

UNIVERSIDADE ESTADUAL DE CAMPINAS

Instituto de Matemática, Estatística e Computação Científica

O PROBLEMA DE TRANSPORTE

MS877 - Projeto Supervisionado II

Autora: Jéssica Moia de Oliveira - RA: 140692

Orientador: Prof. Dr. Aurelio Ribeiro Leite de Oliveira

Campinas/SP

Junho de 2016

Resumo

Nesse projeto é apresentado o Problema de Transporte, um problema de Programação Linear que tem como objetivo minimizar os custos de transporte de produtos, obedecendo limites de oferta e demanda. São descritas e modeladas matematicamente duas formulações para o problema: uma mais geral e uma chamada de formulação clássica. Com o auxílio de um exemplo, será descrito como encontrar a solução do problema utilizando o algoritmo simplex, além de resolver o mesmo exemplo computacionalmente.

Palavras-chave: Problema de Transporte; Programação Linear; Modelagem Matemática; Algoritmo Simplex.

Abstract

In this project the Transportation Problem is presented, a Linear Programming problem which aims to minimize the transportation costs of products, obeying limits of supply and demand. Two formulations for the problem are described and mathematically modeled: one that is more general and one known as classic formulation. With the help of an example, it will be described how to find the solution to the problem using the simplex algorithm, in addition to solving the same example computationally.

Keywords: Transportation Problem; Linear Programming; Mathematical Modeling; Simplex Algorithm.

1 Introdução

O problema de transporte é um problema de Programação Linear caracterizado por atribuir quantidades de uma mercadoria que serão transportadas de um determinado local de origem para um destino. Para tanto, é necessário obedecer a limitação de oferta das origens e também sempre atender no mínimo a demanda de mercadoria dos destinos.

Finalmente, existe um custo associado em transportar uma unidade de mercadoria de cada origem para cada destino e o objetivo final do problema é minimizar o custo total do transporte que respeita as restrições de oferta e demanda.

Devido à abrangência desse problema, ele pode ser aplicado nas mais diferentes áreas, como no transporte de matéria-prima de fornecedores para indústrias, de mercadorias de fábricas para lojas ou centros de distribuição, entre outros.

2 Descrição do Problema

Nessa seção será apresentada a definição do problema de transporte e, além disso, alguns casos específicos do problema.

2.1 Definição

O problema de transporte é uma classe especial de problemas de programação linear que trata do envio de uma mercadoria de origens (por exemplo, fábricas) para destinos (por exemplo, depósitos). O objetivo é determinar a programação de expedição que minimize o custo total de expedição e, ao mesmo tempo, satisfaça os limites de fornecimento e demanda. A aplicação do problema de transporte pode ser estendida a outras áreas de operações, entre elas controle de estoque, programação de empregos e designação de pessoal (TAHA, 2008).

Para chegar à formulação do problema, supomos que existem m origens e n destinos, onde cada origem i tem um limite máximo de fornecimento, ou oferta, denominado o_i e cada destino j tem uma quantidade mínima de demanda d_j . Além disso, o custo de transporte de uma unidade de mercadoria de i para j é igual a c_{ij} .

O problema visa especificar qual a quantidade de mercadoria que deve ser enviada de cada origem para cada destino. Essa variável de decisão será denominada x_{ij} .

A função objetivo do problema é busca minimizar a soma para todo i e j do custo c_{ij} multiplicado pela quantidade x_{ij} . Existem duas restrições que relacionam x_{ij} com a oferta e a demanda: uma que diz que a soma de tudo que sai de uma origem i deve ser no máximo igual à quantidade que pode ser ofertada, e outra que diz que a soma de tudo que chega em um destino j deve ser no mínimo igual à quantidade demandada.

Também é natural assumir que a quantidade de mercadoria x_{ij} é sempre positiva.

Em resumo, a formulação desse problema é escrita da seguinte forma:

$$\text{minimizar} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (1)$$

$$\text{sujeito a:} \quad \sum_{j=1}^n x_{ij} \leq o_i \quad i = 1, \dots, m \quad (2)$$

$$\sum_{i=1}^m x_{ij} \geq d_j \quad j = 1, \dots, n \quad (3)$$

$$x_{ij} \geq 0 \quad i = 1, \dots, m; j = 1, \dots, n. \quad (4)$$

2.2 Factibilidade e Formulação Clássica

A única maneira de um problema do tipo descrito anteriormente ser factível é se o total da quantidade ofertada exceder o total da demanda, ou seja,

$$\sum_{i=1}^n o_i \geq \sum_{j=1}^m d_j. \quad (5)$$

Se essa equação não for satisfeita, não é possível atender toda a demanda com o total de oferta. Se existe oferta suficiente, é fácil observar que as condições do problema podem ser satisfeitas. É conveniente assumir que o total ofertado é igual ao total demandado, o que significa:

$$\sum_{i=1}^n o_i = \sum_{j=1}^m d_j. \quad (6)$$

Dessa maneira, em casos que existe quantidade de oferta em excesso, é possível transformar o problema em um que a oferta é igual à demanda assumindo que a quantidade extra de suprimento pode ser descartada.

