Notas da Aula 01: Introdução à Pesquisa Operacional
Prof. Daniel Caetano

Objetivo: Apresentar o que é Pesquisa Operacional e introduzí-la no âmbito dos Sistemas de Informação

Bibliografia:
- MOREIRA, D.A. *Pesquisa Operacional: Curso Introdutório, 2006*.
- ARENALES, M; ARMENTANO, V; MORABITO, R; YANASSE, H. Pesquisa Operacional. Rio de Janeiro: Editora Campus, 2006

Introdução

A Pesquisa Operacional é área do conhecimento que estuda problemas de como conduzir e coordenar algumas operações dentro de uma organização. Em outras palavras, Pesquisa Operacional (ou PO) é a aplicação de métodos analíticos para auxiliar os executivos a tomar melhores decisões. Tanto quanto possível, a PO busca obter a melhor solução possível, chamada "**solução ótima**" para um dado problema. Chamamos isso de "otimizar" uma operação.

De forma simplificada, otimizar significa encontrar uma combinação de fatores de operação que nos permite o melhor desempenho possível. Em outras palavras, se vamos transportar carga e podemos fazer isso de diversas formas, otimizar é determinar todos as características do transporte que nos trará um menor custo ou tempo, por exemplo. Da mesma forma, se queremos transportar um certo conjunto de dados por uma rede como a internet, onde temos diversos caminhos pelos quais uma informação pode ser transmitida, otimizar significa determinar o modo de transmissão e o caminho da transmissão de forma que a comunicação seja o mais rápida possível.

1. Histórico

O início da Pesquisa Operacional data do início do século XX, tendo o termo Pesquisa Operacional sido cunhado no fim da década de 1930, para descrever a atuação de cientistas na análise de problemas militares.

Seu uso foi muito mais intenso na Segunda Guerra Mundial, devido à necessidade de alocar com urgência e da melhor forma possível diversos recursos escassos como munição e alimento.

Terminadas as grandes guerras, os conhecimentos adquiridos acabaram por ser estendidos para organizações civis, e o uso da Pesquisa Operacional cresceu muito até meados da década de 1970. Nas guerras mais recentes, como as do Iraque, também a Pesquisa Operacional se motrou presente nas estratégias de ocupação e ataque.

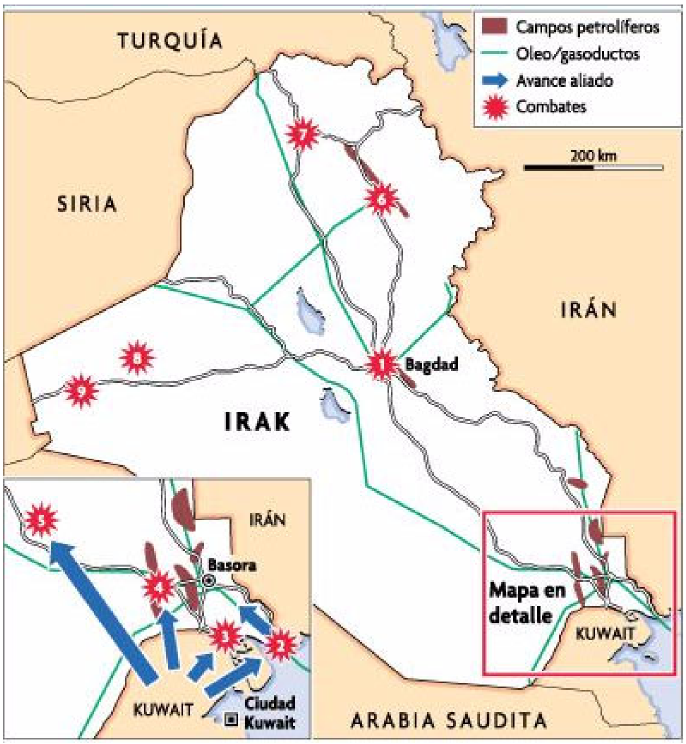


Figura 1 - Avanço das tropas americanas no Iraque (fonte: Reuters)

Ainda na época da Segunda Guerra, surgiram sociedades profissionais de cientistas da área de Pesquisa Operacional. Uma delas é a Operational Research Society (http://www.orsoc.org.uk) e outra é o Institute for Operations Reasearch and the Management Sciences - Informs (http://www.informs.org/). No Brasil existe a Sociedade Brasileira de Pesquisa Operacional - Sobrapo (http://www.sobrapo.org.br/).

2. Conteúdo do Curso

O processo todo de solução de um problema por PO pode ser sintetizado em alguns passos:

1) Definição da situação problema, ou seja, determinar quais são os objetivos desejados, quais são as restrições às soluções, quanto tempo existe para que o problema seja resolvido... e assim por diante. Neste passo informações genéricas e dispersas precisam ser transformadas em informações estruturadas e precisas.

2) Formulação de um modelo quantitativo, ou seja, formalizar todas as informações estruturadas no passo anterior em termos matemáticos, representando as relações entre variáveis dos problemas através de símbolos matemáticos. Na Programação Linear, uma das técnicas usadas pela PO, as relações são expressas por equações e inequações.

3) Resolução do Modelo, ou seja, manipular os valores das variáveis até que se obtenha a melhor solução possível, em termos do objetivo identificado no primeiro passo. É importante lembrar que algumas variáveis podem ser manipuladas livremente. Outras, entretanto, serão calculadas como resultado do processo. Estas últimas são chamadas **variáveis de decisão**.

4) Consideração de Fatores Imponderáveis, ou seja, analisar a solução encontrada e verificar se ela precisa ser modificada para incorporar fatores externos que não tenham sido considerados no modelo matemático.

5) Implementação da solução, ou seja, constatado que a solução é possível na prática, parte-se para a implementação, que deve ter sido projetada para uma transição o mais suave possível.

Neste curso, serão apresentados alguns problemas clássicos de PO e seus modelos decorrentes, apresentando uma introdução aos 3 primeiros destes passos, ficando um aprofundamento a cargo do aluno. Os passos 4 e 5 são deixados para um curso de pós graduação.

Apesar da ênfase às modelagens e aos métodos analíticos de solução, isso não deve ocultar, de forma alguma, o objetivo fundamental de se estudar Pesquisa Operacional, que é o de encontrar as melhors soluções possíveis para problemas práticos.

2.1. Passo 1 - Elaboração de Modelos Matemáticos

Como dito anteriormente, na Pesquisa Operacional buscamos a configuração ótima de um sistema. Entretanto, este ótimo só pode ser obtido a partir de um modelo matemático e, se o modelo não for corretamente desenvolvido, a solução encontrada pode não ser viável na realidade. Quanto melhor o modelo matemático, menor é a chance da solução ótima ser inviável na prática.

Por esta razão, pelo fato de a modelagem ser a parte que requer maior raciocínio e por ser a única etapa da resolução de um problema de PO que não pode ser feita com o auxílio de um computador, este curso dará uma grande ênfase na modelagem matemática. Trataremos apenas de problemas lineares.

2.2. Método Simplex

O Método Simplex é a forma mais geral de se resolver um problema de Programação Linear, que são os problemas mais clássicos e básicos da Pesquisa Operacional. O Método Simplex consiste de uma seqüência de operações sistemáticas que, após o número suficiente de iterações, nos apresenta a solução ótima de um problema de Programação Linear.

Problemas deste tipo podem ser "qual é o caminho mais curto de um ponto a outro?" ou "De qual maneira eu posso aplicar este meu recurso para obter maior lucro?". Pode ainda ser "Quantas unidades de cada tipo de produto eu produzo para maior lucro?" ou "Quais seriam os cardápios não repetidos que satisfariam todas as necessidades diárias de nutrientes e agradassem ao meu paladar?" e uma infinidade de outros.

2.3. O Algoritmo Húngaro

Como será possível observar pelo método, em alguns tipos de problemas (grandes) o método Simplex pode ter desempenho pouco satisfatório; Isso ocorre porque o Simplex é um método genérico, que não tira proveito de nenhuma característica específica de um dado tipo de problema. Assim, para estes casos em que mesmo um computador rápido pode levar dias, meses ou até anos (...) para encontrar uma solução ótima pelo Simplex, muitas vezes são propostas metodologias de cálculo específicas para encontrar uma solução ótima mais rapidamente.

Como um exemplo deste tipo de algoritmo, será apresentado o Algoritmo Húngaro, usado em problemas de atribuição e designação, que classicamente é apresentado como um problema de atribuição de equipes diferentes a projetos distintos, para minimizar o tempo total de conclusão dos projetos (horas pagas).

Entretanto, o Problema de Atribuição é muito comum além desta esfera mais clássica. Em computação distribuída, por exemplo, o computador responsável pelo controle de execução das aplicações deve decidir qual é o outro computador que irá processar uma informação ou executar um dado software. Como cada equipamento pode ter recursos distintos (velocidade de processamento, memória, dispositivos, etc), trata-se de um problema de atribuição que o Sistema Operacional terá de resolver.

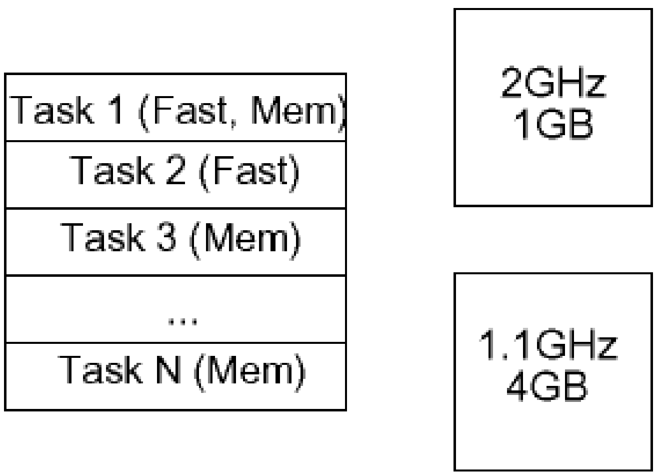


Figura 2 - Qual programa será executado em qual computador?

Hoje também estão se tornando comuns os computadores com dois processadores ou mais (Dual Core, como Core2 Duo, Athlon X2, etc) e isso traz uma tarefa ao Sistema Operacional, antes desnecessária: escolher qual dos processadores (neste caso, todos iguais) irá executar um dado thread (parte de um programa). Essa atribuição vai levar em conta qual a carga de cada processador, quais recursos cada um deles está usando, e assim por diante.

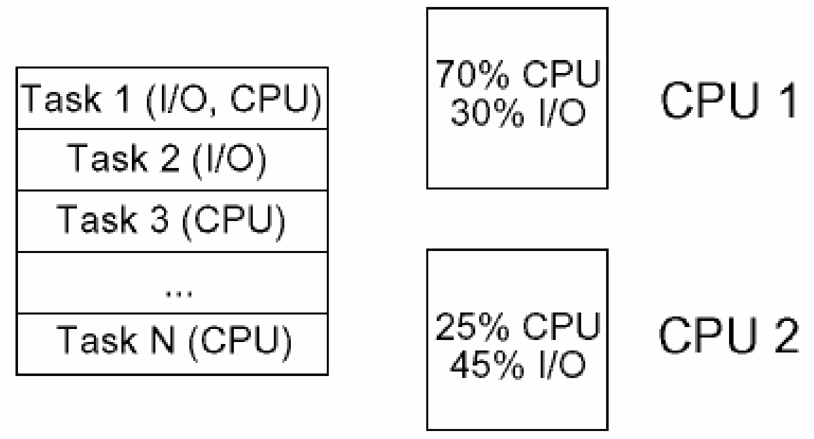


Figura 3 - Qual tarefa será atribuída a cada processador?

3. Bibliografia

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

Notas das Aulas 02 e 03: Modelagem Matemática
Prof. Daniel Caetano

Objetivo: Introduzir os conceitos de modelagem matemática.

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

Introdução

Como visto na aula anteior, a solução de problemas de Pesquisa Operacional, em específico os problemas da Programação Linear, podem ser resolvidos por um processo que pode ser sintetizado da seguinte forma:

1) Definição da situação problema, ou seja, determinar quais são os objetivos desejados, quais são as restrições às soluções, quanto tempo existe para que o problema seja resolvido... e assim por diante. Neste passo informações genéricas e dispersas precisam ser transformadas em informações estruturadas e precisas.

2) Formulação de um modelo quantitativo, ou seja, formalizar todas as informações estruturadas no passo anterior em termos matemáticos, representando as relações entre variáveis dos problemas através de símbolos matemáticos. Na Programação Linear, uma das técnicas usadas pela PO, as relações são expressas por equações e inequações.

3) Resolução do Modelo, ou seja, manipular os valores das variáveis até que se obtenha a melhor solução possível, em termos do objetivo identificado no primeiro passo. É importante lembrar que algumas variáveis podem ser manipuladas livremente. Outras, entretanto, serão calculadas como resultado do processo. Estas últimas são chamadas **variáveis de decisão**.

4) Consideração de Fatores Imponderáveis, ou seja, analisar a solução encontrada e verificar se ela precisa ser modificada para incorporar fatores externos que não tenham sido considerados no modelo matemático.

5) Implementação da solução, ou seja, constatado que a solução é possível na prática, parte-se para a implementação, que deve ter sido projetada para uma transição o mais suave possível.

Nestas aulas e nas seguintes serão focados os passos 1 e 2.

1. Modelagem Matemática para Programação Linear

Os modelos matemáticos que são, provavelmente, os mais populares são aqueles chamados de Modelos de Programação Linear. Estes modelos servem para representar problemas em que a relação entre as variáveis destes problemas possam ser expressas na forma de equações ou inequações lineares. As características fundamentais de um modelo deste tipo são:

1) Existe uma combinação de variáveis que deve ser maximizada ou minimizada, como por exemplo o custo de uma operação industrial ou rentabilidade média de ações. A esta combinação de variáveis dá-se o nome de **função objetivo**. Por exemplo: $7x + 2y$, onde x e y são variáveis de interesse, combinadas na proporção de 7 para 2. As variáveis que aparecem na função objetivo são as chamadas **variáveis de decisão**.

2) A estrutura do problema é tal que existe uma limitação de recursos, não sendo possível ter um lucro tão grande quanto se queira nem um custo tão pequeno quanto se queira. Estas limitações de recursos são expressas como equações ou inequações matemáticas e são chamadas **restrições**.

Assim, sempre que tivermos um problema em que desejamos saber qual é a melhor configuração de operação, ou seja, aquela que traz maior lucro ou reduz os custos, e pudermos representar estes objetivo e restrições através de equações ou inequações lineares, podemos modelá-lo como um problema de Programação Linear para posterior resolução.

1.1. Parâmetros x Variáveis de Decisão: Uma Primeira Noção

Independente do problema, antes de partirmos para a geração de um modelo, é importante ressaltar a diferença entre Parâmetros e Variáveis de Decisão. **Parâmetros são valores que são fornecidos** e nos quais não podemos mexer; devem permanecer como estão. **Variáveis de decisão são valores que podemos alterar**. As variáveis de decisão normalmente são expressas algebricamente, como x_1, x_2, x_3... note os índices.

Por exemplo, se **x** for a variável que indica quanto combustível foi abastecido, é possível dizer que x_1 é a quantidade que foi abastecida no posto 1, x_2 é a quantidade que foi abastecida no posto 2, x_3 é a quantidade que foi abastecida no posto 3... e assim por diante. É importante ressaltar que os **índices** têm um papel importante na especificação do modelo e não há um significado pré-definido para os mesmos. É tarefa do pesquisador descrever o que os índices significam. No caso, é definido que o índice especifica o número do posto no qual ocorreu o abastecimento.

1.2. Como Descrever o Desejo de Minimizar o Gasto com Combustível?

Imaginemos que o problema seja que vai ser realizada uma viagem e é desejável gastar o mínimo com combustível. Na estrada, existem três postos de combustível: p_1, p_2 e p_3. Em cada um dos postos, temos um custo diferente: c_1, c_2 e c_3. O que se deseja saber então é:

Quando eu abasteço no posto 1 (p_1), quanto abasteço no posto 2 (p_2) e quanto abasteço no posto 3 (p_3) para gastar o mínimo possível? Bem, se o desejo é saber quando será abastecido em cada posto, é porque **pretende-se tomar uma decisão** que envolve estes valores. Por esta razão estas são chamadas de **variávies de decisão**. Se chamar de **x** o quanto será abastecido, é possível dizer que x_1 é o quanto será abastecido no posto 1, x_2 é o quanto será abastecido no posto 2 e x_3 é o quanto será abastecido no posto 3.

O valor de x_1, x_2 ou x_3 vai ser o número de litros de combustível abastecido em cada posto. Se nada abastecer em um dado posto, o valor desta variável de decisão será zero. Por exemplo: se a melhor opção for **não** abastecer no posto 2, o valor de x_2 será 0 ($x_2 = 0$).

Com isto em mente, é possível dizer que o valor gasto no posto 1 será o quanto foi abastecido no posto 1, multiplicado pelo valor do litro de combustível no posto 1, ou seja:

$$x_1 * p_1$$

Afinal, se for abastecido 1 litro, será pago 1 vez o preço de um litro. Se for abastecido 2 litros, será pago 2 vezes o preço do litro e assim por diante. Da mesma forma, se nada for abastecido, x_1 valerá zero e nada será pago no posto 1.

Seguindo o mesmo raciocínio, é possível dizer que o custo no posto 2 será $x_2 * p_2$ e que no posto 3 o custo será $x_3 * p_3$. Chamando os custos em cada posto de c_1, c_2 e c_3, podemos reescrever os custos da seguinte forma:

$c_1 = x_1 * p_1$
$c_2 = x_2 * p_2$
$c_3 = x_3 * p_3$

O custo total de abastecimento será $\mathbf{c} = \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3$, que pode ser escrito da seguinte forma:

$$c = x_1 * p_1 + x_2 * p_2 + x_3 * p_3$$

...

Como o desejo é gastar o mínimo possível, ou seja, ter o menor custo possível, é possível dizer que queremos **minimizar** o custo. A maneira formal de dizer isso é:

$$[\text{MIN}]\ x_1 * p_1 + x_2 * p_2 + x_3 * p_3$$

Esta função, que descreve o que desejamos do problema, é chamada de **função objetivo** e, neste caso, x_1, x_2 e x_3 são **variáveis de decisão** e p_1, p_2 e p_3 são os parâmetros.

...

1.3. Restrições

Na seção anterior, foi representado o desejo de gastar o mínimo possível. Entretanto, a solução para o problema apresentado é "fique em casa, não abasteça nada e não gaste nada", já que em lugar algum foi descrito que sair de casa é obrigatório.

Para que o problema representado se assemelhe mais com as necessidades, precisamos adicionar mais informações ao problema. Tais informações são adicionadas na forma de restrições, como por exemplo "Não é possível 'ficar em casa'". É preciso dizer isso usando as **variáveis de decisão** que já estão sendo usadas. No caso, a informação que pode ser usada é um consumo conhecido da viagem, que seria, por exemplo, de 50 litros de combustível.

Como acrescentar esta informação? Bem, se vai ser abastecido x_1 litros no posto 1, x_2 litros no posto 2 e x_3 litros no posto 3, é possível dizer que o total de litros abastecido é:

$$\text{total abastecido} = x_1 + x_2 + x_3$$

...

Ora, o total abastecido precisa ser um valor **maior ou igual** aos 50 litros necessários para a viagem; ou seja, formalmente:

$$x_1 + x_2 + x_3 \geqslant 50$$

...

Com isso, é possível apresentar um primeiro modelo matemático muito simplificado:

$$\begin{array}{ll} \text{[MIN]} & x_1 * p_1 + x_2 * p_2 + x_3 * p_3 \\ \text{Sujeito à:} & x_1 + x_2 + x_3 \geqslant 50 \end{array}$$

Ou ainda:

$$\begin{array}{ll} \text{[MIN]} & x_1 * p_1 + x_2 * p_2 + x_3 * p_3 \\ \text{S.A.} & x_1 + x_2 + x_3 \geqslant 50 \end{array}$$

...

1.4. Condição de Não-Negatividade

Em programação linear, uma restrição sempre existente (mas implícita) é a de que as variáveis de decisão não podem assumir valores negativos. Isso faz todo o sentido do mundo, uma vez que as variáveis de decisão praticamente sempre indicam quantidades. e quantidades negativas não fazem sentido. Por esta razão, podemos sempre incluir estas restrições ao modelo:

$x_1 \geqslant 0$
$x_2 \geqslant 0$
$x_3 \geqslant 0$

Ficando assim, o modelo final:

[MIN] $x_1 * p_1 + x_2 * p_2 + x_3 * p_3$
S.A. $x_1 + x_2 + x_3 \geqslant 50$
$x_1 \geqslant 0$
$x_2 \geqslant 0$
$x_3 \geqslant 0$

2.2. Modelagem Exemplo

Problema (extraído de MOREIRA, 2006):

Uma fábrica produz dois produtos, A e B. Cada um deve ser processado por duas máquinas, M_1 e M_2. Devido à programação de outros produtos que também usam estas máquinas, estão disponíveis para os produtos A e B apenas 24 horas da máquina M_1 e 16 horas da máquina M_2.

Para produzir uma unidade do produto A, são necessárias 4 horas em cada uma das máquinas e para produzir uma unidade do produto B, são necessárias 6 horas em M_1 e 2 horas em M_2. Cada unidade de A vendida gera um lucro de R$ 80,00 e cada unidade de B vendida gera um lucro de R$ 60,00.

Existe uma previsão de demanda máxima de 3 unidades para B, mas nenhuma restrição de demanda para A. Deseja-se saber: quanto produzir de cada produto para maximizar o lucro?

Solução

O primeiro passo que auxilia muito na solução, é fazer um pequeno quadro de informações. Por exemplo:

Produto	**Horas de M_1**	**Horas deM_2**	**Demanda Max**	**Lucro Unitário**
A	4	4	-	80
B	6	2	3	60
Horas Disp.	24	16	-	-

Neste problema, o objetivo é claro: maximizar o lucro. Assim, teremos uma função objetivo de maximização. Mas como podemos descrever esta função objetivo em termos do que temos?

