
Engenharia de Produção – Universidade Federal do Paraná
Mariana Kleina
Engenharia de Produção – Universidade Federal do Paraná
Silvana Pereira Detro
Engenharia de Produção – Universidade Federal do Paraná

Resumo: O problema de alocação de salas (PAS) é um problema recorrente em universidades. Consiste na distribuição das turmas nas salas de aula, sempre objetivando o menor deslocamento que os alunos deverão ter, considerando a quantidade de estudantes em cada turma. Face à complexidade desses problemas, a aplicação de métodos exatos ou de enumeração exaustiva torna-se inviável. Para contornar esse problema, pode se considerar a aplicação de algoritmos implementados a partir de meta-heurísticas. O presente trabalho, apresenta através de um algoritmo em R, o método húngaro, que é utilizado para problemas de otimização. Portanto, pela aplicação do algoritmo se encontrou a melhor distribuição das salas de aula de forma a minimizar o deslocamento dos estudantes.

Palavras-chave: Programação Linear, Otimização, Algoritmo em R, Pesquisa Operacional, Método Húngaro

Linear optimization and application to the in the problem of allocation of classrooms at the university, using the Hungarian method in R language

Abstract: The classrooms allocation problem (CAP) is a recurring problem in universities. It consists of distributing the classes in the classrooms, always aiming at the smallest displacement that the students should have, considering the number of students in each class. Due to the complexity of these problems, the application of exact methods or exhaustive enumeration becomes unfeasible. To circumvent this problem, the application of algorithms implemented from meta-heuristics can be considered. The present work presents, through an algorithm in R, the Hungarian method, which is used for optimization problems. Therefore, through the application of the algorithm, the best distribution of classrooms was found to minimize the displacement of students.

Keywords: Linear Programming, Optimization, Algorithm in R, Operational Research, Hungarian Method

1. Introdução

Uma das maneiras de trabalhar com o problema de alocação de salas é utilizando um método de otimização discreto sobre a matriz C para problemas de alocação chamado de Método Húngaro. Esse nome teve origem em 1955 devido a H.W.Kuhn, professor de programação de linear, que em seus trabalhos fez uma homenagem aos descobridores do algoritmo em 1931, os húngaros: Egerváry e Konig. O Método Húngaro pode ser aplicado em diversos problemas práticos de alocação, desde que seja construído de forma correta, a matriz C e as informações do problema. Com isso, pode também ser implementado computacionalmente, quando o volume de informações do problema for muito grande. (RODRIGUES; VIEIRA; AGUSTINI, 2005)

Neste presente trabalho é realizada a aplicação do problema de alocação de salas usando o Método Húngaro, o problema estudado consiste na designação de salas de aulas para a segunda aula do dia para os alunos do Setor de Educação Profissional e Tecnológica da Universidade Federal do Paraná (UFPR).

De acordo com o planejamento do setor, serão ofertadas 10 disciplinas para as turmas do 2º horário de segunda feira noturno e estarão disponíveis 10 salas para tanto. Será levado em consideração o local de cada turma no 1º horário e a quantidade de alunos que irão se deslocar até as salas. Diante dessa situação, o presente trabalho propõe um meio matemático de alocar essas disciplinas nas salas da melhor forma possível, utilizando especificamente conhecimentos da pesquisa operacional.

Assim, neste trabalho, será apresentado o algoritmo do método húngaro desenvolvido em R para a resolução do problema de alocação de salas.

Este artigo está dividido em 5 seções, a primeira é esta introdução, que descreve o problema geral que pretendemos solucionar, a segunda explica os conceitos empregados na resolução do problema, a terceira seção descreve a metodologia sendo utilizada na resolução do problema, a quarta seção mostra os resultados encontrados na aplicação do método húngaro no problema de alocação de salas, e a última seção debate os resultados encontrados.

2. Materiais e Métodos

Nesta seção serão abordados os conceitos relacionados as técnicas utilizadas, o método utilizado e as informações relevantes para entendimento do objeto de estudo

2.1 Programação Linear

A programação Linear é o problema de otimização mais importante e com mais estudos dedicados da sua área no mundo. Sendo resumido em um processo de minimizar ou maximizar a função objetivo a uma série de igualdades e desigualdades com restrições. Portanto os problemas de programação linear consistem sempre um problema de otimização, ou seja, na alocação de recursos limitados a atividades específicas, de forma ótima e eficiente. (PLOSKAS; SAMARAS, 2015).

