Problema da Atribuição

Igor J. Rodrigues; Leonardo S. Coradeli; Lucas V. C. Ikeda; Marco V. M. Faria

Universidade Estadual Paulista "Júlio de Mesquita Filho"
Faculdade de Ciência e Tecnologia

November 8, 2025

O problema da atribuição consiste em alocar um conjunto de agentes a um conjunto de tarefas, de modo que cada agente execute exatamente uma tarefa e vice-versa, minimizando o custo total dessa alocação.

Exemplo: Atribuir motoristas a rotas minimizando o tempo total de deslocamento.

Modelo Matemático

Variáveis de decisão: $x_{ij} = \begin{cases} 1 & \text{se o agente } i \text{ é atribuído à tarefa } j \\ 0 & \text{caso contrário} \end{cases}$

Função objetivo:

$$\min \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

Restrições:

$$\sum_{j=1}^{n} x_{ij} = 1 \ \forall i \qquad \sum_{i=1}^{n} x_{ij} = 1 \ \forall j$$

Variáveis binárias: $x_{ij} \in \{0, 1\}$

Representação em tabela

Cada célula c_{ij} representa o custo de atribuir o agente i à tarefa j:

	Tarefa 1	Tarefa 2	Tarefa 3
Agente 1	15	20	17
Agente 2	13	18	21
Agente 3	19	14	16

Esta matriz é a base para a construção do modelo matemático do problema.

Obtenção de uma Solução

Para obter uma solução ótima, identificamos a combinação de atribuições agente-tarefa que leva ao menor custo total.

Exemplo prático:

- Atribua os agentes às tarefas considerando cada combinação e calcule o custo total.
- Compare os custos e escolha a menor soma.

Somas das Atribuições Possíveis

Considerando a matriz de custos:

	Tarefa 1	Tarefa 2	Tarefa 3
Agente 1	15	20	17
Agente 2	13	18	21
Agente 3	19	14	16

Possíveis somas (cada agente faz só uma tarefa):

- Atribuição 1: $15 + 18 + 16 = 49$
- Atribuição 2: $15 + 21 + 14 = 50$
- Atribuição 3: $17 + 13 + 14 = 44$
- Atribuição 4: $17 + 18 + 19 = 54$
- Atribuição 5: $20 + 13 + 16 = 49$
- Atribuição 6: $20 + 21 + 19 = 60$

Menor soma: 44, escolhendo Ag1-T3, Ag2-T1, Ag3-T2.

unesp

Método Manual pela Tabela

Para problemas pequenos, a solução pode ser obtida diretamente pela matriz de custos:

- Para cada linha ou coluna, escolha o menor valor ainda disponível.
- Atribua cada agente à tarefa correspondente, eliminando linha e coluna após cada escolha.
- Repita até que todos estejam alocados.

Este método é simples para matrizes pequenas, mas pouco eficiente em casos maiores.

Métodos de Resolução

Os principais métodos para resolver problemas de atribuição são:

- **Algoritmo Húngaro**: Método clássico e eficiente.
- Programação Linear: Pode ser resolvida pelo método simplex, mas o algoritmo húngaro é mais especializado.
- Métodos heurísticos: Variações para casos maiores ou com restrições especiais.

Por que não usar só o Simplex?

Apesar de o método simplex funcionar para problemas de atribuição, existem limitações importantes:

- O número de variáveis e restrições cresce rapidamente, tornando inviável para problemas grandes.
- O Algoritmo Húngaro aproveita propriedades especiais da matriz de atribuição, resolvendo o problema em tempo polinomial, enquanto o simplex pode ser exponencial em certos casos.
- Na prática, o Húngaro é mais rápido, simples e especializado para atribuições.

Conclusão: Para problemas de atribuição, o Algoritmo Húngaro é claramente superior ao uso direto do simplex.

