

# Trabalho de Otimização Inteira

Benício Januário, 12543843 (benicio.januario@usp.br)

Henrique Bovo, 12542539 (henrique.bovo@usp.br)

Henrico Lazuroz, 12543502 (henrico.lazuroz@usp.br)

Victor Fernandes, 12675399 (victorlafernandes@usp.br)

## Introdução

Todos os códigos desse trabalho foram desenvolvidos em um notebook do colab, que pode ser acessado clicando [aqui](#). Os softwares necessários para a realização do projeto, como os solvers *SCIP* e *Gurobi*, foram importados diretamente pelo arquivo python.

## Tarefa 1

Utilizamos o python, junto a biblioteca *PySCIPOpt*, para escrever o problema de localização de facilidades em linguagem de modelagem.

Definimos a variável  $y_i$  como binária, para representar a abertura ou não de uma facilidade, e a variável  $x_{ij}$  como contínua, correspondendo a fração da demanda do cliente  $j$  atendida pela facilidade  $i$ .

Também estabelecemos restrições para a demanda dos clientes e a capacidade das facilidades, assim como a função objetivo, referente a minimização do custo total.

## Tarefa 2

### 1. Explicação

Para facilitar a explicação dessa tarefa, vamos considerar um exemplo numérico.

- Facilidade 1
  - $y_1 = 1$ , facilidade aberta
  - Capacidade: 50
- Facilidade 2
  - $y_2 = 0$ , facilidade não aberta
  - Capacidade: 80
- Cliente 1
  - Demanda: 30
  - $x_{11} = 0.5$ , 50% de sua demanda atendida pela facilidade 1
- Cliente 2
  - Demanda: 20
  - $x_{12} = 1$ , 100% de sua demanda atendida pela facilidade 1
- Cliente 3
  - Demanda: 40
  - $x_{23} = 0$ , 0% de sua demanda atendida pela facilidade 2

A restrição (3) serve para garantir que a capacidade das facilidades seja respeitada. Vamos então calculá-la para os valores do nosso exemplo.

O cliente 1 possui demanda 30, sendo metade dela direcionada à facilidade 1. Logo, consome 15 unidades da sua capacidade. Já o cliente 2 possui demanda 20, totalmente atendida pela facilidade 1. Assim, temos que da capacidade total de 50, apenas 35 é de fato utilizada.

Em contrapartida, o cliente 3 possui demanda 40, sendo que nenhuma parte dela é atendida pela facilidade 2, visto que ela não é aberta.

A abertura da facilidade é uma decisão binária, e  $y_i$  pode ser 0 ou 1. Podemos perceber que se esse valor é 0, a fração da demanda de um cliente atendida por essa facilidade também deve ser 0. Ou seja,  $y_i = 0$  implica que  $x_{ij} = 0$  para todos os valores de  $j$ . Caso  $y_i = 1$ , temos que o valor de  $x_{ij}$  será sempre menor que  $y_i$  para todos os  $j$ , pois representa uma porcentagem que varia de 0 até no máximo 1.

Com isso, podemos concluir que os valores de  $x_{ij}$  quando comparados com  $y_i$  são sempre iguais, no caso da facilidade não ser aberta, ou menores, se for aberta.

No entanto, ao relaxar o problema, a variável  $y_i$  deixa de ser binária, e cenários que contrariam essa conclusão passam a ser permitidos. Por exemplo, um caso hipotético em que  $y_i = 0.3$  e  $x_{ij} = 0.5$  se torna possível.

Dessa forma, a adição da restrição (11) garante que essa condição do problema seja mantida no problema relaxado, possibilitando a poda de diversos casos infactíveis e tornando a solução mais otimizada.

## 2. Solução modelo relaxado

Resolvendo o problema linearmente relaxado considerando o modelo (1) - (5) para as 5 instâncias disponibilizadas.

Resultados obtidos com o SCIP					
	Instância 1	Instância 2	Instância 3	Instância 4	Instância 5
Tempo (s)	19.98	36.61	81.13	270.21	309.65
Valor ótimo	69005.357	75840.634	114633.651	135069.023	0.0
Primal	69005.357	75840.634	114633.651	135069.023	1e+20
Dual	69005.357	75840.634	114633.651	135069.023	0.0
Gap	0.0	0.0	0.0	0.0	1e+20

## 3. Solução modelo relaxado alternativo

Nessa solução alternativa, adicionamos a restrição (11)  $x_{ij} \leq y_i$  e reescrevemos a restrição (3) como  $\sum_{j=1}^m d_j x_{ij} \leq Cap_i$ .

Resultados obtidos com o SCIP					
	Instância 1	Instância 2	Instância 3	Instância 4	Instância 5
Tempo (s)	304.65	301.62	301.89	300.47	Estouro de memória
Valor ótimo	0.0	0.0	0.0	0.0	
Primal	1e+20	1e+20	1e+20	1e+20	
Dual	0.0	0.0	0.0	0.0	
Gap	1e+20	1e+20	1e+20	1e+20	

## 4. Comparando os resultados

Podemos concluir que, na solução relaxada, o SCIP conseguiu obter a solução ótima no tempo determinado para as instâncias 1-4, e na instância 5 estourou o tempo de execução.

