Otimização de Sistemas

Prof. Sandro Jerônimo de Almeida, PhD.

Auction Algorithm
(Algoritmo de Leilão)

Mecanismo de Leilão (*Auction*)

Um único fornecedor (leiloeiro) oferta produtos para vários consumidores

Os consumidores começam a submeter ofertas crescentes para o bem ou serviço anunciado até que reste apenas um único consumidor

Mecanismo de Leilão (*Auction*)

Tipos de Leilão

- Inglês: Primeiro preço, lance aberto
- Primeiro preço, lance fechado
- Holandês: Primeiro preço, lance fechado (decrescente)
- *Vickrey*: Segundo preço, lance fechado

Auction Algorithm

Propostas

Dimitri Bertsekas (MIT) e Castañon (1989)

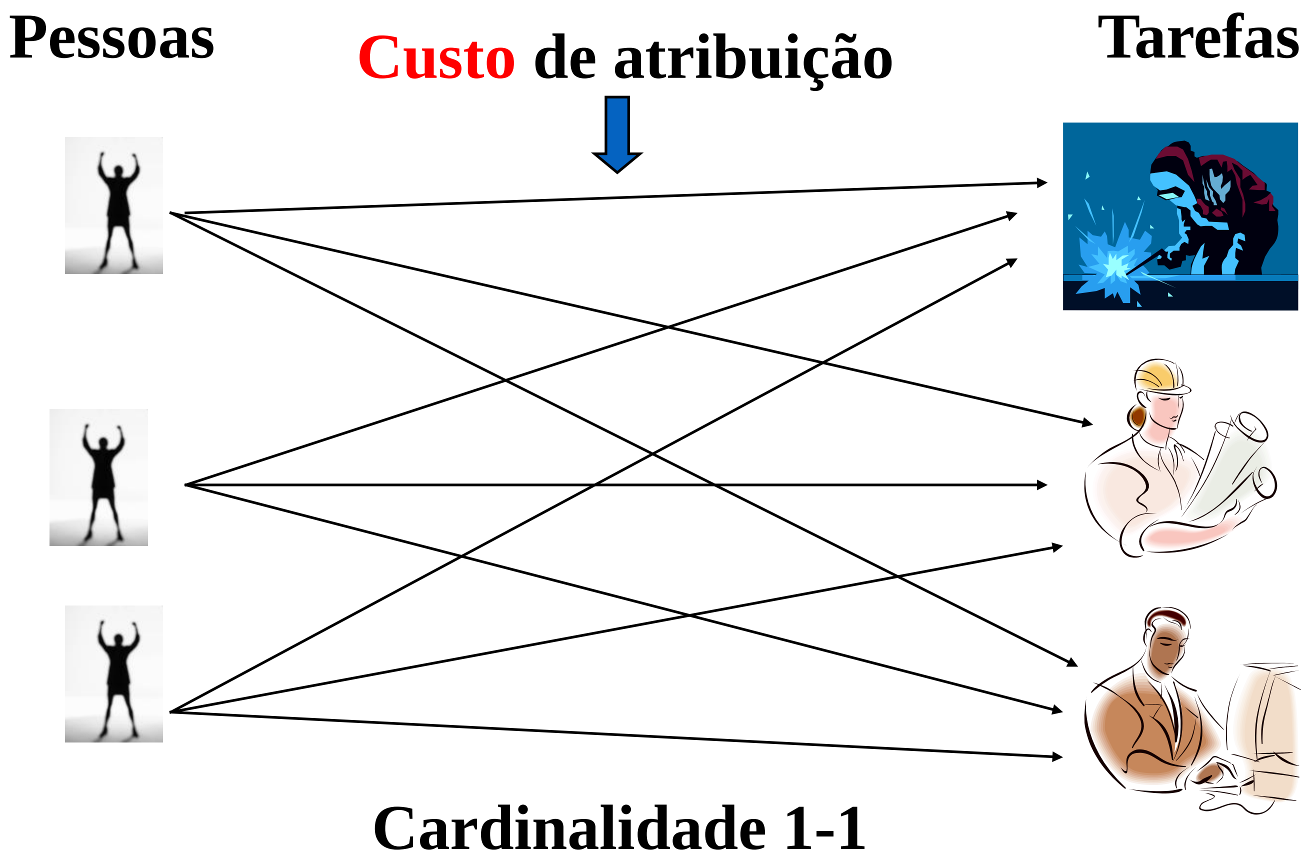
Problemas de fluxo em redes

- Problema de Atribuição (principal)
- Problema de Transporte
- Caminho mínimo
- Fluxo em Redes

