Atividade 3 - Grupo 12 - MO824

Eduardo Barros Innarelli - RA 170161
Gabriel Henriques Siqueira - RA 155446
Vitor Satoru Machi Matsumine - RA 264962

1. Introdução

Este trabalho consiste na apresentação de um modelo de relaxação Lagrangiana do problema 2TSP (*two travelling salesman problem*), assim como a execução de experimentos utilizando o modelo desenvolvido. A modelo foi construído a partir de um modelo matemático inicial, apresentado na atividade, com a dualização da seguinte restrição:

$$\sum_{k\in\{1,2\}} x_e^k \leq 1 \quad \forall e \in E \ ;$$

Para tal, foi empregada programação linear através do software Gurobi em Python, utilizando-se do método do subgradiente e de uma heurística Lagrangiana. Ao longo do documento será apresentado o modelo matemático desenvolvido, a parametrização utilizada no método do subgradiente, a heurística Lagrangiana desenvolvida e os resultados experimentais obtidos.

2. Modelo Matemático

Seja $G(V, E)$ um grafo não-orientado completo, onde V é o conjunto dos vértices, E é o conjunto das arestas e cada aresta $e \in E$ tem um custo associado $c_e : E \rightarrow \mathbb{R}^+$. A relaxação Lagrangiana do modelo de programação linear inteira para o 2TSP proposto é dada por:

$$\begin{aligned}
\min \quad & \sum_{e\in E}\Big(\sum_{k\in\{1,2\}} C_e^k x_e^k - \lambda_e\Big), \quad C_e^k = c_e + \lambda_e && (1)\\
\text{s.a} \quad & \sum_{e\in\delta(i)} x_e^k = 2 && \forall i \in V, \forall k \in \{1,2\} \quad (2)\\
& \sum_{e\in E(S)} x_e^k \leq |S| - 1 && \forall S \subset V, \forall k \in \{1,2\} \quad (3)\\
& x_e^k \in \{0,1\} && \forall e \in E, \forall k \in \{1,2\} \quad (4)
\end{aligned}$$

Onde:

- x_e^k é uma variável de decisão binária associada à presença ($x_e^k = 1$) ou não ($x_e^k = 0$) da aresta e na rota do caixeiro k;
- $\delta(i)$ é o conjunto de arestas que incidem no vértice i;
- $S \subset V$ é um subconjunto próprio de vértices;
- $E(S)$ é o conjunto das arestas cujos dois vértices terminais estão em S;
- λ_e é o multiplicador de Lagrange referente à aresta e;
- $C_e^k = c_e + \lambda_e$ para todo $e \in E$ é o custo lagrangiano.

3. Método do Subgradiente

Desejamos otimizar o limitante inferior que a relaxação lagrangiana retorna. Para isso, é preciso resolver o problema dual lagrangiano da relaxação, que consiste em encontrar os multiplicadores λ_e associados ao limitante ótimo. Uma das maneiras de solucionar o dual lagrangiano é através do método do subgradiente, que implementamos neste trabalho.

Definimos o subgradiente $g_e^{(i)}$ da i-ésima iteração avaliado na solução atual como:

$$g_e^{(i)} = (-1) - (-\sum_{k=1}^{2} (x_e^k)^{(i)}), \text{ para } e \in E.$$

O método resume-se em, a cada iteração i, atualizar os multiplicadores da próxima iteração $\lambda_e^{(i+1)} = \max(0,\ \lambda_e^{(i)} + \alpha^{(i)} \cdot g_e^{(i)})$ dando um passo $\alpha^{(i)}$ pequeno em direção ao subgradiente, com o intuito de aumentar o limitante inferior retornado pelo modelo. Este é otimizado com *lazy constraints*, i.e., as restrições (3) são incluídas dinamicamente conforme soluções do modelo são encontradas. O limitante inferior $Z_{LB}^{(i)}$ obtido é utilizado para gerar um limitante superior $Z_{UB}^{(i)}$ através de uma heurística Lagrangiana (descrita na Seção 4). Mantemos salvo o melhor limitante superior Z_{UB} encontrado até então.