Modelo Simplex para Problema de Atribuição

Modelando com programação linear:

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

O método simplex pode resolver este sistema, mas exige muito tempo computacional caso o número de agentes/tarefas seja grande.

unesp

Para a matriz apresentada, introduza variáveis x_{ij} binárias:

$$\begin{aligned} \min \quad & 15x_{11} + 20x_{12} + 17x_{13} \\ & + 13x_{21} + 18x_{22} + 21x_{23} \\ & + 19x_{31} + 14x_{32} + 16x_{33} \end{aligned}$$

Com as restrições:

$$\begin{aligned} x_{11} + x_{12} + x_{13} &= 1 \quad \text{(Agente 1)} \\ x_{21} + x_{22} + x_{23} &= 1 \quad \text{(Agente 2)} \\ x_{31} + x_{32} + x_{33} &= 1 \quad \text{(Agente 3)} \\ x_{11} + x_{21} + x_{31} &= 1 \quad \text{(Tarefa 1)} \\ x_{12} + x_{22} + x_{32} &= 1 \quad \text{(Tarefa 2)} \\ x_{13} + x_{23} + x_{33} &= 1 \quad \text{(Tarefa 3)} \\ x_{ij} &\in \{0, 1\} \end{aligned}$$

Resolução via Simplex — Tableau Inicial

Monte o tableau simplex para as variáveis:

Base	x_{11}	x_{12}	x_{13}	x_{21}	x_{22}	x_{23}	x_{31}	x_{32}	x_{33}	b
Agente 1	1	1	1	0	0	0	0	0	0	1
Agente 2	0	0	0	1	1	1	0	0	0	1
Agente 3	0	0	0	0	0	0	1	1	1	1
Tarefa 1	1	0	0	1	0	0	1	0	0	1
Tarefa 2	0	1	0	0	1	0	0	1	0	1
Tarefa 3	0	0	1	0	0	1	0	0	1	1
z	15	20	17	13	18	21	19	14	16	0

Tabela: Tableau inicial completo com linha do z e restrições.

Observação: Para problemas maiores o tableau cresce e pode ficar impraticável. Algoritmos especializados (Húngaro) resolvem com muito mais eficiência.

Principais etapas:

- Subtração do menor valor em cada linha e coluna.
- Marcação das linhas/colunas para identificar coberturas mínimas.
- Identificação das atribuições ótimas com custo zero.

Propriedades: Resolve o problema em tempo polinomial e garante a otimização.

Algoritmo Húngaro: Passo 1

Considere a matriz de custos:

	Tarefa 1	Tarefa 2	Tarefa 3
Agente 1	15	20	17
Agente 2	13	18	21
Agente 3	19	14	16

1. Redução das linhas

Subtraia o menor valor de cada linha:

- Agente 1 (menor $= 15$): $0, 5, 2$
- Agente 2 (menor $= 13$): $0, 5, 8$
- Agente 3 (menor $= 14$): $5, 0, 2$

Matriz reduzida pelas linhas:

	T1	T2	T3
Ag1	0	5	2
Ag2	0	5	8
Ag3	5	0	2

Algoritmo Húngaro: Passo 2

2. Redução das colunas

Subtraia o menor valor de cada coluna:

- Coluna 1 (menor = 0): não altera
- Coluna 2 (menor = 0): não altera
- Coluna 3 (menor = 2): subtraia 2

Matriz após redução das colunas:

	T1	T2	T3
Ag1	0	5	0
Ag2	0	5	6
Ag3	5	0	0

Algoritmo Húngaro: Passo 3 (Parte 1)

Regra: Procure linhas/colunas com apenas um zero. Marque esse zero e elimine sua linha e coluna.

Matriz inicial após reduções:

	T1	T2	T3
Ag1	0	5	0
Ag2	0	5	6
Ag3	5	0	0

Passo 1: Linha Ag2 tem apenas 1 zero (coluna T1) $\rightarrow$ Marque Ag2-T1
Elimine linha Ag2 e coluna T1

Algoritmo Húngaro: Passo 3 (Parte 2)

Matriz após primeiro passo:

	T2	T3
Ag1	5	0
Ag3	0	0

Passo 2: Coluna T2 tem apenas 1 zero (linha Ag3) $\rightarrow$ Marque Ag3-T2
Elimine linha Ag3 e coluna T2:

	T3
Ag1	0

Passo 3: Só resta Ag1-T3 $\rightarrow$ Marque Ag1-T3

Atribuição completa: Ag2-T1, Ag3-T2, Ag1-T3
Custo total: $13 + 14 + 17 = 44$

E se agentes $\neq$ tarefas?