Solução ótima (sob cestas condições)

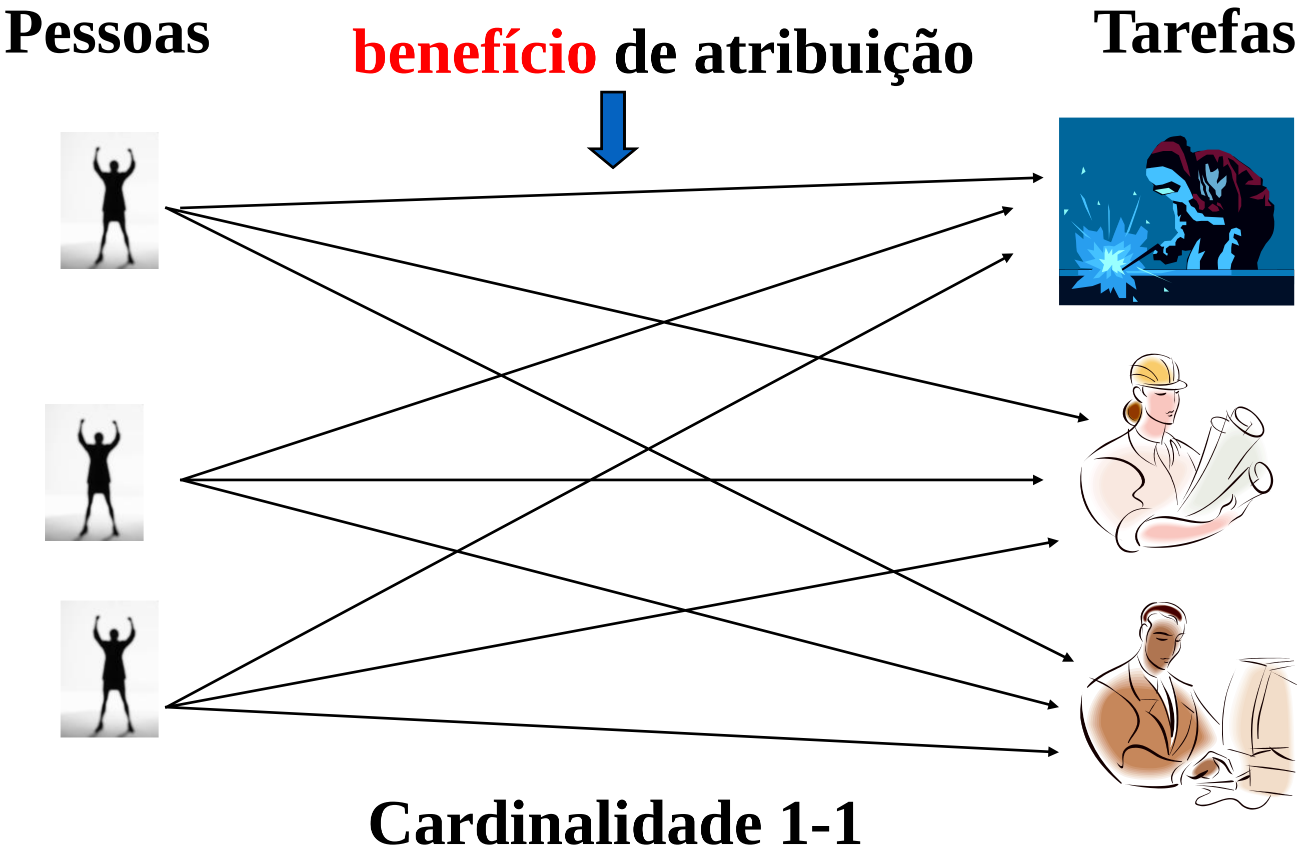
Problema de Atribuição

Como minimizar o custo de atribuição?



Problema de Atribuição

Como maximizar o benefício de atribuição?



Problema de Atribuição

Formulação geral

$$
\begin{aligned}
Minimizar\ z &= \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} \\
\sum_{j=1}^{n} x_{ij} &= 1 \qquad i = 1, \dots, n \\
\sum_{i=1}^{n} x_{ij} &= 1 \qquad j = 1, \dots, n \\
x_{ij} \geq 0 \quad i &= 1, \dots, m;\ j = 1, \dots, n
\end{aligned}
$$

Problema de Atribuição

Exemplo: como alocar policiais a regiões?

