Atualização: 29/03/2019 – Prof. Dr. Daniel Jorge Caetano

Aula 08: O Problema do Transporte: Truques e o Problema da Atribuição

Prof. Daniel Caetano

Objetivo: Descrever como lidar com produção diferente da demanda no Problema do Transporte e apresentar o Algoritmo Húngaro para a solução do Problema da Atribuição

Bibliografia: AHUJA *et al.*; MOREIRA; ARENALES *et al.*

INTRODUÇÃO

Na aula passada foi apresentado o método de solução para o Problema do Transporte. Nessa aula será visto como lidar com o caso de problemas com uma produção diferente da demanda, e, adicionalmente, como resolver o caso específico do Problema da Atribuição pelo Algoritmo Húngaro.

1. PROBLEMA DO TRANSPORTE: OFERTA E DEMANDA DIFERENTES

Conceitos Chave:

- Algoritmos funciona para Oferta = Demanda
- Oferta maior => Sem problemas, entretanto
- Demanda maior => Fábrica Fictícia
    * Oferta = Excesso de Demanda
    * Custo de transporte ZERO

É notório que o algoritmo apresentado funciona apenas no caso em a oferta total (das fábricas) e a demanda total (das lojas) é exatamente a mesma; por outro lado, na prática, isso raramente ocorre. Ou a oferta ou a demanda é maior.

Quando a oferta é maior, o problema pode ser resolvido normalmente. Simplesmente o total transportado será inferior à produção global das fábricas. Entretanto, se a demanda for maior, o problema se torna insolúvel.

Para contornar este problema, basta introduzir uma *fábrica fictícia*, capaz de produzir exatamente a quantidade que falta de unidades para o equilíbrio entre oferta e demanda, e ligada a todas as lojas, com *custo de transporte **ZERO***.

Isso faz com que o problema seja solúvel e o custo de transporte seja calculado (e minimizado) corretamente. Entretanto, as lojas que receberem mercadorias da fábrica fictícia estarão, na verdade, deixando de receber estes produtos.

2. O PROBLEMA DA ATRIBUIÇÃO OU DESIGNAÇÃO

O Problema da Atribuição/Designação é um problema bastante comum, tratando-se de um caso especial de um problema mais geral, o Problema do Transporte da Programação Linear.

Os Problemas de Atribuição/Designação são sempre num formato em que existe um determinado número de atividades a serem processadas e um determinado número de recursos para processá-las; o objetivo é atribuir (ou designar) qual atividade será processada por qual recurso.

Note que este é um tipo genérico de problema; isto significa que os as atividades podem ser produtos, projetos ou trechos de código, por exemplo. Da mesma forma, os recursos podem ser equipamentos, equipes de projetistas ou CPUs, por exemplo.

O objetivo neste problema é alocar as atividades aos recursos de processamento de forma a minimizar a soma do custo de cada processamento.

2.1. Um Problema de Atribuição

Como exemplo, considere o seguinte problema (MOREIRA, 2006, pág. 122, modificado):

Exemplo:

Em uma fábrica temos dois trabalhos T1, T2 e T3, que podem ser processados por 3 máquinas diferentes: M1, M2 e M3. Devido à diferenças tecnológicas nas máquinas, o tempo para que cada uma delas realize cada um dos trabalhos é diferente, estando expressados na tabela abaixo:

Máquina \ Trabalho	T1	T2	T3
M1	10	5	8
M2	12	9	15
M3	9	12	10

2.2. Modelagem Matemática do Problema da Atribuição

Considere para a resolução as variáveis de decisão X_{ij}, sendo i o número da máquina e j o número do trabalho: assim, X_{12} indica se a máquina 1 foi (ou não) atribuída para o trabalho 2; X_{31} indica se a máquina 3 foi (ou não) atribuída ao trabalho 1. Ou seja: quando uma destas variáveis for 1, houve a atribuição. Se ela for 0, não houve. Exemplos:

Se $X_{12} = 1$, a máquina 1 **foi** atribuída ao trabalho 2.
Se $X_{12} = 0$, a máquina 1 **não foi** atribuída ao trabalho 2.
Se $X_{31} = 1$, a máquina 3 **foi** atribuída ao trabalho 1.
Se $X_{31} = 0$, a máquina 3 **não foi** atribuída ao trabalho 1.

