UNIVERSIDADE FEDERAL DO PARANÁ

Discente: Yuri Junqueira Tobias - GRR20211767

Docente: André Guedes

Disciplina: Otimização - CI1238 **Data de entrega:** 28/05/2023

Modelagem e implementação, por programação linear, de uma solução para um problema em específico.

1. Introdução

O presente trabalho consiste na modelagem e implementação, por programação linear, de uma solução para o seguinte problema (Produção de Produtos Químicos): uma empresa química produz **n** tipos de produtos e, para produzir os produtos, utiliza diferentes porções de **m** diferentes compostos. Cada produto **i** tem um valor de venda (por litro), **v_i**. Cada composto **j** tem um preço (por litro), **p_j** e um limite diário de volume **q_j**. A quantidade de uso do composto **j** utilizado na produção do produto **i** é dada por **c_ij**. A empresa assume que toda sua produção será vendida e, com isso, deseja maximizar seu lucro.

Vale ressaltar também que[1] um problema de programação linear consiste em maximizar uma dada função linear dentre um conjunto de todos os vetores que satisfazem um determinado sistema de equações lineares e desigualdades. Além disso, cada programa linear pode ser facilmente transformado para a forma:

maximize cTx sujeito a Ax ≤ b

Sendo cTx a função linear a ser maximizada, também conhecida como **função objetivo;** Ax a matriz que representa o lado esquerdo do sistema de equações e b o lado direito. Ou seja, c é um vetor com os coeficientes das variáveis presentes no vetor x, A é a matriz com os coeficientes das variáveis presentes também em x e, por fim, b é o vetor que contém o resultado das desigualdades para cada equação do sistema linear.

2. Modelagem

Levando em consideração que a empresa deseja maximizar seu lucro e o mesmo varia de acordo com o produto que está sendo vendido, então a ideia da modelagem que será abordada no presente trabalho, seguindo a forma apresentada acima (introdução), consiste em:

- c: lucro por produto;

- x: vetor contendo as variáveis que formam uma bijeção para os produtos;
- A: matriz contendo os coeficientes das restrições, tais coeficientes representam a quantidade de uso de determinado composto em determinado produto;
- b: vetor contendo o limite de uso diário dos compostos;

A partir disso, é possível observar que iremos chegar ao seguinte modelo:

- Max $\sum$i=1^n $\sum$j=1^m (v_i - p_j * c_ij) * x_i

  subject to $\sum$i=1^n (c_ij * x_i) ≤ q_j   para todo j ∈ [1..m]

  n, m, i, j ∈ Z^+

  v_i, p_j, q_j, c_ij ∈ R^+

Com isso, podemos chegar à seguinte conclusão: iremos maximizar o somatório do lucro que cada produto dá, ou seja, o valor de venda do produto (v_i) menos o custo de produzir aquele produto (p_j * c_ij para todo j ∈ [1..m]), vezes a quantidade do produto (x_i). Ademais, a única restrição que temos é a de limitar o quantidade de uso diário dos compostos, e daí temos: todo composto está sujeito ao somatório da sua quantidade de uso em determinado produto (c_ij) vezes a quantidade daquele produto (x_i) para todos os produtos, sendo o limite do composto dado por q_j (valores presentes em b). Neste trabalho assume-se que n e m (quantidade de produtos e compostos) são inteiros positivos e, os demais valores pertencem aos reais positivos.

3. Implementação

O objetivo principal da implementação consiste em gerar uma saída para ser usada pelo *lp_solve* a partir de uma entrada que segue determinado padrão - que será apresentado a seguir. Os formatos de entrada e saída utilizam a entrada e a saída padrões (stdin e stdout), e a entrada pode conter números separados por 1 ou mais espaços, tabs ou fim de linha, tudo conforme solicitado.

A entrada deve seguir o seguinte padrão:

- Inicia com dois números inteiros **n** e **m** indicando a quantidade de produtos e compostos, respectivamente;
- Em seguida temos **n** números indicando os preços de venda (de 1 litro, v_i) de cada produto;
- Após temos **m** linhas, com 2 números em cada, indicando o custo (de 1 litro, p_j) e o limite de produção (q_j) de cada composto usado como matéria prima;
- Logo após temos os **n × m** números indicando a quantidade de cada composto usada na produção de 1 litro de cada produto. As quantidades de cada composto

para a produção do primeiro produto (c11, c12, . . . , c1m), estão na primeira linha, depois as quantidades para o produto 2, e assim até o produto n.

O trabalho está dividido em dois arquivos: producao.c e makefile; e um diretório: inputs. Sendo que o primeiro contém as funções auxiliares e o programa principal para receber a entrada, tratar e gerar a saída para o *lp_solve*. O segundo é apenas para compilar o primeiro e gerar o executável *producao* conforme solicitado. E o diretório vem com alguns exemplos - contendo, inclusive, o exemplo presente no enunciado e que será apresentado a seguir -, e serve apenas para facilitar na hora de gerar entradas, uma vez que basta o usuário redirecionar um dos arquivos ".txt" para o input de *producao*.

