ALGORITMOS APROXIMADOS PARA O PROBLEMA DO CAIXEIRO VIAJANTE

João Augusto, Juliana Araújo

Algoritmos e Estrutura de Dados III, Universidade Federal de São João del-Rei

1 INTRODUÇÃO

Atualmente, inúmeros problemas computacionais encontrados são exponenciais, isto é, problemas em que para se garantir a melhor solução, todas as possibilidades de soluções devem ser encontradas e comparadas. Observa-se então, uma grande dificuldade para encontrar tal solução para grandes instâncias de problemas de otimização em tempo razoável.

Uma das formas de encontrar soluções para problemas exponenciais é utilizar algoritmos aproximados. Com eles, a resposta encontrada pode não ser ótima, porém é garantido que esteja próxima dela, através de uma razão previamente estabelecida e comprovada. Neste trabalho, será apresentado um algoritmo aproximado para o "problema do caixeiro-viajante", baseado na estratégia de Christofides.

2 ESPECIFICAÇÕES DO PROBLEMA

O problema do caixeiro-viajante consiste em um conjunto de cidades e suas respectivas distâncias, onde parte-se de uma cidade inicial arbitrária e visita-se todas as cidades precisamente uma vez, regressando à cidade inicial. A modelagem do problema é feita por uma estrutura denominada grafo. Um grafo é um conjunto de vértices interligados por arestas, como mostra a imagem a seguir.

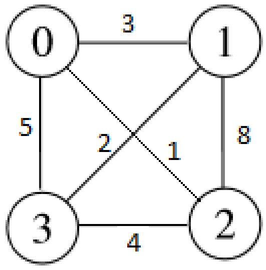

Figura 1 – Grafo não-direcionado completo

Para o problema em questão, os vértices representarão as cidades, bairros, distritos, e etc, do mapa e as arestas são as estradas que ligam os ligam. O grafo utilizado é dito ponderado, e não-direcionado. Ponderado pois cada aresta terá um peso relacionado a distância entre as cidades. Não-direcionado significa que a aresta que liga *u* a *v* é a mesma que liga o vértice *v* a *u*. O grafo também é um grafo completo, pois todos os pares de vértices são adjacentes, ou seja, cada vértice possui uma aresta que o liga a todos os outros vértices do grafo.

3 DESENVOLVIMENTO

3.1 Entradas

TSPLIB (Traveling Sales Problem Library) é uma biblioteca com amostras para o TSP (ou PCV) a partir de várias fontes e de vários tipos. Os dados são dispostos conforme o campo “EDGE_WEIGHT_FORMAT”, podendo assumir formatos explícitos, euclidianos e etc.

A biblioteca foi utilizada como base de dados na realização dos testes, da qual foram-se obtidas informações para a geração do grafo completo e na comparação de resultados, aonde a solução encontrada pelo nosso algoritmo de aproximação era comparada à melhor solução conhecida obtida com o algoritmo de Christofides.

3.2 Saídas

A saída do algoritmo é dada pelo terminal de execução e contém a distância total calculada. Essa distância é o ciclo Euleriano encontrado.

3.3 Implementação

Na implementação do grafo foi utilizada a matriz de adjacência. Cada par de cidade pode ser acessado de forma direta, utilizando apenas os índices da matriz. Nessa posição, estará guardado a distância entre eles. Segue o exemplo de uma matriz de adjacências de um grafo completo ponderado, que possui 4 vértices:

	0	**1**	**2**	**3**
0	0	4	8	6
1	4	0	2	5
2	8	2	0	3
3	6	5	3	0

Figura 2: Representação de grafo em matriz de adjacências

Percebe-se que o grafo é espelhado, já que o mesmo é não-direcionado.

Feita a inserção, inicia-se a implementação do algoritmo de Christofides. De forma breve, os passos desse algoritmo são:

1. Obtenha a árvore geradora mínima *T* para o conjunto de *n* cidades.
2. Construa um casamento mínimo *M* para um conjunto de vértices de grau ímpar em *T*.
3. Encontre um caminho de Euler para o grafo Euleriano obtido com a união de *T* e *M*, e converta o caminho de Euler em um caminho do caixeiro-viajante usando curtos-circuitos.

O detalhamento dos passos e suas respectivas implementações serão apresentadas nos subtópicos a seguir.

