

T1 - Programação Linear

Eduardo Stefanel Paludo (GRR20210581)
Vinícius Oliveira dos Santos (GRR20182592)
Universidade Federal do Paraná (UFPR)
Curitiba, Brasil

I. INTRODUÇÃO

O trabalho desenvolvido consiste em ler dados para o problema de programação linear dos compostos químicos e, com esses dados, gerar uma saída adequada para ser utilizada pelo `lp_solve`. Para isso, foram escritos dois arquivos na linguagem C: o programa principal `main.c`, o arquivo `producao.c` contendo a função que formata os dados, o arquivo de *header* `producao.h` e um `makefile`. O programa foi compilado através do comando `make`, com os parâmetros `-Wall` e `-std=c99`.

II. MODELAGEM

Para modelar o problema, foi pensado primeiro quais seriam as variáveis de decisão. Como o objetivo é maximizar o lucro da empresa por meio da venda de produtos, é muito claro que as variáveis de decisão são as quantidades x_i produzidas de cada produto i . O coeficiente de cada variável é o lucro obtido para cada produto, isto é, o valor de venda v_i menos os custos de produção p_j de cada composto j vezes a quantidade c_{ij} presente no produto. As restrições do problema são os limites q_j de cada composto. Para estabelecer a quantidade que foi produzida de cada composto, foi multiplicada a quantidade c_{ij} desse composto presente em cada produto i vezes a quantidade x_i produzida de cada produto. Além disso, todas as quantidades x_i devem ser não-negativas. Podemos escrever matematicamente esse problema da seguinte maneira:

$$\begin{aligned} \max \quad & \sum_{i=1}^n \left(v_i - \sum_{j=1}^m c_{ij} * p_j \right) * x_i \\ \text{s.t.} \quad & \sum_{i=1}^n c_{ij} * x_i \leq q_j \quad \text{para } j = 1, 2, \dots, m \\ & x_1, x_2, \dots, x_n \geq 0 \end{aligned}$$

III. PROGRAMA PRINCIPAL

Na função `main`, foram criadas duas variáveis que armazenam a quantidade n de produtos e a quantidade m de compostos. Foi criado, também, um vetor de *double* contendo os preços dos n produtos e uma matriz de *double* $m \times 2$, onde cada linha j contém o custo e o limite de cada composto j . Além disso, foi criada outra matriz $n \times m$, onde cada linha i representa um produto, cada coluna j representa um composto, e o valor $c_{i,j}$ é a quantidade do composto j no produto i .

Após os dados serem armazenados na memória, as variáveis criadas são passadas como parâmetro para a função `void lp_solve_format(int n, int m, double* precos, double** compostos, double** quantidades)`, que imprime na saída padrão o problema em um formato adequado para o `lp_solve`.

Esse formato foi gerado escrevendo a função objetivo seguida por cada uma das restrições. Cada equação do problema foi escrita em linhas separadas e separadas por ponto e vírgula.

IV. CONCLUSÃO

Após a conclusão dos algoritmos, foram executados testes com diferentes valores de entrada, como os da especificação do trabalho. Em todos os casos, foi possível obter a solução ótima por meio da execução do programa `lp_solve`, utilizando como entrada o arquivo gerado pelo programa.

REFERÊNCIAS

- [1] J. Matoušek and B. Gärtner, *Understanding and Using Linear Programming*. Springer, 2007.