Ora, temos o lucro que cada unidade de A e B geram: se uma unidade de A for vendida, o lucro será de R$ 80,00. Se uma unidade de B for vendida, o lucro será de R$ 60,00. Se considerarmos o número de unidades de A vendidas como x_A e o número de unidades de B vendidas como x_B, podemos dizer que:

Lucro pelas vendas de A = $80 * x_A$
Lucro pelas vendas de B = $60 * x_B$

Lucro Total = $80 * x_A + 60 * x_B$

Ora, essa é, então, nossa função objetivo, já que queremos maximizar este lucro... E x_A e x_B são as variáveis de decisão. A função objetivo pode ser formalizada como:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Esta é a primeira parte do nosso modelo, mas ele ainda está longe de estar completo... Afinal, da maneira que foi representado podemos definir um lucro infinito, e na prática isso não ocorre! Como contornar isso? Impondo as limitações que o próprio problema apresenta:

- Limitação de Horas de M_1 : 24
- Limitação de Horas de M_2 : 16
- Limitação de Demanda para B : 3

Como podemos escrever isso? Vejamos uma por uma.

<u>Limitação de Horas de M_1 : 24</u>

A máquina M_1 terá de ser compartilhada pela produção de A e B, uma vez que ambos a utilizam. Sabemos que cada unidade de A produzida consome 4 horas de M_1 e cada unidade de B produzida consome 6 horas de M_1. Ora, se multiplicarmos o número de unidades produzidas de A (x_A) por 4, teremos o número de horas de M_1 que é gasto com produção de A e multiplicando o número de unidades produzidas de B (x_B) por 6, teremos o número de horas de M_1 que é gasto com a produção de B:

Tempo de M_1 gasto com produção de A : $4 * x_A$
Tempo de M_1 gasto com produção de B : $6 * x_B$
Tempo total de M_1 : $4 * x_A + 6 * x_B$

Mas a limitação de horas de M_1 é 24 horas, ou seja, podemos usar M_1 por qualquer número de horas, desde que ele não exceda 24 horas. Podemos escrever da seguinte forma:

Tempo total de $M_1 = 4 * x_A + 6 * x_B$
Tempo total de $M_1 \leqslant 24$

Juntando ambos...

$4 * x_A + 6 * x_B \leqslant 24$ <= Restrição do número de horas de M_1

Limitação de Horas de M_2 : 16

A máquina M_2 também terá de ser compartilhada pela produção de A e B. Sabemos que cada unidade de A produzida consome 4 horas de M_2 e cada unidade de B produzida consome 2 horas de M_2. Ora, se multiplicarmos o número de unidades produzidas de A (x_A) por 4, teremos o número de horas de M_2 que é gasto com produção de A e multiplicando o número de unidades produzidas de B (x_B) por 2, teremos o número de horas de M_2 que é gasto com a produção de B:

Tempo de M_2 gasto com produção de A : $4 * x_A$
Tempo de M_2 gasto com produção de B : $2 * x_B$
Tempo total de M_2 : $4 * x_A + 2 * x_B$

Mas a limitação de horas de M_2 é 16 horas, ou seja, podemos usar M_2 por qualquer número de horas, desde que ele não exceda 16 horas. Podemos escrever da seguinte forma:

Tempo total de $M_2 = 4 * x_A + 2 * x_B$
Tempo total de $M_2 \leqslant 16$

Juntando ambos...

$4 * x_A + 2 * x_B \leqslant 16$ <= Restrição do número de horas de M_2

Limitação de Demanda para B : 3

Em tese, podemos produzir qualquer número de unidades de A e B, desde que respeitemos o limite de horas disponível em cada máquina. Entretanto, nos foi fornecida uma informação adicional: a de que caso sejam produzidos mais do que 3 unidades de B, as que excederem este valor não serão vendidas. Unidades não vendidas significam custo para produzir e nenhum lucro. Assim, não podemos deixar isso acontecer, pois isso faria com que o lucro da empresa fosse menor.

Para evitar este problema, basta adicionar uma limitação a mais, indicando que qualquer número de unidades produzidas de B (x_B) é adequado, desde que não exceda 3. Matematicamente:

$x_B \leqslant 3$ <= Restrição de Demanda para B

Condição de Não-Negatividade

Em programação linear, uma restrição sempre implícita é a de que as variáveis de decisão não podem assumir valores negativos. Isso faz todo o sentido do mundo, uma vez que as variáveis de decisão praticamente sempre indicam quantidades... e quantidades negativas não fazem sentido. Por esta razão, incluímos duas restrições no nosso modelo:

$x_A \geqslant 0; x_B \geqslant 0$ <= Restrições de não-negatividade

Modelo Final

Juntando a função objetivo com todas as restrições, temos o modelo matemático final para nosso problema:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Restrições:
$4 * x_A + 6 * x_B \leqslant 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B \leqslant 16$ <= Restrição do número de horas de M_2
$x_B \leqslant 3$ <= Restrição de Demanda para B
$x_A \geqslant 0; x_B \geqslant 0$ <= Restrições de não-negatividade

Onde as variáveis de decisão são x_A e x_B.

3. Exercícios L1

1) Um navio tem um limite de transporte de 300m^3 de carga ou 50t de carga. Ele será usado para transportar dois tipos de carga: a carga A é transportada em unidades de 60m3, que pesam 1t. A carga B é transportada em unidades de 25m^3, e pesam 8t. O lucro pelo transporte de cada unidade de A é R$ 150,00, e o lucro pelo transporte de cada unidade de B é de R$ 72,00. Deseja-se o modelo de programação linear com que se possa obter qual é a melhor composição de carga para a obtenção de máximo lucro.

2) Um computador (1) tem um limite de 4GB (considerado 1GB = 1000MB) de memória e seu usuário pode exectuar até executar até 72 horas de processamento por semana. Todos os dados que serão processados nestas 72 horas da semana precisam ser carregados ao mesmo tempo. Isso significa que tudo tem que caber nos 4GB de memória. Um cliente lhe muitos pacotes de dados, de quatro tipos diferentes:

a) 10 pacotes que exigem 150 MB, 1 hora de processamento cada um, pagando R$ 100,00 por unidade processada.

b) 25 pacotes que exigem 100 MB, 7 horas de processamento cada um, pagando R$ 500,00 por unidade processada.

c) 3 pacotes que exigem 500 MB, 4 horas de processamento cada um, pagando R$ 350,00 por unidade processada.

d) 7 pacotes que exigem 350 MB, 10 horas de processamento cada um, pagando R$ 650,00 por unidade processada.

Deseja-e o modelo de programação linear para definir quais pacotes serão processados para que o maior lucro seja obtido.

3) (livro) Uma empresa do ramo de confecções está considerando quanto deve produzir de seus dois modelos de terno, denominados Executivo Master e Caibem, de forma a maximizar o lucro. É impossível produzir quanto se queira de cada um, pois existem limitações nas horas disponíveis para costura em máquina e acabamento manual. Para a costura, existe um máximo de 180 horas-máquina disponíveis e para o acabamento existe um máximo de 240 homens-hora. Em termos de lucro unitário e produção, os dois modelos de terno apresentam as seguintes características:

a) Executivo Master
- Lucro unitário: R$ 120,00
- horas-máquina de costura por unidade: 2
- homens-hora de acabamento por unidade: 2

b) Caibem
- Lucro unitário: R$ 70,00
- horas-máquina de costura por unidade: 1
- homens-hora de acabamento por unidade: 4

Formule o problema como um modelo de programação linear.

3. Bibliografia

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

Notas da Aula 04: Resolução da L1
Prof. Daniel Caetano

Objetivo: Resolver a primeira lista de exercícios, treinando a concepção de modelos matemáticos para programação linear.

1. Solução do Exercício 1

Problema

1) Um navio tem um limite de transporte de 300m3 de carga ou 50t de carga. Ele será usado para transportar dois tipos de carga: a carga A é transportada em unidades de 60m3, que pesam 1t. A carga B é transportada em unidades de 25m3, e pesam 8t. O lucro pelo transporte de cada unidade de A é R$ 150,00, e o lucro pelo transporte de cada unidade de B é de R$ 72,00. Deseja-se o modelo de programação linear com que se possa obter qual é a melhor composição de carga para a obtenção de máximo lucro.

Solução

O primeiro passo que auxilia muito na solução, é fazer um pequeno quadro de informações. Por exemplo:

Carga	Volume	Peso	Lucro Unitário
A	60	1	150
B	25	8	72
Disponível	300	50	-

Neste problema, o objetivo é claro: maximizar o lucro. Assim, teremos uma função objetivo de maximização. Mas como podemos descrever esta função objetivo em termos do que temos?

Ora, temos o lucro que o transporte cada unidade de A e B geram: se uma unidade de A for transportada, o lucro será de R$ 150,00. Se uma unidade de B for transportada, o lucro será de R$ 72,00. Se considerarmos o número de unidades de A transportadas como x_A e o número de unidades de B transportadas como x_B, podemos dizer que:

Lucro pelo transporte de A = $150 * x_A$
Lucro pelo transporte de B = $72 * x_B$

Lucro Total = $150 * x_A + 72 * x_B$

Ora, essa é, então, nossa função objetivo, já que queremos maximizar este lucro... E x_A e x_B são as variáveis de decisão. A função objetivo pode ser formalizada como:

Função Objetivo:
[MAX] 150 * x_A + 72 * x_B

Esta é a primeira parte do nosso modelo, mas ele ainda está longe de estar completo... Afinal, da maneira que foi representado podemos definir um lucro infinito, e na prática isso não ocorre! Como contornar isso? Impondo as limitações que o próprio problema apresenta:

- Limitação de Volume : 300
- Limitação de Peso : 50

Como podemos escrever isso? Vejamos uma por uma.

Limitação de Volume: 300

O volume será compartilhado pelas cargas A e B, uma vez que ambos terão de ser colocados no mesmo navio. Sabemos que cada unidade de A transportada ocupa 60 m^3 e cada unidade de B transportada ocupa 25 m^3. Ora, se multiplicarmos o número de unidades transportadas de A (x_A) por 60, teremos o volume total ocupado pela carga A e multiplicando o número de unidades transportadas de B (x_B) por 25, teremos o volume total ocupado pela carga B:

Volume total da carga A : 60 * x_A
Volume total da carga B : 25 * x_B
Volume total ocupado : 60 * x_A + 25 * x_B

Mas a limitação de volume é 300 m^3, ou seja, podemos ocupar qualquer volume, desde que não exceda 300 m^3. Podemos escrever isso da seguinte forma:

Volume total ocupado = 60 * x_A + 25 * x_B
Volume permitido $\leqslant$ 300

Juntando ambos...

60 * x_A + 25 * x_B $\leqslant$ 300 <= Restrição de volume

Limitação de Peso: 50

O peso das cargas A e B também se somam para o cálculo do peso total, já que as cargas vão no mesmo navio. Sabemos que cada unidade de A transportada pesa 1t e cada unidade de B transportada ocupa 8t. Ora, se multiplicarmos o número de unidades transportadas de A (x_A) por 1, teremos o peso total da carga A e multiplicando o número de unidades transportadas de B (x_B) por 8, teremos o peso total da carga B:

Peso total da carga A : 1 * x_A
Peso total da carga B : 8 * x_B

Peso total : $1 * x_A + 8 * x_B$

Mas a limitação de peso é 50t, ou seja, o peso total de nossa carga pode ser qualquer um, desde que não exceda 50t. Podemos escrever isso da seguinte forma:

Peso total = $1 * x_A + 8 * x_B$
Peso permitido $\leqslant 50$

Juntando ambos...

$1 * x_A + 8 * x_B \leqslant 50$ <= Restrição de peso

Condição de Não-Negatividade

Em programação linear, uma restrição sempre implícita é a de que as variáveis de decisão não podem assumir valores negativos. Isso faz todo o sentido do mundo, uma vez que as variáveis de decisão praticamente sempre indicam quantidades... e quantidades negativas não fazem sentido. Por esta razão, incluímos duas restrições no nosso modelo:

$x_A \geqslant 0; x_B \geqslant 0$ <= Restrições de não-negatividade

Modelo Final

Juntando a função objetivo com todas as restrições, temos o modelo matemático final para nosso problema:

Função Objetivo:
[MAX] $150 * x_A + 72 * x_B$

Restrições:
$60 * x_A + 25 * x_B \leqslant 300$ <= Restrição de volume
$1 * x_A + 8 * x_B \leqslant 50$ <= Restrição de peso
$x_A \geqslant 0; x_B \geqslant 0$ <= Restrições de não-negatividade

Onde as variáveis de decisão são x_A e x_B.

2. Solução do Exercício 2

Problema

2) Um computador (1) tem um limite de 4GB (considerado 1GB = 1000MB) de memória e seu usuário pode exectuar até executar até 72 horas de processamento por semana. Todos os dados que serão processados nestas 72 horas da semana precisam ser carregados ao mesmo tempo. Isso significa que tudo tem que caber nos 4GB de memória. Um cliente lhe muitos pacotes de dados, de quatro tipos diferentes:

a) 10 pacotes que exigem 150 MB, 1 hora de processamento cada um, pagando R$ 100,00 por unidade processada.

b) 25 pacotes que exigem 100 MB, 7 horas de processamento cada um, pagando R$ 500,00 por unidade processada.

c) 3 pacotes que exigem 500 MB, 4 horas de processamento cada um, pagando R$ 350,00 por unidade processada.

d) 7 pacotes que exigem 350 MB, 10 horas de processamento cada um, pagando R$ 650,00 por unidade processada.

Deseja-e o modelo de programação linear para definir quais pacotes serão processados para que o maior lucro seja obtido.

Solução

O primeiro passo que auxilia muito na solução, é fazer um pequeno quadro de informações. Por exemplo:

Pacote	memória (MB)	horas de proc. (h)	lucro unitáro (R$)	Pacotes disponíveis
A	150	1	100	10
B	100	7	500	25
C	500	4	350	3
D	350	10	650	7
Disponível	4,000	72	-	-

Neste problema, o objetivo é claro: maximizar o lucro. Assim, teremos uma função objetivo de maximização. Mas como podemos descrever esta função objetivo em termos do que temos?

Ora, temos o lucro que o processamento de cada pacote A, B, C, D e E geram: se uma unidade de A for processada, o lucro será de R$ 100,00. Se uma unidade de B for vendida, o lucro será de R$ 500,00... e assim por diante. Se considerarmos o número de unidades de A, B, C e D processadas como x_A, x_B, x_C e x_D respectivamente, podemos dizer que:

Lucro pelo processamento de unidades A = 100 * x_A
Lucro pelo processamento de unidades B = 500 * x_B
Lucro pelo processamento de unidades C = 350 * x_C
Lucro pelo processamento de unidades D = 650 * x_D

Assim, o lucro total será:

Lucro Total = $100 * x_A + 500 * x_B + 350 * x_C + 650 * x_D$

Ora, essa é, então, nossa função objetivo, já que queremos maximizar este lucro... E x_A, x_B, x_C e x_D são as variáveis de decisão. A função objetivo pode ser formalizada como:

Função Objetivo:
[MAX] $100 * x_A + 500 * x_B + 350 * x_C + 650 * x_D$

Esta é a primeira parte do nosso modelo, mas ele ainda está longe de estar completo... Afinal, da maneira que foi representado podemos definir um lucro infinito, e na prática isso não ocorre! Como contornar isso? Impondo as limitações que o próprio problema apresenta:

- Limitação de Memória : 4000
- Limitação de Horas : 72
- Limitação de Unidades A: 10
- Limitação de Unidades B: 25
- Limitação de Unidades C: 3
- Limitação de Unidades D: 7

Como podemos escrever isso? Vejamos uma por uma.

Limitação de Memória: 4000

A memória será compartilhada por todos os pacotes a serem processados nas 72 horas. Assim, precisamos expressar o consumo de memória através das variáveis de decisão x_A, x_B, x_C e x_D. Sabemos que cada unidade de A, por exemplo, consome 150MB de memória. Cada unidade de C, por exemplo, consome 500MB de memória. Ora, se multiplicarmos o número de unidades processadas de A, B, C e D (x_A, x_B, x_C e x_D) pelo seu consumo de memória respectivo, teremos quanta memória os pacotes de um determinado tipo consumiram no total:

Memória consumida por pacotes A: $150 * x_A$
Memória consumida por pacotes B: $100 * x_B$
Memória consumida por pacotes C: $500 * x_C$
Memória consumida por pacotes D: $350 * x_D$
Memória consumida total: $150 * x_A + 100 * x_B + 500 * x_C + 350 * x_D$

Mas a limitação de memória total é 4000, ou seja, podemos consumir qualquer quantidade de memória, desde que não exceda 4000. Podemos escrever isso da seguinte forma:

Memória total necessária: $150 * x_A + 100 * x_B + 500 * x_C + 350 * x_D$
Memória total disponível: 4000

Juntando ambos...

$150 * x_A + 100 * x_B + 500 * x_C + 350 * x_D \leqslant 4000$ <= Restrição de Memória

Limitação de Horas : 72

As horas serão gastar por todos os processamentos (de pacotes A, B, C e D), sendo eles executados sequencialmente. Sabemos que cada unidade de A processada consome 1 hora e cada unidade de D processada consome 10 horas, por exemplo. Ora, se multiplicarmos o número de unidades processadas de A, B, C e D (x_A, x_B, x_C e x_D) pelo seu consumo de horas respectivo, teremos o número de horas consumidas no total devido ao processamento:

Horas consumidas por pacotes A: $1 * x_A$
Horas consumidas por pacotes B: $7 * x_B$
Horas consumidas por pacotes C: $4 * x_C$
Horas consumidas por pacotes D: $10 * x_D$
Horas consumidas total: $1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D$

Mas a limitação de horas é de 72, ou seja, podemos consumir qualquer número de horas, desde que não exceda 72. Podemos escrever isso da seguinte forma:

Horas necessárias: $1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D$
Horas disponíveis: 72

Juntando ambos...

$1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D \leqslant 72$ <= Restrição de Horas

Limitação de Unidades A, B, C e D: 10, 25, 3, 7 (respectivamente)

Além das restrições de hora e memória, temos também algumas restrições na quantidade máxima de cada tipo de unidade que podem ser processados. Por exemplo: só existem 10 unidades A para serem processadas; não faz sentido considerar o processamento de uma décima primeira unidade A. Assim, podemos dizer que o número de unidades processadas de A, B, C e D (x_A, x_B, x_C e x_D) deve ser inferior ou igual ao número de unidades disponíveis:

$x_A \leqslant 10$ <= Restrição de Unidades A
$x_B \leqslant 25$ <= Restrição de Unidades B
$x_C \leqslant 3$ <= Restrição de Unidades C
$x_D \leqslant 7$ <= Restrição de Unidades D

Condição de Não-Negatividade

Em programação linear, uma restrição sempre implícita é a de que as variáveis de decisão não podem assumir valores negativos. Isso faz todo o sentido do mundo, uma vez que as variáveis de decisão praticamente sempre indicam quantidades... e quantidades negativas não fazem sentido. Por esta razão, incluímos quatro restrições no nosso modelo:

$x_A \geqslant 0; x_B \geqslant 0; x_C \geqslant 0 ; x_D \geqslant 0$ <= Não-negatividade

Modelo Final

Juntando a função objetivo com todas as restrições, temos o modelo matemático final para nosso problema:

Função Objetivo:
$[MAX]\ 100 * x_A + 500 * x_B + 350 * x_C + 650 * x_D$

Restrições:
$150 * x_A + 100 * x_B + 500 * x_C + 350 * x_D \leqslant 4000$ <= Restrição de Memória
$1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D \leqslant 72$ <= Restrição de Horas
$x_A \leqslant 10$ <= Restrição de Unidades A
$x_B \leqslant 25$ <= Restrição de Unidades B
$x_C \leqslant 3$ <= Restrição de Unidades C
$x_D \leqslant 7$ <= Restrição de Unidades D
$x_A \geqslant 0; x_B \geqslant 0; x_C \geqslant 0 ; x_D \geqslant 0$ <= Não-negatividade

Onde as variáveis de decisão são x_A, x_B, x_C e x_D.

3. Solução do Exercício 3

Problema

3) (livro) Uma empresa do ramo de confecções está considerando quanto deve produzir de seus dois modelos de terno, denominados Executivo Master e Caibem, de forma a maximizar o lucro. É impossível produzir quanto se queira de cada um, pois existem limitações nas horas disponíveis para costura em máquina e acabamento manual. Para a costura, existe um máximo de 180 horas-máquina disponíveis e para o acabamento existe um máximo de 240 homens-hora. Em termos de lucro unitário e produção, os dois modelos de terno apresentam as seguintes características:

a) Executivo Master
- Lucro unitário: R$ 120,00
- horas-máquina de costura por unidade: 2
- homens-hora de acabamento por unidade: 2

b) Caibem
- Lucro unitário: R$ 70,00
- horas-máquina de costura por unidade: 1
- homens-hora de acabamento por unidade: 4

Formule o problema como um modelo de programação linear.

Solução

O primeiro passo que auxilia muito na solução, é fazer um pequeno quadro de informações. Por exemplo:

Terno	**horas-máquina**	**homens-hora**	**Lucro Unitário**
Executivo Master	2	2	120
Caibem	1	4	70
Disponível	180	240	-

Neste problema, o objetivo é claro: maximizar o lucro. Assim, teremos uma função objetivo de maximização. Mas como podemos descrever esta função objetivo em termos do que temos?

Ora, temos o lucro que a venda de cada Executivo Master (EM) e Caibem (C) geram: se uma unidade de EM for vendida, o lucro será de R$ 120,00. Se uma unidade de C for vendida, o lucro será de R$ 70,00. Se considerarmos o número de unidades de EM transportadas como x_{EM} e o número de unidades de C transportadas como x_C, podemos dizer que:

Lucro pela venda de EM = $120 * x_{EM}$
Lucro pela venda de C = $70 * x_C$

Lucro Total = $120 * x_{EM} + 70 * x_C$

Ora, essa é, então, nossa função objetivo, já que queremos maximizar este lucro... E x_{EM} e x_C são as variáveis de decisão. A função objetivo pode ser formalizada como:

Função Objetivo:
[MAX] $120 * x_{EM} + 70 * x_C$

Esta é a primeira parte do nosso modelo, mas ele ainda está longe de estar completo... Afinal, da maneira que foi representado podemos definir um lucro infinito, e na prática isso não ocorre! Como contornar isso? Impondo as limitações que o próprio problema apresenta:

- Limitação de Horas-Máquina : 180
- Limitação de Homens-Hora : 240

Como podemos escrever isso? Vejamos uma por uma.