A forma matemática clássica de apresentação da programação linear é a seguinte:

$$(\max ou \min) c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

Sujeito as restrições:

$$a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n (\leq = \geq) b_1 \quad (1)$$
$$a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n (\leq = \geq) b_2$$
$$\ldots$$
$$a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n (\leq = \geq) b_m$$
$$x_i \geq 0, i = 1, \ldots, n$$

Esta fórmula descrita é chamada de fórmula primal da programação linear. E tem suas características da seguinte forma:

- A função objetivo deve ser do tipo maximizar ou minimizar;
- Todas as restrições devem ser expressas como equações;
- Todas as variáveis e a constante do lado direito da equação das restrições devem ser não negativa.

Assim, podemos definir o conjunto de equações na forma matricial da seguinte forma:

$$(\max ou \min)\ z = cx$$

Sujeito as restrições:

$$Ax (\leq = \geq) b$$
$$x \geq 0$$

A=$\begin{bmatrix} a_{11} & \ldots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \ldots & a_{mn} \end{bmatrix}$, x=$\begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix}$,, b=$\begin{bmatrix} b_1 \\ \vdots \\ bx_n \end{bmatrix}$, c= $[c_1 \ldots c_n]$.

2.2 O problema do ensalamento

Para tratar especificamente do problema de ensalamento, diversos fatores devem ser considerados. Preliminarmente devem ser consideradas: a quantidade de disciplinas, o número de alunos matriculados, a capacidade das salas e a quantidade de salas.

A solução será dada por uma tabela do melhor alocamento das disciplinas em cada sala de aula, visando minimizar o deslocamento e, portanto, utilizando de maneira eficiente equipamentos e salas disponíveis. No caso em estudo, na Universidade Federal do Paraná especificamente o Setor de Educação Profissional e Tecnológica, o problema não está relacionado aos professores, visto que já se sabe o professor responsável por cada disciplina. Assim, o problema a ser solucionado é relacionado ao deslocamento dos alunos entre disciplinas.

As restrições utilizadas nos problemas se referem da seguinte forma: Uma disciplina só pode ser movida para uma única sala, uma sala somente pode ter uma disciplina por vez e a capacidade da sala não pode ser ultrapassada pela quantidade de alunos nela alocados.

O modelo matemático do problema do ensalamento é semelhante ao modelo de um PPL (1), conforme definido a seguir:

Minimizar $\sum_{i=1}^{nd} \sum_{j=1}^{ns} x_{ij} D_{ij} N_{ij}$ (3a)

Sujeito a: $\sum_{j=1}^{ns} x_{ij} = 1$ $\quad i = 1,2, \ldots, nd$ (3b) (3)

$\sum_{i=1}^{nd} x_{ij} = 1$ $\quad j = 1,2, \ldots, ns$ (3c)

$$N_i x_{ij} \leq C_j \quad i = 1,2, \dots, nd; j = 1,2, \dots, ns \qquad \text{(3d)}$$

Onde nd representa o número de disciplinas em estudo; ns o número de salas disponíveis; D_{ij} representa a distância entre o prédio sede da disciplina e a sala; N_i é o número de alunos matriculados na disciplina; C_j é a capacidade da sala e x_{ij} representa a alocação da disciplina na sala.

A função objetivo, Equação (3a), representa a distância total percorrida pelos alunos para ir da aula anterior até a sala de aula da segunda aula do dia. A primeira restrição, Equação (3b), indica que somente uma sala será alocada para cada disciplina; a segunda restrição, Equação (3c), indica que somente uma disciplina será alocada em cada sala; e a terceira restrição, Equação (3c), indica que a capacidade de alunos de cada sala não será excedida.

O problema do ensalamento, da forma como está descrito em (3) é um problema de programação linear binário, pois cada variável vai assumir valor “1” quando a disciplina for alocada na sala e valor “0” caso contrário.

2.3 Método húngaro

Existem diversas maneiras de resolver este tipo de problema de alocação de salas (PAS) como afirma (SOARES, 2011), Força bruta, *BranchandBound, Heurísticas, SimulatedAnnealing*, porém utilizaremos um algoritmo chamado Método Húngaro descrito na literatura (ANTON; RORRES, 2001).

Este método transforma o problema para uma forma matricial, implicando a construção de uma matriz de custos de tamanho nxn, ou simplesmente de ordem n. Desta forma havendo $n!$ maneiras de alocar disciplinas às salas devido ao fato de que há n maneiras de alocar a primeira tarefa, $n-1$ maneiras de alocar a segunda tarefa, $n-2$ maneiras de alocar a terceira tarefa e assim por diante, até que encontremos 1 maneira de alocar a última tarefa. Dessa forma teremos $n.(n-1).(n-2).(n-3). \dots .3.2.1 = n!$ formas de alocação, sendo uma dentre elas a forma ótima de alocação.