3.3.1 Árvore geradora mínina (AGM)

Árvore geradora mínima é um subgrafo que contém todos os vértices do grafo original, tendo um conjunto de arestas de menor custo que conecte todos os vértices do grafo (todo par de vértices ainda é alcançável por algum caminho).

O algoritmo utilizado para gerar a árvore geradora mínima é o Prim. Primeiramente, é definido um vértice inicial para a sua construção. Por padrão, foi definido que este é o primeiro vértice inserido no gráfico. A cada iteração, é procurada a aresta de menor peso (menor distância entre as cidades) que conecte um vértice da árvore a outro que ainda não esteja na árvore. Esse vértice é adicionado e o processo é repetido. O processo termina quando todos os vértices fazem parte da árvore ou quando não se pode encontrar uma aresta que satisfaça a condição. Abaixo, a exemplificação de um grafo e sua respectiva árvore geradora mínima.

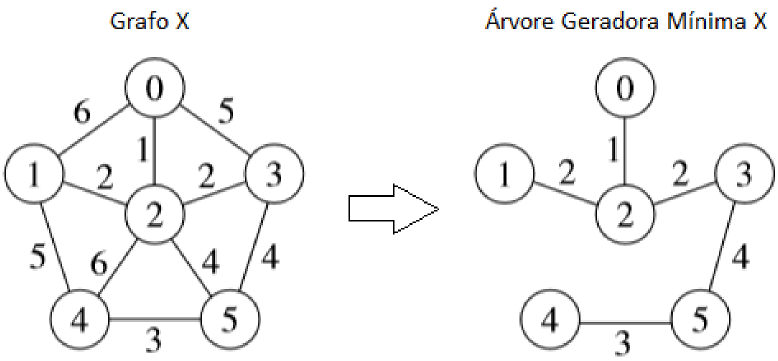

Figura 3: Grafo e AGM

Após a obtenção desse conjunto de arestas, é criado um novo grafo com a AGM.

3.3.2 Casamento mínimo

O grau de um vértice é o número de arestas que incidem sobre ele. Christofides concluiu que existe sempre um número par de vértices de grau ímpar. Sendo assim, para obter um grafo conectado em que todos os vértices possuem grau par é necessário apenas que se faça um “casamento” entre os vértices de grau ímpar. Isso aumenta em um o grau de cada vértice ímpar, já os de grau par não mudam.

Porém, o casamento efetuado neste trabalho não foi o mínimo. Foi criada uma heurística que, a partir de um vértice ímpar, procura outro vértice ímpar adjacente com quem faz um casamento de menor peso. Essa aresta então é acrescentada no grafo que contém a AGM. Isso é feito para o restante dos vértices até que todos já não possuam grau ímpar.

3.3.3 Caminho Euleriano com curto-circuitos

Um grafo Euleriano é um grafo conectado no qual todo vértice tem grau par. Esse grafo possui um caminho Euleriano, um ciclo que passa por todas as arestas exatamente uma vez. Percebe-se que um grafo Euleriano foi obtido a partir da AGM, no passo anterior.

O caminho Euleriano foi obtido usando a busca em profundidade. A busca em profundidade é realizada partindo de um vértice inicial, e explorando o máximo possível de cada um de seus ramos antes de retroceder. No fim, todos os vértices foram visitados. Esse caminho realizado será contabilizado. Porém, o caminho feito quando se retrocede não será. É nesse ponto que é inserido o curto-circuito.

As entradas utilizadas respeitam a desigualdade triangular, onde a ligação direta de dois vértices por meio de uma única aresta não pode ser maior que um caminho que os ligam. O curto-circuito realiza essa “quebra” de rota. Quando a busca em profundidade retrocede, em vez de contabilizarmos esse caminho até um novo vértice, apenas ligamos o último vértice visitado ao próximo que deve ser visitado, por meio de uma única aresta, e então adiciona-se o peso da mesma na distância percorrida. É garantido que o peso dessa aresta é menor que o peso do caminho de retrocesso que seria feito até ela, devido a desigualdade triangular.