Limitação de Horas-Máquina : 180

As horas-máquina serão compartilhadas pela produção dos ternos EM e C. Sabemos que cada unidade de EM produzida gasta 2 horas-máquina e cada unidade de C produzida gasta 1 hora-máquina. Ora, se multiplicarmos o número de unidades produzidas de EM (x_{EM}) por 2, teremos o número de horas-máquina consumidas pela produção de EM e multiplicando o número de unidades produzidas de C (x_C) por 1, teremos o número de horas-máquina consumidas pela produção de C:

Horas-máquina gastos por EM : $2 * x_{EM}$
Horas-máquina gastos por C : $1 * x_C$
Horas-máquina gastas : $2 * x_{EM} + 1 * x_C$

Mas a limitação de horas-máquina é de 180, ou seja, podemos gastar qualquer número de horas-máquina, desde que não exceda 180. Podemos escrever isso da seguinte forma:

Horas-máquina necessárias = $2 * x_{EM} + 1 * x_C$
Horas-máquina disponíveis $\leqslant 180$

Juntando ambos...

$2 * x_{EM} + 1 * x_C \leqslant 180$ <= Restrição de horas-máquina

Limitação de Homens-Hora : 240

Os homens-hora serão compartilhados pela produção dos ternos EM e C. Sabemos que cada unidade de EM produzida gasta 2 homens-hora e cada unidade de C produzida gasta 4 homens-hora. Ora, se multiplicarmos o número de unidades produzidas de EM (x_{EM}) por 2, teremos o número de homens-horas consumidos pela produção de EM e multiplicando o número de unidades produzidas de C (x_C) por 4, teremos o número de homens-hora consumidas pela produção de C:

Homens-hora gastos por EM : $2 * x_{EM}$
Homens-hora gastos por C : $4 * x_C$
Homens-hora gastos : $2 * x_{EM} + 4 * x_C$

Mas a limitação de homens-hora é de 240, ou seja, podemos gastar qualquer número de homens-hora, desde que não exceda 240. Podemos escrever isso da seguinte forma:

Homens-horas necessários = $2 * x_{EM} + 4 * x_C$
Homens-horas disponíveis $\leqslant 240$

Juntando ambos...

$2 * x_{EM} + 4 * x_C \leqslant 240$ <= Restrição de homens-hora

<u>Condição de Não-Negatividade</u>

Em programação linear, uma restrição sempre implícita é a de que as variáveis de decisão não podem assumir valores negativos. Isso faz todo o sentido do mundo, uma vez que as variáveis de decisão praticamente sempre indicam quantidades... e quantidades negativas não fazem sentido. Por esta razão, incluímos duas restrições no nosso modelo:

$x_{EM} \geqslant 0$; $x_C \geqslant 0$ <= Restrições de não-negatividade

<u>Modelo Final</u>

Juntando a função objetivo com todas as restrições, temos o modelo matemático final para nosso problema:

Função Objetivo:
[MAX] $120 * x_{EM} + 70 * x_C$

Restrições:
$2 * x_{EM} + 1 * x_C \leqslant 180$ <= Restrição de horas-máquina
$2 * x_{EM} + 4 * x_C \leqslant 240$ <= Restrição de homens-hora
$x_{EM} \geqslant 0$; $x_C \geqslant 0$ <= Restrições de não-negatividade

Onde as variáveis de decisão são x_{EM} e x_C.

Notas da Aula 05: Motivação: Solução Gráfica de Problemas
Prof. Daniel Caetano

Objetivo: Apresentar graficamente a solução de um problema de programação linear.
Bibliografia:
- MOREIRA, D.A. **Pesquisa Operacional: Curso Introdutório**. Ed. Pioneira, 2007.

Introdução

Apesar de não ser qualquer tipo de problema de programação linear que permite uma solução gráfica, alguns deles permitem e a apresentação deste tipo de solução pode ser bastante positivo para a compreensão dos problemas em si, da modelagem matemática e até mesmo o funcionamento do algoritmo Simplex que será apresentado nas próximas aulas.

Com este objetivo, esta aula será devotada à resolução gráfica de um dos problemas apresentados na segunda aula. Adicionalmente serão feitos alguns comentários com relação à análise dos resultados obtidos.

1. A Modelagem e Solução

Problema (extraído de MOREIRA, 2006):

Uma fábrica produz dois produtos, A e B. Cada um deve ser processado por duas máquinas, M_1 e M_2. Devido à programação de outros produtos que também usam estas máquinas, estão disponíveis para os produtos A e B apenas 24 horas da máquina M_1 e 16 horas da máquina M_2.

Para produzir uma unidade do produto A, são necessárias 4 horas em cada uma das máquinas e para produzir uma unidade do produto B, são necessárias 6 horas em M_1 e 2 horas em M_2. Cada unidade de A vendida gera um lucro de R$ 80,00 e cada unidade de B vendida gera um lucro de R$ 60,00.

Existe uma previsão de demanda máxima de 3 unidades para B, mas nenhuma restrição de demanda para A. Deseja-se saber: quanto produzir de cada produto para maximizar o lucro?

Modelo Final

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Sujeito a:
$4 * x_A + 6 * x_B \leqslant 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B \leqslant 16$ <= Restrição do número de horas de M_2
$x_B \leqslant 3$ <= Restrição de Demanda para B
$x_A \geqslant 0; x_B \geqslant 0$ <= Restrições de não-negatividade

Onde as variáveis de decisão são x_A e x_B.

1.1. Solução Gráfica

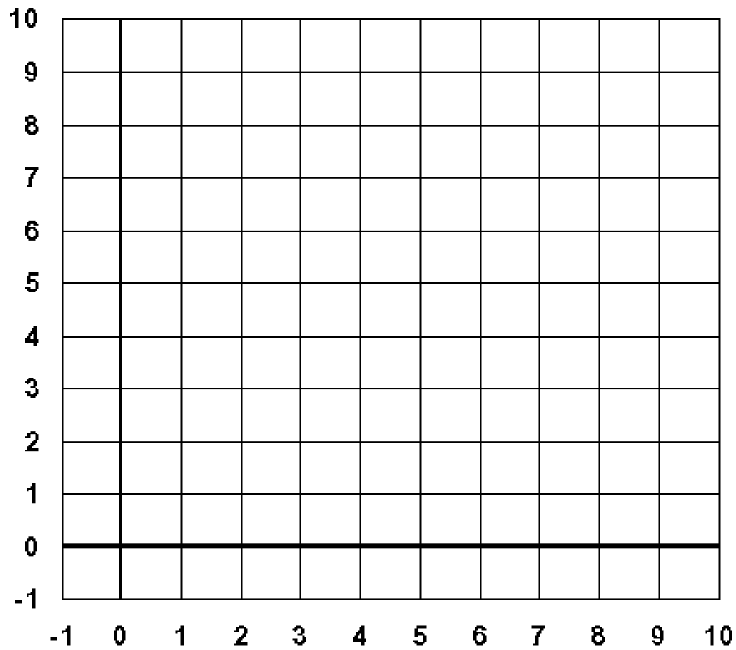
Sempre que um problema de programação linear tiver apenas duas variáveis, será possível resolvê-lo graficamente. Embora não seja a forma mais rápida de resolver um problema e seja um tanto limitada, é uma maneira interessante de entender o mecanismo de solução de problemas de programação linear.

A idéia por trás da solução gráfica é delimitar a área em que todas as soluções possíveis se encontram e então buscar, neste espaço - chamado **Espaço de Soluções** - a melhor solução possível.

Bem, se o desejado é encontrar as soluções **possíveis** e o que limita as soluções são as **restrições**, então iremos usar estas últimas para delimitar o espaço de soluções possíveis. Vejamos em mais detalhe as restrições:

$4 * x_A + 6 * x_B \leqslant 24$	<= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B \leqslant 16$	<= Restrição do número de horas de M_2
$x_B \leqslant 3$	<= Restrição de Demanda para B
$x_A \geqslant 0$; $x_B \geqslant 0$	<= Restrições de não-negatividade

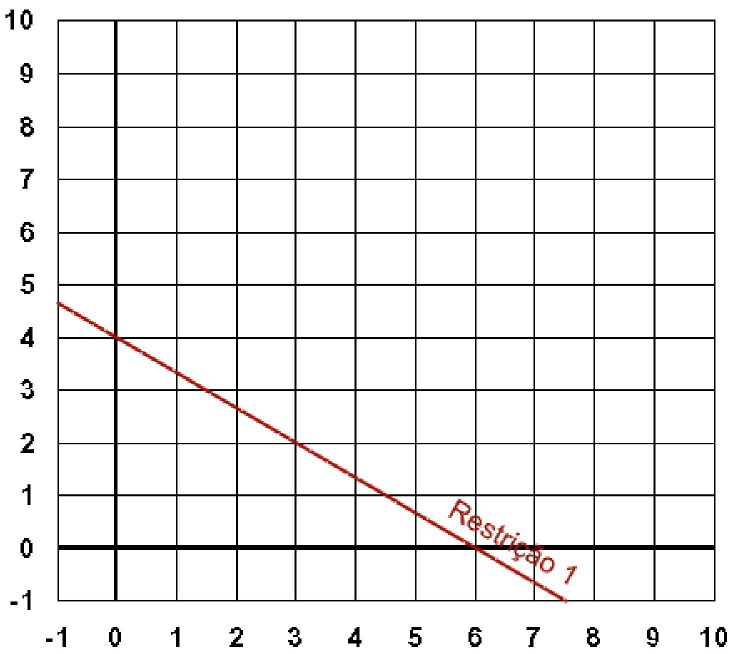
O primeiro passo é desenhar um plano cartesiano, onde iremos traçar, uma a uma, as áreas representadas pelas inequações:



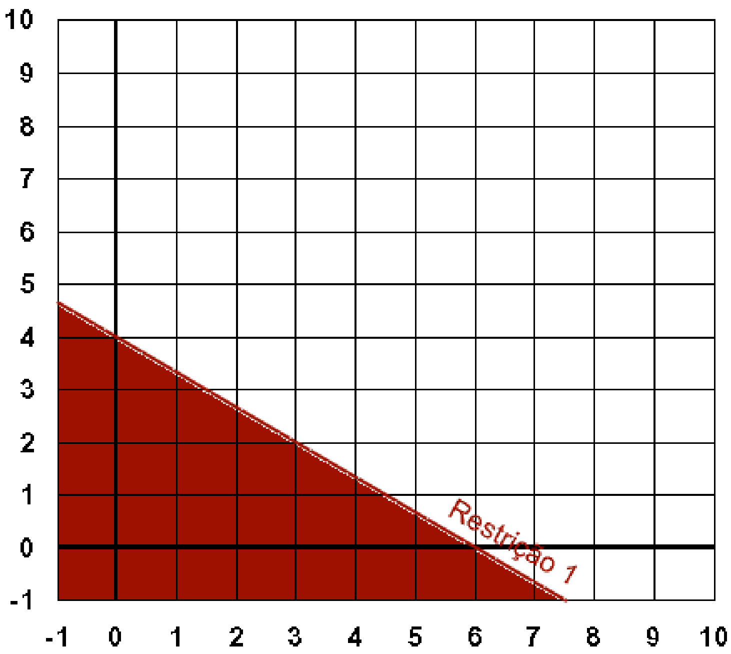
Agora, tracemos a reta equivalente à primeira restrição, $4 * x_A + 6 * x_B \leqslant 24$, que é a reta $4 * x_A + 6 * x_B = 24$. A tabela para esta construção é:

X (x_A)	**Y (x_B)**
0	6
4	0

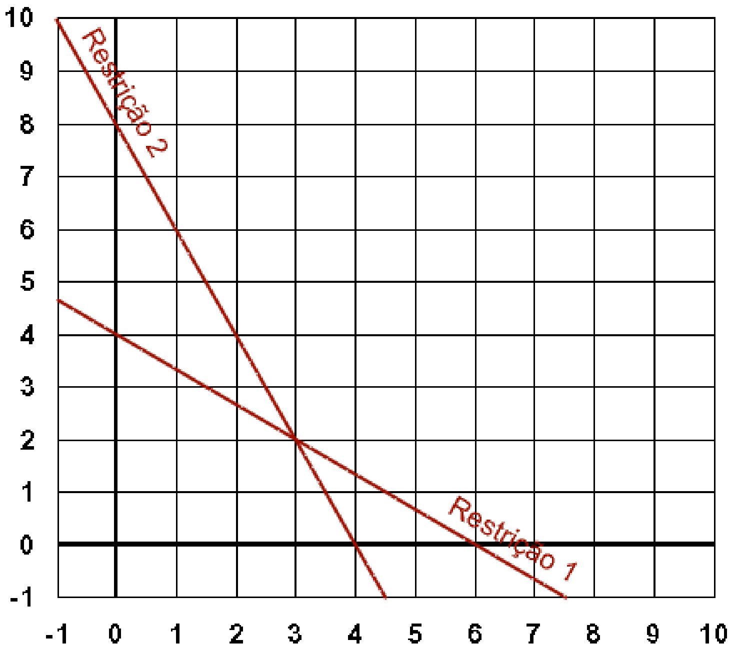
Traçando esta reta no gráfico, temos:



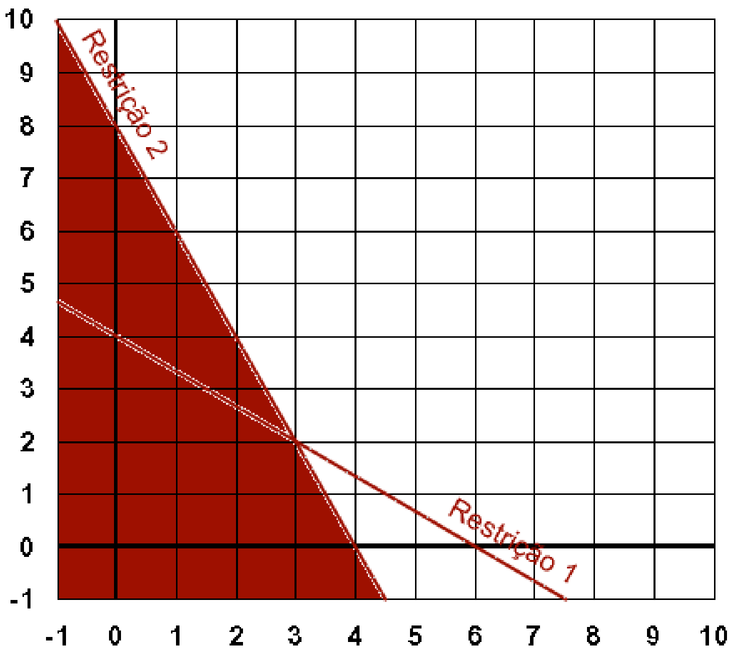
Entretanto, não se trata de uma equação (que dá a reta) e sim de uma **inequação**, que delimita um plano. A representação para a inequação $4 * x_A + 6 * x_B \leqslant 24$ é, então:



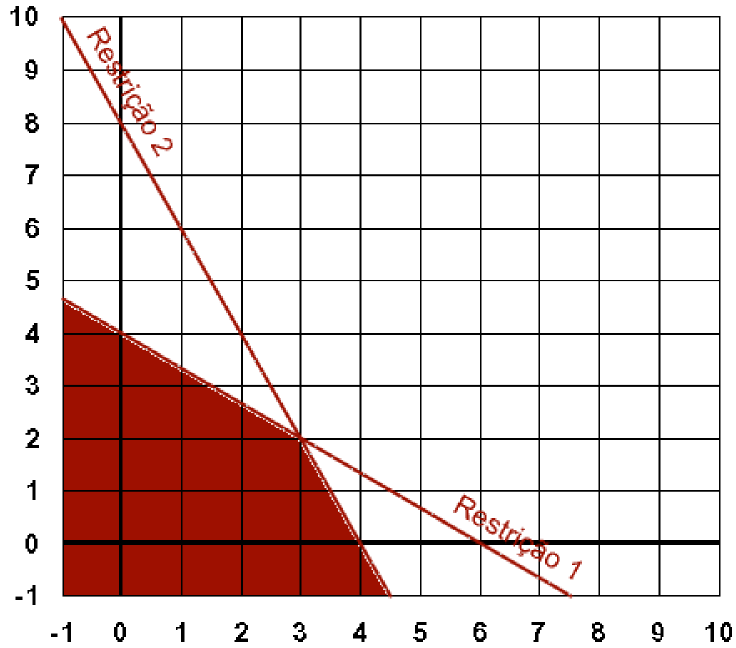
Agora, vamos traçar a equação relacionada à segunda restrição, $4 * x_A + 2 * x_B \leqslant 16$, no mesmo gráfico que foi traçada a reta anterior. Para facilitar a visualização, foi eliminado temporariamente o preenchimento da área na próxima figura.



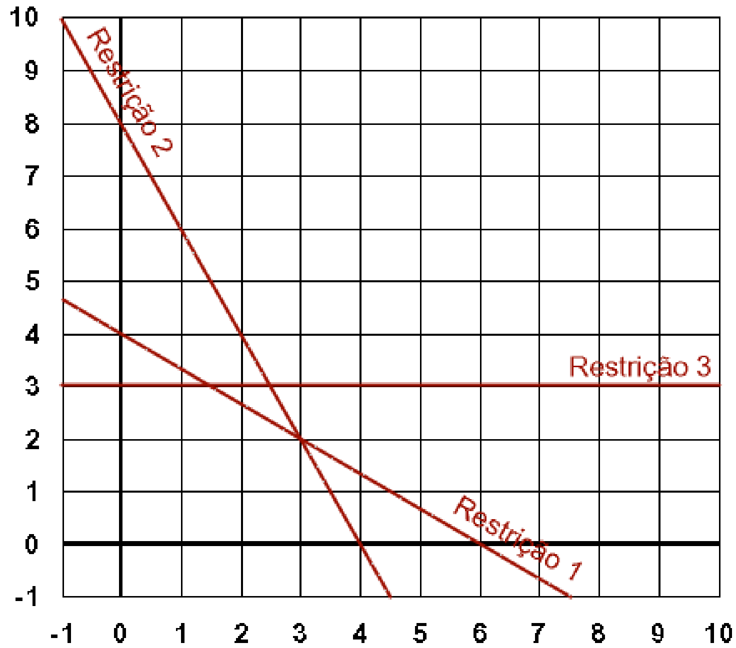
Entretanto, mais uma vez não se trata de uma equação e sim de uma **inequação**, que delimita um plano. A representação para a inequação $4 * x_A + 2 * x_B \leqslant 16$ é, então:



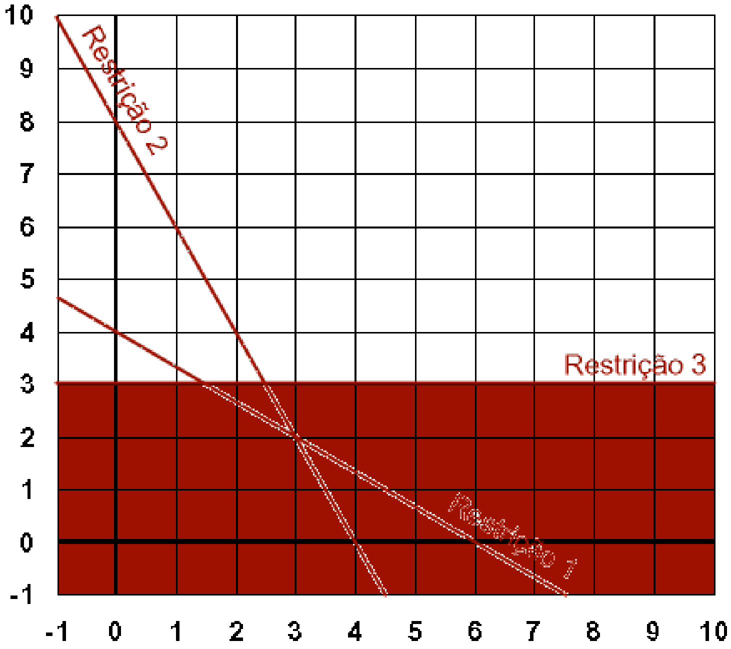
Entretanto, note que uma parte da área permitida pela Restrição 2 não é permitira pela Restrição 1 (área acima da reta da restrição 1 e abaixo da restrição 2). Assim, a área permissível pelas duas restrições será representada na próxima figura, mostrando como acrescentar a restrição 2 **reduziu** o espaço de soluções:



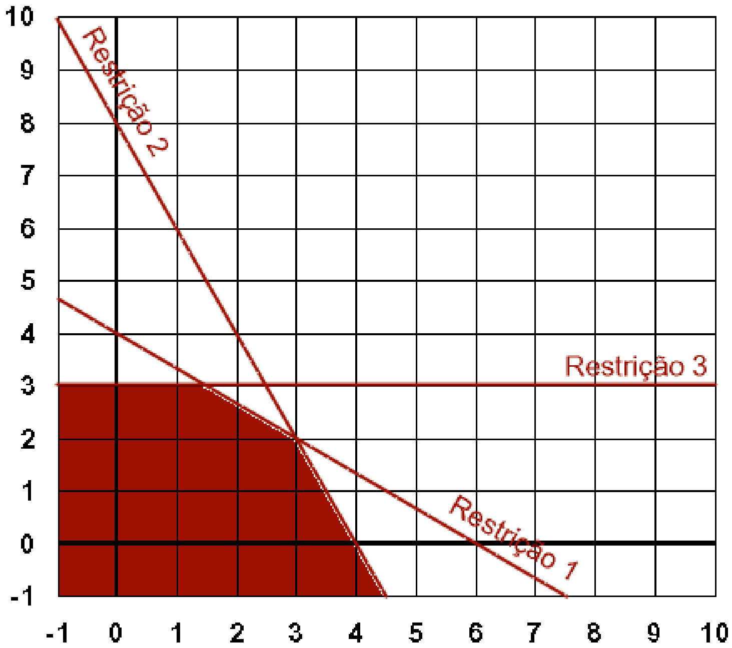
A terceira restrição é mais simples: $x_B \leqslant 3$. A equação associada é $x = 3$, que será representada na próxima figura:



E na próxima figura, será marcada a área permissível apenas pela restrição 3, ignorando as outras duas restrições:

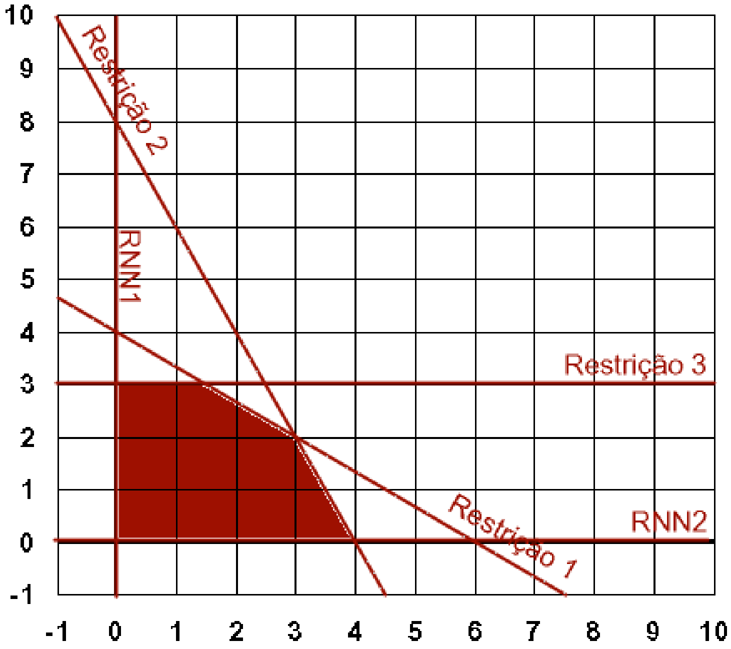


E agora, marcando apenas a área permissível ao mesmo tempo por todas as três restrições:



Entretanto, é possível observar que a área está se estendendo por regiões negativas tanto no eixo X (x_A) quanto no eixo Y (x_B). Isto está incorreto, já que temos duas restrições adicionais: as restrições de não negatividade: $x_A \geqslant 0$; $x_B \geqslant 0$.