Para o tamanho do passo $\alpha^{(i)}$, adotamos a expressão sugerida em aula:

$$\alpha^{(i)} = \pi^{(i)} \frac{(Z_{UB} - Z_{LB}^{(i)})}{\sum_{e \in E} (g_e^{(i)})^2} ,$$

onde $\pi^{(0)} = 2$ e $\pi^{(i)} = \max(0.1,\ 0.99 \cdot \pi^{(i-1)})$, se $i > 1$. O método termina por optimalidade (mais especificamente se o *gap* entre os limitantes for menor que 10^{-6}) ou se o limite de tempo de 30 minutos for atingido. O Algoritmo 1 apresenta um pseudo-código descrevendo passo-a-passo o algoritmo.

Algoritmo 1: Método do Subgradiente

Entrada: Modelo relaxado e custos c_e das arestas
Saída: Melhores limitantes encontrados

início
  Inicialize $\pi \leftarrow 2,\ Z_{LB} \leftarrow 0,\ Z_{UB} \leftarrow \infty,\ \lambda_e \leftarrow 0$ para cada aresta $e \in E$
  enquanto $\frac{Z_{UB}-Z_{LB}}{Z_{UB}} \geq 10^{-6}$ *e o tempo total de execução não atingir* 30 *minutos* **faça**
    $Z_{LB},\ x \leftarrow$ Resolva a relaxação lagrangiana do 2TSP
    $Z'_{UB} \leftarrow$ Resolva a heurística lagrangiana do 2TSP
    $Z_{UB} \leftarrow \min(Z_{UB},\ Z'_{UB})$
    para *cada aresta* $e \in E$ **faça**
      $g_e \leftarrow \sum_{k=1}^{2} x_e^k - 1$
      $\lambda_e \leftarrow \max(0,\ \lambda_e + \pi \cdot \frac{(Z_{UB}-Z_{LB})}{\sum_{e \in E}(g_e)^2} \cdot g_e)$
    $\pi \leftarrow \max(0.1,\ 0.99 \cdot \pi)$
  retorna Z_{LB} *e* Z_{UB}

4. Heurística Lagrangiana

Na heurística Lagrangiana desenvolvida, tomamos um limitante inferior $Z_{LB}^{(i)}$, obtido através da solução do modelo relaxado na i-ésima iteração, como base para obter um limitante superior $Z_{UB}^{(i)}$.

Separamos os respectivos *tours* $k \in \{1, 2\}$ da solução obtida do problema relaxado. Fixamos o primeiro *tour* e substituímos as arestas do segundo *tour* que já estão presentes no primeiro. Para a substituição das arestas, removemos um par de arestas não consecutivas (i, j) e (k, l), que estão presentes no primeiro *tour*, e as substituímos por duas arestas (i, k) e (j, l), ou (i, l) e (j, k), caso adicionar (i, k) forme um *tour* menor do que $|V|$ (nesse caso o mesmo ocorreria com (j, l)). Com essa substituição, adicionamos pelo menos uma aresta que não está no primeiro *tour*. Continuamos a repetir essa substituição enquanto os pares podem ser encontrados. Para melhorar os resultados escolhemos, de forma gulosa, os pares que serão trocados por arestas de custo mínimo.

Após aplicar todas as substituições possíveis, ainda podem sobrar arestas pertencentes ao primeiro *tour*. Para remover essas arestas, aplicamos trocas como as anteriores, mas apenas uma das arestas escolhidas deve pertencer ao primeiro *tour*. Nesse caso, para garantir que iremos sempre remover uma aresta pertencente ao primeiro *tour*, escolhemos o par de arestas de forma a garantir que nenhuma das duas arestas adicionadas pertençam ao primeiro *tour*.

O custo deste novo *tour* é calculado e somado ao custo do *tour* fixado. O processo é então repetido fixando-se o segundo *tour* e modificando-se o primeiro. O valor de $Z_{UB}^{(i)}$ é então definido pelo menor valor total dentre as modificações dos *tours* realizadas. O Algoritmo 2 apresenta um pseudo-código descrevendo passo-a-passo o algoritmo.

