Relatorio T1 - otimização

Thales Gabriel[1], Matheus Telles Batista[2]

,2[1]Departamento de Informática Universidade Federal do Paraná – UFPR Curitiba, Brasil

Keywords
Otimização, Simplex, Programação Linear,

1. Introdução do Problema

O problema apresentado é um no qual há uma empresa com uma fábrica em uma cidade (origem) e um depósito em outra cidade (destino). Esta empresa produz um tipo de produto vendido por peso em toneladas. Com a garantia de vender toda a produção, a empresa obterá um ganho fixo de p por tonelada transportada até o depósito, descontando os custos de produção. O objetivo é enviar o máximo possível desse produto para o depósito diariamente.

Para chegar ao destino, é necessário perpassar por várias cidades e a ida de uma cidade a outra demanda recursos, dependendo da quantidade de tonelada enviada por aquela rota.

Os recursos, são vendidos por pacotes, então não é possível comprá-los individualmente.

Por exemplo:

Se temos a seguinte situação:

$$\text{Cidades} = \{1, 2, 3, 4\};$$
$$\text{Rotas} = \{\{1, 2\}, \{1, 3\}, \{2, 3\}, \{2, 4\}, \{3, 4\}\}$$

	Pacote 1	**Pacote 2**
recurso 1	4	5
recurso 2	2	2
recurso 3	0	1
Preço do Pacote	10	20

Table 1
Quantidade de recursos e preço dos pacotes

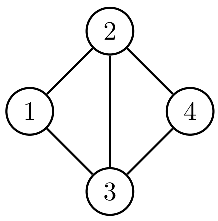

Figure 1: Grafo com cada vértice representando uma cidade e cada aresta representando uma possível rota

Como já dito na apresentação do trabalho:

"Para este exemplo um plano ótimo tem ganho de 150, onde são comprados 5 pacotes do tipo 1 e passando 2 toneladas, de 1 para 3, de 3 para 2 e de 2 para 4 e 0 nas outras rotas."

Otimização
mtb21@inf.ufpr.br (M. T. Batista)
https://github.com/MTlls (M. T. Batista)

2. Desenvolvimento

2.1. Problema do Fluxo

É possível perceber que o problema além de envolver achar uma rota ótima com o máximo de toneladas enviadas, também inclui a minimização do custo ao comprar os recursos.

Houve uma maior dificuldade na modelagem/generalização do problema do fluxo, que já foi apresentado no Livro "*Understanding and Using Linear Programming*" na seção "*2.2 Flow in a Network*"

Entender que em uma cidade x, as rotas $\{a, x\}$ $\{x, b\}$ são respectivamente, caminhos no qual x **recebe** e **envia** as cargas foi crucial para a realização de um fluxo íntegro de toneladas. O que justifica na modelagem do problema a seguinte linha no PL:

$$\sum_{e_i=\{a,v\}\in E} x_{e_i} - \sum_{e_j=\{v,b\}\in E} x_{e_j} = 0 \quad \text{para } v = 2, \ldots, n-1$$

No código, quem é responsável por isso é a função `void rotas(vector<vector<int>> m)`

2.2. Problema da Dieta

A modelagem do problema necessitou de várias variáveis, principalmente pela necessidade de separar os recursos necessários por rota, por pacote, por pacote comprado e por todas as rotas (se não foi nomeado algum, desculpe). A fim de manter uma integridade entre o que os recursos comprados e os que serão usufruídos nas rotas (esse problema em específico é resolvido nas funções:

```
void recurso_necessario(vector<vector<int>> m, int qtd_recursos);
void recurso_comprado (vector<vector<int>> m, int qtd_recursos);
void recurso_usado(int qtd_recursos);
```

O Livro "*Understanding and Using Linear Programming*" apresenta um problema que se assemelha bastante com este "balanceamento" de recursos de forma ótima (seção *2.1 Optimized Diet: Wholesome and Cheap?*).

O que foi necessário foi explicitar **todas** as restrições de compra, tal que os recursos estejam disponíveis para que haja um caminho até a cidade destino (cidade n).

2.3. Problema do Módulo

Há um momento em que é necessário trabalhar com módulos, mais especificamente no momento em que a variável $x_{e_i} \leq 0$, o que é totalmente possível visto que pode ocorrer casos onde a rota $\{a, b\}$ vai no sentido contrário (de b para a). Para isso, usamos como base o problema da seção "*2.4 Fitting a Line*" (o livro você já sabe) no qual é necessário em algum momento computar o erro entre dois pontos, ou seja, $|(x_i, y_i) - (x_j, y_j)|$.

Basicamente a solução que temos no problema do livro é:

$$\begin{aligned} \mathrm{e}_i &\geq f(x_i) - y_i \\ \mathrm{e}_i &\geq -(f(x_i) - y_i) \end{aligned}$$

Pode-se trazer isso para o problema atual, pois é verdade que o módulo de um número sempre será maior ou igual a ele mesmo e seu oposto. Em termos matemáticos:

$$\forall x, \ |x| \geq -x \quad \text{e} \quad |x| \geq x$$

A função responsável por isso é `void modulo_rota(vector<vector<int>> m)`

3. Modelagem do Problema

O problema pode ser formulado da seguinte maneira:

maximize: $pW - C$

sujeito a:

$$
\begin{aligned}
0 &= \sum_{e_i=\{a,v\}\in E} x_{e_i} - \sum_{e_j=\{v,b\}\in E} x_{e_j} \quad \text{para } v = 2, ..., n-1 \\
W &= \sum_{e_i=\{1,a\}\in E} |x_{e_i}| \\
C &= \sum_{i=1}^{q} v_i \cdot y_i \\
s_j &= \sum_{i=1}^{q} y_i \cdot d_{j,i} \quad , \forall 1 \leq j \leq l \\
t_j &= \sum_{e_i\in E} r_{j,e_i} \cdot |x_{e_i}| \, , \forall 1 \leq j \leq l \\
s_j &= \sum_{i=1}^{q} d_{j,i} \cdot y_i \quad , \forall 1 \leq j \leq l \\
t_i &\leq s_i \quad , \forall 1 \leq i \leq l \\
c_{e_i} &\leq x_{e_i} \leq c_{e_i} \\
y_i &\geq 0 \quad , \forall 1 \leq j \leq l
\end{aligned}
$$

Onde:

W é a quantidade de toneladas do produto enviadas para o depósito (destino);
C é o custo total dos pacotes de recursos comprados;
p é o ganho por tonelada de produto transportada até o depósito;
c_{e_i} é a capacidade de carga diária da rota e_i;
r_{l,e_i} é a quantidade de recursos do tipo l necessários por tonelada na rota e_i;
t_l é a quantidade de recursos do tipo l necessários por tonelada em E;
y_q é a quantidade de pacotes comprados do tipo q;
v_q é quanto custa o pacote q;
s_l é a quantidade recursos do tipo l comprados;
$d_{l,q}$ é a quantidade recursos do tipo l disponíveis no pacote q;
x_{e_i} é a quantidade de toneladas de produto transportadas na rota e_i;
e_i é uma rota, que representada um par $\{a, b\}$, sendo a ou b a origem ou o destino com $1 \leq i \leq m$, sendo m o número de cidades;
E é o conjunto que contém todas as rotas;