O arquivo producao.c contém, além da função principal(main), 1 estrutura (struct) e 4 funções auxiliares:

- **Struct**: contém todos os valores/atributos necessários para a produção e contém a descrição de cada um no próprio código;
- **leEntrada(...)**: lê os valores da entrada padrão (stdin) e faz as respectivas atribuições e alocações de memória;
- **calculaLucros(...)**: calcula quanto cada produto dá de lucro e armazena os valores no vetor de lucros presente na estrutura(struct);
- **imprimeLPSolve(...)**: gera a saída no formato de entrada para o *lp_solve*;
- **deletaProducao(...)**: libera o espaço de memória alocado para os atributos da estrutura (struct);
- **main()**: instancia uma estrutura (struct) e chama todas as funções auxiliares na sequência apresentada acima;
  Obs: todas as funções auxiliares recebem apenas a estrutura (struct);

4. Exemplos

4.1 - Todos os produtos dão lucros diferentes

Considere uma entrada com n = 3 produtos e m = 4 compostos, da seguinte forma:

		compostos				
	n/m	1	2	3	4	valor
pro	1	0,2	0,5	1,0	0,1	10
du	2	1,0	0,1	0,3	0,1	7
tos	3	0,4	0,2	0,2	0,0	3
	custo	1	2	5	10	
	limite	1000	2000	500	2000	

Figura 1: tabela de exemplo presente no enunciado do trabalho

A mesma surge a partir da seguinte entrada (padrão do arquivo input1.txt):

- 3 4
  10 7 3
  1 1000
  2 2000
  5 500
  10 2000
  0.2 0.5 1.0 0.1
  1.0 0.1 0.3 0.1
  0.4 0.2 0.2 0.0

O programa producao.c para esta instância irá gerar a seguinte saída:

- max: + 2.80x1 + 3.30x2 + 1.20x3 ;

  + 0.20x1 + 1.00x2 + 0.40x3 <= 1000.00 ;
  + 0.50x1 + 0.10x2 + 0.20x3 <= 2000.00 ;
  + 1.00x1 + 0.30x2 + 0.20x3 <= 500.00 ;
  + 0.10x1 + 0.10x2 + 0.00x3 <= 2000.00 ;

Quando aplicarmos a saída acima como entrada do lp_solve, o mesmo irá gerar a seguinte saída:

- Value of objective function: 3755.31914894

  Actual values of the variables:
  x1 212.766
  x2 957.447
  x3 0

Para este exemplo um plano ótimo tem ganho de R$ 3.755,31, onde são produzidos 212,766 litros do produto 1, 957,447 litros do produto 2 e 0 litros do produto 3 (de acordo com o enunciado). Como podemos perceber, a saída do *lp_solve* para a saída do executável *producao* corresponde ao resultado esperado.

4.2 - Caso em que nem todos os produtos dão lucro e os que dão, dão o mesmo lucro, porém possuem custo de produção diferente

Considere uma entrada com n = 3 produtos e m = 4 compostos, da seguinte forma:

		Compostos				
	n/m	1	2	3	4	valor
Produtos	1	0,5	0,5	0,5	0,5	3
	2	0,5	0,5	0,5	0,5	6
	3	1	0,5	1	0,5	8
	custo	2	2	2	2	
	limite	200	200	200	200	

Figura 2: tabela com as entradas pro caso 2

O programa producao.c para esta instância irá gerar a seguinte saída:

- max: + -1.00x1 + 2.00x2 + 2.00x3 ;

  + 0.50x1 + 0.50x2 + 1.00x3 <= 200.00 ;
  + 0.50x1 + 0.50x2 + 0.50x3 <= 200.00 ;
  + 0.50x1 + 0.50x2 + 1.00x3 <= 200.00 ;
  + 0.50x1 + 0.50x2 + 0.50x3 <= 200.00 ;

Quando aplicarmos a saída acima como entrada do lp_solve, o mesmo irá gerar a seguinte saída:

- Value of objective function: 800.00000000

  Actual values of the variables:
  x1 0
  x2 400
  x3 0

Para este exemplo um plano ótimo tem ganho de R$ 800,00, onde são produzidos 0 litros do produto 1, 400 litros do produto 2 e 0 litros do produto 3. Como podemos perceber, uma vez que o primeiro produto dá prejuízo e o último, apesar de gerar o mesmo lucro que o segundo, gasta mais compostos em sua produção, a solução ótima acaba por utilizar somente do composto 2, que é o que dá lucro e gasta menos compostos em sua produção (aresta mais inclinada).