Fazendo essa simplificação, o problema se transforma no que é chamada de formulação clássica do problema de transporte, escrita da seguinte forma:

$$\text{minimizar} \quad \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (7)$$

$$\text{sujeito a:} \quad \sum_{j=1}^n x_{ij} = o_i \quad i = 1, \dots, m \quad (8)$$

$$\sum_{i=1}^m x_{ij} = d_j \quad j = 1, \dots, n \quad (9)$$

$$x_{ij} \geq 0 \quad i = 1, \dots, m; j = 1, \dots, n. \quad (10)$$

2.3 Um caso especial do Problema de Transporte

Existe um problema que é um caso especial do problema do transporte, chamado problema da designação (ou alocação) que se resume a alocar m agentes a n tarefas, onde cada agente i executa exatamente uma tarefa j e uma tarefa j só pode ser realizada exclusivamente por um agente i . O objetivo desse problema é minimizar o custo c_{ij} associado que cada agente tem para realizar as

tarefas.

Vale ressaltar que esse problema tem como premissa que todos os agentes são capazes de realizar todas as tarefas. Além disso, aqui a variável de decisão é binária, assumindo valor 1 se o agente i for designado à tarefa j e 0 caso contrário.

Dessa maneira, o problema é modelado da seguinte forma:

$$\text{minimizar} \quad \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (11)$$

$$\text{sujeito a:} \quad \sum_{j=1}^n x_{ij} = 1 \quad i = 1, \dots, m \quad (12)$$

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, n \quad (13)$$

$$x_{ij} \in \{0, 1\} \quad i = 1, \dots, m; j = 1, \dots, n. \quad (14)$$

Se compararmos esse conjunto de equações com o conjunto do problema de transporte de formulação clássica, podemos ver que as seguintes modificações foram feitas: tanto a oferta o_i quanto a demanda d_j assumiram valor 1 e a variável de decisão x_{ij} que antes podia assumir qualquer valor maior que 0 passou a ser binária.

O fato de que todas as quantidades fornecidas e demandadas serem iguais a 1 levou ao desenvolvimento de um algoritmo de solução simples denominado método húngaro. Embora o novo método de solução pareça não ter relação alguma com o problema do transporte, na realidade a raiz do algoritmo é o método simplex, exatamente como a do problema de transporte (TAHA, 2008).

O problema da designação, além de ser um problema de Programação Linear, é também um problema de Programação Inteira, já que sua variável de decisão só assume os valores inteiros 0 ou 1.

3 Método de Solução para o Problema

Como visto em Taha (2008), aqui será apresentado um método de solução para o problema do transporte, onde as etapas são paralelas às etapas do algoritmo simplex.

3.1 Etapas do Método

1. Determine uma solução básica *inicial* viável e passe para a etapa 2.
2. Use a condição de otimalidade do método simplex para determinar a *variável que entra* entre todas as variáveis não básicas. Se a condição de otimalidade for satisfeita, pare. Caso contrário, passe para a etapa 3.
3. Use a condição de viabilidade do método simplex para determinar a *variável que sai* entre todas as variáveis básicas atuais e ache a nova solução básica. Volte para a etapa 2.

Do item 1, vemos que primeira etapa do método envolve encontrar uma solução básica inicial. Essa solução será determinada utilizando o método do canto noroeste, explicado em 3.1.1.

3.1.1 Etapa 1 Através do Método do Canto Noroeste

As etapas do método são as seguintes, segundo Goldberg e Luna (2005):