Apresentamos a seguir os passos do método húngaro sintetizado em um algoritmo:

Algoritmo 1: Algoritmo do método Húngaro
Entrada: Matriz de custos: $C = (c_{i,j})$ de ordem n.
Saída: Matriz de alocação ótima O= $(o_{i,j})$ de ordem n.
Início
Subtrair a menor entrada de cada linha de C de todas as entradas da mesma linha;
Subtrair a menor entrada de cada coluna de C de todas as entradas da mesma coluna;
Riscar um traço ao longo de linhas e colunas de tal modo que todas as entradas zero da matriz sejam riscadas e utilizando um número mínimo de traços ;
se $m = n$ então
A matriz resultante é de alocação ótima O;
Sair
enquanto $m < n$ faça

9	Determinar o menor valor não riscada por nenhum traço (k);
10	Subtrair k de todas as entradas não riscadas;
11	Somar k a todas as entradas riscadas tanto horizontal quanto verticalmente;
12	Retornar à linha 4.

3. Metodologia

O estudo foi feito através do software R (R Core Team 2022), e para importação das ferramentas do método húngaro, o pacote utilizado foi o BRKGA. Para isso utilizou-se os seguintes conjuntos de variáveis:

Tabela 1 – Quantidade de alunos por disciplinas

Disciplina	Quantidade de alunos na 1ª aula	Quantidade de alunos na 2ª aula
D1	46	54
D2	34	51
D3	47	35
D4	35	40
D5	25	42
D6	48	31
D7	40	20
D8	35	51
D9	41	50
D10	47	24

Fonte: Universidade Federal do Paraná (2022)

Tabela 2 – Capacidade das salas de aula

Sala	Capacidade
S1	44
S2	44
S3	42
S4	44
S5	30
S6	55
S7	52
S8	50
S9	51
S10	50

Fonte: Universidade Federal do Paraná (2022)

Tabela 3 – Local da 1ª aula

Disciplina	Sala
D1	S9
D2	S2
D3	S10
D4	S8
D5	S5
D6	S7
D7	S4
D8	S3
D9	S1
D10	S6

Fonte: Universidade Federal do Paraná (2022)

Figura 1 – Deslocamento entre as salas

DE/PARA	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	1	2	3	3	3	3	3	3	3	3
S2	2	1	3	3	3	3	3	3	3	3
S3	3	3	1	2	2	2	2	2	2	2
S4	3	3	2	1	2	2	2	2	2	2
S5	3	3	2	2	1	2	2	2	2	2
S6	3	3	2	2	2	1	2	2	2	2
S7	3	3	2	2	2	2	1	2	2	2
S8	3	3	2	2	2	2	2	1	2	2
S9	3	3	2	2	2	2	2	2	1	2
S10	3	3	2	2	2	2	2	2	2	1

Fonte: Os autores (2022)