- Distância dos policiais para as regiões (*d*)
- Para o problema de minimização, basta considerar a própria distância no processo de busca de solução

--	Região A	Região B	Região C
Policial 1	3	5	4
Policial 2	4	3	5
Policial 3	1	4	2

Problema de Atribuição

Problema de Atribuição de Maximização

- Estratégia 1 $a_{ij} = 1 / d_{ij}$

--	Região A	Região B	Região C
Policial 1	1/3 = 0,33	1/5 = 0,2	1/4 = 0,25
Policial 2	1/4 = 0,25	1/3 = 0,33	1/5 = 0,2
Policial 3	1/1 = 1,0	1/4 = 0,25	1/2 = 0,2

- Estratégia 2 $a_{ij} = - d_{ij}$

Auction Algorithm

Lógica do algoritmo

- Cada origem é vista como como pessoa *i*, e cada destino como o objeto *j* em um leilão
- Cada pessoa *i* possui um interesse a_{ij} por um objeto j, que inicialmente tem um preço p_j
- O valor real de um objeto *j* para uma pessoa *i* é

$$v_r = a_{ij} - p_j$$

Auction Algorithm

Lógica do algoritmo

Benefício
(receita $ com transporte de noivos)

Objeto

Chofer
(Casamento)

R$ 250.000 em 5 anos

1

Preço
R$ 1.000.000,00

R$ 50.000 em 5 anos

2

Preço
R$ 30.000,00

$$V_{ferrari} = a_{11} - p_1 = 250.000 - 1.000.000 = -750.000$$
$$V_{palio} = a_{12} - p_2 = 50.000 - 30.000 = 20.000$$

Auction Algorithm

Satisfação e condição de parada

- Uma pessoa estará satisfeita se ela sempre ficar alocada ao objeto que traga o maior valor real

$$v_r = max\,\{a_{ij} - p_j\} \mid j = \{1, 2, \dots n\}$$

- Objetivo é otimizar as atribuições, considerando o custo/benefício de cada relação (pessoa → objeto)

Auction Algorithm

Lances e aumento de preço

- A cada iteração as pessoas dão lances por objetos, ficando alocado a esses objetos.
- Os lances são baseados na diferença entre o primeiro e segundo objeto com maior valor real para uma pessoa

$$v_j = \max\{a_{ij} - p_j\} \mid j = \{1, 2, \ldots n\} \quad \longleftarrow \textit{Melhor objeto}$$

$$w_j = \max\{a_{ij} - p_j\} \mid j \neq j_i \quad \longleftarrow \textit{Segundo melhor objeto}$$