4.3 - Caso em que ninguém dá lucro

Considere uma entrada com n = 3 produtos e m = 4 compostos, da seguinte forma:

		Compostos				
	n/m	1	2	3	4	valor
Produtos	1	0,5	1	0,4	0,2	5
	2	1	0,3	0,3	0,5	4
	3	1	0,5	0,1	0,6	4
	custo	1	3	2	6	
	limite	2000	1500	1000	4000	

Figura 3: tabela com as entradas pro caso 3

O programa producao.c para esta instância irá gerar a seguinte saída:

- max: + -0.50x1 + -1.50x2 + -2.30x3 ;

  + 0.50x1 + 1.00x2 + 1.00x3 <= 2000.00 ;
  + 1.00x1 + 0.30x2 + 0.50x3 <= 1500.00 ;
  + 0.40x1 + 0.30x2 + 0.10x3 <= 1000.00 ;
  + 0.20x1 + 0.50x2 + 0.60x3 <= 4000.00 ;

Quando aplicarmos a saída acima como entrada do lp_solve, o mesmo irá gerar a seguinte saída:

- Value of objective function: 0

  Actual values of the variables:
  x1 0
  x2 0
  x3 0

Para este exemplo um plano ótimo tem ganho de R$ 0, onde são produzidos 0 litros de todos os produtos. Como podemos perceber, uma vez que nenhum produto dá lucro, muito pelo contrário, todos dão prejuízo, então o melhor caso é aquele em que você não vende nenhum produto.

4.4 - Caso em que todos os produtos dão o mesmo lucro e gastam a mesma quantidade de compostos

Considere uma entrada com n = 3 produtos e m = 4 compostos, da seguinte forma:

		Compostos				
	n/m	1	2	3	4	valor
Produtos	1	0,5	0,5	0,5	0,5	5
	2	0,5	0,5	0,5	0,5	5
	3	0,5	0,5	0,5	0,5	5
	custo	2	2	2	2	
	limite	2000	1500	1000	4000	

Figura 4: tabela com as entradas pro caso 4

O programa producao.c para esta instância irá gerar a seguinte saída:

- max: + 1.00x1 + 1.00x2 + 1.00x3 ;

  + 0.50x1 + 0.50x2 + 0.50x3 <= 2000.00 ;
  + 0.50x1 + 0.50x2 + 0.50x3 <= 1500.00 ;
  + 0.50x1 + 0.50x2 + 0.50x3 <= 1000.00 ;
  + 0.50x1 + 0.50x2 + 0.50x3 <= 4000.00 ;

Quando aplicarmos a saída acima como entrada do lp_solve, o mesmo irá gerar a seguinte saída:

- Value of objective function: 2000.00000000

  Actual values of the variables:
  x1 2000
  x2 0
  x3 0

Para este exemplo um plano ótimo tem ganho de R$ 2000,00, onde são produzidos 2000 litros do produto 1 e 0 litros dos produtos 2 e 3. Como podemos perceber, uma vez que todos os produtos possuem o mesmo valor e utilizam a mesma quantidade de todos os compostos em sua produção, então o melhor caso acaba sendo vender apenas o máximo possível de um(qualquer) dos produtos.

4.5 - Caso em que um dos produtos gasta apenas o composto mais custoso e com maior limite para produção

Considere uma entrada com n = 3 produtos e m = 4 compostos, da seguinte forma:

		Compostos				
	n/m	1	2	3	4	valor
Produtos	1	0	0	0	1	9
	2	0,5	0,5	0,5	0,5	12
	3	0,2	0,5	0,1	1	15
	custo	2	4	6	7	
	limite	2000	1500	1000	4000	

Figura 2: tabela com as entradas pro caso 5

O programa producao.c para esta instância irá gerar a seguinte saída:

- max: + 2.00x1 + 2.50x2 + 5.00x3 ;

  + 0.00x1 + 0.50x2 + 0.20x3 <= 2000.00 ;
  + 0.00x1 + 0.50x2 + 0.50x3 <= 1500.00 ;
  + 0.00x1 + 0.50x2 + 0.10x3 <= 1000.00 ;
  + 1.00x1 + 0.50x2 + 1.00x3 <= 4000.00 ;

Quando aplicarmos a saída acima como entrada do lp_solve, o mesmo irá gerar a seguinte saída:

- Value of objective function: 17000.00000000

  Actual values of the variables:
  x1 1000
  x2 0
  x3 3000

Para este exemplo um plano ótimo tem ganho de R$ 17.000,00, onde são produzidos 1000 litros do produto 1, 0 litros do produto 2 e 3000 litros do produto 3. Como podemos perceber, uma vez que o produto 1 utiliza apenas o último composto em sua produção e o mesmo possui o maior limite, então o melhor caso é produzir o máximo do produto que dá mais lucro dentre os produtos restantes (o 3 nesse caso) e, por fim, produzir o máximo do produto 1.

5. Referências:

[1]: Understanding and Using Linear Programming. Jiří Matoušek, Bernd Gärtner. 2007.