Primeiro Trabalho Prático (CI1238)

Caio Henrique Ramos Rufino e Frank Wolff Hannemann

8 de abril de 2024

1 Introdução

Este relatório irá demonstrar uma implementação da modelagem do problema "Transporte de carga com pacotes de recursos"[1].

2 Modelagem

2.1 Função Objetiva

Para modelar a função objetiva, tomamos como base a função $pW - C$, que indica diretamente o lucro, e adaptamos de uma maneira que a quantidade de toneladas entregues e o custo fossem subsituídos pelas variáveis que seriam utilizadas nas restrições. Dessa forma, chegamos à seguinte função objetiva:

$$\sum_{j=2}^{n} p * r_{1,j}[se\ r_{1,j}\ existe] - \sum_{u=1}^{q} qtd\_pacote_u * v_u$$

Onde n é o número de cidades, q o número de pacotes, o primeiro somatório representa a soma de todas as rotas que tem como origem a primeira cidade, e o segundo, a soma de todos os pacotes multiplicados pelos seus respectivos custos.

2.2 Lidando com o módulo

Considerando que as rotas são bidirecionais, o fluxo de cada uma delas poderia assumir tanto valores positivos quanto negativos (por convenção, todas as rotas são tratadas como positivas na direção em que são lidas na entrada inicial). Entretanto, o fluxo não pode ser negativo, mas usar a função de módulo para lidar com essa questão faria com que tivéssemos restrições não lineares. Então, introduzimos duas variáveis não negativas $y_{i,j}$ e $z_{i,j}$, que representam, respectivamente, a parte positiva e a "negativa" do fluxo de cada rota $r_{i,j}$. Essa ideia foi baseada na resolução do problema *2.3 Ice Cream All Year Round*[2].

2.3 Restrições dos Pacotes

As primeiras restrições doos pacotes dizem respeito ao fato de que o número de pacotes não pode ser negativo, então, para cada pacote q_u:

$$q_u \geq 0$$

Além disso, é necessário considerar a quantidade de cada recurso l que é gasto ao passar por cada uma das rotas. Assim, para uma quantidade k de recursos, teremos:

$$\sum_{l=1}^{k}(\sum_{e=1}^{m}(y_e + z_e) * r_{el} \ \leq \sum_{u=1}^{q} qtd\_pacote_u * qtd\_recurso_l \ )$$

O primeiro somatório representa os k recursos, o segundo representa o fluxo de cada rota e multiplicado pela quantidade do recurso r_{el} que será gasto ao passar pela respectiva rota e. Já o somatório do lado direito da desigualdade representa a quantidade de cada pacote u que deve ser comprado, multiplicado pela quantidade do recurso l que está naquele pacote.

2.4 Restrições das Rotas

Primeiramente, o fluxo de cada rota, representado por $f_{i,j}$, foi limitado por sua respectiva capacidade $c_{i,j}$, então, para cada fluxo, temos:

$$f_{i,j} \leq c_{i,j}$$
$$f_{i,j} \geq -c_{i,j}$$

De maneira análoga, as variáveis $y_{i,j}$ e $z_{i,j}$ também foram limitadas pelas capacidades de suas rotas, mas, como ambas são variáveis positivas, elas só estão limitadas positivamente:

$$y_{i,j} \leq c_{i,j}$$
$$y_{i,j} \geq 0$$
$$z_{i,j} \leq c_{i,j}$$
$$z_{i,j} \geq 0$$

Considerando que nada do produto a ser entregue pode ser guardado nas cidades, tudo o que entrar em uma cidade deve também sair. Podemos representar isso da seguinte forma, considerando uma cidade $a \in n = \{2, ..., n-1\}$:

$$\sum_{i=1}^{n} f_{i,a} \ = \sum_{j=1}^{n} f_{a,j}$$

O primeiro somatório representa a soma de todo o fluxo que chega em a, e o segundo de todo o fluxo que sai de a. Haverá uma equação dessa para cada cidade n, exceto para a primeira e a última, já que nehum fluxo chega na primeira, e nem sai da última.