1. Inicie pela célula (1, 1) do quadro de transporte.
2. Faça $x_{11} = \min(\tilde{o}_1, \tilde{d}_1)$ e substitua \tilde{o}_1 por $\tilde{o}_1 - x_{11}$ e \tilde{d}_1 por $\tilde{d}_1 - x_{11}$.
 Se $\tilde{o}_1 > \tilde{d}_1$, vá para a célula (1, 2) e faça $x_{12} = \min(\tilde{o}_1, \tilde{d}_2)$ e substitua \tilde{o}_1 por $\tilde{o}_1 - x_{12}$ e \tilde{d}_2 por $\tilde{d}_2 - x_{12}$.
 Se $\tilde{o}_1 < \tilde{d}_1$, vá para a célula (2, 1) e faça $x_{21} = \min(\tilde{o}_2, \tilde{d}_1)$ e substitua \tilde{o}_2 por $\tilde{o}_2 - x_{21}$ e \tilde{d}_1 por $\tilde{d}_1 - x_{21}$.
 Se $\tilde{o}_1 = \tilde{d}_1$ (degeneração) faça $x_{11} = 0$ e vá para (2, 2) ou (1, 3).
3. Se uma solução já foi alcançada, pare. Caso contrário, desconsidere linhas e colunas satisfeitas e aplique as regras anteriores fazendo a célula corrente ocupar a posição da célula (1, 1).

Para ilustrar esse método será considerado um exemplo encontrado em Taha (2008).

3.2 Um exemplo numérico

A SunRay Transport Company despacha caminhões de grãos provenientes de três silos para quatro moinhos. As quantidades fornecidas (em cargas de caminhão) e a demanda (também em cargas de caminhão), aliadas aos custos unitários de transporte por caminhão nas diferentes rotas, estão resumidas na Tabela 1. Os custos unitários de transporte c_{ij} estão em centenas de dólares. O modelo procura a programação de expedição de custo mínimo x_{ij} entre o silo i e o moinho j ($i = 1, 2, 3$ e $j = 1, 2, 3, 4$).

Tabela 1: Exemplo

	Moinho 1	Moinho 2	Moinho 3	Moinho 4	Oferta
Silo 1	10	2	20	11	15
Silo 2	12	7	9	20	25
Silo 3	4	14	16	18	10
Demanda	5	15	15	15	

Podemos escrever esse problema da seguinte maneira:

$$\begin{aligned}
 \text{minimizar} \quad & 10x_{11} + 2x_{12} + 20x_{13} + 11x_{14} + 12x_{21} + 7x_{22} + 9x_{23} + 20x_{24} \\
 & + 4x_{31} + 14x_{32} + 16x_{33} + 18x_{34} + 5x_{41} + 15x_{42} + 15x_{43} + 15x_{44} \\
 \text{sujeito a:} \quad & x_{11} + x_{12} + x_{13} + x_{14} = 15 \\
 & x_{21} + x_{22} + x_{23} + x_{24} = 25 \\
 & x_{31} + x_{32} + x_{33} + x_{34} = 10 \\
 & x_{11} + x_{21} + x_{31} + x_{41} = 5 \\
 & x_{12} + x_{22} + x_{32} + x_{42} = 15 \\
 & x_{13} + x_{23} + x_{33} + x_{43} = 15 \\
 & x_{14} + x_{24} + x_{34} + x_{44} = 15 \\
 & x_{ij} \geq 0 \quad (i = 1, 2, 3 \text{ e } j = 1, 2, 3, 4).
 \end{aligned}$$

Aplicando o método do canto noroeste, para realizar a etapa 1 de 3.1:

- $x_{11} = \min(\tilde{o}_1, \tilde{d}_1) = \min(15, 5) = 5 \Rightarrow \tilde{o}_1 = \tilde{o}_1 - 5 = 15 - 5 = 10; \tilde{d}_1 = \tilde{d}_1 - 5 = 5 - 5 = 0$.
0. Como $\tilde{o}_1 > \tilde{d}_1$:
 $x_{12} = \min(\tilde{o}_1, \tilde{d}_2) = \min(10, 15) = 10 \Rightarrow \tilde{o}_1 = \tilde{o}_1 - 10 = 10 - 10 = 0; \tilde{d}_2 = \tilde{d}_2 - 10 = 15 - 10 = 5$.
5. Como $\tilde{o}_1 < \tilde{d}_2$:
 $x_{22} = \min(\tilde{o}_2, \tilde{d}_2) = \min(25, 5) = 5 \Rightarrow \tilde{o}_2 = \tilde{o}_2 - 5 = 25 - 5 = 20; \tilde{d}_2 = \tilde{d}_2 - 5 = 5 - 5 = 0$.
0. Como $\tilde{o}_2 > \tilde{d}_2$:
 $x_{23} = \min(\tilde{o}_2, \tilde{d}_3) = \min(20, 15) = 15 \Rightarrow \tilde{o}_2 = \tilde{o}_2 - 15 = 20 - 15 = 5; \tilde{d}_3 = \tilde{d}_3 - 15 = 15 - 15 = 0$.
0. Como $\tilde{o}_2 > \tilde{d}_3$:
 $x_{24} = \min(\tilde{o}_2, \tilde{d}_4) = \min(5, 15) = 5 \Rightarrow \tilde{o}_2 = \tilde{o}_2 - 5 = 5 - 5 = 0; \tilde{d}_4 = \tilde{d}_4 - 5 = 15 - 5 = 10$.
10. Como $\tilde{o}_2 < \tilde{d}_4$:
 $x_{34} = \min(\tilde{o}_3, \tilde{d}_4) = \min(10, 10) = 10$.