No próximo gráfico estão traçadas as restrições de não-negatividade e a área final já está delimitada:



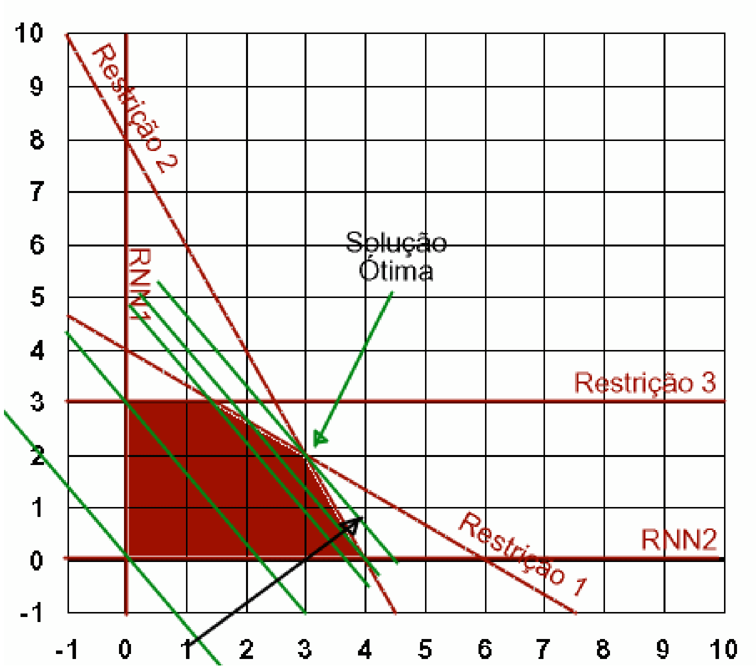
Bem, assim temos delimitada toda a região de soluções possíveis. Mas qual delas está correta? Bem, neste momento convém apresentar uma importante propriedade:

"A solução ótima de um problema está em um dos pontos extremos da região permissível"

Em outras palavras, a solução ótima está em um dos "cantos" do espaço de soluções possíveis. A tabela abaixo apresenta os diversos pontos extremos do gráfico acima. Sua determinação é feita através das equações das retas que se cruzam para formar cada um deles:

Ponto Extremo	x_A	x_B	Função Objetivo: $80*x_A + 60*x_B$
1	0	0	0
2	4	0	320
3	3	2	360
4	1.5	3	300
5	0	3	180

Pelos valores calculados para a função objetivo, é possível ver que o ponto extremo 3 ($x_A = 3$ e $x_B = 2$) tem a melhor solução pois maximiza o lucro. Entretanto, esta não é a únic aforma de verificar a melhor solução. Pelo próprio gráfico é possível avaliar qual é o ponto extremo da melhor solução, se plotarmos a família de retas representada pela função objetivo. Por exemplo: **$80*x_A + 60*x_B = 0$, $80*x_A + 60*x_B = 180$**... e assim por diante, até encontrarmos o último ponto da figura que a reta da função objetivo toca, como indicado na figura a seguir.



2. Análises Possíveis

Através da representação gráfica dos problemas, podemos verificar porque algumas soluções indesejadas podem ocorrer. A seguir analisaremos algumas destas situações e também veremos um pouco sobre o que é uma "análise de sensibilidade".

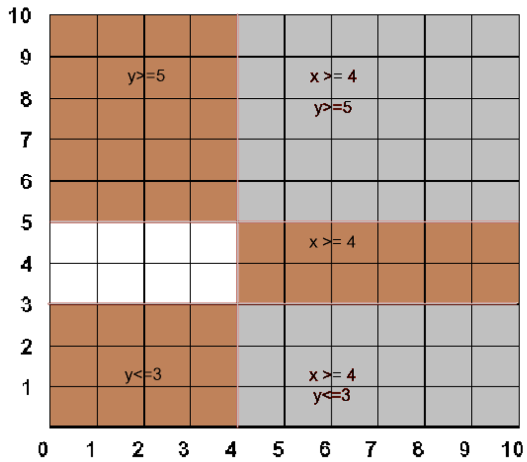
2.1. Restrições Incompatíveis

Em alguns problemas podemos não ter solução alguma (solução impossível), fato este causado por incompatibilidade entre as restrições. Por exemplo:

[MAX] 1*x + 2*y

Sujeito a: $x \geq 4$
$y \geq 5$
$y \leq 3$

Certamente temos um problema aqui: y não pode ser, ao mesmo tempo, maior ou igual a cinco e menor ou igual a três. Entretanto, a incompatibilidade nem sempre é tão óbvia. É possível observar no gráfico a seguir como esta incompatibilidade de fato existe. Estão pintadas as áreas permissíveis a partir de cada restrição: note como não há nenhuma área que atenda simultaneamente às três restrições.



Como não há nenhum ponto extremo que obedeça às três restrições, este problema é de solução impossível.

2.2. Solução sem Fronteiras

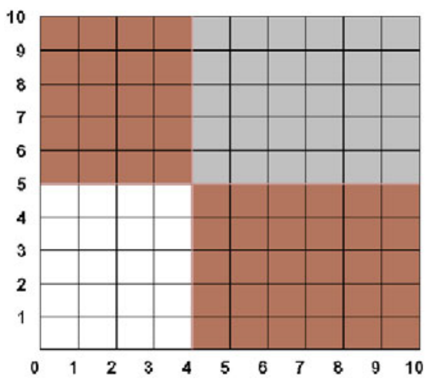
Se temos um problema como o apresentado a seguir:

[MAX] 1*x + 2*y

Sujeito a: $x \geqslant 4$
$y \geqslant 5$

Temos um problema chamado "Solução sem Fronteiras". O que isto significa? Que não há um máximo definido para a função objetivo, ela pode ser tão grande quanto se deseje, já que seu valor aumenta com o crescimento de X e Y e ambos têm seu máximo ilimitado.

No gráfico a seguir, esta situação é representada, valendo a pena notar que as regiões sombreadas não se limitam à área apresentada, estendendo-se infinitamente para cima e para a direita. Por esta razão, não é possível identificar o ponto extremo de máximo, que se daria justamente quando X e Y tiverem o valor infinito. Note que este é um problema que fere um dos princípios da programação linear, que não serve, obviamente, para resolver problemas ilimitados como o apresentado pelo modelo matemático acima representado.



2.3. Restrições Redundantes

Se temos um problema como o apresentado a seguir:

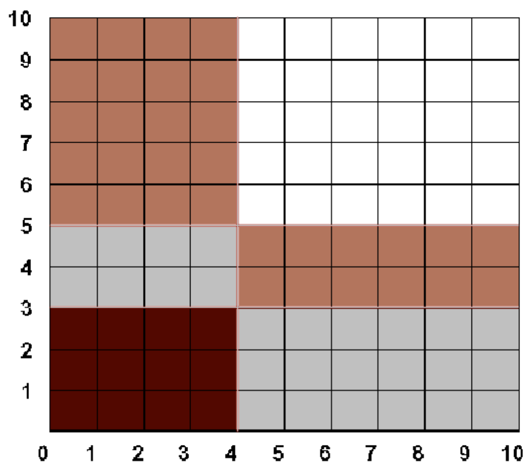
[MAX] 1*x + 2*y

Sujeito a: $x \leqslant 4$
$y \leqslant 5$
$y \leqslant 3$

Temos o que chamaos de restrições redundantes, no caso, para a variável Y. A restrição $y \leqslant 3$ claramente já "inclui" a restrição $y \leqslant 5$, uma vez que se $y \leqslant 3$ for respeitada,$y \leqslant 5$ também sempre o será.

A representação deste problema em gráfico pode ser visualizada na próxima figura. Convém observar, entretanto, que este não é um "problema" em si, já que não atrapalha a solução do problema de programação linear.

É interessante, porém, eliminar as restrições redundantes que se identifique, a fim de simplificar o problema matemático a ser resolvido.



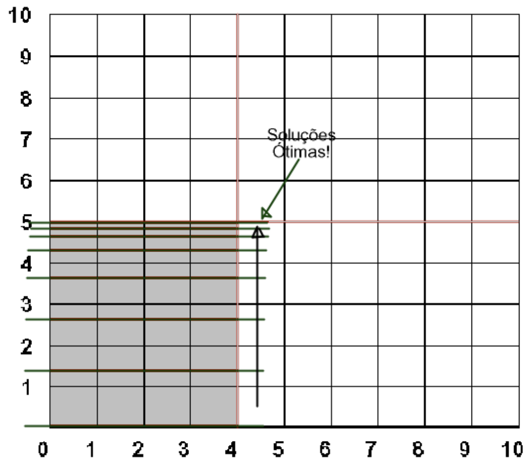
2.4. Soluções Alternativas

Em alguns problemas é impossível determinar um único ponto de extremo que seja a soluçao ótima. isso ocorre na situação em que a reta que representa a função objetivo é **paralela** à uma das restrições. Por exemplo:

[MAX] 1*y

Sujeito a: $x \leqslant 4$
$y \leqslant 5$

A representação gráfica a seguir mostra que, quando a reta da função objetivo toca o extremo da área de soluções possíveis, um segmento de reta inteiro fica marcado (e não apenas um ponto). Por esta razão, existem infinitas soluções ótimas para este problema, sendo qualquer uma delas aceitável de acordo com o modelo matemático apresentado.



2.5. Análise de Sensibilidade

Normalmente, quando obtemos uma solução ótima para um problema, temos uma preocupação adicional: a solução encontrada será implementada na prática. Bem, mas por que razão isso é um problema?

Bem, ocorre que alguns parâmetros que especificamos no modelo podem não ser muito precisos. Por exemplo, no exercício apresentado, o número de horas disponíveis da Máquina 1 foi apresentado como sendo 24 horas. Mas o que ocorreria se, por algum problema, ficassem disponíveis apenas 23 horas da máquina M1? Isso mudaria muito a solução? Qual seria esta mudança? Teríamos que reprogramar as operações para maximizar lucro?

Da mesma forma, uma empresa concorrente poderia colocar no mercado um produto similar ao nosso, fazendo com que o preço de um de nossos produtos caísse. Neste caso, será que a operação da nossa fábrica precisa ser reajustada? Até que valor seria possível abaixar o preço de nosso produto sem a necessidade de alterar a programação de produção?

Estas análises podem ser feitas também com análises gráficas. Entretanto, deixaremos este tipo de análise mais para o futuro, quando a modelagem e o processo de solução de problemas já for melhor compreendido por todos.

3. Bibliografia

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

Notas da Aula 06: Modelagem na Forma Padrão para o Simplex
Prof. Daniel Caetano

Objetivo: Apresentar as modificações na modelagem matemática necessárias para a especificação de um modelo na forma padrão.

Bibliografia:

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

Introdução

A solução gráfica para problemas de programação linear, vista nas aulas anteriores, é bastante elucidativa. Entretanto, é uma forma desajeitada de resolver problemas, além de se tornar complexa ou impossível de ser aplicada para problemas com mais de duas variáveis de decisão.

Uma forma alternativa é o uso do Método Simplex para a resolução dos problemas de programação linear. O Método Simplex é um método sistemático, baseado em um *tableau*, onde indicamos todos os dados do problema e, realizando algumas operações, encontramos a solução ótima.

Embora o Simplex use cálculos bastante simples, sua seqüência é bastante tediosa. Esta característica faz com que seja interessante criar programas para resolver problemas pelo Método Simplex. Entretanto, é necessário aprender todos os passos do Simplex, verificando suas qualidades e quando surgem problemas, possibilitando uma correta interpretação dos resultados quando a solução por encontrada por meio de um software.

Porém, apesar de termos visto, nas aulas anteriores, a maneira de converter um problema real em um modelo matemático, tais modelos ainda não estão corrretamente preparados para sua resolução pelo Simplex. Para que o modelo mamtemático se adapte às necessidades do Simplex, ainda precisamos fazer algumas modificações em sua forma, sem alterar o seu significado matemático. A forma final, pronta para o Simplex, é chamada de Forma Padrão.

1. Requisitos do Simplex para a Modelagem

O primeiro problema que teremos ao pensar em usar Simplex é que o Método Simplex faz algumas exigências com relação aos dados; em especial, para que o método seja iniciado, precisamos encontrar uma solução inicial que seja possível. A forma como encontraremos essa solução implicará em algumas mudanças nas restrições.

Adicionalmente, o Método Simplex exige que não tenhamos qualquer tipo de inequações como restrições (com $\leqslant$ ou $\geqslant$), apenas equações como restrições (com sinal =). Apesar de isso parecer restritivo, veremos truques que permitem fazer com que transformemos as inequações em equações sem mudarmos seu significado matemático. Veremos neste aula como lidar com restrições do tipo $\leqslant$ e em aulas posteriores analisaremos o caso das restrições $\leqslant$, além de vermos que mesmo as restrições de igualdade precisarão de um pequeno ajuste.

Além disso, todas as variáveis do problema precisam estar presentes em todas as restrições e na função objetivo. Como já vimos, isso nem sempre ocorre "naturalmente", já que freqüentemente algumas restrições envolvem apenas uma ou duas variáveis. Neste caso, também a solução será simples e nada será modificado no significado matemático das restrições.

Uma outra restrição é que "nenhum lado direito" as restrições seja menor que zero. O que é o "lado direito" de uma restrição? Simples: se movermos todas as variáveis para o lado esquerdo, e todos os números constantes para o lado direito, em geral termos um número (uma constante) solitária do lado direito. Isto é que chamamos de "lado direito".

$$x_A + x_B + 7 = y_A + y_B$$

$$||$$

$$x_A + x_B - y_A - y_B = -7$$

Neste caso, "-7" é o lado direito... e, neste caso, ele está melhor que zero. O Simplex não aceita uma restrição nesta forma e, por isso, veremos uma maneira de corrigir este problema, sem alterar o significado matemático da restrição.

Finalmente, algo que não faz parte da "forma padrão", mas que também é uma modificação dos dados do modelo (sem alterar seu significado matemático), é a transformação de uma "minimização" em "maximização". Isso é útil para que possamos aprender apenas o Método Simplex para maximizar e, sempre que encontrarmos um problema de "minimização" o transformaremos no problema de "maximização" equivalente.

2. A Modelagem

Voltando ao Problema extraído de MOREIRA (2006):

Uma fábrica produz dois produtos, A e B. Cada um deve ser processado por duas máquinas, M_1 e M_2. Devido à programação de outros produtos que também usam estas máquinas, estão disponíveis para os produtos A e B apenas 24 horas da máquina M_1 e 16 horas da máquina M_2.

Para produzir uma unidade do produto A, são necessárias 4 horas em cada uma das máquinas e para produzir uma unidade do produto B, são necessárias 6 horas em M_1 e 2 horas em M_2. Cada unidade de A vendida gera um lucro de R$ 80,00 e cada unidade de B vendida gera um lucro de R$ 60,00.

Existe uma previsão de demanda máxima de 3 unidades para B, mas nenhuma restrição de demanda para A. Deseja-se saber: quanto produzir de cada produto para maximizar o lucro?

Modelo Final

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Sujeito a:
$4 * x_A + 6 * x_B \leqslant 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B \leqslant 16$ <= Restrição do número de horas de M_2
$1 * x_B \leqslant 3$ <= Restrição de Demanda para B

as variáveis de decisão são x_A e x_B e aqui não iremos mais representar as restrições de não-negatividade, porque elas são implícitas no Simplex.

2.1. Resolvendo o Problema das Inequações

O que devemos fazer para eliminar as desigualdades, neste caso? Uma coisa é certa: não podemos fazer isso:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Sujeito a (RESTRIÇÕES INCORRETAS!):
$4 * x_A + 6 * x_B = 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B = 16$ <= Restrição do número de horas de M_2
$1 * x_B = 3$ <= Restrição de Demanda para B

Simplesmente porque, se isso fosse feito, estaríamos **modificando** o modelo de uma forma que **modifica o problema** a ser resolvido. Em muitos casos, se fizéssemos o que foi indicado acima, o problema seria até mesmo insolúvel. Suponhamos, por exemplo, que a solução ótima do problema original fosse x_A=2 e x_B=2. Neste caso, pela restrição de hora de M_1, teríamos:

$$4 * x_A + 6 * x_B = 24 \;\Rightarrow\; 4*2 + 6*2 = 24 \;\Rightarrow\; 8+12 = 24 \;\Rightarrow\; 20 = 24 \;!?!$$

Não! **20 ≠ 24!** Como resolver a questão, então? A solução é usar um pequeno truque: acrescentar uma variável a mais em cada restrição:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} = 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B + 1 * x_{S2} = 16$ <= Restrição do número de horas de M_2
$1 * x_B + 1 * x_{S3} = 3$ <= Restrição de Demanda para B

Observemos agora que, mesmo com o sinal de igual (=), não estamos mais restringindo as soluções. Por exemplo, suponhamos novamente que a solução ótima do problema original fosse x_A=2 e x_B=2.

Agora, pela restrição de horas de M_1 teremos:

$4 * x_A + 6 * x_B + 1 * x_{S1} = 24$ <= Restrição do número de horas de M_1
$4 * 2 + 6 * 2 + 1 * x_{S1} = 24$
$8 + 12 + x_{S1} = 24$
$20 + x_{S1} = 24$
$x_{S1} = 24 - 20$
$x_{S1} = 4$

Ou seja, na nossa solução suposta ótima x_A=2 e x_B=2 implicou em um x_{S1} = 4, mas a solução voltou a ser possível, ainda que tenhamos um sinal de igual. Mas o que representa esta variável x_{S1} ? Ora, ela representa o número de horas que **sobraram** da máquina M_1. x_{S1} é chamada uma **variável de folga**. Folga, em inglês, é Slack, daí o índice S na variável.

Repare, também, como as variáveis de folga também precisam obedecer às restrições de não-negatividade, já que não faz sentido sobrar "-10 horas" da máquina M_1, por exemplo. Repare que o raciocínio todo acima pode ser realizado para as outras restrições, envolvendo x_{S2} e x_{S3}.

2.2. Resolvendo o Problema das Variáveis Faltantes

Este problema é bem mais fácil de ser resolvido: basta pegar o modelo matemático e acrescentar em todas as variáveis faltantes em cada equação com coeficiente zero. Assim, o modelo que era:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} = 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B + 1 * x_{S2} = 16$ <= Restrição do número de horas de M_2
$1 * x_B + 1 * x_{S3} = 3$ <= Restrição de Demanda para B

Se torna:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B + 0 * x_{S1} + 0 * x_{S2} + 0 * x_{S3}$

Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} + 0 * x_{S2} + 0 * x_{S3} = 24$ <= Restrição M_1
$4 * x_A + 2 * x_B + 0 * x_{S1} + 1 * x_{S2} + 0 * x_{S3} = 16$ <= Restrição M_2
$0 * x_A + 1 * x_B + 0 * x_{S1} + 0 * x_{S2} + 1 * x_{S3} = 3$ <= Restrição B

2.3. Resolvendo o Problema de Lado Direito Negativo

Apesar de não ser o caso deste problema, há situações em que o lado direito de uma restrição aparece com um valor negativo. Como foi, por exemplo, apresentado anteriormente:

$$x_A + x_B - y_A - y_B = -7$$

O lado direito tem o valor negativo **-7**, e isso não é aceitável. Neste caso, basta multiplicar a restrição toda por **-1**, invertendo todos os sinais e tornando o lado direito positivo:

$$[\ x_A + x_B - y_A - y_B = -7\ ] * (-1) \quad \Rightarrow \quad - x_A - x_B + y_A + y_B = 7$$

2.4. Transformando Problemas de Minimização em Problemas de Maximização

Apesar de existir um procedimento de cálculo Simplex para minimizar uma função objetivo, isso implicaria o aprendizado de dois procedimentos: o de maximização e o de minimização. Para evitar este inconveniente, há um "truque" muito simples que transforma problemas de minimização em maximização, que é a simples multiplicação de toda a função objetivo por **-1**.