$$\Upsilon_i = v_j - w_j \quad \longleftarrow \textit{Lance}$$

Auction Algorithm

Para cada Pessoa P não atribuída faça

Identifique o 1° melhor objeto e o 2° melhor objeto para P

Defina lance = 1ºMelhor Objeto (P) - 2º Melhor Objeto(P) + ε

Aumente o preço do 1º Melhor Objeto em lance

Desfaça a atribuição de quem estava atribuído ao 1º Melhor Objeto

Atribua a pessoa P ao 1ºMelhor Objeto

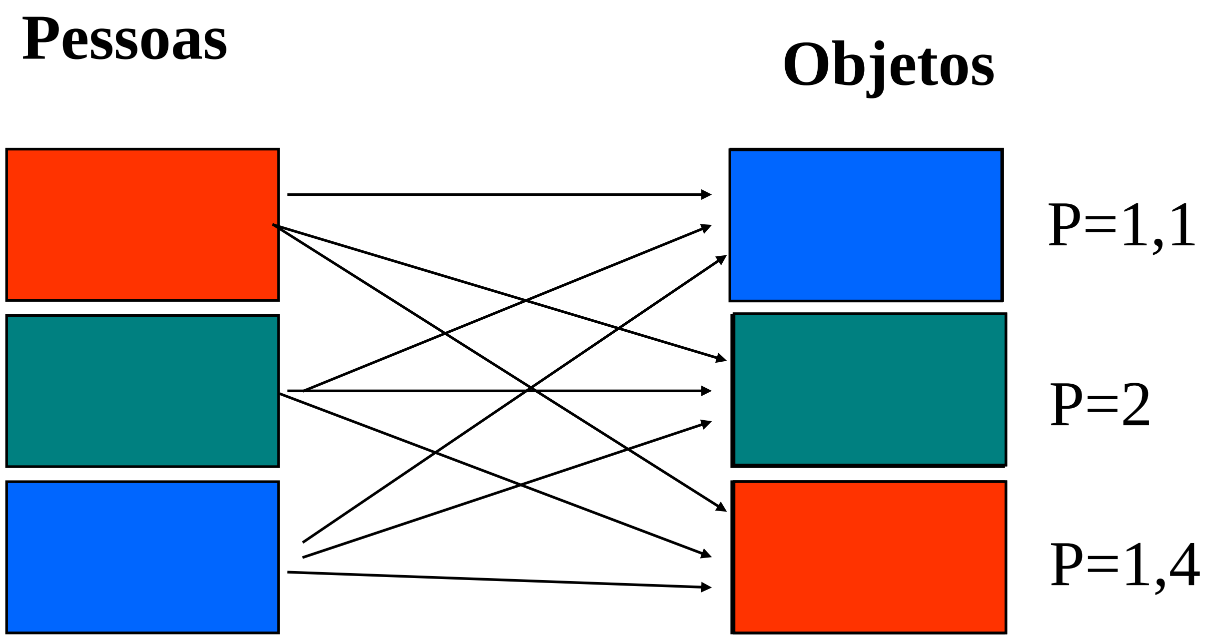
Fim para

ε = constante

Quando $\varepsilon < 1/n$, a solução é ótima.

Auction Algorithm

Animação



Atribuição completa! Todos pessoas estão felizes!

Auction Algorithm

Exemplo: maximizar o benefício de alocação

- Pessoas devem ser alocadas aos objetos

--	Objeto A	Objeto B	Objeto C
Pessoa 1	6	3	2
Pessoa 2	2	7	5
Pessoa 3	3	10	1

Preço inicial
$P_A = 0$
$P_B = 0$
$P_C = 0$

$\varepsilon = 0{,}1$

Artifício

Valores assumidos

- Inicialmente as pessoas 2 e 3 tem interesse pelo mesmo objeto

Auction Algorithm

Execução do algoritmo – Passo-a-Passo

Iteração	Alocação	Incremento $y = v - w + \varepsilon$ $\varepsilon = 0{,}1$	Novo preço $Pj = Pj + y$	*Desatribuição*
#1				
#2				

--	Objeto A	Objeto B	Objeto C
Pessoa 1	6	3	2
Pessoa 2	2	7	5
Pessoa 3	3	10	1

Auction Algorithm

Execução do algoritmo – Passo-a-Passo

Iteração	Alocação	Incremento $y = v - w + \varepsilon$ $\varepsilon = 0{,}1$	Novo preço $P_j = P_j + y$	*Desatribuição*
#1	(1, A)	=(6-0)-(3-0) + 0,1 = 3,1	= 0 + 3,1 = 3,1	---
	(2,B)	=(7-0)–(5-0) + 0,1 = 2,1	= 0 + 2,1 = 2,1	---
	(3,B)	=(10-2,1)-(1-0) +0,1 = 7	= 2,1 + 7 = 9,1	(2,B)
#2	(2,C)	=(5-0)-(2-3,1)+0,1=6,2	=0+6,2=6,2	--