Portanto a solução básica inicial é: $x_{11} = 5, x_{12} = 10, x_{22} = 5, x_{23} = 15, x_{24} = 5$ e $x_{34} = 10$.
 E o custo associado z é: $z = 5 * 10 + 10 * 2 + 5 * 7 + 15 * 9 + 5 * 20 + 10 * 18 = 520$.

Abaixo temos a tabela com os valores de x determinados pelo método do canto noroeste.

Tabela 2: Solução inicial do canto noroeste

	Moinho 1	Moinho 2	Moinho 3	Moinho 4	Oferta
Silo 1	5	10			15
Silo 2		5	15	5	25
Silo 3				10	10
Demanda	5	15	15	15	

Depois de determinar a solução inicial, segue-se para as etapas 2 e 3 do método explicitadas em 3.1.

Na etapa 2, é necessário determinar a variável que entra na base entre as variáveis que não fazem parte da solução básica inicial. Essa determinação será feita usando o método dos multiplicadores.

Nesse método, são associados os multiplicadores u_i e v_j com a linha i e a coluna j da tabela simplex de transporte. Esses multiplicadores satisfazem as seguintes equações: $u_i + v_j = c_{ij}$ para cada variável x_{ij} básica. O método instrui a estabelecer qualquer u_i arbitrário igual a zero e resolver para as variáveis que restarem.

Nesse exemplo, fazendo $u_1 = 0$, temos a seguinte resolução das equações:

Tabela 3: Resolução das equações

Variável básica	Equação (u, v)	Solução
x_{11}	$u_1 + v_1 = 10$	$u_1 = 0 \rightarrow v_1 = 10$
x_{12}	$u_1 + v_2 = 2$	$u_1 = 0 \rightarrow v_2 = 2$
x_{22}	$u_2 + v_2 = 7$	$v_2 = 2 \rightarrow u_2 = 5$
x_{23}	$u_2 + v_3 = 9$	$u_2 = 5 \rightarrow v_3 = 4$
x_{24}	$u_2 + v_4 = 20$	$u_2 = 5 \rightarrow v_4 = 15$
x_{34}	$u_3 + v_4 = 18$	$v_4 = 15 \rightarrow u_3 = 3$

O passo seguinte é usar u_i e v_j calculados para avaliar as variáveis não básicas calculando $u_i + v_j - c_{ij}$ para cada variável x_{ij} não básica. Esse resultado está na Tabela 4.

Tabela 4: Resultado das avaliações das variáveis não básicas

Variável não básica	$u_i + v_j - c_{ij}$
x_{13}	$u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
x_{14}	$u_1 + v_4 - c_{14} = 0 + 15 - 11 = 4$
x_{21}	$u_2 + v_1 - c_{21} = 5 + 10 - 12 = 3$
x_{31}	$u_3 + v_1 - c_{31} = 3 + 10 - 4 = \mathbf{9}$
x_{32}	$u_3 + v_2 - c_{32} = 3 + 2 - 14 = -9$
x_{33}	$u_3 + v_3 - c_{33} = 3 + 4 - 16 = -9$

Como no problema de transporte o objetivo é minimizar o custo total, a variável que entra na base é a que tem o coeficiente mais positivo. Pela Tabela 4 podemos ver então que a variável que entra na base é x_{31} . Agora é necessário realizar a etapa 3 do método, ou seja, determinar a variável que sai da base.

Para isso, é preciso analisar como ficou a tabela com os cálculos da primeira iteração.

Tabela 5: Primeira iteração

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Oferta
$u_1 = 0$	5	10	<i>-16</i>	<i>4</i>	15
$u_2 = 5$	<i>3</i>	5	15	5	25
$u_3 = 3$	<u>9</u>	<i>-9</i>	<i>-9</i>	10	10
Demanda	5	15	15	15	

Na Tabela 5, os valores em negrito são da solução básica inicial, os em itálico são resultado dos cálculos das variáveis não básicas e o valor sublinhado indica que é a variável naquela posição que entra na base.

Selecionar essa variável para entrar na base significa que é desejável despachar mercadorias por essa rota porque ela diminui o custo total. No próximo passo da etapa 3, fixa-se o valor da variável x_{31} como α e então o valor máximo de α é determinado tendo como base duas condições: os limites de oferta e os requisitos de demanda continuam satisfeitos e os despachos por todas as rotas permanecem não negativos. Isso significa basicamente que o valor máximo de α deve obedecer as restrições do problema.