Desta forma:

$$\{ \text{[MIN]}\ x_A + x_B - y_A - y_B \} * (-1) \Rightarrow \text{[MAX]}\ - x_A - x_B + y_A + y_B$$

Sendo ambos os problemas matematicamente equivalentes (os valores calculados para x_A, x_B, y_A e y_B serão iguais para ambas as funções objetivo).

3. Soluções Básicas e Não-Básicas: Encontrando uma Solução Inicial

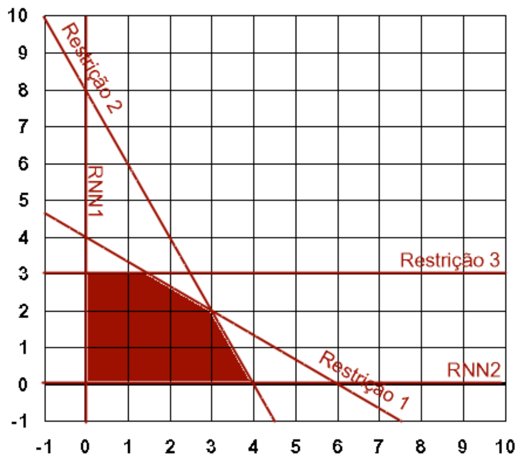
Retomemos o modelo matemático apresentado anteriormente:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B + 0 * x_{S1} + 0 * x_{S2} + 0 * x_{S3}$
Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} + 0 * x_{S2} + 0 * x_{S3} = 24$ <= Restrição M_1
$4 * x_A + 2 * x_B + 0 * x_{S1} + 1 * x_{S2} + 0 * x_{S3} = 16$ <= Restrição M_2
$0 * x_A + 1 * x_B + 0 * x_{S1} + 0 * x_{S2} + 1 * x_{S3} = 3$ <= Restrição B

Observando esta formaulação, é possível verificar que temos 5 incógnitas e apenas 3 equações. Como temos mais incógnitas do que equações, o problema é indeterminado, isto é, não é possível determinar uma solução única para ele. Entretanto, **se escolhermos duas variáveis quaisquer e fixarmos seus valores**, teremos então 3 equações e 3 incógnitas, tornando-se um sistema determinado, possibilitando o cálculo das variáveis restantes.

Por facilidade nas contas, fixaremos os valores das duas variáveis escolhidas como sendo 0 (zero). As variáveis escolhidas arbitrariamente para terem seu valor definido como zero formam o que chamamos de "solução não-básica" e as variáveis restantes, cujos valores foram calculados, formam a chamada "solução básica". Note que a solução básica definida desta forma pode não ser viável, isto é, desrespeitar as restrições previamente impostas.

Na aula anterior vimos a área de soluções possíveis para este problema:



Ainda na aula anterior, vimos que a solução ótima estava sempre num ponto extremo. Assim, para verificar um resultado interessante, vamos analisar os valores das variáveis nos pontos extremos, sendo o ponto A o ponto (0,0) e os pontos B, C, D e E são seqüenciais, no sentido horário:

Ponto	X_A	X_B	X_{S1}	X_{S2}	X_{S3}
A	**0**	**0**	24	16	3
B	**0**	3	6	10	**0**
C	1,5	3	**0**	4	**0**
D	3	2	**0**	**0**	1
E	4	**0**	8	**0**	3

Observe que em todos os pontos extremos **há sempre duas variáveis iguais a zero**! E os pontos extremos também representam **todas as soluções básicas possíveis**.

Assim, pode-se dizer, de forma genérica, que sempre que tivermos um problema de programação linear com **n** incógnitas e **m** equações, em todos os extremos da região de soluções possíveis teremos **(n-m)** incógnitas com valor igual a zero.

3.1. A Solução Inicial

Como futuramente precisaremos encontrar sempre uma solução inicial, convém analisarmos como encontrar tal solução inicial.

O primeiro aspecto é que não convém que a solução inicial seja complexa, o que a tornaria de difícil cálculo. É desejável que a solução inicial seja a mais simples possível. Adicionalmente, se esta solução inicial puder ser similar para todos os problemas, tanto melhor.

Curiosamente, existe uma solução inicial que se encaixa em todas estas características. E é uma solução tão simples e comum que a chamamos de "solução trivial": se arbitrarmos que as variáveis de decisão originais valem ZERO, tornando-as variáveis **não-básicas** (fora da solução), teremos uma solução de cálculo imediato para o problema. Vejamos:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B + 0 * x_{S1} + 0 * x_{S2} + 0 * x_{S3}$

Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} + 0 * x_{S2} + 0 * x_{S3} = 24$ <= Restrição M_1
$4 * x_A + 2 * x_B + 0 * x_{S1} + 1 * x_{S2} + 0 * x_{S3} = 16$ <= Restrição M_2
$0 * x_A + 1 * x_B + 0 * x_{S1} + 0 * x_{S2} + 1 * x_{S3} = 3$ <= Restrição B

As variáveis de decisão são x_A e x_B. Tornemo-as iguais a zero: $x_A = x_B = 0$. Substituindo os valores de x_A e x_B nas restrições, teremos:

$4 * 0 + 6 * 0 + 1 * x_{S1} + 0 * x_{S2} + 0 * x_{S3} = 24$ <= Restrição M_1
$4 * 0 + 2 * 0 + 0 * x_{S1} + 1 * x_{S2} + 0 * x_{S3} = 16$ <= Restrição M_2
$0 * 0 + 1 * 0 + 0 * x_{S1} + 0 * x_{S2} + 1 * x_{S3} = 3$ <= Restrição B

Resolvendo os cálculos, isso pode ser reescrito da seguinte forma:

$0 + 0 + x_{S1} + 0 + 0 = 24$ <= Restrição M_1
$0 + 0 + 0 + x_{S2} + 0 = 16$ <= Restrição M_2
$0 + 0 + 0 + 0 + x_{S3} = 3$ <= Restrição B

Ora, limpando este monte de zeros, teremos:

$x_{S1} = 24$ <= Restrição M_1
$x_{S2} = 16$ <= Restrição M_2
$x_{S3} = 3$ <= Restrição B

E, como havíamos arbitrado, $x_A = x_B = 0$. Estes cinco valores compõem a solução inicial (que, normalmente, não é uma solução ótima).

4. Exercício L2

Sobre os problemas da L1 (disponíveis em http:/www.caetano.eng.br/aulas/fb/IPO/ , no arquivo das aulas 2 e 3):

1. Coloque os três problemas modelados na L1 na forma padrão.
2. Encontre as soluções iniciais, variáveis básicas e não básicas para cada um deles.

Entrega na próximas aula. Vale **nota**.

6. Bibliografia

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

Notas da Aula 07: Modelagem na Forma Padrão para o Simplex
Prof. Daniel Caetano

Objetivo: Resolver a segunda lista de exercícios, sedimentando o processo de modelagem matemática.

Bibliografia:

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

Resolução do Exercício L2

1. Coloque os três problemas modelados na L1 na forma padrão.

Problema 1

Modelagem original do primeiro problema:

Função Objetivo:
[MAX] $150 * x_A + 72 * x_B$

Restrições:
$60 * x_A + 25 * x_B \leqslant 300$ <= Restrição de volume
$1 * x_A + 8 * x_B \leqslant 50$ <= Restrição de peso

a) Eliminando as desigualdades das Restrições

Como temos duas restrições do tipo menor ou igual, podemos transformá-las em igualdades acrescentando variáveis de sobra em cada uma das restrições:

$60 * x_A + 25 * x_B \leqslant 300 \quad \Rightarrow \quad 60 * x_A + 25 * x_B + 1 * x_{S1} = 300$
$1 * x_A + 8 * x_B \leqslant 50 \quad \Rightarrow \quad 1 * x_A + 8 * x_B + 1 * x_{S2} = 50$

Onde x_{S1} significa o volume não utilizado do navio e x_{S2} o peso livre do navio.

b) Fazendo com que todas as equações tenham todas as variáveis

Basta acrescentar em todas as equações as variáveis faltantes, com coeficiente zero:

[MAX]$150 * x_A +$	$72 * x_B +$	$0 * x_{S1} +$	$0 * x_{S2}$	
$60 * x_A +$	$25 * x_B +$	$1 * x_{S1}$	$0 * x_{S2}$	$= 300$
$1 * x_A +$	$8 * x_B +$	$0* x_{S1}$	$1* x_{S2}$	$= 50$

Problema 2

Modelagem original do segundo problema:

Função Objetivo:
[MAX] $100 * x_A + 500 * x_B + 350 * x_C + 650 * x_D$

Restrições:

$150 * x_A + 100 * x_B + 500 * x_C + 350 * x_D \leqslant 4000$	<= Restrição de Memória
$1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D \leqslant 72$	<= Restrição de Horas
$x_A \leqslant 10$	<= Restrição de Unidades A
$x_B \leqslant 25$	<= Restrição de Unidades B
$x_C \leqslant 3$	<= Restrição de Unidades C
$x_D \leqslant 7$	<= Restrição de Unidades D
$x_A \geqslant 0; x_B \geqslant 0; x_C \geqslant 0 ; x_D \geqslant 0$	<= Não-negatividade

a) Eliminando as desigualdades das Restrições

Como temos duas restrições do tipo menor ou igual, podemos transformá-las em igualdades acrescentando variáveis de sobra em cada uma das restrições:

$150*x_A+100*x_B+500*x_C+350*x_D \leqslant 4000$	=>	$150*x_A+100*x_B+500*x_C+350*x_D+x_{S1} = 4000$
$1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D \leqslant 72$	=>	$1 * x_A + 7 * x_B + 4 * x_C + 10 * x_D + x_{S2} = 72$
$1* x_A \leqslant 10$	=>	$1 * x_A + x_{S3} = 10$
$1 * x_B \leqslant 25$	=>	$1 * x_B + x_{S4} = 25$
$1 * x_C \leqslant 3$	=>	$1 * x_C + x_{S5} = 3$
$1 * x_D \leqslant 7$	=>	$1 * x_D + x_{S6} = 7$

Onde x_A significa o número de pacotes A a serem processados, x_B significa o número de pacotes B a serem processados, x_C significa o número de pacotes C a serem processados, x_D significa o número de pacotes D a serem processados.

b) Fazendo com que todas as equações tenham todas as variáveis

Basta acrescentar em todas as equações as variáveis faltantes, com coeficiente zero:

$$
\begin{array}{ll}
\text{[MAX]} & 100*x_A + 500*x_B + 350*x_C + 650*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} \\
 & 150*x_A + 100*x_B + 500*x_C + 350*x_D + 1*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} = 4000 \\
 & 1*x_A + 7*x_B + 4*x_C + 10*x_D + 0*x_{S1} + 1*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} = 72 \\
 & 1*x_A + 0*x_B + 0*x_C + 0*x_D + 0*x_{S1} + 0*x_{S2} + 1*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} = 10 \\
 & 0*x_A + 1*x_B + 0*x_C + 0*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 1*x_{S4} + 0*x_{S5} + 0*x_{S6} = 25 \\
 & 0*x_A + 0*x_B + 1*x_C + 0*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 1*x_{S5} + 0*x_{S6} = 3 \\
 & 0*x_A + 0*x_B + 0*x_C + 1*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 1*x_{S6} = 7
\end{array}
$$

Problema 3

Modelagem original do terceiro problema:

Função Objetivo:
[MAX] $120 * x_{EM} + 70 * x_C$

Restrições:
$2 * x_{EM} + 1 * x_C \leqslant 180$ <= Restrição de horas-máquina
$2 * x_{EM} + 4 * x_C \leqslant 240$ <= Restrição de homens-hora

<u>a) Eliminando as desigualdades das Restrições</u>

Como temos duas restrições do tipo menor ou igual, podemos transformá-las em igualdades acrescentando variáveis de sobra em cada uma das restrições:

$2 * x_{EM} + 1 * x_C \leqslant 180 \Rightarrow 2 * x_{EM} + 1 * x_C + 1 * x_{S1} = 180$
$2 * x_{EM} + 4 * x_C \leqslant 240 \Rightarrow 2 * x_{EM} + 4 * x_C + 1 * x_{S2} = 240$

Onde x_{S1} significa o número de horas/máquina não usado e x_{S2} o número de homens/hora não usados.

<u>b) Fazendo com que todas as equações tenham todas as variáveis</u>

Basta acrescentar em todas as equações as variáveis faltantes, com coeficiente zero:

[MAX]$120 * x_{EM}$ +	$70 * x_C$ +	$0 * x_{S1}$ +	$0 * x_{S2}$	
$2 * x_{EM}$ +	$1 * x_C$ +	$1 * x_{S1}$	$0 * x_{S2}$	= 180
$2 * x_{EM}$ +	$4 * x_C$ +	$0 * x_{S1}$	$1 * x_{S2}$	= 240

<u>2. Encontre as soluções iniciais, variáveis básicas e não básicas para cada um.</u>

Problema 1

[MAX]$150 * x_A$ +	$72 * x_B$ +	$0 * x_{S1}$ +	$0 * x_{S2}$	
$60 * x_A$ +	$25 * x_B$ +	$1 * x_{S1}$	$0 * x_{S2}$	= 300
$1 * x_A$ +	$8 * x_B$ +	$0 * x_{S1}$	$1 * x_{S2}$	= 50

Solução inicial: $x_A = x_B = 0$; $x_{S1} = 300$; $x_{S2} = 50$
Variáveis não-básicas: x_A e x_B
Variáveis básicas: x_{S1} e x_{S2}

Problema 2

$$
\begin{array}{ll}
[MAX] & 100*x_A + 500*x_B + 350*x_C + 650*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} \\
& 150*x_A + 100*x_B + 500*x_C + 350*x_D + 1*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} = 4000 \\
& 1*x_A + 7*x_B + 4*x_C + 10*x_D + 0*x_{S1} + 1*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} = 72 \\
& 1*x_A + 0*x_B + 0*x_C + 0*x_D + 0*x_{S1} + 0*x_{S2} + 1*x_{S3} + 0*x_{S4} + 0*x_{S5} + 0*x_{S6} = 10 \\
& 0*x_A + 1*x_B + 0*x_C + 0*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 1*x_{S4} + 0*x_{S5} + 0*x_{S6} = 25 \\
& 0*x_A + 0*x_B + 1*x_C + 0*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 1*x_{S5} + 0*x_{S6} = 3 \\
& 0*x_A + 0*x_B + 0*x_C + 1*x_D + 0*x_{S1} + 0*x_{S2} + 0*x_{S3} + 0*x_{S4} + 0*x_{S5} + 1*x_{S6} = 7
\end{array}
$$

Solução inicial: $x_A = x_B = x_C = x_D = 0$;
$x_{S1} = 4000$; $x_{S2} = 72$; $x_{S3} = 10$; $x_{S4} = 25$; $x_{S5} = 3$; $x_{S6} = 7$
Variáveis não-básicas: x_A, x_B, x_C, x_D
Variáveis básicas: x_{S1}, x_{S2}, x_{S3}, x_{S4}, x_{S5}, x_{S6}

Problema 3

$$
\begin{array}{llllll}
[MAX] & 120 * x_{EM} + & 70 * x_C + & 0 * x_{S1} + & 0 * x_{S2} & \\
& 2 * x_{EM} + & 1 * x_C + & 1 * x_{S1} & 0 * x_{S2} & = 180 \\
& 2 * x_{EM} + & 4 * x_C + & 0 * x_{S1} & 1 * x_{S2} & = 240
\end{array}
$$

Solução inicial: $x_{EM} = x_C = 0$; $x_{S1} = 180$; $x_{S2} = 240$
Variáveis não-básicas: x_{EM} e x_C
Variáveis básicas: x_{S1} e x_{S2}

3. Bibliografia

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

Notas da Aula 08: Artifícios de Modelagem e Variáveis Artificiais
Prof. Daniel Caetano

Objetivo: Apresentar artifícios para a resolução de alguns problemas de modelagem.

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

Introdução

Até o presente momento foram modelados e resolvidos problemas considerados "bem comportados", isto é, que de início já possuiam praticamente todas as características para que funcionem bem com o Método Simplex (que veremos na próxima aula).

Assim como já aprendemos a lidar com um dos problemas nas aulas anteriores, em que precisamos das variáveis de folga para transformar as restrições do tipo $\leqslant$ em restrições do tipo "=", algumas outras situações podem surgir e igualmente necessitarem de um "truque" como o das variáveis de folga.

Nesta aula veremos alguns destes truques, em que situações eles são usados e a forma de aplicá-los.

1. Lado Direito das Restrições Negativo

Como vimos anteriormente, o lado direito das restrições não pode ser negativo. Vamos relembrar como corrigir este problema. Se, ao modelar, tivermos equações do seguinte tipo:

$$5X_A - 6X_B \leqslant -17$$
$$2X_A + 1X_B \geqslant -3$$
$$-4X_A - 4X_B = -16$$

Basta usar um truque simples: multiplicar todas as restrições por -1, sembrando de inverter o sinal das desigualdades:

$$[\ 5X_A - 6X_B \leqslant -17\ ]\quad *(-1) \quad \Rightarrow \quad -5X_A + 6X_B \geqslant +17$$
$$[\ 2X_A + 1X_B \geqslant -3\ ]\quad *(-1) \quad \Rightarrow \quad -2X_A - 1X_B \leqslant +3$$
$$[-4X_A - 4X_B = -16\ ]\quad *(-1) \quad \Rightarrow \quad +4X_A + 4X_B = +16$$

Lembre-se: apenas o **lado direito** não pode ser negativo. Os *coeficientes das variáveis* **podem**.

2. Restrições do Tipo $\geqslant$

Este tipo de restrição é bastante comum em problemas de minimização, embora apareça também em alguns problemas de maximização.

Assim como fomos obrigados a fazer modificações nas restrições do tipo $\leqslant$, com as variáveis de folga, no caso das restrições do tipo $\geqslant$ teremos que usar também algum tipo de variável de ajuste.

Pense, por exemplo, na restrição abaixo:

$4X_A + 10X_B \geqslant 45$

Qualquer valor de X_A e X_B que tornem a soma maior ou igual a 45 terá resolvido o problema. Por exemplo, se $X_A=4$ e $X_B=3$, temos

$4*4 + 10*3 = 16 + 30 = 46$ que é maior ou igual a 45.

Assim, podemos dizer que $X_A=4$ e $X_B=3$ é uma solução possível para o problema. Mas, quando transformamos a inequação em uma equação, o problema surge:

$4X_A + 10X_B = 45$

Esta eqüação não mais admite $X_A=4$ e $X_B=3$ como resposta (porque 46 não é igual a 45), significando que modificamos o problema original... e, por causa de nosso objetivo final - que é o de resolver o problema real - não podemos realizar mudanças na modelagem que modifiquem o problema! A solução aqui, similar ao já visto no caso da restrição do tipo $\leqslant$, é usar uma variável a mais, só que agora ela aparecerá do lado direito:

$4X_A + 10X_B \geqslant 45 \quad => \quad 4X_A + 10X_B = 45 + X_E$

Sempre poderemos escolher um valor de X_E que torne a igualdade verdadeira. Esta variável é chamada de *variável de excesso*, que pode ser encarada como uma variável de folga negativa, ou uma "variável de falta":

$4X_A + 10X_B = 45 + X_E \quad => \quad 4X_A + 10X_B - X_E = 45$

Infelizmente isso não resolve nosso problema, novamente por causa da tal solução inicial em que zeramos as variáveis de decisão (X_A e X_B, no caso) para encontrar uma solução possível. Vejamos o que ocorre neste caso, ao fazer $X_A = X_B = 0$:

$4X_A + 10X_B - X_E = 45 => 4*0 + 10*0 - X_E = 45 => -X_E = 45 => X_E = -45$

Mas X_E não pode ter um valor negativo! Como resolver este problema?

Neste caso o truque é inserir **outra** variável e dizer que $X_A = X_B = X_E = 0$:

$4X_A + 10X_B - X_E + A_1 = 45 \Rightarrow 0 + A_1 = 45 \Rightarrow A_1 = 45$ OK!

Entretanto, esta variável a ser inserida não tem qualquer tipo de significado físico, ela é um artifício matemático para que possamos encontrar uma solução simples. Por esta razão, este tipo de variável é chamada de *variável artificial*. Estas variáveis terão uma implicação na resolução do Simplex que veremos mais adiante.

3. Restrições do Tipo =

Aparentemente, como queremos uma restrição que seja uma igualdade, ao nos depararmos com uma restrição deste tipo temos a sensação que nada precisamos fazer. Entretanto, isso não é verdade.

Você é capaz de se lembrar que um dos pré-requisitos para o Simplex é calcular uma solução possível? Pois é. Esta solução era conseguida "zerando" as variáveis de decisão, encontrando a solução "trivial". Vamos ver o que acontece ao "zerarmos" X_A e X_B na restrição abaixo:

$4X_A + 4X_B = 16 \Rightarrow 4*0 + 4*0 = 16 \Rightarrow 0 = 16$!?!?!?!?

Este resultado é absurdo, correto? Então, para possibilitar encontrar uma solução inicial facilmente - sem causar este absurdo, da mesma forma que com as restrições do tipo $\geqslant$, inserimos uma variável artificial nas restrições do tipo igualdade, ou seja:

$4X_A + 4X_B + A_1 = 16 \Rightarrow 4*0 + 4*0 + A_1 = 16 \Rightarrow 0 = 16 - A_1 \Rightarrow A_1 = 16$ OK!

Mais uma vez, o uso de variáveis artificiais traz implicações no Simplex.

4. Implicações dos Artifícios de Modelagem na Função Objetivo

Assim como modificávamos a Função Objetivo com as variáveis de folga, inserindo-as com coeficiente 0, também as variáveis de excesso entrarão na Função Objetivo com coeficiente zero, pela mesma razão anterior: para que todas as variáveis apareçam na Função Objetivo.

Mas, e as variáveis artificiais? Bem, como o próprio nome diz, estas variáveis não fazem parte do problema, são apenas um artifício para facilitar o cálculo de uma solução inicial de forma simples, para que possamos dar partida no Simplex. Por esta razão, é preciso garantir no método Simplex que estas variáveis sejam retiradas da base.

