MO824 | Atividade 2

Nome Adolfo Schneider **RA** 234991
Nome Pedro Mota **RA** 185853
Nome Fabio Stori **RA** 196631

1. Introdução

O objetivo desta atividade consiste na modelagem em programação linear e otimização do *k-similar Travelling Salesman Problem* (kSTSP) utilizando *lazy constraints* no software Gurobi. O problema modelado é descrito a seguir:

Seja um grafo não-orientado completo G(V, E), onde V é o conjunto dos vértices e E é o conjunto das arestas. Em cada aresta $e \in E$ há dois custos $c^1_e, c^2_e : E \rightarrow R+$. O objetivo do problema consiste em encontrar dois ciclos Hamiltonianos com custo total mínimo, tal que pelo menos k arestas do grafo sejam visitadas por ambos os ciclos. O parâmetro k define a similaridade entre os ciclos, de modo que $k = |V|$ implica que uma solução factível corresponda a dois ciclos Hamiltonianos disjuntos nas arestas, enquanto $k = 0$ resulta em uma solução factível ser qualquer par de ciclos Hamiltonianos

2. Modelo

Para modelar o problema apresentado, usou-se três variáveis de decisão:

x^1_i : variável binária, 1 se a aresta *i* está faz parte do ciclo hamiltoniano 1, e 0 caso o contrário

x^2_i : variável binária, 1 se a aresta *i* está faz parte do ciclo hamiltoniano 2, e 0 caso o contrário

y_i : variável binária, 1 se a aresta *i* está em ambos os ciclos, 0 caso o contrário

A grandeza a ser minimizada neste problema, é a soma do custo total dos dois ciclos. A *Equação 1* expressa o custo total dos ciclos de forma matemática. Onde $c^1(i)$ e $c^2(i)$ indicam o custo da aresta para o ciclo 1 e 2, respectivamente.

$$\sum_{i \in E} x^1_i \cdot c^1(i) + x^2_i \cdot c^2(i) \qquad (1)$$

As *Equações 2* e *3* expressam a restrição de que cada vértice deve ter duas arestas incidentes (uma "de entrada" e outra "de saída").

$$\sum_{i \in \delta(i)} x^1_i = 2, \forall i \in V \qquad (2)$$

$$\sum_{i \in \delta(i)} x_i^2 = 2, \forall i \in V \quad (3)$$

As restrições das *Equações 4* e *5* eliminam subciclos ilegais ao impedir que em qualquer subconjunto de vértices existam mais do que |s|-1 arestas, onde *S* é um subconjunto próprio de vértices.

$$\sum_{i \in E(S)} x_i^1 \leq |S| - 1, \forall S \subset V \quad (4)$$

$$\sum_{i \in E(S)} x_i^2 \leq |S| - 1, \forall S \subset V \quad (5)$$

As *equações 6, 7, 8* e *9* expressam as restrições de similaridades entre os ciclos. As *Equações 6, 7 e 8* expressam a definição de que, se $y_i = 1$, a aresta i pertence aos dois ciclos e que caso i pertença a apenas 1 ciclo então $y_i = 0$. Já a *Equação 9* define a restrição de que pelo menos k arestas estão presentes nos dois ciclos (x_i^1 e x_i^2).

$$x_i^1 \geq y_i \quad (6)$$

$$x_i^2 \geq y_i \quad (7)$$

$$x_i^1 + x_i^2 - 1 \geq y_i \quad (8)$$

$$\sum_{i \in E} y_i \geq k \quad (9)$$

3. Resultados

Implementamos o modelo construído para o problema com a biblioteca gurobipy, implementação do software Gurobi para a linguagem de programação Python. Na execução dos experimentos computacionais, foi utilizado um computador com 32GB de memória ram, processador Intel® Core™ i5-8265U CPU @ 1.60GHz × 8 e sistema operacional Ubuntu 20.04.

O código implementado utiliza-se das coordenadas fornecidas e executa o modelo um total de 12 vezes, variando entre as possíveis combinações de N e K, onde

$$N = \{100, 150, 200, 250\} \quad (8)$$

$$K = \{0, N/2, N\} \quad (9)$$

conforme definido no enunciado da atividade. Os resultados para cada instância do modelo são apresentados na **Tabela 1**, e **Gráficos 1** e **2**.

N	K	Custo Ótimo	Gap de Otimalidade	Tempo de Execução (segundos)
100	0	1630	0.0000%	10.71
100	50	2102	0.0000%	115.75
100	100	3463	0.0000%	9.02
150	0	1966	0.0000%	52.61
150	75	2748	0.0000%	693.45
150	150	4780	0.0000%	42.61
200	0	2308	0.0000%	153.94
200	100	3390*	-	1800**
200	200	6003	0.0000%	49.27
250	0	2597	0.0000%	540.69
250	125	3912*	-	1800**
250	250	6999	0.0000%	426.44

Tabela 1: Tabela com os dados de execução do modelo para cada conjunto de N e K iterados.
*Último valor exibido antes da interrupção pelo limite de tempo
**Execução não chegou ao fim e alcançou o tempo limite de 30 minutos de execução