Essas condições determinam o valor de α e a variável que sai da base da seguinte maneira: constrói-se um circuito fechado que começa e termina em $(3, 1)$ (célula da variável que entra na base), onde os segmentos que conectam o circuito devem ser sempre horizontais ou verticais, nunca diagonais. Além disso, com exceção da variável que entra na base, cada canto do circuito deve coincidir com uma variável básica.

A quantidade α é designada a $(3, 1)$ e, para que oferta e demanda continuem satisfeitas, alterna-se entre subtrair e somar α nos cantos sucessivos do circuito.

O circuito para x_{31} é mostrado a seguir.

Tabela 6: Circuito fechado para x_{31}

	$v_1 = 10$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Oferta
$u_1 = 0$	$5-\alpha$	$10+\alpha$			15
$u_2 = 5$		$5-\alpha$	15	$5+\alpha$	25
$u_3 = 3$	α			$10-\alpha$	10
Demanda	5	15	15	15	

Então para $\alpha \geq 0$ os novos valores das variáveis permanecem não negativos se:

$$x_{11} = 5 - \alpha \geq 0 \quad (15)$$

$$x_{22} = 5 - \alpha \geq 0 \quad (16)$$

$$x_{34} = 10 - \alpha \geq 0. \quad (17)$$

O valor máximo de α que satisfaz as inequações (15)-(17) é 5, o que ocorre quando ambas x_{11} e x_{22} são zero. Considerando que podemos escolher qualquer uma das duas para sair da base, a escolha será x_{11} .

Portanto, foram selecionadas $x_{31} = 5$ para entrar na base e x_{11} para sair da base. Lembrando que é necessário ajustar os valores das variáveis básicas nos cantos do circuito.

Agora repetimos o cálculo dos multiplicadores, da mesma maneira que antes.

Tabela 7: Equações para a segunda iteração

Variável básica	Equação (u, v)	Solução
x_{12}	$u_1 + v_2 = 2$	$u_1 = 0 \rightarrow v_2 = 2$
x_{22}	$u_2 + v_2 = 7$	$v_2 = 2 \rightarrow u_2 = 5$
x_{23}	$u_2 + v_3 = 9$	$u_2 = 5 \rightarrow v_3 = 4$
x_{24}	$u_2 + v_4 = 20$	$u_2 = 5 \rightarrow v_4 = 15$
x_{31}	$u_3 + v_1 = 4$	$u_3 = 3 \rightarrow v_1 = 1$
x_{34}	$u_3 + v_4 = 18$	$v_4 = 15 \rightarrow u_3 = 3$

Avaliando as variáveis não básicas:

Tabela 8: Avaliação das variáveis não básicas para a segunda iteração

Variável não básica	$u_i + v_j - c_{ij}$
x_{11}	$u_1 + v_1 - c_{11} = 0 + 1 - 10 = -9$
x_{13}	$u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
x_{14}	$u_1 + v_4 - c_{14} = 0 + 15 - 11 = 4$
x_{21}	$u_2 + v_1 - c_{21} = 5 + 1 - 12 = -6$
x_{32}	$u_3 + v_2 - c_{32} = 3 + 2 - 14 = -9$
x_{33}	$u_3 + v_3 - c_{33} = 3 + 4 - 16 = -9$

A Tabela 9 resume os cálculos para a segunda iteração.

Tabela 9: Segunda iteração

	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Oferta
$u_1 = 0$	0	15			15
$u_2 = 5$		0	15	10	25
$u_3 = 3$	5			5	10
Demanda	5	15	15	15	

Portanto aqui vemos que a variável que entra na base é x_{14} . Faz-se novamente o circuito fechado, agora para x_{14} .

Tabela 10: Circuito fechado para x_{14}

	$v_1 = 1$	$v_2 = 2$	$v_3 = 4$	$v_4 = 15$	Oferta
$u_1 = 0$		$15 - \alpha$		α	15
$u_2 = 5$		$0 + \alpha$	15	$10 - \alpha$	25
$u_3 = 3$	5			5	10
Demanda	5	15	15	15	

Então para $\alpha \geq 0$ os novos valores das variáveis permanecem não negativos se:

$$x_{12} = 15 - \alpha \geq 0 \quad (18)$$

$$x_{24} = 10 - \alpha \geq 0. \quad (19)$$

Ou seja, o valor máximo de α que satisfaz (18)-(19) é 10. Dessa maneira a variável que sai da base é x_{24} .