Por fim, devemos igualar os fluxos $f_{i,j}$ às variáveis adicionais criadas para lidar com o módulo ($y_{i,j}$ e $z_{i,j}$):

$$f_{i,j} = y_{i,j} - z_{i,j}$$

Como $y_{i,j}$ representa a parte positiva, e $z_{i,j}$ a parte negativa, é necessário que ocorra a subtração para obtermos o valor de $f_{i,j}$ fora do módulo, assim, seu sinal representará o sentido da rota.

3 Programa em C

3.1 Estrutura do código

O código está dividido nos diretórios `src`, que contém os arquivos `.c`, e `include` contendo os arquivos `.h`.

3.2 Implementação da modelagem

Para auxiliar na implementação da modelagem, foram usadas duas estruturas de dados: "Pacote" para representar os pacotes e "Rota" para representar as rotas.

A estrutura "Pacote" é composta por um inteiro que representa o custo do pacote e por um vetor que representa a quantidade de cada recurso que aquele pacote possui.

Similar à estrutura "Pacote", a estrutura "Rota" possui um vetor que representa a quantidade de cada recurso que aquela rota exige. Além disso, também é composta por três inteiros, cada um representando o inicio, fim e capacidade da rota, respectivamente.

3.3 Modo de uso

Para compilar o programa, basta rodar o comando `make` no diretório principal, que irá gerar o arquivo executável `transporte`. O programa recebe a entrada na formatação especificada pelo trabalho[1] pela entrada STDIN e imprime a saída em STDOUT.

4 Exemplos

Exemplo 1

O primeiro exemplo foi tirado da própria especificação do trabalho, a seguir estão as informações de entrada: Considere $n = 4$ cidades, $m = 5$ rotas, $k = 3$ recursos, $q = 2$ pacotes e $p = 100$ de ganho. As rotas e os pacotes são descritos na tabela abaixo:

	rotas					pacotes	
recursos	1,2	1,3	2,3	2,4	3,4	1	2
1	1	2	1	2	3	4	5
2	1	2	2	1	4	2	2
3	5	0	0	0	6	0	1
	5	2	5	2	5	10	20
	capacidades					custos	

Considerando o exemplo fornecido na descrição do trabalho[1] como referência, obtivemos um resultado que apresenta um 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, conforme esperado.

Exemplo 2

O segundo exemplo é fictício e representa um caso aleatório e serve para testar se a modelagem está sendo feita corretamente. Considere n = 7 cidades, m = 10 rotas, k = 3 recursos, q = 2 pacotes e p = 300 de ganho. As rotas e os pacotes são descritos na tabela abaixo:

	rotas										pacotes	
recursos	1, 2	1, 3	1, 4	2, 3	2, 5	3, 6	4, 5	4, 6	5, 7	6, 7	1	2
1	1	3	0	1	0	1	2	1	2	0	4	5
2	1	2	1	5	2	2	1	1	1	1	2	2
3	5	0	4	0	0	2	2	1	0	0	0	1
	3	1	1	1	1	3	4	4	4	1	10	20
	capacidades										custos	

Para esse exemplo de entradas, obtivemos os seguintes valores: 740 de ganho, onde são comprados 0 pacotes do tipo 1 e 23 pacotes do tipo 2 e passando 1 tonelada de 1 para 3, 1 para 4, 2 para 3, 2 para 5, 6 para 4 e 6 para 7, 2 toneladas de 1 para 2, de 3 para 6 e de 4 para 5 e 3 toneladas de 5 para 7.

Referências

1 GUEDES, A. **Primeiro Trabalho Prático**. [s.n.], 2024. Disponível em: <https://www.inf.ufpr.br/andre/Disciplinas/CI1238-2024-1/trabalho1.pdf>.

2 MATOUŠEK, J.; GÄRTNER, B. **Understanding and Using Linear Programming**. [S.l.]: Springer, 2007.