Para fazer isso, existem dois mecanismos:

1) **O Método do M grande**, onde as variáveis artificiais entram "prejudicando" a Função Objetivo na proporção de um valor M maior que qualquer outro valor existente no problema;
2) **O Método das Duas Fases**, sendo que na primeira fase substituímos a função objetivo pela minimização da soma de todas as variáveis artificiais e, ao obter o resultado final, voltamos a função objetivo original e continuamos a resolver o problema.

O primeiro método pode ser visto no livro, e envolve algumas mudanças no procedimento de cálculo. O segundo método é um pouco mais longo (pois envolve duas soluções seqüenciais), mas o procedimento não será diferente do que veremos no cálculo do Simplex tradicional. Ambos os métodos fogem ao escopo deste curso.

5. Exercício L3

Modele e apresente na forma padrão (pronto para o Simplex) o seguinte problema abaixo:

1) Para realizar a instalação de terminais de computador, uma empresa pode usar os esforços de dois funcionários: Pedro e João. O salário de Pedro é R$ 25,00 por hora e o de João é de R$ 40,00 por hora. Pedro consegue instalar um terminal em meia hora (0,5 hora) e João em 15 minutos (0,25 hora). É necessário instalar um total de 40 terminais, sendo que Pedro deve instalar pelo menos 10 deles. Sabe-se que nenhum dos dois funcionários pode trabalhar mais do que 8 horas em um dia. Deseja-se minimizar o custo total da instalação.

1. Modele matematicamente o problema acima.
2. Coloque o problema na forma padrão, pronto para ser resolvido pelo Simplex.

6. Bibliografia

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

Notas da Aula 09: Usando o Método Simplex para Solução de Problemas
Prof. Daniel Caetano

Objetivo: Apresentar o Método Simplex para solução de problemas de Programação Linear.

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

Introdução

A solução gráfica para problemas de programação linear, vista nas aulas anteriores, é bastante elucidativa. Entretanto, é uma forma desajeitada de resolver problemas, além de se tornar complexa ou impossível de ser aplicada para problemas com mais de duas variáveis de decisão.

Uma forma alternativa é o uso do Método Simplex para a resolução dos problemas de programação linear. O Método Simplex é um método sistemático, baseado em um *tableau*, onde indicamos todos os dados do problema e, realizando algumas operações, encontramos a solução ótima.

Embora o Simplex use cálculos bastante simples, sua seqüência é bastante tediosa. Esta característica faz com que seja interessante criar programas para resolver problemas pelo Método Simplex. Entretanto, é necessário aprender todos os passos do Simplex, verificando suas qualidades e quando surgem problemas, possibilitando uma correta interpretação dos resultados quando a solução por encontrada por meio de um software.

1. A Modelagem

Voltando ao Problema extraído de MOREIRA (2006) :

Uma fábrica produz dois produtos, A e B. Cada um deve ser processado por duas máquinas, M_1 e M_2. Devido à programação de outros produtos que também usam estas máquinas, estão disponíveis para os produtos A e B apenas 24 horas da máquina M_1 e 16 horas da máquina M_2.

Para produzir uma unidade do produto A, são necessárias 4 horas em cada uma das máquinas e para produzir uma unidade do produto B, são necessárias 6 horas em M_1 e 2 horas em M_2. Cada unidade de A vendida gera um lucro de R$ 80,00 e cada unidade de B vendida gera um lucro de R$ 60,00.

Existe uma previsão de demanda máxima de 3 unidades para B, mas nenhuma restrição de demanda para A. Deseja-se saber: quanto produzir de cada produto para maximizar o lucro?

Cujo **modelo final** era:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B$

Sujeito a:
$4 * x_A + 6 * x_B \leqslant 24$ <= Restrição do número de horas de M_1
$4 * x_A + 2 * x_B \leqslant 16$ <= Restrição do número de horas de M_2
$1 * x_B \leqslant 3$ <= Restrição de Demanda para B

Sendo as variáveis de decisão x_A e x_B.

O primeiro problema que temos ao pensar em usar Simplex é que o Método Simplex faz algumas exigências com relação aos dados; em especial, quanto às restrições. Como vimos anteriormente, o Método Simplex exige que os dados estejam na chamada "forma padrão". Assim, toda modelagem precisa ser transformada para a forma padrão antes que o problema possa ser resolvido pelo Método Simplex.

Como visto em aulas anteriores, a forma padrão para este modelo é:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B + 0 * x_{S1} + 0 * x_{S2} + 0 * x_{S3}$

Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} + 0 * x_{S2} + 0 * x_{S3} = 24$ <= Restrição M_1
$4 * x_A + 2 * x_B + 0 * x_{S1} + 1 * x_{S2} + 0 * x_{S3} = 16$ <= Restrição M_2
$0 * x_A + 1 * x_B + 0 * x_{S1} + 0 * x_{S2} + 1 * x_{S3} = 3$ <= Restrição B

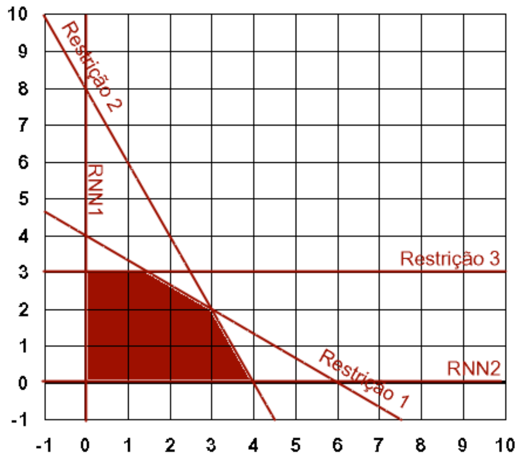
2. Soluções Básicas e Não-Básicas

Pela formulação anterior, é possível verificar que temos 5 incógnitas e apenas 3 equações. Como temos mais incógnitas do que equações, o problema é indeterminado, isto é, não é possível determinar uma solução única para ele.

Entretanto, se escolhermos duas variáveis quaisquer e fixarmos seus valores, teremos então 3 equações e 3 incógnitas, tornando-se um sistema determinado, possibilitando o cálculo das variáveis restantes, possibilitando encontrar a já conhecida "solução inicial viável".

Por facilidade nas contas, fixaremos os valores das duas variáveis escolhidas como sendo 0 (zero). As variáveis escolhidas arbitrariamente para terem seu valor definido como zero formam o que chamamos de "**solução não-básica**" e as variáveis restantes, cujos valores foram calculados, formam a chamada "**solução básica**". Note que a solução básica definida desta forma pode não ser viável, isto é, pode desrespeitar as restrições previamente impostas. Para corrigir esta situação, usamos os truques já vistos anteriormente.

Algumas aulas atrás, vimos a área de soluções possíveis para este problema:



Ainda naquela aula, vimos que a solução ótima estava sempre num ponto extremo. Assim, para verificar um resultado interessante, vamos analisar os valores das variáveis nos pontos extremos, sendo o ponto A o ponto (0,0) e os pontos B, C, D e E são seqüenciais, no sentido horário:

Ponto	X_A	X_B	X_{S1}	X_{S2}	X_{S3}
A	**0**	**0**	24	16	3
B	**0**	3	6	10	**0**
C	1,5	3	**0**	4	**0**
D	3	2	**0**	**0**	1
E	4	**0**	8	**0**	3

Observe que em todos os pontos extremos há sempre duas variáveis iguais a zero!E os extremos também representam **todas as soluções básicas possíveis**.

Assim, pode-se dizer que forma genérica que sempre que tivermos um problema de programação linear com **n** incógnitas e **m** equações, em todos os extremos da região de soluções possíveis teremos **(n-m)** incógnitas com valor igual a zero.

3. O Método Simplex

A idéia por trás do Método Simplex explora justamente esta característica que acabamos de enunciar. A idéia é procurar o conjunto de variáveis que, quando igualadas a zero, fazem com que o sistema tenha o maior valor na função objetivo.

Para iniciar o Simplex precisamos de uma solução possível, ainda que ela esteja longe de ser a melhor solução. A partir desta solução, o método permite que "naveguemos" ao londo dos extremos do espaço de soluções possíveis, até chegar à solução ótima. O ponto de partida costuma ser a **solução trivial**, ou seja, aquela em que a origem do espaço é a solução. Em outras palavras, aquela em que **as variáveis de decisão são zero**.

O Método Simplex, entretanto, não é "cego", isto é, não explora os extremos do espaço de soluções de forma aleatória. Da "solução trivial", o processo irá para o próximo extremo contíguo que fornecer o maior incremento na função objetivo (no caso de um problema de maximização).

Cada solução será apresentada na forma de uma tabelinha denominada "*tableau*". Em cada tableau serão realizados alguns cálculos que permitirão gerar o próximo tableau, que representa a próxima solução (o extremo seguinte do espáco de soluções possíveis). O processo é repetido até que qualquer mudança piore o resultado da função objetivo, ao invés de melhorar.

O algoritmo é:

1) Monta-se o tableau da solução inicial, que corresponde à origem;
2) Aplicam-se cálculos no tableau, cujo resultado é um segundo tableau;
3) Realiza-se um teste para verificar se a solução é ótima;
4) Caso não seja, repetem-se os cálculos no tableau, gerando o próximo tableau.

3.1. Exemplo de Aplicação do Método Simplex

Neste primeiro exemplo, vamos resolver o problema já apresentado, que possuía apenas restrições do tipo $\leqslant$, que já foram devidamente convertidos para a forma padrão. O modelo matemático na forma padrão encontrado anteriormente foi:

Função Objetivo:
[MAX] $80 * x_A + 60 * x_B + 0 * x_{S1} + 0 * x_{S2} + 0 * x_{S3}$

Sujeito a:
$4 * x_A + 6 * x_B + 1 * x_{S1} + 0 * x_{S2} + 0 * x_{S3} = 24$ <= Restrição M_1
$4 * x_A + 2 * x_B + 0 * x_{S1} + 1 * x_{S2} + 0 * x_{S3} = 16$ <= Restrição M_2
$0 * x_A + 1 * x_B + 0 * x_{S1} + 0 * x_{S2} + 1 * x_{S3} = 3$ <= Restrição B

Iremos agora construir o primeiro tableau.

3.1.1. Construção do Primeiro Tableau

O primeiro passo é construir uma pequena tabela. O número de linhas será o número de restrições mais quatro. Assim, em nosso problema teremos uma tabela de 7 linhas. O número de colunas é igual ao número de variáveis mais quatro, ou seja, em nosso caso, o tableau terá 9 colunas. O aspecto do primeiro tableau deve ser:

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4	6	1	0	0	24	
0	**X_{S2}**	4	2	0	1	0	16	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Na primeira linha, temos a contribuição de cada variável para a função objetivo, sendo que X_{S1} a X_{S3} em nada contribuem, obviamente.

A segunda linha é basicamente uma linha de título, contendo os nomes das variáveis e algumas outras informações. A primeira coluna, **c_j**, indica a contribuição de cada variável na solução para o valor da função objetivo. No caso, as variáveis na solução nada contribuem na função objetivo. A segunda coluna, **variáveis na solução**, indica quais variáveis estão na solução atual. No caso, são as variáveis de folga, pois serão sempre as variáveis iniciais do Simplex.

Ainda na segunda linha, aparecem os nomes das variáveis da função objetivo e restrições, além da coluna **b_j**, que representa o lado direito das equações, ou seja, o "número depois do igual" nas restrições da modelagem matemática. Finalmente, há a coluna **b_j/a_{ij}**, que será usada durante o cálculo do Simplex.

Nas linhas seguintes temos os coeficientes de cada variável, retirados diretamente do modelo matemático, sendo que cada linha é relativa a uma restrição. Note que, como as variáveis na solução são as variáveis de folga, o valor da coluna **b_j** indica os recursos disponíveis, ociosos, relativo a cada uma das variáveis de folga.

Finalmente, as duas últimas linhas. As linhas Z e C-Z são também usadas no cálculo. A idéia é indicar na linha Z quanto se retira da função objetivo por aumentar uma unidade desta variável. Já a linha CZ indica quanto se acrescenta na função objetivo por aumentar uma unidade desta variável.

Para preencher a linha Z, usaremos as linhas relativas às variáveis na solução. Multiplicaremos todos os coeficientes das variáveis X_A, X_B, X_{S1}, X_{S2} e X_{S3}, além de b_j pelo c_j da linha. Feito isso para cada elemento das duas linhas, somam-se os resultados coluna a coluna e o valor resultante é o valor da linha Z:

Passo 1: multiplicando os elementos da linha **j** pelo valor de $\mathbf{c_j}$

		80	60	0	0	0		Linha
$\mathbf{c_j}$	**Variáveis na Solução**	$\mathbf{X_A}$	$\mathbf{X_B}$	$\mathbf{X_{S1}}$	$\mathbf{X_{S2}}$	$\mathbf{X_{S3}}$	$\mathbf{b_j}$	$\mathbf{b_j / a_{ij}}$
0	$\mathbf{X_{S1}}$	4 (* 0)	6 (* 0)	1 (* 0)	0 (* 0)	0 (* 0)	24 (* 0)	
0	$\mathbf{X_{S2}}$	4 (* 0)	2 (* 0)	0 (* 0)	1 (* 0)	0 (* 0)	16 (* 0)	
0	$\mathbf{X_{S3}}$	0 (* 0)	1 (* 0)	0 (* 0)	0 (* 0)	1 (* 0)	3 (* 0)	
	Linha Z							
	Linha C-Z							

Passo 2: Somando os resultados coluna a coluna

		80	60	0	0	0		Linha
$\mathbf{c_j}$	**Variáveis na Solução**	$\mathbf{X_A}$	$\mathbf{X_B}$	$\mathbf{X_{S1}}$	$\mathbf{X_{S2}}$	$\mathbf{X_{S3}}$	$\mathbf{b_j}$	$\mathbf{b_j / a_{ij}}$
0	$\mathbf{X_{S1}}$	4 (0)	6 (0)	1 (0)	0 (0)	0 (0)	24 (0)	
0	$\mathbf{X_{S2}}$	4 (0)	2 (0)	0 (0)	1 (0)	0 (0)	16 (0)	
0	$\mathbf{X_{S3}}$	0 (0)	1 (0)	0 (0)	0 (0)	1 (0)	3 (0)	
	Linha Z	0+0+0	0+0+0	0+0+0	0+0+0	0+0+0	0+0+0	
	Linha C-Z							

Passo 3: Resultado da linha Z calculado

		80	60	0	0	0		Linha
$\mathbf{c_j}$	**Variáveis na Solução**	$\mathbf{X_A}$	$\mathbf{X_B}$	$\mathbf{X_{S1}}$	$\mathbf{X_{S2}}$	$\mathbf{X_{S3}}$	$\mathbf{b_j}$	$\mathbf{b_j / a_{ij}}$
0	$\mathbf{X_{S1}}$	4 (0)	6 (0)	1 (0)	0 (0)	0 (0)	24 (0)	
0	$\mathbf{X_{S2}}$	4 (0)	2 (0)	0 (0)	1 (0)	0 (0)	16 (0)	
0	$\mathbf{X_{S3}}$	0 (0)	1 (0)	0 (0)	0 (0)	1 (0)	3 (0)	
	Linha Z	0	0	0	0	0	0	
	Linha C-Z							

O cálculo da linha C-Z é mais simples: basta subtrair o valor de Z correspondente a cada variável do coeficiente da variável na função objetivo:

Passo 4: Calculando C-Z

		80	60	0	0	0		Linha
$\mathbf{c_j}$	**Variáveis na Solução**	$\mathbf{X_A}$	$\mathbf{X_B}$	$\mathbf{X_{S1}}$	$\mathbf{X_{S2}}$	$\mathbf{X_{S3}}$	$\mathbf{b_j}$	$\mathbf{b_j / a_{ij}}$
0	$\mathbf{X_{S1}}$	4	6	1	0	0	24	
0	$\mathbf{X_{S2}}$	4	2	0	1	0	16	
0	$\mathbf{X_{S3}}$	0	1	0	0	1	3	
	Linha Z	0	0	0	0	0	0	
	Linha C-Z	80-0	60-0	0-0	0-0	0-0		

Passo 5: C-Z calculado

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	X_A	X_B	X_{S1}	X_{S2}	X_{S3}	b_j	b_j / a_{ij}
0	X_{S1}	4	6	1	0	0	24	
0	X_{S2}	4	2	0	1	0	16	
0	X_{S3}	0	1	0	0	1	3	
	Linha Z	0	0	0	0	0	0	
	Linha C-Z	80	60	0	0	0		

Agora nosso tableau está completo. Mas como saber se esta solução, na qual X_A=0 e X_B=0, é uma solução ótima?

Bem, como a linha C-Z indica o quanto é possível aumentar na função objetivo com o acréscimo de uma unidade em uma dada variável, a solução ótima terá sido encontrada quando todos os valores na linha C-Z forem nulos (zero) ou negativos, indicando que não adianta aumentar uma unidade em variável nenhuma, o resultado da função objetivo não melhorará. Assim, a regra é: **"Quando todos os valores da linha C-Z forem nulos ou negativos, foi atingida a solução ótima"**.

No caso acima, temos valores positivos em duas colunas: na coluna do X_A e do X_B, indicando que ainda podemos melhorar a função objetivo.

3.1.2. Construção do Segundo Tableau

Para a construção do segundo tableau, precisamos selecionar duas coisas:

1) Uma variável para entrar na solução
2) Uma variável para sair da solução

O primeiro problema é de simples solução: a **variável que entra é aquela cujo valor de C-Z é o mais alto** (pois é aquela que mais contribui com o aumento da função objetivo). No nosso caso, essa variável é a X_A, que apresenta um incremento de 80 por unidade na função objetivo.

O segundo problema já é um pouco mais complicado e é agora que a coluna b_j/a_{ij} apresenta sua função. Para o cálculo, divide-se o b_j de cada linha pelo coeficiente da variável que entra na solução. Considerando que bj é a quantidade de um dado recurso disponível e aij é a quandidade deste recurso necessária para cada unidade da variável que vai entrar, a idéia é procurar qual das linhas é mais restritiva. No nosso caso, buscamos a restrição que limita mais a produção.

Em outras palavras, buscamos o **menor valor de b_j/a_{ij}**. A linha que tiver o menor valor nesta coluna, indicará a variável que sai (veja na coluna Variáveis na Solução).

Passo 6: Calculando b_j/a_{ij}

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4	6	1	0	0	24	24/4
0	**X_{S2}**	4	2	0	1	0	16	16/4
0	**X_{S3}**	0	1	0	0	1	3	3/0
	Linha Z	0	0	0	0	0	0	
	Linha C-Z	80	60	0	0	0		

Variável que Entra — Valor de b_j

Passo 7: b_j/a_{ij} calculado, identificada variável que sai

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4	6	1	0	0	24	6
0	**X_{S2}**	4	2	0	1	0	16	4
0	**X_{S3}**	0	1	0	0	1	3	∞
	Linha Z	0	0	0	0	0	0	
	Linha C-Z	80	60	0	0	0		

Variável que Entra — Valor de b_j

Ou seja, como o menor valor de b_j/a_{ij} é 4, a variável que sai é a X_{S2}, entrando a variável X_A em seu lugar. A linha da variável que sai recebe o nome de "linha principal" e o elemento no cruzamento da coluna da variável que entra com a linha principal é chamado de "elemento pivô", sendo que neste caso este elemento vale 4.

Agora, substituímos X_{S2} por X_A (lembrando de substituir cj pelo coeficiente da variável que entra na função objetivo), retiramos os valores da coluna b_j/a_{ij}, das linhas Z e C-X e, em seguida, teremos que recalcular todas as linhas do tableau. O primeiro passo é dividir todos os elementos da linha principal pelo elemento pivô:

Passo 8: Recalculando a linha principal

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4	6	1	0	0	24	
80	**X_A**	4/4	2/4	0/4	1/4	0/4	16/4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Passo 9: Linha principal recalculada

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4	6	1	0	0	24	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

O passo seguinte é recalcular as outras linhas. Neste caso, as outras linhas são referentes às variáveis XS1 e XS3. Vejamos linha por linha.

Recalculando X_{S1}:

Primeiramente, determinamos o pivo da linha da variável X_{S1}, que é o cruzamento da linha de X_{S1} com a coluna da variável que está entrando, no caso, X_A. No caso, este pivô também vale 4.

Passo 10: Pivô da linha da variável X_{S1}

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4	6	1	0	0	24	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

A idéia é, então, subtrair de cada elemento da linha X_{S1} o valor do pivô desta linha multiplicado peloo elemento correspondente da linha principal, como indicado no tableau abaixo:

Passo 11: Recalculando a linha da variável X_{S1}

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	4 -4*1	6 -4*1/2	1 -4*0	0 -4*1/4	0 -4*0	24 -4*4	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Passo 11: Linha da variável X_{S1} recalculada

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0	4	1	-1	0	8	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Recalculando X_{S3}:

Novamente, determinamos o pivo da linha, agora da variável X_{S3}, que é o cruzamento da linha de X_{S3} com a coluna da variável que está entrando, no caso, X_A. No caso, este pivô vale 0.