Portanto, foram selecionadas $x_{14} = 10$ para entrar na base e x_{24} para sair da base. Novamente é necessário ajustar os valores das variáveis básicas nos cantos do circuito.

Fazendo o cálculo dos multiplicadores para a terceira iteração:

Tabela 11: Equações para a terceira iteração

Variável básica	Equação (u, v)	Solução
x_{12}	$u_1 + v_2 = 2$	$u_1 = 0 \rightarrow v_2 = 2$
x_{14}	$u_1 + v_4 = 11$	$u_1 = 0 \rightarrow v_4 = 11$
x_{22}	$u_2 + v_2 = 7$	$v_2 = 2 \rightarrow u_2 = 5$
x_{23}	$u_2 + v_3 = 9$	$u_2 = 5 \rightarrow v_3 = 4$
x_{31}	$u_3 + v_1 = 4$	$u_3 = 7 \rightarrow v_1 = -3$
x_{34}	$u_3 + v_4 = 18$	$v_4 = 11 \rightarrow u_3 = 7$

E avaliando as variáveis não básicas:

Tabela 12: Avaliação das variáveis não básicas para a terceira iteração

Variável não básica	$u_i + v_j - c_{ij}$
x_{11}	$u_1 + v_1 - c_{11} = 0 - 3 - 10 = -13$
x_{13}	$u_1 + v_3 - c_{13} = 0 + 4 - 20 = -16$
x_{21}	$u_2 + v_1 - c_{21} = 5 - 3 - 12 = -10$
x_{24}	$u_2 + v_4 - c_{24} = 5 + 11 - 20 = -4$
x_{32}	$u_3 + v_2 - c_{32} = 7 + 2 - 14 = -5$
x_{33}	$u_3 + v_3 - c_{33} = 7 + 4 - 16 = -5$

Como agora todos os valores de $u_i + v_j - c_{ij}$ são negativos para as variáveis não básicas, chegamos na solução ótima.

A Tabela 13 resume os cálculos para a terceira iteração.

Tabela 13: Terceira iteração

	$v_1 = -3$	$v_2 = 2$	$v_3 = 4$	$v_4 = 11$	Oferta
$u_1 = 0$		5		10	15
$u_2 = 5$		10	15		25
$u_3 = 7$	5			5	10
Demanda	5	15	15	15	

Finalmente, a Tabela 14 resume a solução ótima.

Tabela 14: Resumo da solução ótima

Do Silo	Para o Moinho	Número de Cargas (Caminhões)
1	2	5
1	4	10
2	2	10
2	3	15
3	1	5
3	4	5

Logo, o valor do custo total de transporte é: $5 * 2 + 10 * 11 + 10 * 7 + 15 * 9 + 5 * 4 + 5 * 18 = 435$.

4 Uma Solução Computacional

Aqui será apresentada uma maneira de resolver o exemplo 3.2 utilizando o software AIMMS. Vale ressaltar que apesar de resolver o problema somente para os valores desse exemplo, a modelagem permite encontrar a solução para qualquer problema de transporte que siga a formulação clássica.

5 Resolvendo o Exemplo 3.2 no AIMMS

5.1 Declarações

Primeiro, é necessário declarar os *conjuntos* do problema. No caso, as origens e os destinos com índices de domínio i e j respectivamente.










Type	Set	        
Identifier	Origem	
Index domain		
Subset of		
Text		
Index	i	
Parameter		
Property		
Order by		

Figura 1: O conjunto de origens










Type	Set	        
Identifier	Destino	
Index domain		
Subset of		
Text		
Index	j	
Parameter		
Property		
Order by		

Figura 2: O conjunto de destinos

Depois, declara-se os *parâmetros* de oferta, demanda e o custo unitário da mercadoria, chamados de $Oferta(i)$, $Demanda(j)$ e $CustoUnitario(i, j)$, respectivamente.






Type	Parameter	        
Identifier	Oferta	
Index domain	i	
Text		
Range		
Unit		
Default		
Property		

Figura 3: O parâmetro de oferta










Type	Parameter	        
Identifier	Demanda	
Index domain	j	
Text		
Range		
Unit		
Default		
Property		