Em contrapartida, para a solução alternativa, o solver não conseguiu obter a solução ótima para nenhuma das instâncias no tempo determinado.

## Tarefa 3

Resultados obtidos ao resolver as instâncias utilizando o *SCIP* com tempo limite de 5 minutos:

Resultados obtidos com o SCIP					
	Instância 1	Instância 2	Instância 3	Instância 4	Instância 5
Tempo (s)	300.43	300.87	301.28	302.45	304.44
Valor ótimo	69335.527	75932.681	114899.854	140164.597	339838.053
Primal	69335.527	75932.681	114899.854	140164.597	339838.053
Dual	69042.858	75870.061	114676.192	135101.215	0.0
Gap	0.0042	0.0008	0.0019	0.0374	1e+20

## Tarefa 4

Resultados obtidos ao resolver as instâncias utilizando o *Gurobi* com tempo limite de 5 minutos:

Resultados obtidos com o Gurobi					
	Instância 1	Instância 2	Instância 3	Instância 4	Instância 5
Tempo (s)	8.15	300.10	285.03	300.27	300.36
Valor ótimo	69055.247	75932.681	114736.905	135143.455	162103.483
Primal	69048.364	75893.925	114726.509	135107.970	161954.007
Dual	69048.364	75893.925	114726.509	135107.970	161954.007
Gap	0.0	0.0	0.0	0.0	0.0

## Aplicação do problema de localização de facilidades

### 1. Descrição do problema

O problema que escolhemos foi a localização de hospitais públicos de acordo com a demanda da população de São Carlos. Consideramos que a cidade já possui uma unidade da Santa Casa, além do hospital universitário da UFScar. O objetivo é auxiliar a prefeitura em uma possível construção de um novo hospital, fornecendo a localização ótima para tal de acordo com os custos de cada bairro e a necessidade dos pacientes.

### 2. Variáveis e parâmetros

Utilizamos uma variável binária para determinar se um hospital é aberto ou não, e uma variável contínua correspondente a fração da demanda de cada paciente atendida pelos hospitais.

Como parâmetros temos o custo de instalação e a capacidade de cada hospital, a demanda de atendimento dos pacientes de cada região e o custo de transporte de cada bairro para os hospitais.

### 3. Função objetivo e restrições

A nossa função objetivo visa minimizar o custo total de distribuição dos hospitais, com base no custo fixo de cada e dos custos de transporte dos pacientes até o mesmo.

Elaboramos as seguintes restrições:

- Todos os pacientes devem ter suas demandas por atendimento médico atendidas
- A capacidade de cada um dos hospitais é respeitada, evitando um colapso no atendimento
- As hospitais 1 (Santa Casa) e 2 (Federal) já estão abertos
- Limita a abertura de apenas um novo hospital
- Domínio das variáveis

# Toy problem

## 1. Modelagem

$$\text{minimize } z = \sum_{i=1}^n f_i y_i + \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij} \quad (1)$$

sujeito a

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, \dots, m \quad (2)$$

$$\sum_{j=1}^m d_j x_{ij} \leq \text{Cap}_i y_i \quad i = 1, \dots, n \quad (3)$$

$$y_1 = 1 \quad (4)$$

$$y_2 = 1 \quad (5)$$

$$\sum_{i=1}^n y_i = 3 \quad (6)$$

$$y_i \in \{0, 1\} \quad i = 1, \dots, n \quad (7)$$

$$0 \leq x_{ij} \leq 1 \quad i = 1, \dots, n; j = 1, \dots, m \quad (8)$$

## 2. Dados utilizados

- Capacidade dos hospitais: [52, 149, 14, 146, 8]
- Custo fixo de construção dos hospitais: [15, 11, 34, 31, 19]
- Matriz do custo de transporte dos pacientes para os hospitais:

$$\begin{bmatrix} 3 & 2 & 8 & 4 & 5 \\ 7 & 8 & 5 & 6 & 1 \\ 4 & 3 & 7 & 10 & 6 \\ 6 & 2 & 5 & 7 & 1 \\ 5 & 6 & 5 & 3 & 8 \\ 2 & 4 & 2 & 8 & 7 \\ 4 & 4 & 7 & 4 & 7 \\ 9 & 10 & 10 & 9 & 4 \\ 2 & 2 & 7 & 5 & 7 \\ 4 & 8 & 7 & 7 & 9 \end{bmatrix}$$

- Demandas dos pacientes: [5, 5, 3, 1, 5, 2, 2, 1, 4, 3]