Passo 12: Determinação do pivô da linha X_{S2}

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0	4	1	-1	0	8	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Mais uma vez, subtrairemos de cada elemento da linha X_{S3} o valor do pivô desta linha multiplicado pelo elemento correspondente da linha principal, como indicado no tableau abaixo:

Passo 13: Recalculando a linha da variável X_{S3}

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0	4	1	-1	0	8	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0 -0*1	1 -0*1/2	0 -0*0	0 -0*1/4	1 -0*0	3 -0*4	
	Linha Z							
	Linha C-Z							

Passo 14: Linha da variável X_{S3} recalculada

		80	60	0	0	0		Linha
c_j	**Variáveis na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0	4	1	-1	0	8	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Agora nosso tableau está atualizado e podemos recalcular a linha Z e C-Z:

Passo 15: Calculando a nova linha Z

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0 (*0)	4 (*0)	1 (*0)	-1 (*0)	0 (*0)	8 (*0)	
80	**X_A**	1(*80)	1/2(*80)	0 (*80)	1/4(*80)	0 (*80)	4 (*80)	
0	**X_{S3}**	0 (*0)	1 (*0)	0 (*0)	0 (*0)	1 (*0)	3 (*0)	
	Linha Z	0+80+0	0+40+0	0+0+0	0+20+0	0+0+0	0+320+0	
	Linha C-Z							

Passo 16: Linha Z calculada e calculando a nova linha C-Z

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0 (*0)	4 (*0)	1 (*0)	-1 (*0)	0 (*0)	8 (*0)	
80	**X_A**	1(*80)	1/2(*80)	0 (*80)	1/4(*80)	0 (*80)	4 (*80)	
0	**X_{S3}**	0 (*0)	1 (*0)	0 (*0)	0 (*0)	1 (*0)	3 (*0)	
	Linha Z	80	40	0	20	0	320	
	Linha C-Z	80-80	60-40	0-0	0-20	0-0		

Passo 17: Tableau 2 finalizado

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0 (*0)	4 (*0)	1 (*0)	-1 (*0)	0 (*0)	8 (*0)	
80	**X_A**	1(*80)	1/2(*80)	0 (*80)	1/4(*80)	0 (*80)	4 (*80)	
0	**X_{S3}**	0 (*0)	1 (*0)	0 (*0)	0 (*0)	1 (*0)	3 (*0)	
	Linha Z	80	40	0	20	0	320	
	Linha C-Z	0	20	0	-20	0		

3.1.3. Construção do Terceiro Tableau

Para a construção do terceiro tableau, novamente precisamos identificar a variável que vai entrar na solução e a variável que vai sair da solução. O C-Z mais alto é agora o da variável X_B, sendo ela a entrar na solução. Calculemos os bj/aij de cada linha, de forma a identificar a variável que sai:

Passo 18: Indicação da variável que entra e variável que sai

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
0	**X_{S1}**	0	4	1	-1	0	8	8/4 = 2
80	**X_A**	1	1/2	0	1/4	0	4	4/0,5 = 8
0	**X_{S3}**	0	1	0	0	1	3	3/1 = 3
	Linha Z	80	40	0	20	0	320	
	Linha C-Z	0	20	0	-20	0		

O menor valor de **b_j/a_{ij}** foi 2, na linha de X_{S1}. Assim, é esta a variável que sai, como indicado no tableau anterior, sendo o pivô da linha principal o valor 4 também indicado.

Os próximos passos são a substituição da variável, ajuste do c_j, eliminação dos conteúdos da coluna b_j/a_{ij} e das linhas Z e C-Z, seguindo-se o recálculo da linha principal, dividindo todos seus elementos pelo valor do pivô:

Passo 19: Cálculo da nova linha principal

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0/4=0	4/4=1	1/4	-1/4	0/4=0	8/4=2	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Passo 20: Linha principal já recalculada

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1	1/2	0	1/4	0	4	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

O pivô da linha X_A é 1/2. Recalculemos a linha de X_A, subtraindo de cada elemento dela o seu pivô multiplicado pelo elemento da linha principal:

Passo 21: Recalculando a linha X_A

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1 -1/2*0	1/2 -1/2*1	0 -1/2*1/4	1/4 -1/2*-1/4	0 -1/2*0	4 -1/2*2	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

Passo 22: Linha X_A recalculada

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1	0	-1/8	3/8	0	3	
0	**X_{S3}**	0	1	0	0	1	3	
	Linha Z							
	Linha C-Z							

O pivô da linha X_{S3} é 1. Recalculemos a linha de X_{S3}, subtraindo de cada elemento dela o seu pivô multiplicado pelo elemento da linha principal:

Passo 23: Recalculando a linha X_{S3}

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1	0	-1/8	3/8	0	3	
0	**X_{S3}**	0 -1*0	1 -1*1	0 -1*1/4	0 -1*-1/4	1 -1*0	3 -1*2	
	Linha Z							
	Linha C-Z							

Passo 24: Linha X_{S3} recalculada

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1	0	-1/8	3/8	0	3	
0	**X_{S3}**	0	0	-1/4	1/4	1	1	
	Linha Z							
	Linha C-Z							

Agora nosso tableau está atualizado e podemos recalcular a linha Z e C-Z:

Passo 25: Calculando a nova linha Z

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0 (*60)	1 (*60)	1/4(*60)	-1/4(*60)	0 (*60)	2 (*60)	
80	**X_A**	1 (*80)	0 (*80)	-1/8(*80)	3/8 (*80)	0 (*80)	3 (*80)	
0	**X_{S3}**	0 (*0)	0 (*0)	-1/4(*0)	1/4 (*0)	1 (*0)	1 (*0)	
	Linha Z	0+80+0	60+0+0	15-10+0	-15+30+0	0+0+0	120+240+0	
	Linha C-Z							

Passo 26: Calculando a nova linha C-Z

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1	0	-1/8	3/8	0	3	
0	**X_{S3}**	0	0	-1/4	1/4	1	1	
	Linha Z	80	60	5	15	0	360	
	Linha C-Z	80-80	60-60	0-5	0-15	0-0		

Passo 27: Tableau final

		80	60	0	0	0		Linha
c_j	**Var. na Solução**	**X_A**	**X_B**	**X_{S1}**	**X_{S2}**	**X_{S3}**	**b_j**	**b_j / a_{ij}**
60	**X_B**	0	1	1/4	-1/4	0	2	
80	**X_A**	1	0	-1/8	3/8	0	3	
0	**X_{S3}**	0	0	-1/4	1/4	1	1	
	Linha Z	80	60	5	15	0	360	
	Linha C-Z	0	0	-5	-15	0		

Como na linha C-Z não há qualquer valor maior que zero, esta é a solução ótima. A solução é indicada pelas variáveis na coluna "Variáveis na Solução" e seus respectivos valores estão na coluna b_j, sendo que as variáveis que não estão em nenhuma linha da tabela têm, por definição, valor igual a zero.

Assim, a solução ótima para o problema é:
$X_A = 3$
$X_B=2$
$X_{S1} = 0$
$X_{S2} = 0$
$X_{S3} = 1$
O que significa que produziremos 3 unidades de A, 2 unidades de B, esgotando as horas de máquina 1 e 2 disponíveis, mas não atendendo completamente a demanda de B.

4. Bibliografia

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

Notas da Aula 10: Resolução da L3
Prof. Daniel Caetano

Objetivo: Resolver o problema da Lista 3, apresentando artifícios de modelagem.

Bibliografia:

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

Resolução do Exercício L3

Enunciado:

Modele e apresente na forma padrão (pronto para o Simplex) o seguinte problema abaixo:

1) Para realizar a instalação de terminais de computador, uma empresa pode usar os esforços de dois funcionários: Pedro e João. O salário de Pedro é R$ 25,00 por hora e o de João é de R$ 40,00 por hora. Pedro consegue instalar um terminal em meia hora (0,5 hora) e João em 15 minutos (0,25 hora). É necessário instalar um total de 40 terminais, sendo que Pedro deve instalar pelo menos 10 deles. Sabe-se que nenhum dos dois funcionários pode trabalhar mais do que 8 horas em um dia. Deseja-se minimizar o custo total da instalação.

1. Modele matematicamente o problema acima.

Função Objetivo:

Como queremos minimizar o custo de instalação, a primeira coisa é determinar como podemos calcular este custo. Bem, temos dois funcionários, sendo que ambos recebem um valor específico por hora. Assim, se designarmos:

x_P = número de horas trabalhadas por Pedro
x_J = número de horas trabalhadas por João

Sabendo que o salário de Pedro é R$ 25,00 por hora e o de João é de R$ 40,00 por hora, o custo de cada um dos funcionários será:

Pedro: $25 * x_P$ (em reais)
João: $40 * x_J$ (em reais)

Assim, o custo total será:

Custo = $25*x_P + 40*x_J$

Como queremos minimizar o custo, a função objetivo fica:

[MIN] $25*x_P + 40*x_J$

Restrições:

Há, basicamente, 4 restrições:

- Número total de máquinas a instalar: 40
- Número mínimo de máquinas que serão instaladas por Pedro: 10
- Número máximo de horas de trabalho de Pedro: 8
- Número máximo de horas de trabalho de João: 8

- Número total de máquinas a instalar: 40

Primeiro é preciso descobrir quantas máquinas Pedro e João instalam por hora. Pelo enunciado, Pedro instala um terminal em meia hora. Em outras palavras, ele instala 2 máquinas em uma hora. Como x_P é número de horas trabalhadas por Pedro, o número total de máquinas instaladas por Pedro é:

Máquinas Instaladas por Pedro: $2*x_P$

Da mesma forma, é dito que João instala uma máquina em 15 minutos, ou seja, ele instala 4 máquinas em uma hora. Como x_J é número de horas trabalhadas por João, o número total de máquinas instaladas por João é:

Máquinas Instaladas por João: $4*x_J$

Assim, o total de máquinas instaladas é:

$2*x_P + 4*x_J$

Que deve ser exatamente igual a 40... ou seja:

$2*x_P + 4*x_J = 40$ <= Restrição do número de máquinas a instalar.

- Número mínimo de máquinas que serão instaladas por Pedro: 10

Já vimos que o número de máquinas instaladas por Pedro $2*x_P$. O que esta restrição diz é que este número precisa ser pelo menos igual a 10, ou seja, ele é maior ou igual a 10. Assim:

$2*x_P \geqslant 10$ <= Restrição do mínimo de máquinas para Pedro

- Número máximo de horas de trabalho de Pedro: 8

Como x_P é o número de horas trabalhadas por Pedro, este valor deve ser menor que 8:

$x_P \leqslant 8$ <= Número máximo de horas de trabalho para Pedro
$x_J \leqslant 8$ <= Número máximo de horas de trabalho para João

Modelo Final

F.O.:

$[MIN]\ 25*x_P + 40*x_J$

S.A.:

$2*x_P + 4*x_J = 40$ <= Restrição do número de máquinas a instalar.
$2*x_P \geqslant 10$ <= Restrição do mínimo de máquinas para Pedro
$x_P \leqslant 8$ <= Número máximo de horas de trabalho para Pedro
$x_J \leqslant 8$ <= Número máximo de horas de trabalho para João

2. Coloque o problema na forma padrão, pronto para ser resolvido pelo Simplex.

Primeiramente, vamos colocar variáveis de folga e excesso nas restrições do tipo ⩽ e ⩾ respectivamente, sendo que apenas com isso as restrições já podem se tornar igualdades:

$$\begin{array}{llllllll}
\text{F.O.:} & [MIN] & 25*x_P & +40*x_J & & & & \\
\text{S.A.:} & & 2*x_P & +4*x_J & & & = 40 & (=) \\
 & & 2*x_P & & -1*x_{E1} & & = 10 & (\geqslant) \\
 & & 1*x_P & & +1*x_{S1} & & = 8 & (\leqslant) \\
 & & & 1*x_J & & +1*x_{S2} & = 8 & (\leqslant)
\end{array}$$

Sinal Antigo ---^

Entretanto, este problema ainda não admite solução direta para $x_P = x_J = 0$, algo que desejamos para poder iniciar o Simplex. As restrições problemáticas são aquelas cujos sinais originais eram ⩾ ou =, uma vez que com a igualdade teremos resultados estranhos ao fazer $x_P = x_J = 0$:

$2*x_P + 4*x_J = 40 \Rightarrow (x_P = x_J = 0) \Rightarrow 2*0 + 4*0 = 40 \Rightarrow 0 = 40$!?!?!!?
$2*x_P - 1*x_{E1} = 10 \Rightarrow (x_P = x_J = 0) \Rightarrow 2*0 -1 *x_{E1}= 10 \Rightarrow x_{E1}= -10$!?!?!!?

Assim, precisamos adicionar variáveis artificiais nestas restrições, de forma a tornar a solução $x_P = x_J = 0$ possível. Com isso, o problema fica:

$$
\begin{array}{llllllllll}
\text{F.O.:} & [\text{MIN}] & 25*x_P & + 40*x_J & & & & & & \\
\text{S.A.:} & & 2*x_P & + 4*x_J & & & & +1*a_1 & & = 40 \\
 & & 2*x_P & & - 1*x_{E1} & & & & +1*a_2 & = 10 \\
 & & 1*x_P & & & + 1*x_{S1} & & & & = 8 \\
 & & & 1*x_J & & & + 1*x_{S2} & & & = 8
\end{array}
$$

Agora a modelagem está quase completa, faltando apenas fazer com que todas as variáveis apareçam em todas as linhas, o que é conseguido acrescentando-as com coeficiente zero onde elas não aparecem:

$$
\begin{array}{llllllllll}
\text{F.O.:} & [\text{MIN}] & 25*x_P & + 40*x_J & + 0*x_{E1} & + 0*x_{S1} & + 0*x_{S2} & +0*a_1 & +0*a_2 & \\
\text{S.A.:} & & 2*x_P & + 4*x_J & + 0*x_{E1} & + 0*x_{S1} & + 0*x_{S2} & +1*a_1 & +0*a_2 & = 40 \\
 & & 2*x_P & +0*x_J & - 1*x_{E1} & + 0*x_{S1} & + 0*x_{S2} & +0*a_1 & +1*a_2 & = 10 \\
 & & 1*x_P & +0*x_J & + 0*x_{E1} & + 1*x_{S1} & + 0*x_{S2} & +0*a_1 & +0*a_2 & = 8 \\
 & & 0*x_P & 1*x_J & + 0*x_{E1} & + 0*x_{S1} & + 1*x_{S2} & +0*a_1 & +0*a_2 & = 8
\end{array}
$$

O último truque a ser aplicado é a conversão de um problema de minimização para um de maximização, bastando, para isso, multiplicar a função objetivo por -1:

$$
\begin{array}{llllllllll}
\text{F.O.:} & [\text{MAX}] & -25*x_P & - 40*x_J & - 0*x_{E1} & - 0*x_{S1} & - 0*x_{S2} & - 0*a_1 & - 0*a_2 & \\
\text{S.A.:} & & 2*x_P & + 4*x_J & + 0*x_{E1} & + 0*x_{S1} & + 0*x_{S2} & +1*a_1 & +0*a_2 & = 40 \\
 & & 2*x_P & +0*x_J & - 1*x_{E1} & + 0*x_{S1} & + 0*x_{S2} & +0*a_1 & +1*a_2 & = 10 \\
 & & 1*x_P & +0*x_J & + 0*x_{E1} & + 1*x_{S1} & + 0*x_{S2} & +0*a_1 & +0*a_2 & = 8 \\
 & & 0*x_P & 1*x_J & + 0*x_{E1} & + 0*x_{S1} & + 1*x_{S2} & +0*a_1 & +0*a_2 & = 8
\end{array}
$$

3. Bibliografia

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

Notas da Aula 10a: Ferramentas Computacionais
Prof. Daniel Caetano

Objetivo: Apresentar o uso do Excel/Solver para a resolução de um problema de otimização.

1. Apresentação do Problema

1) Para realizar a instalação de terminais de computador, uma empresa pode usar os esforços de dois funcionários: Pedro e João. O salário de Pedro é R$ 25,00 por hora e o de João é de R$ 40,00 por hora. Pedro consegue instalar um terminal em meia hora (0,5 hora) e João em 15 minutos (0,25 hora). É necessário instalar um total de 40 terminais, sendo que Pedro deve instalar pelo menos 10 deles. Sabe-se que nenhum dos dois funcionários pode trabalhar mais do que 8 horas em um dia. Deseja-se minimizar o custo total da instalação.

Como já foi visto, a modelagem matemática deste problema é:

F.O.: [MIN] $25*x_P + 40*x_J$
S.A.: $2*x_P + 4*x_J = 40$ <= Restrição do número de máquinas a instalar.
$2*x_P \geqslant 10$ <= Restrição do mínimo de máquinas para Pedro
$x_P \leqslant 8$ <= Número máximo de horas de trabalho para Pedro
$x_J \leqslant 8$ <= Número máximo de horas de trabalho para João

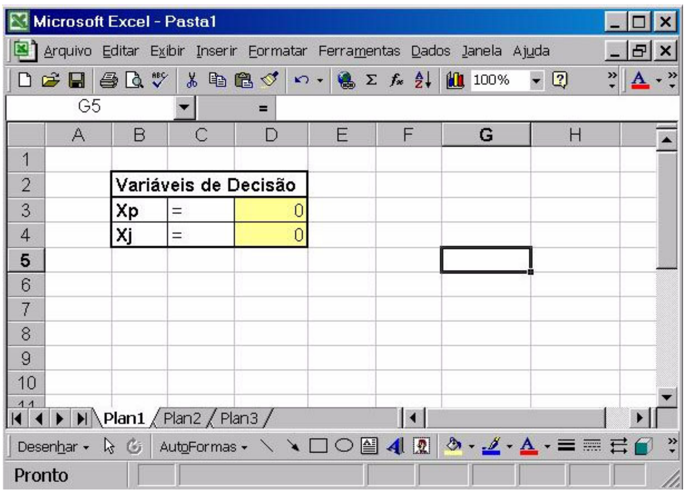
2. Usando o Computador para Resolver o Problema

Para resolver um problema deste tipo, existem inúmeras ferramentas. Algumas delas, as mais poderosas (LINDO, LINGO, GAMS, etc) exigem que o problema seja transformado para a forma padrão antes de ser fornecido para o computador.

Entretanto, há ferramentas mais simples no mercado, como aquela que vem no próprio Excel, chamada Solver. O Solver é uma biblioteca do Excel que implementa algoritmos de programação linear e não linear. Curiosamente, para o uso do Excel não é necessário usar a forma padrão, como veremos a seguir.

2.1. Preparando a Planilha para o Uso do Solver

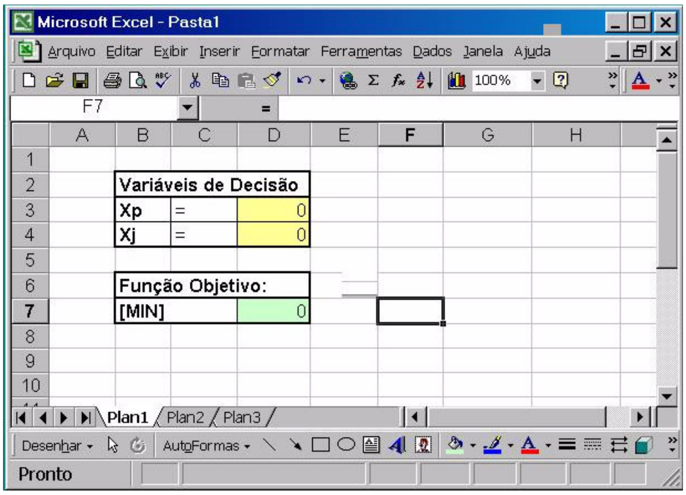
Ao abrir o Excel, temos uma planilha em branco. Mas podemos preenchê-la de uma forma a facilitar nosso trabalho para seu uso posterior no solver. A idéia é listar primeiramente todas as variáveis de decisão, com seu nome e valor inicial (normalmente 0), como apresentado na tela a seguir. Note que estão marcados em amarelo os valores das variáveis de decisão.



O próximo passo é representar a Função Objetivo, como na figura abaixo, lembrando que no local onde aparece o valor da função objetivo é preciso escrever a fórmula da mesma, baseada no valor das células de decisão já indicadas, sendo que Xp é a célula D3 e Xj é a célula D4, a fórmula $25*x_P + 40*x_J$ fica:

= 25*D3 + 40*D4

Note que está marcado em verde o valor da função objetivo.

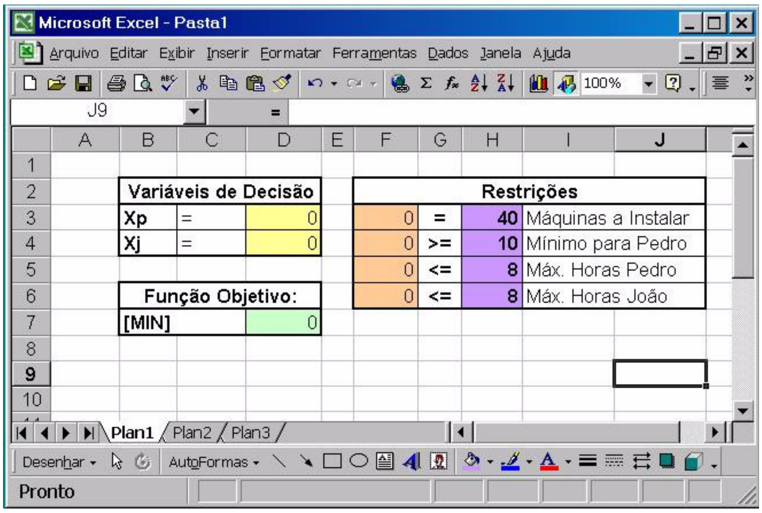


Falta agora representar as restrições, sendo que elas devem ser representadas em 3 colunas: a primeira contém as fórmulas do lado esquerdo de cada restrição, convertidas para o formato Excel, a segunda contem o sinal (igual, maior ou igual, menor ou igual) e a terceira contem o lado direito das restrições (o número).

O lado direito das restrições $2*x_P + 4*x_J = 40$, $2*x_P \geqslant 10$, $x_P \leqslant 8$ e $x_J \leqslant 8$ ficam, no formado do Excel, respectivamente:

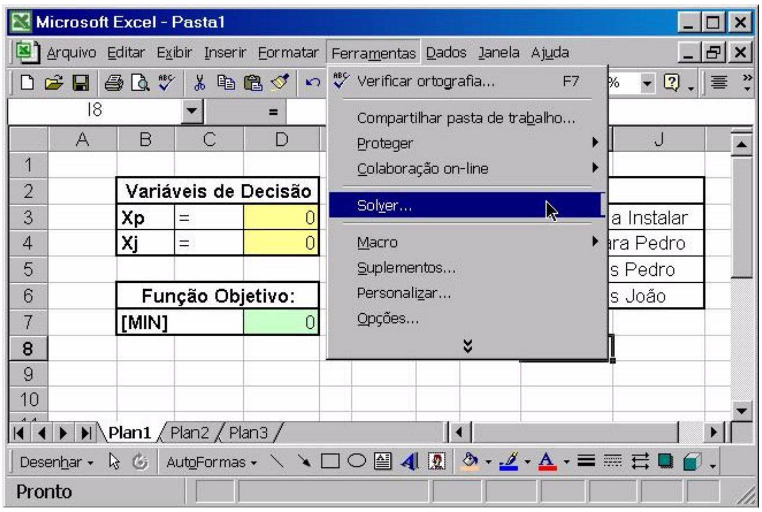
= 2*D3 + 4*D4
= 2*D3
= D3
= D4

Quando colocadas no Excel, as restrições deverão ficar como na figura a seguir. Note que os valores das restrições estão em vermelho e os limites das restrições estão em roxo.