Figura 4: O parâmetro de demanda

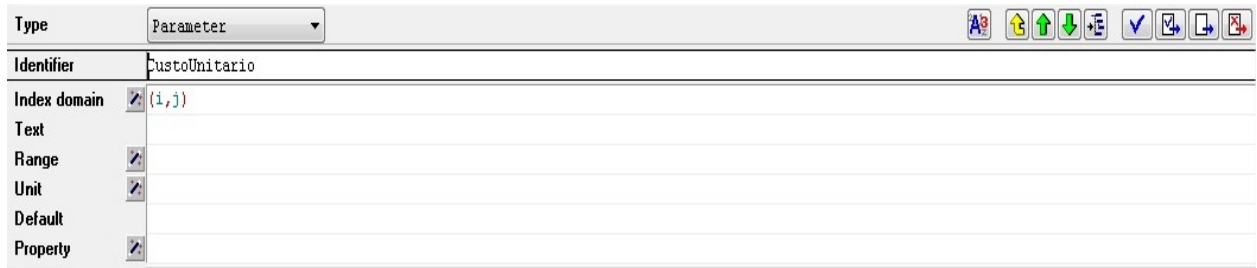


Figura 5: O parâmetro de custo

E então a *variável* de decisão nomeada de $Transporte(i, j)$ e também a função a ser minimizada, a do custo total de transporte, chamada $CustoTotalTransporte$. Essa função é declarada como uma *variável sem índices*.

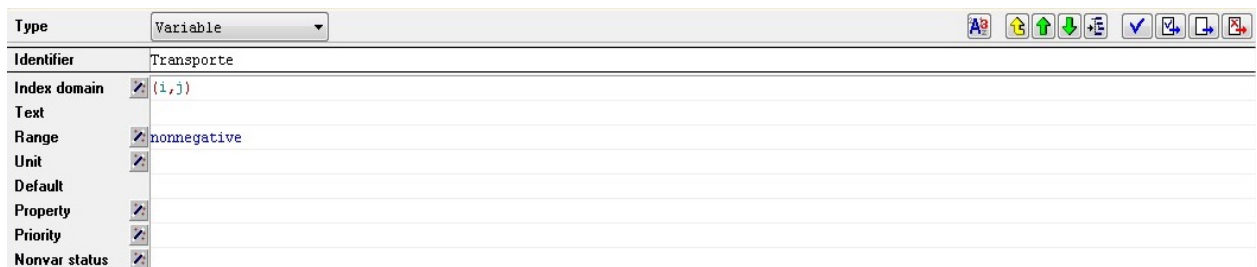


Figura 6: A variável de decisão

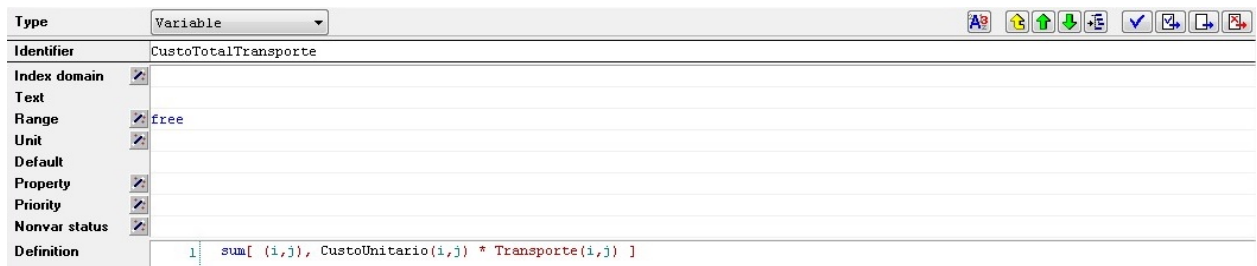


Figura 7: Função objetivo

Vê-se na Figura 6 que o valor da variável da decisão já é definido como não negativo e na Figura 7 que a variável de custo é declarada já com a sua definição, ou seja,

$$\sum_i \sum_j \text{CustoUnitario}(i, j) \text{Transporte}(i, j).$$

Em seguida são declaradas as duas *restrições* de oferta e demanda, $RestricaoOferta(i)$ e $RestricaoDemanda(j)$, respectivamente.

Type	Constraint
Identifier	RestricaoOferta
Index domain	i
Text	
Unit	
Property	
Definition	$\sum [j, Transporte(i,j)] = Oferta(i)$

Figura 8: A restrição de oferta

Type	Constraint
Identifier	RestricaoDemanda
Index domain	j
Text	
Unit	
Property	
Definition	$\sum [i, Transporte(i,j)] = Demanda(j)$

Figura 9: A restrição de demanda

Conforme as figuras 8 e 9, as restrições são declaradas da seguinte maneira:

$$\sum_j Transporte(i, j) = Oferta(i)$$

$$\sum_i Transporte(i, j) = Demanda(j).$$

Finalmente, declara-se o que é chamado *Mathematical Program* no AIMMS, que recebe as informações de quais variáveis e restrições serão consideradas, se o problema é de máximo ou de mínimo e qual é a função objetivo.