Note que as variáveis X_{ij} só podem assumir valores 0 ou 1, já que não faz sentido dizer que uma máquina foi "meio" atribuída a uma atividade e "meio" atribuída a outra. Assim, como cada máquina só pode ser atribuída a um trabalho, a conseqüência é que se $X_{11} = 1$, então X_{12} e X_{13} precisam ser, obrigatoriamente, iguais a 0. Isso significa dizer que se a máquina 1 foi usada para o trabalho 1 ($X_{11} = 1$) esta máquina 1 não pode ser usada para o trabalho 2 e 3 ($X_{12} = X_{13} = 0$). Da mesma forma, se $X_{12} = 1$, então $X_{11} = X_{13} = 0$... ou ainda, se $X_{13} = 1$, então $X_{11} = X_{12} = 0$. Ora, como é possível ver, a soma dos valores de $X_{1j} = 1$, sempre! Isso pode ser descrito assim:

$$X_{11} + X_{12} + X_{13} = 1, \qquad X_{1j} \in \{ 0, 1\}$$

O que foi dito sobre a máquina 1, pode também ser dito sobre a máquina 2 e sobre a máquina 3:

$$X_{21} + X_{22} + X_{23} = 1, \qquad X_{2j} \in \{ 0, 1\}$$
$$X_{31} + X_{32} + X_{33} = 1, \qquad X_{3j} \in \{ 0, 1\}$$

Juntando todas estas restrições, resulta:

$$X_{11} + X_{12} + X_{13} = 1$$
$$X_{21} + X_{22} + X_{23} = 1$$
$$X_{31} + X_{32} + X_{33} = 1$$

$$X_{ij} \in \{ 0, 1\}$$

Por outro lado, é sabido também que os trabalhos só podem ser designados para uma máquina de cada vez, também não fazendo sentido dizer que um trabalho foi "meio" atribuído a uma máquina e "meio" atribuído a outra. Assim, ocorre que se $X_{11} = 1$, então X_{21} e X_{31} precisam ser, obrigatoriamente, iguais a 0. Isso significa dizer que se o trabalho 1 foi alocado para a máquina 1 ($X_{11} = 1$) este mesmo trabalho não pode ser alocado para as máquians 2 e 3 ($X_{21} = X_{31} = 0$). Da mesma forma, se $X_{21} = 1$, então $X_{11} = X_{31} = 0$... ou ainda, se $X_{31} = 1$, então $X_{11} = X_{21} = 0$. Ora, como é possível ver, a soma dos valores de $X_{i1} = 1$, sempre! Isso pode ser descrito da seguinte maneira:

$$X_{11} + X_{21} + X_{31} = 1, \qquad X_{i1} \in \{ 0, 1\}$$

O que foi dito sobre o trabalho 1, pode também ser dito sobre o trabalho 2 e sobre o trabalho 3:

$X_{12} + X_{22} + X_{32} = 1, \qquad X_{i2} \in \{ 0, 1\}$
$X_{13} + X_{23} + X_{33} = 1, \qquad X_{i3} \in \{ 0, 1\}$

Juntando todas estas restrições, teremos:

$X_{11} + X_{21} + X_{31} = 1$
$X_{12} + X_{22} + X_{32} = 1$
$X_{13} + X_{23} + X_{33} = 1$

$X_{ij} \in \{ 0, 1\}$

Com isso, pode-se apresentar a definição completa das restrições:

$X_{11} + X_{12} + X_{13} = 1$ <= Máquina 1 só pega um trabalho
$X_{21} + X_{22} + X_{23} = 1$ <= Máquina 2 só pega um trabalho
$X_{31} + X_{32} + X_{33} = 1$ <= Máquina 3 só pega um trabalho
$X_{11} + X_{21} + X_{31} = 1$ <= Trabalho 1 só está em uma máquina
$X_{12} + X_{22} + X_{32} = 1$ <= Trabalho 2 só está em uma máquina
$X_{13} + X_{23} + X_{33} = 1$ <= Trabalho 3 só está em uma máquina

$X_{ij} \in \{ 0, 1\}$

Mas ainda falta a definição de uma função objetivo! Bem, a função objetivo é a soma do custo de cada atribuição realizada. Como X_{ij} indica exatamente se uma atribuição foi feita ou não, basta multiplicar o custo de cada atribuição (dados pelo problema) pela variável X_{ij}, que identifica se aquela atribuição foi feita:

Máquina \ Trabalho	T1	T2	T3
M1	10	5	8
M2	12	9	15
M3	9	12	10

Exemplo 1:
Custo da atribuição da M1 ao T1: **10**
Variável que indica se esta atribuição foi feita (1 se sim, 0 se não): $\mathbf{X_{11}}$
Custo final desta atribuição: $10^* X_{11}$

Exemplo 2:
Custo da atribuição da M2 ao T3: **15**
Variável que indica se esta atribuição foi feita (1 se sim, 0 se não): $\mathbf{X_{23}}$
Custo final desta atribuição: $15^* X_{23}$

Jutando o custo das 9 possíveis atribuições, define-se a função objetivo:

F.O.:
[MIN] $10X_{11} + 5X_{12} + 8X_{13} + 12X_{21} + 9X_{22} + 15X_{23} + 9X_{31} + 12X_{32} + 10X_{33}$
S.A.: $X_{11} + X_{12} + X_{13} = 1$
$X_{21} + X_{22} + X_{23} = 1$
$X_{31} + X_{32} + X_{33} = 1$
$X_{11} + X_{21} + X_{31} = 1$
$X_{12} + X_{22} + X_{32} = 1$
$X_{13} + X_{23} + X_{33} = 1$

$X_{ij} \in \{ 0, 1\}$

Enfim, o modelo completo. Entretanto, ainda falta colocar este modelo na forma padrão, o que acrescentaria uma variável artificial em cada restrição, ampliando o número de variáveis de 9 para 15.

Observe como um problema simples e pequeno, de atribuição de 3 máquinas a 3 trabalhos, tornou-se um modelo matemático enorme, com 15 variáveis (colunas do Simplex) e 6 restrições (linhas do Simplex).

Para piorar, o problema exige que as respostas sejam números inteiros, o que provavelmente impede o Simplex (sozinho) de resolvê-lo, sendo necessário um algoritmo que engloba o Simplex, chamado "Branch and Bound" que pode vir a ter que re-executar diversas vezes o Simplex, podendo ter que adicionar até mais 9 restrições (linhas do Simplex), uma para cada variável de decisão, totalizando 15 variáveis e 15 restrições.

Não é preciso ir muito longe para perceber que esse problema pode demorar um bom tempo para ser resolvido... e que, para problemas muito maiores, a execução será inviável.

É para resolver este tipo de problema que foi criado o algoritmo húngaro, que será apresentado a seguir.

3. O ALGORITMO HÚNGARO

Assim como o Simplex, o Algoritmo Húngaro pode ser descrito como uma seqüência de operações matemáticas que, quando aplicadas, revelam a solução ótima para um problema de atribuição.

O Algoritmo Húngaro é baseado na matriz dos tempos/custos de atribuição que se deseja minimizar. Além disso, ele é baseado no fato de existir **igual número de tarefas e máquinas** (ou seja, o número de linhas é igual ao de colunas). Há uma pressuposição que **qualquer tarefa pode ser alocada a qualquer máquina** e, finalmente, pressupõe que os **dados são custos** (ou outra grandeza qualquer) **que precisam ser minimizados**.