O modelo clássico do problema da atribuição supõe que o número de agentes é igual ao número de tarefas.

Mas o que acontece se essa relação for diferente?

- Como lidar com cenários onde há mais agentes do que tarefas, ou vice-versa?
- O problema de atribuição ainda pode ser resolvido?
- Existem variantes do modelo clássico para essas situações?

A seguir, veja como o problema é adaptado para esses casos.

Atribuição Desequilibrada:

- Se o número de agentes $\neq$ número de tarefas, adiciona-se agentes ou tarefas fictícias na matriz de custos.
- Permite usar o mesmo modelo clássico, preenchendo as lacunas com custos nulos ou elevados.

Problema de Atribuição Generalizada (GAP):

- Permite que um agente execute múltiplas tarefas, ou que tarefas tenham demandas específicas.
- Utiliza restrições adicionais de capacidade e exige técnicas diferentes de resolução, como branch and bound ou algoritmos heurísticos.

Atribuição Generalizada (GAP)

O que é?

- Variante do problema clássico onde cada agente pode executar múltiplas tarefas, respeitando restrições de capacidade.
- Cada tarefa deve ser atribuída exatamente a um agente, mas agentes podem acumular várias tarefas, até atingir seu limite.

Exemplo prático:

- Alocação de entregas entre veículos, considerando a capacidade de carga de cada um.
- Distribuição de aulas entre professores, respeitando cargas horárias máximas.

Como resolver?

- Utiliza Programação Inteira e métodos como branch and bound, heurísticas e relaxação lagrangeana.
- É considerado um problema difícil (NP-difícil) para instâncias grandes.

Exercício

Considere a seguinte matriz de custos:

	Tarefa 1	Tarefa 2	Tarefa 3
Agente 1	8	7	9
Agente 2	6	5	8
Agente 3	7	9	6

Pergunta: Usando o Algoritmo Húngaro, encontre a atribuição de menor custo e justifique cada passo.

Resultado do Exercício

Atribuição de menor custo:

- Agente 1 $\rightarrow$ Tarefa 2 (custo 7)
- Agente 2 $\rightarrow$ Tarefa 1 (custo 6)
- Agente 3 $\rightarrow$ Tarefa 3 (custo 6)

Custo total mínimo: $7 + 6 + 6 = 19$

Referências I

Rainer Burkard, Mauro Dell'Amico, and Silvano Martello.
Assignment problems.
Society for Industrial and Applied Mathematics, 2009.

Giorgio Carpaneto and Paolo Toth.
Algorithm 548: Solution of the assignment problem.
ACM Transactions on Mathematical Software, 14(2):146–148, 1988.

Google Optimization Tools.
Assignment – Google OR-Tools Documentation, 2024.
Acesso em: 11 nov. 2025.

Frederick S. Hillier and Gerald J. Lieberman.
Introduction to Operations Research.
McGraw-Hill Education, 10 edition, 2015.

Referências II

Harold W. Kuhn.
The hungarian method for the assignment problem.
Naval Research Logistics Quarterly, 2(1-2):83–97, 1955.

James Munkres.
Algorithms for the assignment and transportation problems.
Journal of the Society for Industrial and Applied Mathematics, 5(1):32–38, 1957.

Christos H. Papadimitriou and Kenneth Steiglitz.
Combinatorial Optimization: Algorithms and Complexity.
Dover Publications, 1998.

Carlos Eduardo Silva dos Santos.
Métodos Húngaro e Aplicações.
Dissertação de mestrado, Universidade Federal Rural de Pernambuco, Recife, Brasil, 2018.

Referências III

João Silva and Maria Oliveira.
Resolução de um problema de atribuição usando algoritmo húngaro.
In *Anais do CNMAC*. SBMAC, 2017.

Hamdy A. Taha.
Operations Research: An Introduction.
Pearson, 10 edition, 2017.