Algoritmo 2: Heurística Lagrangiana

Entrada: Custos c_e das arestas, par de *tours* hamiltonianos T_1 e T_2
Saída: Custo do par de *tours* corrigidos com a substituição de arestas repetidas

1 **início**
2   **para** *cada par de aresta* $\{(i,j),(k,l)\} \subset T_1$ *(ordenadas pelo custo das substituições)* **faça**
3   **se** *a adição de* (i,k) *em* T_2 *não forma um ciclo de tamanho menor que* $|V|$ **então**
4   Adicione (i,k) e (j,l) a T_2
5   **senão**
6   Adicione (i,l) e (j,k) a T_2
7   **enquanto** *existir uma aresta* $(i,j) \in T_1 \cap T_2$ **faça**
8   Escolha uma aresta (k,l) de tal forma que a troca realizada a seguir não adicione arestas de T_1
9   **se** *a adição de* (i,k) *em* T_2 *não forma um ciclo de tamanho menor que* $|V|$ **então**
10   Adicione (i,k) e (j,l) a T_2
11   **senão**
12   Adicione (i,l) e (j,k) a T_2
13   $Z_{UB,1}^{(i)} \leftarrow$ custo dos *tours* T_1 e T_2
14   Repita as linhas 2 até 13 trocando T_1 por T_2 para obter $Z_{UB,2}^{(i)}$
15   **retorna** *o mínimo entre* $Z_{UB,1}^{(i)}$ *e* $Z_{UB,2}^{(i)}$

---

5. Geração de instâncias

As instâncias foram geradas de maneira uniforme e aleatória para cada uma das quantidades de vértices $|V| = \{100, 150, 200, 250, 300\}$. Em cada instância foram gerados $|V|$ pontos (x, y) aleatórios com $x, y \in [0.0, 1.0]$, que por sua vez foram utilizadas para criar arestas cujo peso corresponde a distância Euclidiana entre um par de pontos.

Após todas as instâncias serem geradas, elas são serializadas e salvas em um arquivo "**.pkl**" utilizando-se da biblioteca ***pickle*** para posteriormente serem recuperadas facilmente e utilizadas nos experimentos.

6. Resultados dos experimentos

Testamos o modelo relaxado e o modelo original para $2TSP$ com as instâncias geradas. Em ambos os testes, limitamos o tempo de execução para cada instância em 30 minutos.

As Tabelas 1 e 2 mostram os resultados obtidos com o modelo relaxado e com o modelo original, respectivamente. Observamos que não houve vantagem em utilizar o modelo relaxado para as instâncias testadas. Apesar dos limites encontrados para o modelo relaxado serem compatíveis com intervalos apresentados na solução do problema original, o mesmo obteve limitantes inferiores e superiores melhores que o modelo relaxado, inclusive encontrando o resultado exato para as instâncias de tamanhos 100 e 150 em um tempo inferior a 30 minutos.

Tabela 1: Resultados Obtidos (problema relaxado)

Número de vértices	Limitante Inferior	Limitante Superior	Tempo de Execução (s)
100	18.04	21.13	1800.00
150	20.77	26.08	1800.09
200	23.64	30.41	1800.01
250	24.38	34.55	1800.00
300	25.14	36.01	1800.01

Tabela 2: Resultados Obtidos (problema original)

Número de vértices	Limitante Inferior	Limitante Superior	Tempo de Execução (s)
100	19.18	19.18	75.97
150	23.19	23.19	321.93
200	27.04	27.26	1800.09
250	28.76	33.66	1800.68
300	30.53	35.13	1802.86

Acreditamos que a qualidade dos resultados está relacionada ao fato das restrições relaxadas não serem restrições complicadoras: o problema relaxado ainda envolve encontrar dois ciclos Hamiltonianos independentes, o que é computacionalmente difícil para as instâncias geradas. Uma relaxação mais promissora em termos de complexidade pode surgir de também relaxar as restrições de grau-2 (2), pois o problema relaxado passaria a ser encontrar as duas 1-árvores independentes de menor custo do grafo. Um possível próximo passo deste trabalho seria implementar e testar essa relaxação, baseada na relaxação lagrangiana proposta por Held & Karp (1970) para o TSP.