Grupo	Heitor Rodrigues Sabino Ranulfo Mascari Neto	202120101 202120792	10A 10A
Disciplina	Algoritmos em Grafos		
Tema	Relatório - Trabalho Prático - Projeto Final		

Macroentrega 1

Introdução

Neste relatório será abordado a questão do algoritmo de resolução da Macroentrega 2 e a lógica pensada por trás deste algoritmo. Os comentários acerca da Macroentrega 1 estarão presentes no código para facilitar a compreensão do mesmo.

Desenvolvimento

Ao analisarmos o problema referente à Macroentrega 2, identificamos como uma "variação" do Problema do Caixeiro-Viajante (PCV), onde temos que: partir de um ponto de origem (depósito), traçar uma rota na qual passamos pelos pontos de coleta e, após isso, passar pelos seus respectivos pontos de entrega e, ao final, voltar ao ponto de origem, isso passando pela menor rota possível. Entretanto, há mais uma variável a ser considerada, as janelas de tempo, tanto do depósito, quanto do ponto de coleta, quanto do cliente. Por conta dessa variável a ser considerada, torna-se uma "variação" do PCV. Pelo fato de as arestas serem ponderadas com o tempo, temos que além de traçar a menor rota, obedecer às janelas de tempo. Pensando de modo simplista, seria somente usar vários caminhões e resolver o problema, mas temos que considerar também a menor quantidade de caminhões possíveis. Dessa forma, temos que fazer com que cada caminhão seja usado da melhor maneira possível.

Tendo em vista essa problemática, pensamos em estratégias para diminuir nosso problema em partes, logo pensamos em "dividir" o grafo em regiões. Assim, teríamos regiões de ponto de entrega, ao invés de analisarmos cada ponto de entrega. Obtemos as regiões por meio da técnica de *machine learning k-means clustering*. Cada cluster tem o tamanho da capacidade de um caminhão, dessa forma o caminhão apenas precisa passar nos pontos de coleta dos respectivos pontos de entrega daquele cluster e partir para ele. Porém, podemos ter um grafo que contenha muitos clusters, e enviar um caminhão para cada cluster não seria viável, pois quando um caminhão termina seu cluster ele volta para o depósito, sendo que este caminhão poderia, ao invés de voltar para o depósito, fazer mais entregas (clusters). Então,

pensamos em trabalhar com uma ideia progressiva, mandando um caminhão e testando se ele consegue completar todas as entregas, senão concluiu todas as entregas dentro da janela de tempo, mandamos mais um caminhão na próxima iteração, ou seja, enquanto uma solução atual não for válida, será inserido mais um caminhão na frota de veículos, até o momento em que conseguimos fazer todas as entregas dentro dos prazos dos pontos de coleta, entrega e do depósito.

Pseudocódigo

```
para cada c ∈ G.Clusters
        c.disponível = verdadeiro
        c.tempo = tempo de roteirização do cluster

Ordene os G.Clusters em ordem crescente de tempo de roteirização

|M| = 0
solução = Ø
para cada v ∈ M
        v.posição = depósito
fazer {
        |M| = |M| + 1
        enquanto(c.disponível ∈ G.Clusters)
                para cada v ∈ M
                        v.trajetória += roteirização(v.posição, cluster disponível
        mais próximo do veículo)
                        v.posição = último da trajetória
                        v.tempo += c.tempo
                        c.disponível = falso
                        ordena(v.tempo ∈ M)
        para cada v ∈ M
                v.trajetória += depósito
                solução.v = v.trajetória
                v.tempo += tempo(v.posição, depósito).tempo
} enquanto(!função_de_verificação(solução))
```

<tempo de roteirização do cluster>: Definição do melhor tempo de rota da origem para os pontos de coleta e entrega com base em sua trajetória(T).

$$\min\left(\sum_{\substack{v^{(i)} \in T \\ i=0}}^{T}\left(\begin{matrix} v_{tempo}^{(i)} + \\ \left(v^{(i)}, v^{(i+1)}\right)_{tempo} \end{matrix}\right)\right) + v_{tempo}^{(i)}$$

<roteirização(<origem>, <conjunto de vértices do cluster>)>: Função que retorna o conjunto de vértices da melhor rota do ponto de origem até todos os vértices do cluster.