Os passos do Algoritmo Húngaro são:

1) Desenho da tabela de custos.
2) Seleção do menor valor de cada linha.
3) Subtração deste número de todos da mesma linha que ele.
4) Seleção do menor valor de cada coluna.
5) Subtração deste número de todos da mesma coluna que ele.
6) Determinação da ordem da matriz.
7) Traçar o *menor número* de retas horizontais e verticais que passem por todos os zeros.
8) Comparar o número de retas necessário com a ordem da matriz.
9) Se o número de retas for menor que a ordem da matriz, deve ser selecionado o menor número não coberto pelas retas. Se for igual, o algoritmo acabou... passo 13.
10) Subtração deste número de todos que não estiverem cobertos por retas.
11) Adição do número selecionado no passo 9 em todas as interseções de retas.
12) Voltar ao passo 7.
13) Determinação da atribuição.

3.1. Aplicação do Algoritmo Húngaro a um Problema

Para aplicar o algoritmo, será usado o mesmo exemplo (MOREIRA, 2006, pág. 122, modificado) visto anteriormente.

Exemplo:

Em uma fábrica temos dois trabalhos T1, T2 e T3, que podem ser processados por 3 máquinas diferentes: M1, M2 e M3. Devido à diferenças tecnológicas nas máquinas, o tempo para que cada uma delas realize cada um dos trabalhos é diferente, estando expressados na tabela abaixo:

Máquina \ Trabalho	T1	T2	T3
M1	10	5	8
M2	12	9	15
M3	9	12	10

Com base neste problema será apresentado o algoritmo húngaro para o problema de atribuição, visando encontrar qual máquina deve realizar qual trabalho de forma a minimizar o número total de horas de máquina gastos.

3.1.1. Aplicação do Algoritmo

O Algoritmo Húngaro é composto de várias etapas, que serão vistas a seguir, para resolver o problema previamente apresentado. Segue, agora, um guia detalhado de como executar cada etapa, passo a passo.

Passo 1: Desenho da tabela de custos

Máquina \ Trabalho	T1	T2	T3
M1	10	5	8
M2	12	9	15
M3	9	12	10

Passo 2: Seleciona-se o menor valor de cada linha...

Máquina \ Trabalho	T1	T2	T3
M1	10	>>5<<	8
M2	12	>>9<<	15
M3	>>9<<	12	10

Passo 3: Subtrai-se este número de todos as células da mesma linha que ele

Máquina \ Trabalho	T1	T2	T3
M1	5	0	3
M2	3	0	6
M3	0	3	1

Passo 4: Seleciona-se o menor valor de cada coluna

Máquina \ Trabalho	T1	T2	T3
M1	5	>>0<<	3
M2	3	0	6
M3	>>0<<	3	>>1<<

Passo 5: Subtrai-se este número de todos as células da mesma coluna que ele

Máquina \ Trabalho	T1	T2	T3
M1	5	0	2
M2	3	0	5
M3	0	3	0

Passo 6: Determina-se a ordem da matriz

A ordem da matriz é 3, já que ela é uma matriz 3x3.

Passo 7: Traça-se o |menor| número de retas (h/v) que passem por todos os zeros

Máquina \ Trabalho	T1	T2	T3
M1	5	0|	2
M2	3	0|	5
M3	-----------0-----------	-----------3|-----------	-----------0-----------

Passo 8: Compara-se o número de retas com a ordem da matriz

Há duas retas: a que passa na linha M3 e a que passa na coluna T2. Como o número de retas é MENOR que a ordem (três), segue-se para o passo 9. Caso o número fosse igual a três, a seqüência seria a partir do passo 13. Note que se o número de retas for MAIOR que a ordem, certamente há erro no traçado das retas.