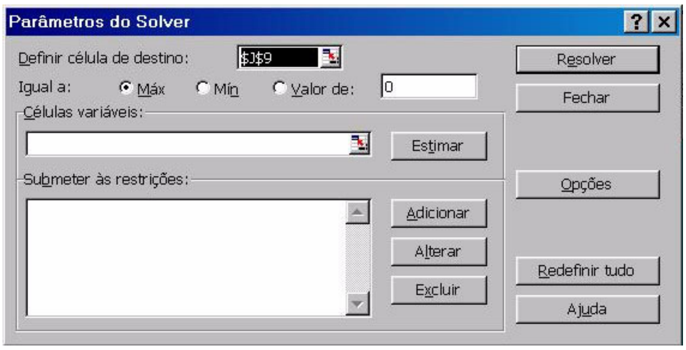


2.1. Iniciando e Preparando o Solver

Agora que a planilha está pronta, está na hora de iniciar o Solver. O solver se encontra no menu ferramentas, como indicado na figura a seguir:



Caso esta opção não esteja disponível, será necessário instalar o módulo do Solver. Clicando nesta opção, aparecerá a janela principal do Solver, representada abaixo:



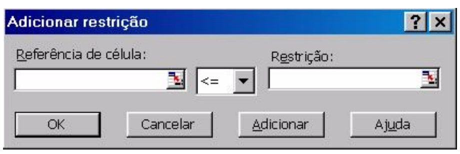
Antes de mais nada, devemos entrar em "Opções" para configurar o tipo de problema que queremos resolver, já que o Solver serve para muitos outros tipos de problema. A janela inicial de Opções é:

As mudanças que faremos são: marcar "Presumir modelo linear" e "Presumir não negativos". Depois disso, clique em OK.

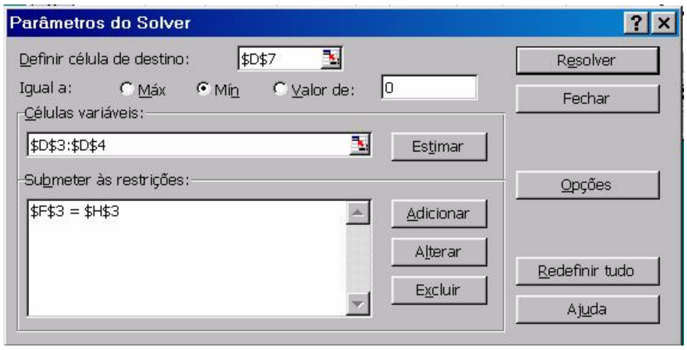
Após isso, será apresentada novamente a janela do Solver. No campo "Definir Célula Destino", clique no botão e indique a célula em verde. No campo "Igual a" marque "Min", que indica que queremos minimizar o valor.

No campo "Células Variáveis", clique no botão e indique as células em amarelo (todas ao mesmo tempo).

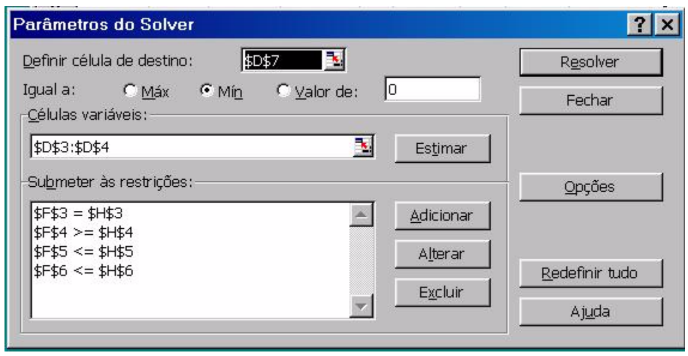
No campo "Submeter às Restrições" teremos um pouco mais de trabalho. Clique no botão "Adicionar". A janela abaixo será apresentada:



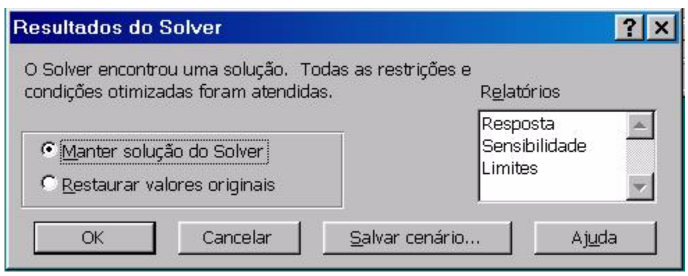
Vamos adicionar a primeira restrição. No campo "Referência da célula", clique no botão e selecione a primeira célula laranja (F3). No campo sem nome que contém o sinal, indique o mesmo sinal da restrição, que no caso é o igual (=). No campo "Restrição", clique no botão e indique a primeira célula roxa (H3). Feitoisso, clique no botão OK e a primeira restrição estará adicionada, como indicado na figura abaixo:



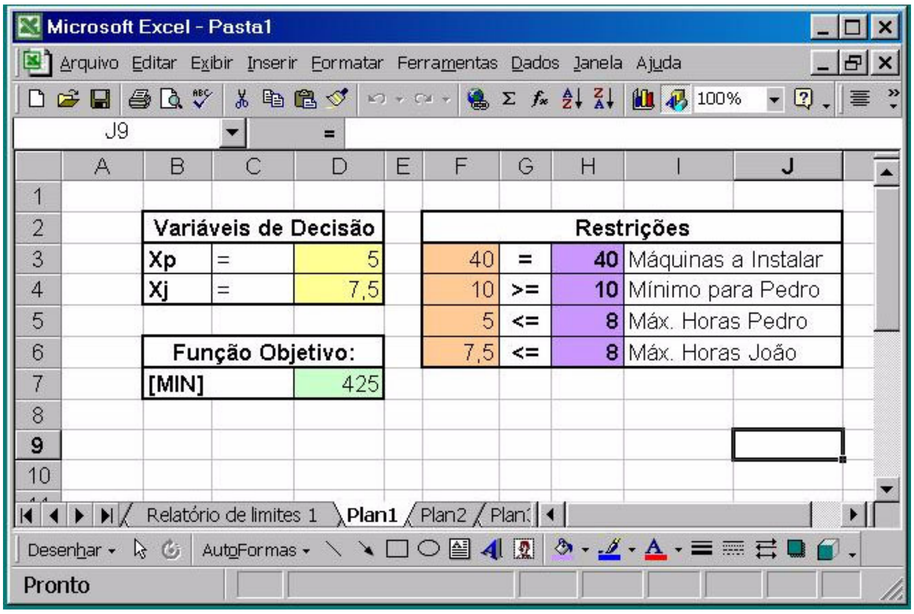
Repita o processo para as outras 3 restrições. No final, a janela do solver terá a seguinte aparência:



Neste momento, basta clicar no botão "Resolver". Após isso, aparecerá a seguinte janela:



Após decidir se quer visualizar algum dos relatórios (Resposa, Sensibilidade e Limites), clique em Ok. Neste ponto, a planilha do Excel mostrará a solução final, como apresentado na figura a seguir.



Ou seja: a solução é que Xp = 5 (Pedro trabalhará 5 horas) e Xj = 7,5 (João trabalhará 7,5 horas). O custo final será R$ 425,00. Estes valores também são apresentados no relatório de respostas.

O "Relatório de Sensibilidade" e "Relatório de Limites" apresentam algumas informações adicionais sobre variações que poderiam ocorrer sem que a solução fosse modificada.

Notas da Aula 11: O Problema da Atribuição/Designação
Prof. Daniel Caetano

Objetivo: Apresentar a modelagem do Problema da Atribuição como base para posterior apresentação do algoritmo húngaro.

Bibliografia:

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

Introdução

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.

1. Um Problema de Atribuição

Vejamos um exemplo (MOREIRA, 2006, pág. 122, modificado) para compreender melhor qual é este tipo de problema.

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. Modelagem Matemática do Problema da Atribuição

Consideremos para a resolução as variáveis de decisão X_{ij}, sendo i o número da máquina e j o número do trabalho: 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. Assim:

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, temos que se $X_{11} = 1$, 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 parao 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 escrito assim:

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

Bem, o que é 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, teremos:

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

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

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, temos que se $X_{11} = 1$, 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 escrito da seguinte maneira:

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

Bem, o que é dito sobre o trabalho 1, pode também ser dito sobre o trabalho 2 e sobre o trabalho 3:

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

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, temos 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} diz se uma atribuição foi feita, 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: $15^* X_{23}$

Jutanto o custo das 9 possíveis atribuições, definimos 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 isso pode demorar um bom tempo... e que, para problemas muito maiores, a execução será inviável.

A solução para isso é o uso do algoritmo húngaro, que será apresentado na aula que vem, e é específico para problemas de atribuição e designação.

3. Exercício L4 (Livro)

Em uma empresa de construção civil, há três projetos que podem ser alocados a três equipes diferentes. Tanto o tempo de experiência das equipes quanto suas orientações técnicas são distintas, de modo que o tempo de término de cada projeto dependerá da equipe específica ao qual ele for alocado. A matriz a seguir mostra os tempos para cada equipo e projeto. Modele como um problema de programação linear (para o Simplex, mas não precisa colocar na forma padrão) e aplique o Algoritmo Húngaro para chegar à alocação ótima, ou seja, o menor número de horas pagas de desenvolvimento.

	Projeto A	Projeto B	Projeto C
Equipe 1	15	24	21
Equipe 2	17	22	18
Equipe 3	23	29	30

Bibliografia

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

Notas da Aula 12: O Algoritmo Húngaro
Prof. Daniel Caetano

Objetivo: Apresentar o algoritmo húngaro, para solução do problema da atribuição.

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

Introdução

Como foi visto anteriormente, o problema da atribuição/designação pode ser modelado matematicamente como um problema de programação linear, visando sua solução pelo método Simplex.

Entretanto, a solução pelo método Simplex pode ser excessivamente demorada em problemas muito grandes, com milhares de atividades a serem atribuidas. Por esta razão, foi criado um algoritmo chamado "Algoritmo Húngaro" que é capaz de resolver problemas de atribuição com um esforço computacional bastante menor que o Simplex.

Nesta aula veremos como solucionar o problema apresentado na aula anterior com o uso do Algorítmo Húngaro.

1. 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) que precisa ser minimizada.

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.

2. Aplicação do Algoritmo Húngaro a um Problema

Para aplicar o algoritmo, voltemos ao exemplo (MOREIRA, 2006, pág. 122, modificado) da aula anterior.

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.

2.1. Aplicação do Algoritmo

O Algoritmo Húngaro é composto de várias etapas, que serão vistas a seguir, para resolver o problema previamente apresentado. Vejamos 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

Temos 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, segue-se para o passo 9. Caso o número fosse igual a três, poderíamos seguir para o passo 13. Note que se o número de retas for MAIOR que a ordem, 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

Localize a primeira linha ou coluna que aparece apenas UM zero e faça a atribuição. 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:

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

Localize a próxima linha ou coluna que aparece apenas UM zero e faça a 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. Tornando o Algoritmo Húngaro Genérico

Vimos anteriormente que, para que o Algoritmo Húngaro funcione, temos a necessidade de obedecer alguns critérios... mas e quando isso não ocorre? Neste caso, iremos usar alguns truques para "forçar" os critérios necessários.

Número de Linhas e Colunas Diferente: Neste caso criam-se 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.

Bibliografia

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

Notas da Aula 13: Resolução da Lista L4
Prof. Daniel Caetano

Objetivo: Apresentar a resolução da listsa.

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

1. Exercício L4

Em uma empresa de construção civil, há três projetos que podem ser alocados a três equipes diferentes. Tanto o tempo de experiência das equipes quanto suas orientações técnicas são distintas, de modo que o tempo de término de cada projeto dependerá da equipe específica ao qual ele for alocado. A matriz a seguir mostra os tempos para cada equipo e projeto. Modele como um problema de programação linear (para o Simplex, mas não precisa colocar na forma padrão) e aplique o Algoritmo Húngaro para chegar à alocação ótima, ou seja, o menor número de horas pagas de desenvolvimento.

	Projeto A	Projeto B	Projeto C
Equipe 1	15	24	21
Equipe 2	17	22	18
Equipe 3	23	29	30

Modelagem Matemática

Consideremos para a resolução as variáveis de decisão X_{ij}, sendo i o número da equipe e j a letra do projeto: X_{1B} indica se a equipe 1 foi (ou não) atribuída para o projeto B; X_{3A} indica se a equipe 3 foi (ou não) atribuída ao projeto A. Ou seja: quando uma destas variáveis for 1, houve a atribuição. Se ela for 0, não houve. Assim:

Se $X_{1B} = 1$, a equipe 1 **foi** atribuída ao projeto B.
Se $X_{1B} = 0$, a equipe 1 **não foi** atribuída ao projeto B.
Se $X_{3A} = 1$, a equipe 3 **foi** atribuída ao projeto A.
Se $X_{3A} = 0$, a equipe 3 **não foi** atribuída ao projeto A.

Note que as variáveis X_{ij} só podem assumir valores 0 ou 1, já que não faz sentido dizer que uma equipe foi "meio" atribuída a uma atividade e "meio" atribuída a outra. Assim, como cada equipe só pode ser atribuída a um projeto, temos que se $X_{1A} = 1$, X_{1B} e X_{1C} precisam ser, obrigatoriamente, iguais a 0. Isso significa dizer que se a equipe 1 foi usada para o projeto A ($X_{1A} = 1$) esta equipe 1 não pode ser usada para os projetos B e C ($X_{1B} = X_{1C} = 0$). Da mesma forma, se $X_{1B} = 1$, então $X_{1A} = X_{1C} = 0$... ou ainda, se $X_{1C} = 1$, então $X_{1A} = X_{1B} = 0$. Ora, como é possível ver, a soma dos valores de $X_{1j} = 1$, sempre! Isso pode ser escrito assim:

$X_{1A} + X_{1B} + X_{1C} = 1, \qquad X_{1j} \in \{0, 1\}$

Bem, o que é dito sobre a equipe 1, pode também ser dito sobre a equipe 2 e sobre a equipe 3:

$X_{2A} + X_{2B} + X_{2C} = 1, \qquad X_{2j} \in \{0, 1\}$
$X_{3A} + X_{3B} + X_{3C} = 1, \qquad X_{3j} \in \{0, 1\}$

Juntando todas estas restrições, teremos:

$X_{1A} + X_{1B} + X_{1C} = 1$
$X_{2A} + X_{2B} + X_{2C} = 1$
$X_{3A} + X_{3B} + X_{3C} = 1$

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

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

$X_{1A} + X_{2A} + X_{3A} = 1, \qquad X_{iA} \in \{0, 1\}$

Bem, o que é dito sobre o projeto A, pode também ser dito sobre o projeto B e sobre o projeto C:

$X_{1B} + X_{2B} + X_{3B} = 1, \qquad X_{iB} \in \{0, 1\}$
$X_{1C} + X_{2C} + X_{3C} = 1, \qquad X_{iC} \in \{0, 1\}$

Juntando todas estas restrições, teremos:

$X_{1A} + X_{2A} + X_{3A} = 1$
$X_{1B} + X_{2B} + X_{3B} = 1$
$X_{1C} + X_{2C} + X_{3C} = 1$

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

Com isso, temos a definição completa das restrições:

$X_{1A} + X_{1B} + X_{13} = 1$ <= Equipe 1 só pega um projeto
$X_{2A} + X_{2B} + X_{23} = 1$ <= Equipe 2 só pega um projeto
$X_{3A} + X_{3B} + X_{33} = 1$ <= Equipe 3 só pega um projeto
$X_{1A} + X_{2A} + X_{3A} = 1$ <= Projeto A só está em uma equipe
$X_{1B} + X_{2B} + X_{3B} = 1$ <= Projeto B só está em uma equipe
$X_{1C} + X_{2C} + X_{3C} = 1$ <= Projeto C só está em uma equipe

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

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} diz se uma atribuição foi feita, basta multiplicar o custo de cada atribuição (dados pelo problema) pela variável X_{ij} que identifica se aquela atribuição foi feita:

	Projeto A	Projeto B	Projeto C
Equipe 1	15	24	21
Equipe 2	17	22	18
Equipe 3	23	29	30

Exemplo 1:
Custo da atribuição da E1 ao PA: **15**
Variável que indica se esta atribuição foi feita (1 se sim, 0 se não): $\mathbf{X_{1A}}$
Custo final desta atribuição: $15* X_{1A}$

Exemplo 2:
Custo da atribuição da E2 ao PC: **18**
Variável que indica se esta atribuição foi feita (1 se sim, 0 se não): $\mathbf{X_{2C}}$
Custo final desta atribuição: $18* X_{2C}$

Jutanto o custo das 9 possíveis atribuições, definimos a função objetivo:

F.O.:
[MIN] $15X_{1A} + 24X_{1B} + 21X_{1C} + 17X_{2A} + 22X_{2B} + 18X_{2C} + 23X_{3A} + 29X_{3B} + 30X_{3C}$
$X_{1A} + X_{1B} + X_{13} = 1$
$X_{2A} + X_{2B} + X_{23} = 1$
$X_{3A} + X_{3B} + X_{33} = 1$
$X_{1A} + X_{2A} + X_{3A} = 1$
$X_{1B} + X_{2B} + X_{3B} = 1$
$X_{1C} + X_{2C} + X_{3C} = 1$

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

Resolução pelo Algoritmo Húngaro

Passo 1: Desenho da tabela de custos

	Projeto A	Projeto B	Projeto C
Equipe 1	15	24	21
Equipe 2	17	22	18
Equipe 3	23	29	30

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

	Projeto A	Projeto B	Projeto C
Equipe 1	>>15<<	24	21
Equipe 2	>>17<<	22	18
Equipe 3	>>23<<	29	30

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

	Projeto A	Projeto B	Projeto C
Equipe 1	0	9	6
Equipe 2	0	5	1
Equipe 3	0	6	7

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

	Projeto A	Projeto B	Projeto C
Equipe 1	>>0<<	9	6
Equipe 2	0	>>5<<	>>1<<
Equipe 3	0	6	7

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

	Projeto A	Projeto B	Projeto C
Equipe 1	0	4	5
Equipe 2	0	0	0
Equipe 3	0	1	6

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

	Projeto A	Projeto B	Projeto C
Equipe 1	0 |	4	5
Equipe 2	---------- 0 |----------	---------- 0 ----------	---------- 0 ----------
Equipe 3	0 |	1	6

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

Temos duas retas: a que passa na linha Equipe2 e a que passa na coluna ProjetoA. Como o número de retas é MENOR que a ordem, segue-se para o passo 9.

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

<table>
<tr><th></th><th>Projeto A</th><th>Projeto B</th><th>Projeto C</th></tr>
<tr><td>Equipe 1</td><td>0 |</td><td>4</td><td>5</td></tr>
<tr><td>Equipe 2</td><td>---------- 0 |----------</td><td>---------- 0 ----------</td><td>---------- 0 ----------</td></tr>
<tr><td>Equipe 3</td><td>0 |</td><td>>>1<<</td><td>6</td></tr>
</table>

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

<table>
<tr><th></th><th>Projeto A</th><th>Projeto B</th><th>Projeto C</th></tr>
<tr><td>Equipe 1</td><td>0 |</td><td>3</td><td>4</td></tr>
<tr><td>Equipe 2</td><td>---------- 0 |----------</td><td>---------- 0 ----------</td><td>---------- 0 ----------</td></tr>
<tr><td>Equipe 3</td><td>0 |</td><td>0</td><td>5</td></tr>
</table>

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

<table>
<tr><th></th><th>Projeto A</th><th>Projeto B</th><th>Projeto C</th></tr>
<tr><td>Equipe 1</td><td>0 |</td><td>3</td><td>4</td></tr>
<tr><td>Equipe 2</td><td>---------- 1 |----------</td><td>---------- 0 ----------</td><td>---------- 0 ----------</td></tr>
<tr><td>Equipe 3</td><td>0 |</td><td>0</td><td>5</td></tr>
</table>

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

<table>
<tr><th></th><th>Projeto A</th><th>Projeto B</th><th>Projeto C</th></tr>
<tr><td>Equipe 1</td><td>0 |</td><td>3 |</td><td>4</td></tr>
<tr><td>Equipe 2</td><td>-----------1 |-----------</td><td>-----------0 |-----------</td><td>------------0------------</td></tr>
<tr><td>Equipe 3</td><td>0 |</td><td>0 |</td><td>5</td></tr>
</table>

Como o número de retas é igual à ordem da matriz, segue-se para o passo 13.

Passo 13: Determina-se a atribuição

Localize a primeira linha ou coluna que aparece apenas UM zero e faça a atribuição. Neste caso: A linha Equipe1 tem apenas UM zero (ProjetoA). Assim, o Projeto A será atribuído à Equipe 1, e essa linha e coluna podem ser eliminadas:

	Projeto A	Projeto B	Projeto C
Equipe 1	0	3	4
Equipe 2	1	0	0
Equipe 3	0	0	5

	Projeto B	Projeto C
Equipe 2	0	0
Equipe 3	0	5

Localize a próxima linha ou coluna que aparece apenas UM zero e faça a atribuição. Neste exemplo: A linha Equipe3 tem apenas UM zero (coluna ProjetoB). Assim, o Projeto B será atribuído à Equipe 3, e essa linha e coluna podem ser eliminadas:

	Projeto B	Projeto C
Equipe 2	0	0
Equipe 3	0	5

	Projeto C
Equipe 2	0

Finalmente, o Projeto C será atribuído à Equipe 2. Solução:

PA => E1, tempo 15 meses
PB => E3, custo 29 meses
PC => E2, custo 18 meses

Custo final total: 62 meses.

Bibliografia

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