- Resultado da alocação

    (Pessoa 1, Objeto A) = 6

    (Pessoa 2, Objeto C) = 5

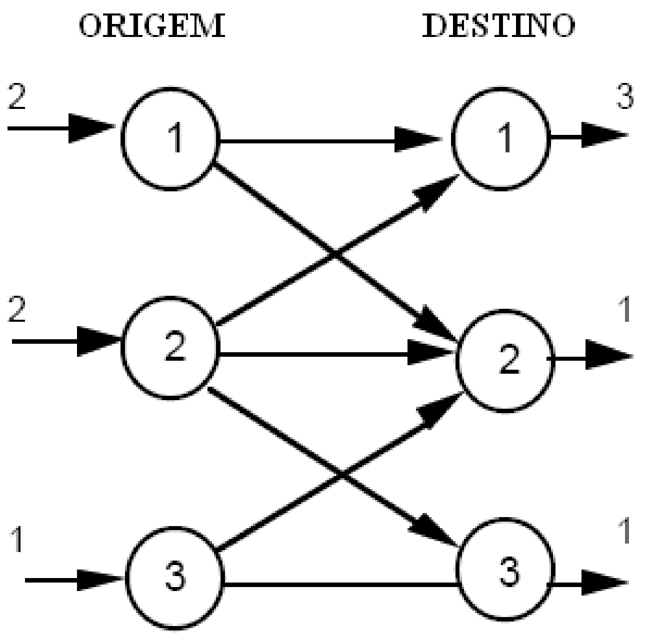
    (Pessoa 3, Objeto B) = 10

Benefício final da alocação

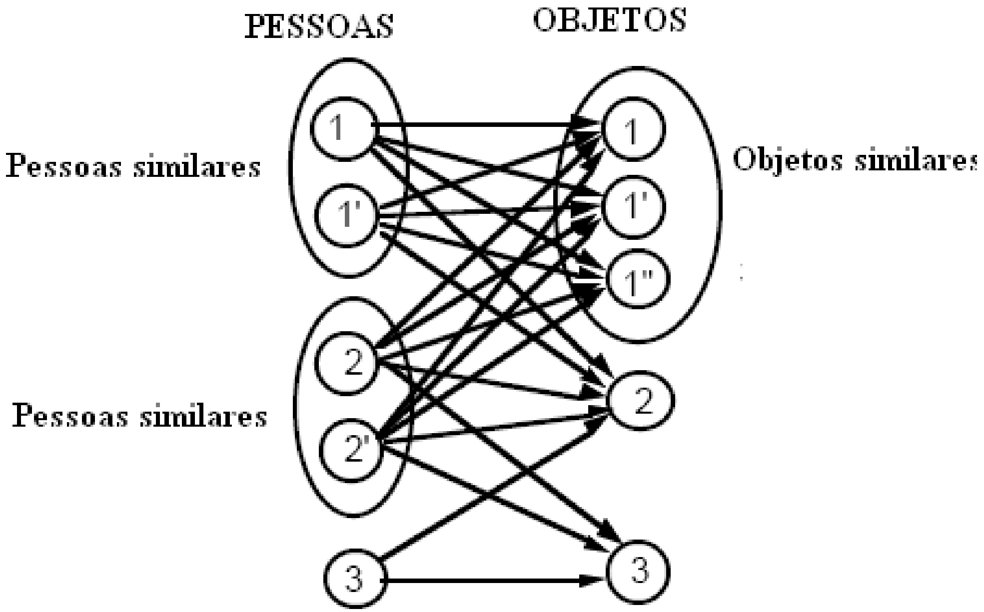
6 + 5 + 10 = 21

Problema de Transporte

Transformação no problema de atribuição



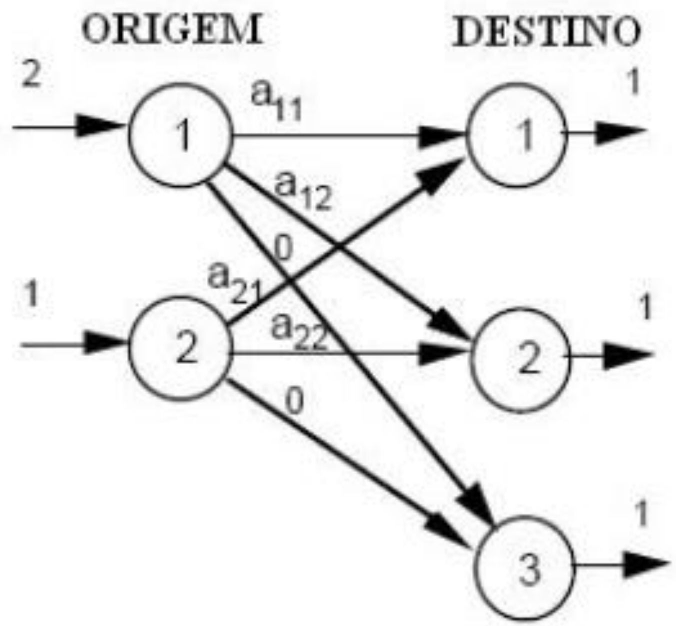
Problema de Transporte



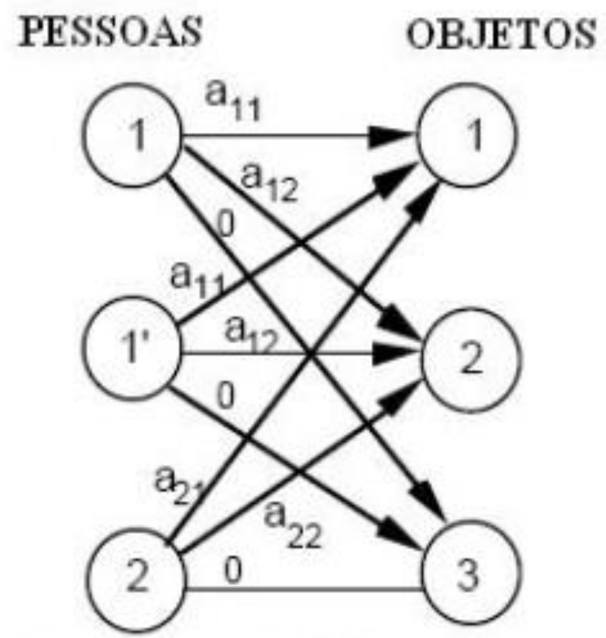
Problema de Atribuição

Problema de Transporte

Transformação no problema de atribuição



Problema de Transporte



Problema de atribuição equivalente com pessoas similares

Problema de Transporte

Exercício: resolver os seguinte problema de transporte (minimizar custos)

Demanda dos Destinos

Oferta

	1	**3**
2	5	3
2	5	7

$\varepsilon = 0{,}2$

Problema de Transporte

1° Transformar em problema de atribuição

Demanda dos Destinos

Oferta

		C	D
		1	**3**
A	**2**	5	3
B	**2**	5	7

	C	D1	D2	D3
A1				
A2				
B1				
B2				

2° Transformar o problema de minimização em maximização

	C	D1	D2	D3
A1				
A2				
B1				
B2				

Problema de Transporte

1° Transformar em problema de atribuição

Demanda dos Destinos

		C	D
		1	**3**
Oferta A	**2**	5	3
Oferta B	**2**	5	7

	C	D1	D2	D3
A1	5	3	3	3
A2	5	3	3	3
B1	5	7	7	7
B2	5	7	7	7

2° Transformar o problema de minimização em maximização

	C	D1	D2	D3
A1	-5	-3	-3	-3
A2	-5	-3	-3	-3
B1	-5	-7	-7	-7
B2	-5	-7	-7	-7

Auction Algorithm

3° Resolver o problema de atribuição (maximização)