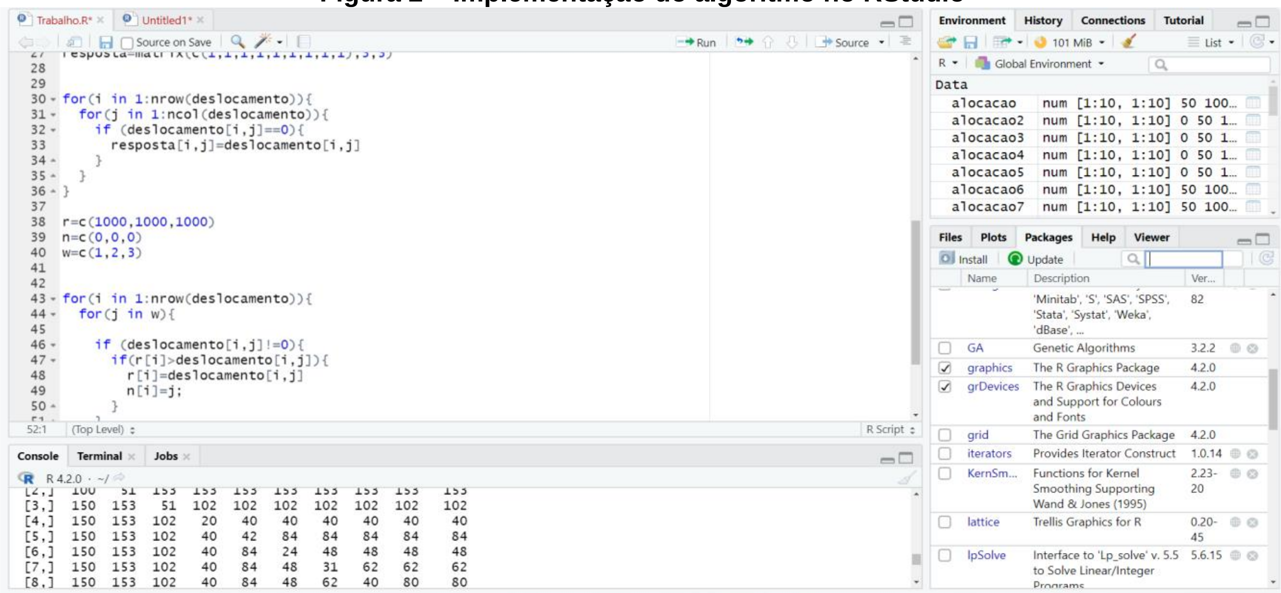
4. Resultados

Os resultados foram obtidos pela inserção de todas as informações da seção anterior no pacote de análise do R.

A Figura 2 mostra o algoritmo sendo executado no software de desenvolvimento em linguagem R. O pacote BRKGA foi instalado seguindo o manual de instalação do software.

A Figura 3 é a primeira saída do software, os valores que anteriormente são nulos, devido a capacidade da sala ser menor que a quantidade de alunos são substituídos pelo valor de: 1000.

Figura 2 – Implementação do algoritmo no RStudio



Fonte: Os autores (2022)

Figura 3 – Matriz de custo no método húngaro

This is the original cost matrix:

1,000	1,000	1,000	1,000	1,000	150	150	1,000	150	150
1,000	1,000	1,000	1,000	1,000	153	153	1,000	153	1,000
1,000	1,000	1,000	1,000	1,000	102	102	1,000	102	1,000
150	153	102	20	40	40	40	40	40	40
150	1,000	1,000	40	1,000	84	84	84	84	84
150	153	102	40	84	24	48	48	48	48
150	153	102	40	1,000	48	31	62	62	62
150	1,000	102	40	1,000	48	62	40	80	80
1,000	1,000	1,000	1,000	1,000	48	1,000	1,000	1,000	1,000
1,000	1,000	1,000	1,000	1,000	48	62	1,000	108	35

Fonte: Os autores (2022)

Na figura 4, os resultados retirados do software foram colocados em tabela para melhor compreensão.

Figura 4 – Método Húngaro etapa 1

DE/PARA	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1	50	100	150	150	150	150	150	150	150	150
S2	100	51	153	153	153	153	153	153	153	153
S3	150	153	51	102	102	102	102	102	102	102
S4	150	153	102	20	40	40	40	40	40	40
S5	150	153	102	40	42	84	84	84	84	84
S6	150	153	102	40	84	24	48	48	48	48
S7	150	153	102	40	84	48	31	62	62	62
S8	150	153	102	40	84	48	62	40	80	80
S9	150	153	102	40	84	48	62	80	54	108
S10	150	153	102	40	84	48	62	80	108	35

Fonte: Os autores (2022)

Figura 5 – Método húngaro etapa 2

DE/PARA	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1						150	150		150	150
S2						153	153		153	
S3						102	102		102	
S4	150	153	102	20	40	40	40	40	40	40
S5	150			40		84	84	84	84	84
S6	150	153	102	40	84	24	48	48	48	48
S7	150	153	102	40		48	31	62	62	62
S8	150		102	40		48	62	40	80	80
S9						48				
S10						48	62		108	35

Fonte: Os autores (2022)

Na Figura 5, o método húngaro foi aplicado e já retirou os valores que eram provisoriamente 1000. Foram feitas as retiradas conforme os passos 2 ao 7 do algoritmo apresentado no Algoritmo 1

Figura 6 – Deslocamento ótimo de alunos entre as salas

DE/PARA	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
S1						150	150	150	150	150
S2						153	153		153	
S3						102	102		102	
S4	150	153	102	20	40	40	40	40	40	40
S5	150			40		84	84	84	84	84
S6	150	153	102	40	84	24	48	48	48	48
S7	150	153	102	40		48	31	62	62	62
S8	150		102	40		48	62	40	80	80
S9						48				
S10						48	62		108	35

Fonte: Os autores (2022)