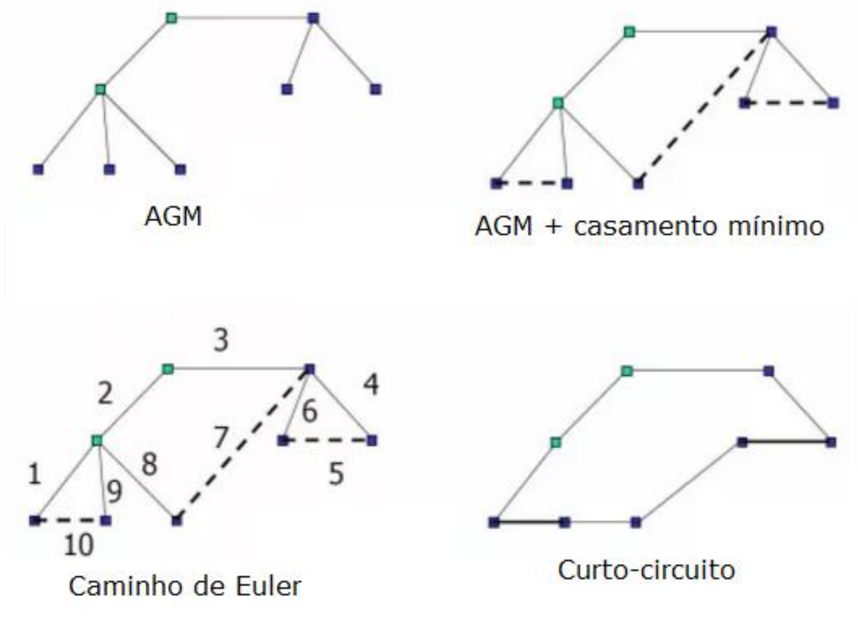

Figura 4: Passos do algoritmo de Christofides

A implementação do curto-circuito foi feita em conjunto com a busca em profundidade. Quando haveria o retrocesso, algumas variáveis foram utilizadas a fim de guardar as informações necessárias, realizou o retrocesso sem acrescentar os pesos das arestas, adicionou o curto-circuito e então retomava-se a busca em profundidade.

3.4 Razão de aproximação

Como já dito, um algoritmo aproximado é um algoritmo que gera soluções aproximadas. Para ser útil, é preciso obter um limite para a razão entre a solução ótima e a produzida pelo algoritmo aproximado.

A partir da AGM, podemos derivar um limite inferior para o problema do cacheiro-viajante (PCV). Ao retirar uma aresta do caminho ótimo do PVC, temos uma árvore geradora que consiste de um caminho que visita todos as cidades. Logo, o caminho do PVC é necessariamente maior do que o comprimento da AGM. Conclui-se que o limite inferior para o custo desse caminho é a AGM.

Já a desigualdade triangular permite utilizar a AGM para obter um limite superior para a razão de aproximação com relação ao comprimento do caminho ótimo. Utiliza-se o algoritmo da busca em profundidade, que visita todos os vértices e nenhuma aresta é visitada mais de duas vezes. Assim, é obtido um caminho que visita todas as cidades cujo custo é menor ou igual a duas vezes o caminho da AGM. Como o caminho é maior que o custo da AGM, definido anteriormente, então o caminho obtido é no máximo duas vezes o custo do caminho ótimo.

Já Christofides propôs uma melhoria no algoritmo anterior utilizando o conceito de casamento mínimo com pesos. Utilizando o casamento mínimo, a razão de aproximação garantida para o pior caso é de Ra <= 3/2. Como neste trabalho foi utilizado o conceito de casamento, porém não é garantido que seja mínimo, concluímos que não é possível garantir que a razão de aproximação do algoritmo proposto é *Ra <= 3/ 2, e sim que é Ra <= 2.*

4 ANÁLISE DE COMPLEXIDADE

Sendo *n* o número de cidades e, consequentemente o número de vértices, segue a análise de complexidade das funções utilizadas:

- Funções que criam, liberam e inserem no grafo: são constantes devido a utilização da matriz de adjacências, portanto O(1).