Passo 9: Caso "no. de retas < ordem", seleciona-se o menor valor "não coberto"

Máquina \ Trabalho	T1	T2	T3
M1	5	0|	>>2<<
M2	3	0|	5
M3	-----------0-----------	-----------3|-----------	-----------0-----------

Passo 10: Subtrai-se valor selecionado no passo 9 de todos os "não cobertos"

Máquina \ Trabalho	T1	T2	T3
M1	3	0|	0
M2	1	0|	3
M3	-----------0-----------	-----------3|-----------	-----------0-----------

Passo 11: Soma-se o valor selecionado no passo 9 na interseção das retas

Máquina \ Trabalho	T1	T2	T3
M1	3	0|	0
M2	1	0|	3
M3	-----------0-----------	-----------5|-----------	-----------0-----------

Passo 12: Traça-se o |menor| número de retas que passem por todos os zeros

Máquina \ Trabalho	T1	T2	T3
M1	----------- 3 -----------	----------- 0 |-----------	----------- 0 -----------
M2	1	0 |	3
M3	----------- 0 -----------	----------- 3 |-----------	----------- 0 -----------

Caso o número de retas ainda seja menor que a ordem, volta-se para o passo 8 e reprete-se até conseguir um número de retas igual à ordem da matriz.

Passo 13: Determina-se a atribuição

A primeira linha ou coluna que aparece apenas UM zero deve ser localizada e a atribuição feita no ponto em que aparece o zero. Neste exemplo: A linha M2 tem apenas UM zero (coluna T2). Assim, o trabalho T2 será atribuído à máquina M2, e essa linha e coluna podem ser eliminadas (já que a M2 e o T2 não poderão ser atribuídos a mais nada):

Máquina \ Trabalho	T1	T2	T3
M1	3	0	0
M2	1	0	3
M3	0	3	0

Resultando em:

Máquina \ Trabalho	T1	T3
M1	3	0
M3	0	0

Localizando a próxima linha ou coluna que aparece apenas UM zero, define-se a próxima atribuição. Neste exemplo: A linha M1 tem apenas UM zero (coluna T3). Assim, o trabalho T3 será atribuído à máquina M1, e essa linha e coluna podem ser eliminadas:

Máquina \ Trabalho	T1	T3
M1	3	0
M3	0	0

Resultando em:

Máquina \ Trabalho	T1
M3	0

Finalmente, o trabalho T1 será atribuído à máquina M3. Solução:

T1 => M3, custo 9
T2 => M2, custo 9
T3 => M1, custo 8

Custo final total: 26 horas.

3.2. Tornando o Algoritmo Húngaro Genérico

Foi visto anteriormente que, para que o Algoritmo Húngaro funcione, existe a necessidade de obedecer alguns critérios... mas e quando isso não ocorre? Neste caso, serão usados alguns truques para "forçar" os critérios necessários.

Número de Linhas e Colunas Diferente: Neste caso devem ser criadas linhas ou colunas fictícias (conforme o caso), preenchendo o custo de todas as células desta linha/coluna acrescentada como 0.

Problema de Maximização ao invés de minimização: Neste caso, procura-se o maior número na matriz. Encontrado este número, em cada célula indicar o resultado da operação "Maior_Número - Valor_Original_Da_Célula_Atual".

Alocações Impossíveis: Quando alguma alocação é impossível, basta indicar seu custo com um valor excessivamente alto. Normalmente a letra "M" é usada para representar este valor.

4. BIBLIOGRAFIA

MOREIRA, D.A. **Pesquisa Operacional: Curso Introdutório**. [S.I.]: Ed. Thomson Pioneira, 2007.

ARENALES, M; ARMENTANO, V; MORABITO, R; YANASSE, H. **Pesquisa Operacional**. Rio de Janeiro: Editora Campus, 2006

AHUJA, R.K; MAGNANTI, T.L; ORLIN, J.B. **Network Flows**: Theory, Algorithms and Applications. New Jersey: Prentice Hall, 1993.