Figura 7 – Software fazendo os cálculos entre as matrizes

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0   49   99  130  110  126  119  110  110   115
 [2,]   50    0  102  133  113  129  122  113  113   118
 [3,]  100  102    0   82   62   78   71   62   62    67
 [4,]  100  102   51    0    0   16    9    0    0     5
 [5,]   98  100   49   18    0   58   51   42   42    47
 [6,]  100  102   51   20   44    0   17    8    8    13
 [7,]  100  102   51   20   44   24    0   22   22    27
 [8,]  100  102   51   20   44   24   31    0   40    45
 [9,]   86   88   37    6   30   10   17   26    0    59
[10,]  100  102   51   20   44   24   31   40   68     0
> alocacao4
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0   49   99  130  110  126  119  110  110   115
 [2,]   50    0  102  133  113  129  122  113  113   118
 [3,]  100  102    0   82   62   78   71   62   62    67
 [4,]  100  102   51    0    0   16    9    0    0     5
 [5,]   98  100   49   18    0   58   51   42   42    47
 [6,]  100  102   51   20   44    0   17    8    8    13
 [7,]  100  102   51   20   44   24    0   22   22    27
 [8,]  100  102   51   20   44   24   31    0   40    45
 [9,]   86   88   37    6   30   10   17   26    0    59
[10,]  100  102   51   20   44   24   31   40   68     0
```

Fonte: Os autores (2022)

Figura 7 – Resultado obtido no software

```
Console  Terminal ×  Jobs ×
R 4.2.0 · ~/
> alocacao5
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
 [1,]    0    0    0    0    0  150    0    0    0     0
 [2,]    0    0    0    0    0    0  153    0    0     0
 [3,]    0    0    0    0    0    0    0  102    0     0
 [4,]    0    0    0    0   40    0    0    0    0     0
 [5,]    0    0    0   40    0    0    0    0    0     0
 [6,]    0    0    0    0    0    0    0    0   48     0
 [7,]    0  153    0    0    0    0    0    0    0     0
 [8,]  150    0    0    0    0    0    0    0    0     0
 [9,]    0    0  102    0    0    0    0    0    0     0
[10,]    0    0    0    0    0    0    0    0    0    35
> opvalue
[1] "The optimal value equals 973."
>
```

Fonte: Os autores (2022)

O resultado encontrado apresentou um valor de 973. Isso nos mostra que teremos um deslocamento total de 973, sendo a multiplicação do número de alunos pela distância. O algoritmo nos retornou também qual caminho devemos escolher de modo a minimizar o deslocamento considerando as variáveis descritas nas tabelas 1,2 e 3 e na Figura 1.

5. Discussão

O artigo descreveu a implementação de um algoritmo do método húngaro na linguagem R, exemplificando as alterações que foram necessárias para que a capacidade das salas fosse uma restrição também inserida no desenvolvimento do projeto, é através de uma pequena quantidade de interações foi prosseguir com a execução proposta.

O motivo do trabalho foi prático, visto a real dificuldade de alocação de salas no setor estudado da Universidade Federal do Paraná, e a metodologia utilizada foi a considerada mais prática e rápida pela literatura. Portanto o resultado encontrado foi de encontro ao esperado, uma vez que a tabela final foi dada como resposta, assim como o valor de deslocamento.

A pesquisa ficou limitada ao estudo em apenas um pequeno setor com poucas disciplinas, podendo ser estudado em setores maiores para encontrar possíveis dificuldades na implementação do algoritmo. As sugestões para os próximos trabalhos relacionados ao tema, são: a colocação da distância real entre as salas, colocação dos obstáculos (escadas, rampas e espaços descobertos) como dificultadores de movimentação entre blocos, e ampliação do algoritmo para que seja definidas as disciplinas alocadas na 1ª aula juntamente as disciplinas que serão alocadas na 2ª aula.

Referências

ANTON, H.; RORRES, C. **Álgebra linear com aplicações**. [S.l.]: Bookman Porto Alegre, 2001. v. 8.

PLOSKAS, N.; SAMARAS, N. **Efficient GPU-based implementations of simplex type algorithms**.APPLIED MATHEMATICS AND COMPUTATION, v. 250, p. 552–570, 2015.

R CORE TEAM (2022). R: **A language and environment for statistical computing**. R Foundation for Statistical Computing, Vienna, Austria. URL https://www.R-project.org/

RODRIGUES, L. B.; VIEIRA, F. B. P.; AGUSTINI, E. O método húngaro de otimização para o problema da alocação de tarefas. **FAMAT em Revista**, Uberlândia, n. 4, 2005.

SOARES, H. C. de A. **Um estudo sobre o Problema de Alocação**. 2011. Dissertação (Trabalho de Conclusão de Curso) — UNIFESP, São José dos Campos - SP, 2011.

UNIVERSIDADE FEDERAL DO PARANÁ. **Grade Horária 2022/1.** Disponível em: <http:// http://www.sept.ufpr.br/portal.html> Acesso em: 16 set. 2022