- Função que gera a árvore geradora mínima por meio do algoritmo de Prim: como percorre cada vértice e analisa cada ligação que ele faz, há dois laços aninhados que vão de 0 ao número total de vértices (n). Portanto, o custo do algoritmo no pior caso é $O(n^2)$.
- Funções que realizam a busca em profundidade: há uma função auxiliar que inicializa o vetor com 0. Esse vetor guarda o momento em que os vértices foram visitados. Depois a função que de fato realiza a busca em profundidade é chamada. Essa função sai de um vértice e visita todos seus adjacentes recursivamente. Desse modo, no pior caso sua complexidade é $O(|n| + |A|)$, onde A é o número de arestas.
- Leitura dos arquivos: a leitura do cabeçalho do arquivo de entrada é desconsiderada. A inserção de dados no grafo é feita de forma linear à execução da leitura, de tal forma que após um elemento ser lido é feita uma transformação $O(1)$ pra obtenção dos pesos respectivos às arestas. Para a inserção dos pesos no grafo, o cálculo é feito de um vértice para todos. Como os dados estão armazenados em um vetor de tamanho *n,* utiliza-se uma estrutura de repetição para encontrar cada um dos vértices, e depois é utilizada mais estrutura de mesmo tipo aninhada para encontrar a distância dele a todos os outro. Portanto, sua complexidade é $O(n^2)$.

A complexidade total do algoritmo será $Max(O(1) + O(n^2) + O(|n| + |A|) + O(n^2))$, onde *n* é o número de vértices presentes no arquivo de leitura. Deste modo, a complexidade do algoritmo de aproximação proposto é $O(n^2 + |A|)$.

5 ANÁLISE DE RESULTADOS

Foram executados testes para dois tipos de entradas: tipo Geo e tipo Euclidianas. Nos gráficos abaixo, podemos conferir os resultados do algoritmo de aproximação para testes de três entradas dentro de cada tipo. O objetivo é compará-las aos resultados otimizados previamente fornecidos, obtidos do algoritmo de Christofides. Também foi adicionado para fim de comparação, uma coluna com o dobro dos valores dos resultados otimizados. Previamente, foi definido que o algoritmo apresentado tem sua razão maior ou igual a do algoritmo de Christofides, e no máximo duas vezes maior que o melhor caminho do PCV.

Portanto, era necessário que o os resultados desse algoritmo fossem menor que o dobro dos resultados otimizados.