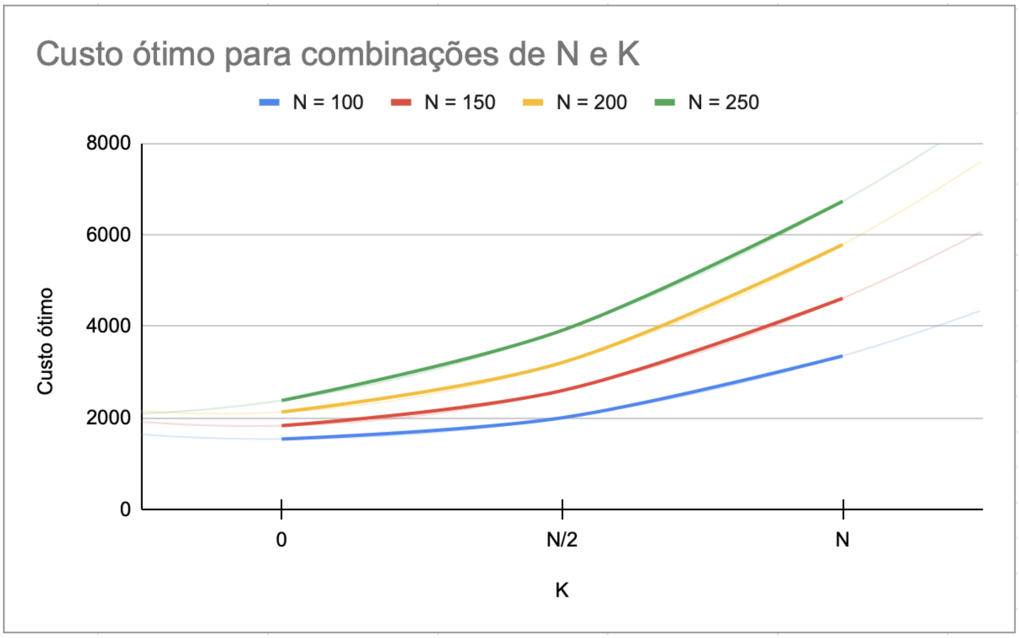

Gráfico 1: Gráfico mostrando o custo ótimo do modelo para cada conjunto de N e K.

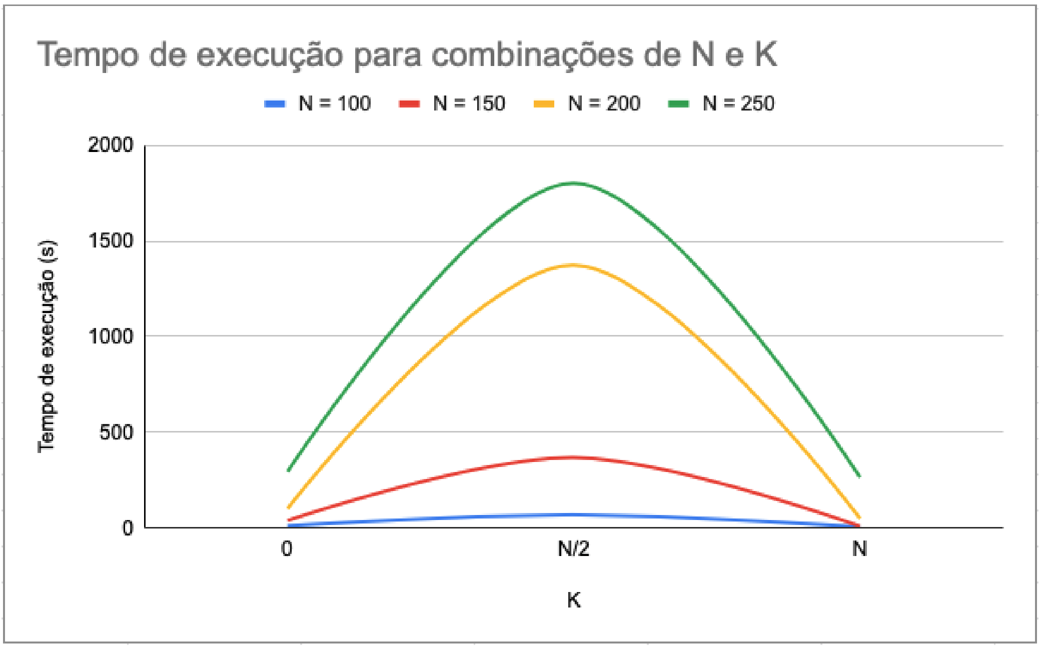

Gráfico 1: Gráfico mostrando o tempo de execução do modelo para cada conjunto de N e K.

3. Análise

A **Tabela 1** e **Gráfico 1** mostram um padrão de crescimento do custo ótimo para as instâncias do modelo. Como era esperado, quanto maior o valor de N, maior o valor do custo, já que há mais vértices para serem visitados. Além disso, o custo ótimo é proporcional ao grau de similaridade , K, entre os ciclos. Para todos os valores de N, o custo mínimo é menor quando K = 0, e maior quando K = N. Isso acontece porque quando **K = 0** é possível escolher a melhor combinação de arestas que formam os dois ciclos com custo total mínimo, sem a necessidade ter uma ou mais arestas em comum nos ciclos, o que poderia aumentar o custo da solução para satisfazer essa restrição. Já quando K = N os dois ciclos precisam conter as mesmas arestas, diminuindo o número de soluções factíveis e de custos menores para satisfazer essa restrição.

A **Tabela 1** e **Gráfico 2** também mostram um padrão de crescimento do tempo de execução para as instâncias do modelo. O **Gráfico 2** mostra que para todos os valores de K, um aumento em N gera um aumento no tempo de execução. Além disso, observando o tempo de execução para as 3 variações de **K** para um mesmo **N**, é possível perceber um menor tempo de execução quando K = N em todos os casos. Isso acontece porque quando **K = N**, os dois ciclos devem ser exatamente iguais, simplificando o problema a encontrar apenas um ciclo hamiltoniano.

4. Conclusão

A partir dos resultados e análise deste relatório, é possível concluir que: o custo mínimo é proporcional ao valor de N. E que para um mesmo valor de N, quanto maior o grau de similaridade K entre os ciclos, maior será o custo mínimo. Além disso, o tempo de execução também é proporcional ao valor de N. E que para um mesmo N, o tempo de execução é mínimo quando K = N, e máximo para K = N/2.