Type	Mathematical Progr
Identifier	PlanoMenorCustoTransporte
Objective	CustoTotalTransporte
Direction	minimize
Constraints	AllConstraints
Variables	AllVariables
Text	
Type	Automatic
Violation penalty	

Figura 10: O programa matemático

Ficamos então com as seguintes declarações:

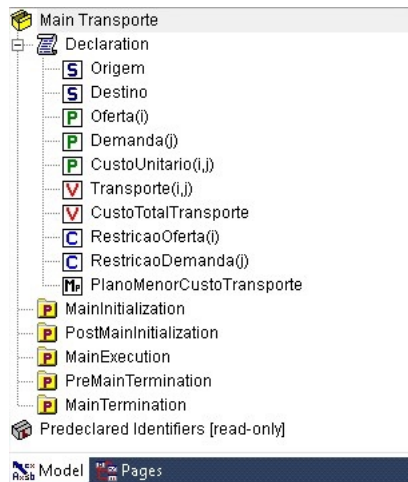


Figura 11: Declarações do problema

5.2 Preenchendo Valores e Resolvendo o Problema

São preenchidos os valores dos parâmetros do problema conforme os do Exemplo 3.2.

i		j	
Silo 1	15	Moinho 1	5
Silo 2	25	Moinho 2	15
Silo 3	10	Moinho 3	15
		Moinho 4	15

Figura 12: Valores das ofertas e demandas, respectivamente

	j	Moinho 1	Moinho 2	Moinho 3	Moinho 4
i					
Silo 1		10	2	20	11
Silo 2		12	7	9	20
Silo 3		4	14	16	18

Figura 13: Valores dos custos unitários

Então para resolver o problema basta abrir a aba *MainExecution* da Figura 11 e escrever o seguinte código:

```

Procedure      MainExecution
Body
1  solve PlanoMenorCustoTransporte;
2
3  if( PlanoMenorCustoTransporte.ProgramStatus <> 'Optimal' ) then
4      empty Transporte, CustoTotalTransporte;
5  endif;
6

```

Figura 14: Código para resolver o problema

Depois, deve-se clicar com o botão direito em cima desta aba e então em *RunProcedure*, para rodar o programa. Pressionando *Ctrl + p*, abre-se uma aba de progresso que tem algumas

informações sobre o problema e a solução ótima.

```

Progress
-----
READY
AIMMS      : Transporte.ams
Math.Program : PlanoMenorCustoTransporte
# Constraints : 8
# Variables  : 13
# Nonzeros   : 37
Model Type   : LP
Direction    : minimize

SOLVER      : XA 15
Phase       : Primal Simplex
Iterations   : 6
Objective    : 435
Best Solution : 435
Solving Time : 0.14 sec (Peak Mem: 2.0 Mb)
Program Status : Optimal
Solver Status : Normal completion

Total Time   : 0.33 sec
Memory Used  : 96.8 Mb
Memory Free  : 2233.2 Mb
    
```

Figura 15: Aba de progresso

Observe que *Best Solution* : 435, portanto a solução obtida computacionalmente é a mesma da encontrada em 3.2.

Para ter certeza disso, vemos que o resultado da variável $Transporte(i, j)$, ou seja, o número de cargas (caminhões) é o seguinte:

	j	Moinho 1	Moinho 2	Moinho 3	Moinho 4
i					
Silo 1			5		10
Silo 2			10	15	
Silo 3		5			5

Figura 16: Número de cargas

Comparando com a Tabela 14 da seção 3.2 podemos concluir que os resultados obtidos foram iguais.

6 Conclusão

O problema de transporte é um problema de Programação Linear que tem uma aplicação abrangente em diversas áreas relacionadas à Pesquisa Operacional. Nesse projeto, foram modeladas duas formulações para tal problema e com uma delas, a formulação clássica, foi possível chegar a um caso específico do mesmo, chamado de problema da designação.

Além disso, com o auxílio de um exemplo, foram explicadas duas maneiras de chegar à solução do problema de transporte: uma com o algoritmo simplex e a outra com o software AIMMS. Um problema pequeno pode ser resolvido com poucas iterações do algoritmo simplex, mas conforme o número de origens, destinos e conseqüentemente variáveis aumentam fica claro que essa maneira de resolver é inviável. O programa desenvolvido computacionalmente é a melhor escolha nesses casos, visto que resolve problemas maiores em segundos.

Referências Bibliográficas

- Goldberg, M. C.; Luna, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*. Elsevier Editora LTDA., 2005.
- Taha, H. A. *Pesquisa Operacional*. Pearson Prentice Hall, 2008.