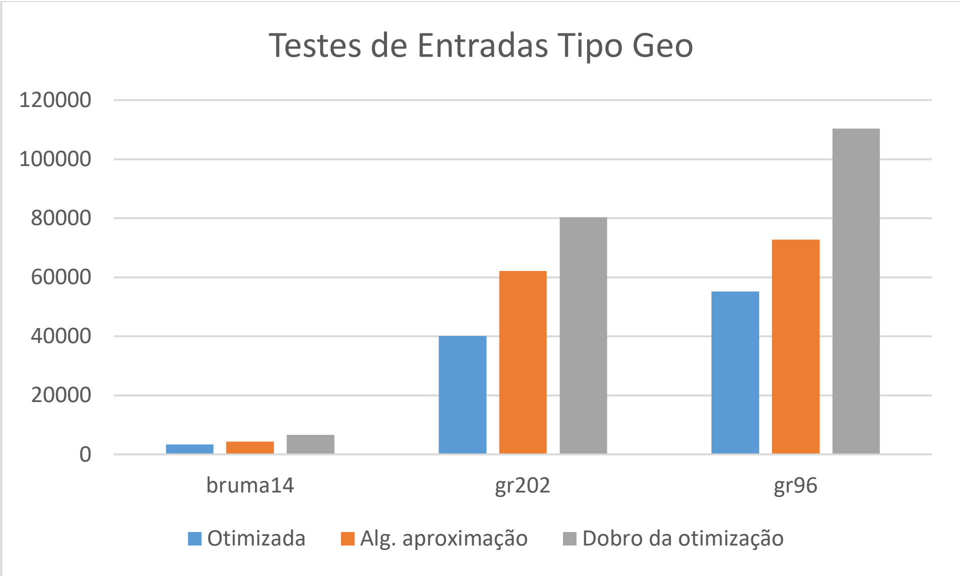

Figura 5: Testes de Entradas Tipo Geo

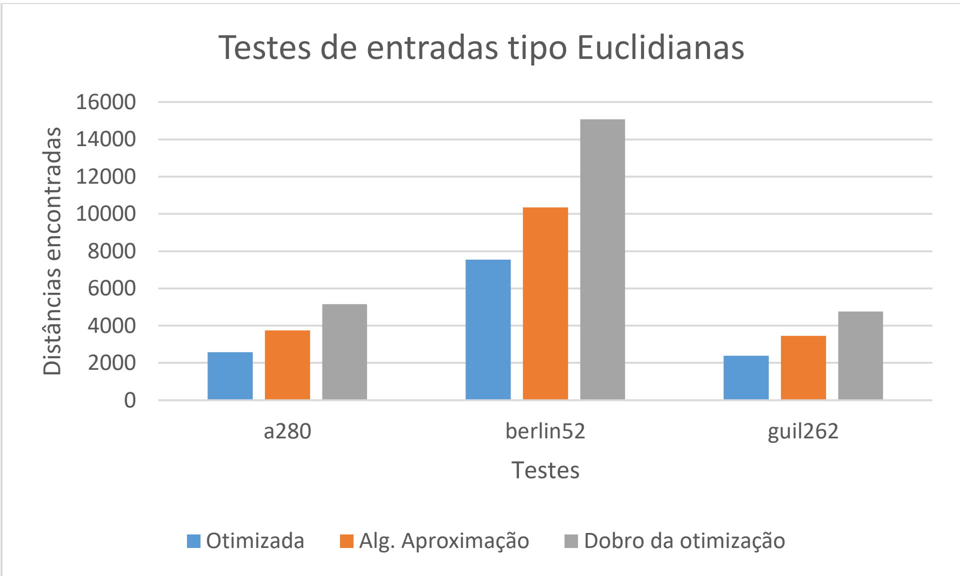

Figura 6: Testes de Entradas Tipo Euclidianas

Como pode ser observado, o algoritmo de aproximação proposto obteve valores maiores que os otimizados que seguiram a implementação de Christofides. E também respeitam a restrição que foi observada anteriormente, de que seriam menor que o dobro dos valores otimizados.

O gráfico abaixo tem o objetivo de demonstrar a eficiência dos atalhos obtidos com o curto-circuito. Observa-se primeiramente o número de arestas presentes na árvore geradora mínima. Ao aplicar a busca em profundidade com curto-circuitos, gerando atalhos, as arestas percorridas não representam o total de arestas que o grafo possui. Os dados foram obtidos fazendo a média dos testes realizados com os tipos Geo e Euclidianas.

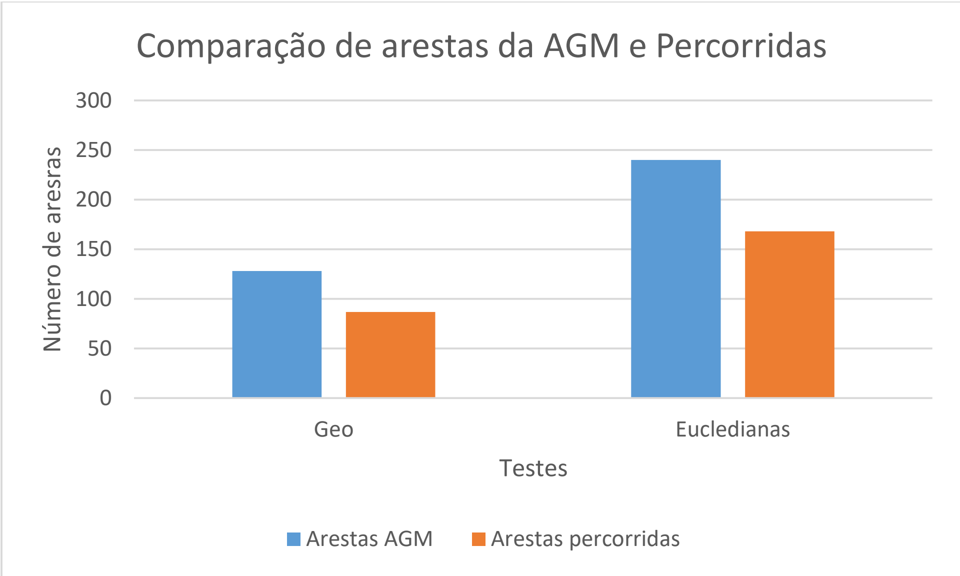

Figura 7: Comparação de arestas da AGM e arestas percorridas

Os tempos de usuário e sistema obtidos com os testes foram nulos. Observa-se que o número de vértices não eram grandes o suficiente para gerar tempos maiores que zero.

6 CONCLUSÃO

A implementação apresentada baseada na estratégia do algoritmo de Christofides possibilitou a obtenção de resultados aproximados em tempo polinomial para um problema exponencial. Ainda foi possível estabelecer uma razão de aproximação satistatória, tendo o resultado apresentado sua razão de aproximação garantida para o pior caso Ra <= 2.