Universidade de São Paulo
Escola Politécnica

PNV5761 - Programação Matemática Aplicada a Problemas de Transporte

Prova 1

Prof. Marco Antonio Brinati

Guilherme Fernandes Alves - 10774361

São Paulo, Agosto de 2024

1 Questão 1

- N: número total de semanas
- n_t: número de páginas disponíveis na semana $t \in \{1, \dots, N\}$
- m: número de clientes ou pedidos
- i_k: semanal inicial para veiculação do pedido k
- d_k: duração do pedido k
- a_k: tamanho do pedido k
- p_k: preço do pedido k
- K_1: grupo de pedidos imutáveis
- $\bar{K}_1$: grupo de pedidos que podem sofrer algum atraso, porém sob penalidade

Embora o enunciado não deixe claro, estamos assumindo que a união dos conjuntos K_1 e $\bar{K}_1$ é igual ao conjunto inicial K. Ademais, não há intersecção entre os conjuntos K_1 e $\bar{K}_1$. Podemos formalizar através da equação 1.

$$K_1 \cup \bar{K}_1 = K \quad e \quad K_1 \cap \bar{K}_1 = \emptyset \tag{1}$$

A equação 1 é importante para garantir que não hajam pedidos pertencentes aos dois subconjuntos, pois se um pedido é mutável, ele não poderá ser imutável; bem como para garantir que não hajam pedidos que estejam fora dos dois subconjuntos.

1.1 Variáveis de decisão

Vamos propor uma variável de decisão para o subconjunto K_1 e duas outras variáveis para o conjunto $\bar{K}_1$.

Seja x_k uma variável binária que indica se o pedido $k \in K_1$ foi aceito ou não.
Seja $\bar{x}_k$ uma variável binária que indica se o pedido $k \in \bar{K}_1$ foi aceita, sem atraso, ou não.
Seja $\bar{y}_k$ uma variável binária que indica se o pedido $k \in \bar{y}_1$ foi aceita, com atraso, ou não.

1.2 Restrições

1.2.1 Cada pedido pode ser aceito somente uma única vez

Como introduzimos duas variáveis de decisão para o subconjunto $\bar{K}_1$, precisamos de alguma forma compatibilizar essas duas variáveis, assim:

$$\bar{x}_k + \bar{y}_k \leq 1; \quad \forall k \in \bar{K}_1 \tag{2}$$

Por outro lado, para o conjunto imutável K_1, como x_k e uma variável binária, já estamos garantindo que cada pedido será aceito no máximo uma única vez.

1.2.2 Duração

Para o caso sem atrasos, queremos que a semana inicial somada a duração não ultrapasse o limite de semanas disponíveis. Assim:

$$i_k + d_k \cdot x_t \leq N; \quad \forall k \in K_1 \tag{3}$$

Por outro lado, para os pedidos do subconjunto $\bar{K}_1$, teremos que considerar os casos de que o pedido pode ser aceito com atraso ou não, assim temos:

Para o caso com atraso:

$$y_k \cdot d_k \leq N - i_k - 1 \tag{4}$$

Para o caso sem atraso:

$$\bar{x}_k \cdot d_k \leq N - i_{ik} \tag{5}$$

Para o caso genérico: Vamos juntar as equações 4 e 5 em uma só:

$$d_k \cdot (\bar{x}_k + y_k) \leq N - i_k - y_k; \quad \forall k \in \bar{K}_1 \tag{6}$$

1.2.3 Capacidade

Vamos introduzir variáveis intermediarias:

- b_{kt}: variável binária que indica se o pedido k, que não teve atraso, esta ativo na semana t
- c_{kt}: variável binária que indica se o pedido k, que teve algum atraso, esta ativo na semana t

Em cada semana, não podemos ultrapassar o limite de numero de páginas n_t disponíveis.

$$\sum_{k \in K_1} a_k \cdot b_{kt} \cdot x_k + \sum_{k \in \bar{K}_1} \bar{x}_k \cdot a_k \cdot b_{kt} + \sum_{k \in \bar{K}_1} \bar{y}_k \cdot a_k \cdot c_{kt} \leq n_t; \quad \forall t \in \{1, \ldots, N\} \tag{7}$$

1.3 Função objetivo

Queremos maximizar a receita auferida pela empresa através dos preços cobrados por cada pedido. Assim:

$$max \left[\sum_{k \in K_1} p_k \cdot x_k + \sum_{k \in \bar{K}_1} (p_k \cdot \bar{x}_k + 95\% \cdot p_k \cdot \bar{y}_k) \right] \tag{8}$$

Note que sofremos uma penalização de 5% do preço caso haja algum atraso.

1.4 Espaço das variáveis

Todas as nossas variáveis de decisão são binárias, mas precisamos defini-las com respeito aos diferentes índices.

- $x_k \in \{0, 1\}; \quad \forall k \in K_1$
- $\bar{x} \in \{0, 1\}; \quad \forall k \in \bar{K}_1$
- $\bar{y} \in \{0, 1\}; \quad \forall k \in \bar{K}_1$
- $b_{kt} \in \{0, 1\}; \quad \forall k \in K \quad \forall t \in \{1, \ldots, N\}$
- $c_{kt} \in \{0, 1\}; \quad \forall k \in \bar{K}_1 \quad \forall t \in \{1, \ldots, N\}$

2 Questão 2

Problema inicial:

- n candidatos a postos de bombeiros
- m locais/zonas a serem construídas

No item c) da questão 9, foram instalados r ($r < n$) postos, de modo que somente algumas zonas foram cobertas duplamente, porem todas as zonas foram cobertas ao menos uma vez.

Seja δ o numero de zonas cobertas duplamente com a solução do item c), dado que δ já é conhecido e fixo, com $0 < \delta < m$. Defini-se:

- $n' = n - r =$ numero de postos candidatos ainda disponíveis
- $m' = m - \delta =$ numero de zonas com apenas cobertura simples

O problema se resume ao item a) da questão 9, consistindo em cobrir m' zonas utilizando n' candidatos.

2.1 Variáveis de decisão

Seja x_i uma variável binária que indica se o local $i \in \{1, \ldots, n'\}$ foi escolhido para ser construído ou não.

2.2 Função objetivo

Queremos minimizar a quantidade de postos construídos, dado que o enunciado não fala sobre custos associados a cada posto.

$$\min \sum_{i=1}^{m'} x_i \tag{9}$$

2.3 Restrições

Precisamos garantir que todas as regiões m' sejam cobertas ao utilizamos os candidatos remanescentes. Repare que os postos que já foram construídos não serão incluídos nesta contagem, pois já não fazem mais parte do problema.

$$\sum_{i=1}^{n'} a_{ij} \cdot x_i \geq 1; \quad \forall j \in \{1, \ldots, m'\} \tag{10}$$

2.4 Espaço das variáveis

Definimos que a variável x_i é uma variável binária, podemos então formalizar que:

- $x_i \in \{0, 1\}; \quad \forall i \in \{1, \ldots, n'\}$