## 3. Solução obtida

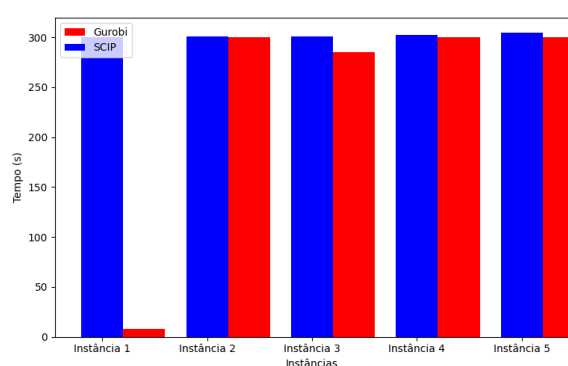
O valor ótimo encontrado foi de 73, que é correspondente a ambos os limitantes primal e dual, e o hospital selecionado para ser aberto foi o 5. Com isso, a distribuição dos pacientes pelos hospitais ficou da seguinte forma:

- Hospital Santa Casa: Paciente 5, Paciente 6, Paciente 7, Paciente 9, Paciente 10
- Hospital da Federal: Paciente 1, Paciente 3
- Novo Hospital: Paciente 2, Paciente 4, Paciente 8

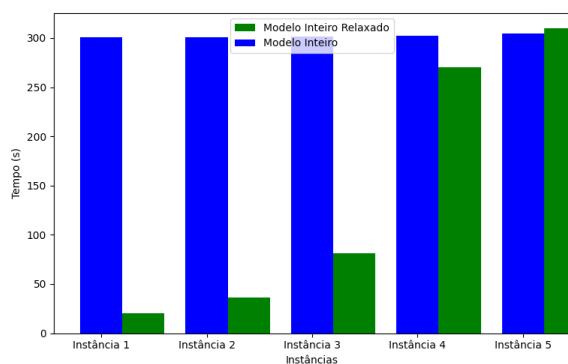
## Análise de Resultados

Com base nos dados obtidos, pudemos obter algumas comparações entre os solves Gurobi e Scip, para os modelos inteiro e relaxado.

## 1. Tempo

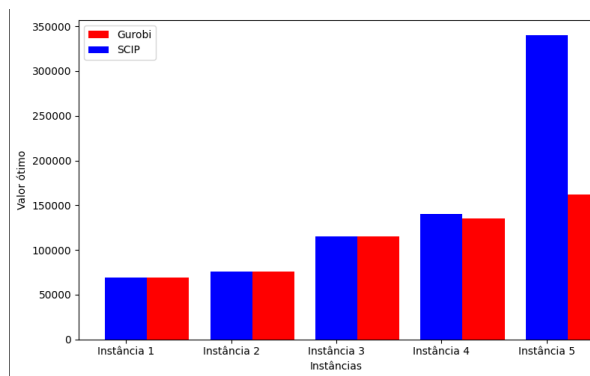


Com base no gráfico apresentado, podemos observar que os solvers Gurobi (em vermelho) e Scip (em azul) obtiveram resultados semelhantes em questão do tempo, com exceção para a instância 1, onde o Gurobi foi muito mais performático (8.15s) com relação ao Scip (300.43s).



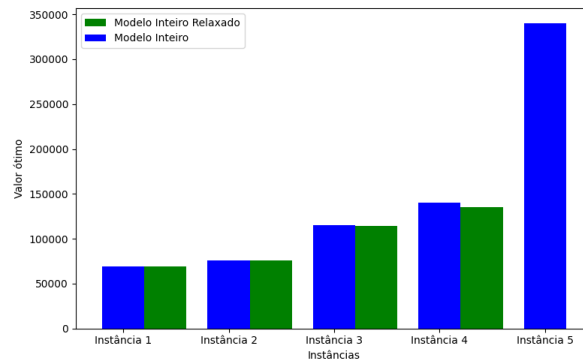
Já para o modelo relaxado (em verde), utilizando o Scip, as instâncias 1, 2 e 3 obtiveram uma grande vantagem com relação ao modelo inteiro (em azul). Já nas instâncias 4 e 5, o tempo de execução foi semelhante entre os dois modelos.

## 2. Valor Ótimo



Para o valor ótimo, foi possível concluir que houve uma vantagem do Gurobi (em vermelho) com relação ao Scip (em azul) apenas para a instância 5, onde o Gurobi foi capaz de melhor minimizar a solução.

No modelo relaxado (em verde), pudemos notar que os valores ótimos com relação ao modelo inteiro (em azul) foram muito semelhantes, com exceção da instância 5, onde apenas o modelo inteiro obteve uma solução.



No *Toy problem*, o solver não teve muita dificuldade em achar a solução ótima, e demorou apenas 0.01 segundos. Ademais, podemos perceber que a distribuição dos pacientes foi feita de maneira consideravelmente uniforme entre os hospitais, representando uma solução que dificulta o colapso do sistema de saúde na cidade.

## Conclusão

Portanto, temos que as propostas exigidas pelo trabalho foram concluídas de maneira esperada. Os dois solvers tiveram um desempenho semelhante, gerando soluções condizentes com as instâncias dadas. Além disso, a pesquisa para o desenvolvimento de uma aplicação voltada para a cidade de São Carlos nos auxiliou em enxergar as possibilidades que os conceitos de otimização aprendidos nessa disciplina possuem em ser aplicados no cotidiano real.