Iteração	Alocação	Incremento y = v – w + ε	Novo preço Pj = Pj + y	*Desatribuição*

$\varepsilon = 0{,}2$

	C	D1	D2	D3
A1	**-5**	**-3**	**-3**	**-3**
A2	**-5**	**-3**	**-3**	**-3**
B1	**-5**	**-7**	**-7**	**-7**
B2	**-5**	**-7**	**-7**	**-7**

Preço
C = 0
D1 = 0
D2 = 0
D3 = 0

Auction Algorithm

3° Resolver o problema de atribuição (maximização)

Iteração	Alocação	Incremento y = v – w + ε	Novo preço Pj = Pj + y	*Desatribuição*
#1	A1, D1	(-3-0) – (-3-0) + 0,2 = 0,2	0,2	--
#1	A2, D2	(-3-0) – (-3-0) + 0,2 = 0,2	0,2	--
#1	B1, C	(-5-0) – (-7-0) + 0,2 = 2,2	2,2	--
#1	B2, D3	(-7-0) – (-5 – 2,2) + 0,2 =0,4	0,4	--

$\varepsilon = 0{,}2$

	C	D1	D2	D3
A1	**-5**	**-3**	**-3**	**-3**
A2	**-5**	**-3**	**-3**	**-3**
B1	**-5**	**-7**	**-7**	**-7**
B2	**-5**	**-7**	**-7**	**-7**

Preço
C = 2,2
D1 = 0,2
D2 = 0,2
D3 = 0,4

4° Mostrar o resultado

(A,D) = 2 x 3 = 6
(B,C) = 1 x 5 = 5
(B,D) = 1 x 7 = 7

Problema de transporte
Custo final da alocação
6 + 5 + 7 = 18

Complexidade de Algoritmos

Problema de Atribuição

- Algoritmo Húngaro: $O(n^3)$
- *Auction Algorithm*: $O(\frac{\max aij}{\boldsymbol{\varepsilon}}\ n^2)$

Problema de Transportes

- *TransAuction*: O((M + N)^{3}log (C min {M,N})),

  Onde C = max {|a_{ij} || j Є A(i)}.

Auction Algorithm

Referências

[1] D. P. Bertsekas. Auction algorithm for network flow problems: A tutorial introduction. 0(LIDSP- 2108):1–54, Maio 1992.

[2] D. P. Bertsekas, Castanon D.A. The Auction algorithm for the transformation problem 0(LIDSP- 2108):1–54, Fevereiro 1989.