PONTIFÍCIA UNIVERSIDADE CATÓLICA DE MINAS GERAIS
Instituto de Ciências Exatas e de Informática

Relatório Trabalho Prático N.02 - K-centros

Resumo

Este relatório descreve e compara a implementação de dois métodos para determinar K centros em um grafo completo com custos nas arestas, sendo um método para determinar o raio de forma exata e outro de forma aproximada. Os métodos implementados recebem um arquivo contendo os vértices de origem e destino de cada aresta e o número de K centros a serem encontrados, e exibem os centros e o raio encontrado.

1 INTRODUÇÃO

O problema dos k-centros consiste em, dado um conjunto de V vértices, as distâncias entre eles e um inteiro k, encontrar um subconjunto C com k vértices que minimiza o raio da solução. Esse raio é definido como a maior distância entre qualquer vértice do grafo e o centro do conjunto C que lhe for mais próximo.

Encontrar a solução exata para este problema pode ser computacionalmente inviável para grafos grandes, o que nos força a buscar diferentes estratégias. Por isso, este trabalho foca em explorar e comparar duas soluções distintas para o problema.

Este relatório apresenta a implementação de dois algoritmos, um de força bruta, que encontra uma solução exata ao testar todas as combinações possíveis de k centros. E que embora garanta o melhor resultado, tem um custo computacional muito alto, o que o restringe a instâncias pequenas. E um algoritmo de aproximação baseado no algoritmo de Gonzalez, que faz uma busca gulosa para encontrar os k-centros. Este método é muito mais rápido e busca uma solução "boa o suficiente"de forma eficiente, tornando viável a resolução para grafos maiores. Para avaliar e comparar as duas abordagens, ambas as implementações foram testadas utilizando as 40 instâncias de teste da OR-Library, conforme especificado no escopo do trabalho. A análise final foca nas diferenças de desempenho em eficácia (o quão bom é o raio encontrado) e eficiência (o quão rápido o algoritmo executa), destacando as vantagens e desvantagens de cada método.

2 DESENVOLVIMENTO

2.1 Representação do Grafo

```
Map<Integer, Map<Integer, Integer>> grafo = new HashMap<>();
```

O grafo direcionado foi representado por meio de uma estrutura de dados baseada em um mapa, em que cada chave corresponde a um vértice do grafo e o valor associado a ele é uma lista de pares representando as arestas de saída e o peso da aresta.

Foi executado um algoritmo para encontrar os caminhos mínimos entre todos os vértices do grafo, baseado no algoritmo de Floyd-Warshall.

2.2 Solução Exata

A primeira abordagem implementada para resolver o problema dos k-centros foi um algoritmo de força bruta. O objetivo deste método é garantir a obtenção da solução ótima, ou seja, o menor raio possível, sem deixar nenhuma possibilidade sem testar.

O algoritmo consiste em gerar todas as combinações possíveis de k vértices que podem formar o conjunto de centros. Para cada uma dessas combinações, o algoritmo calcula o raio da solução correspondente, que é a maior distância de qualquer vértice do grafo ao seu centro mais próximo dentro daquela combinação específica. Ao longo do processo, a melhor solução encontrada (a combinação de centros com o menor raio) é armazenada e atualizada sempre que uma solução melhor é descoberta.

Para implementar a geração de todas as combinações, foi utilizada uma função recursiva que constrói as combinações de k centros. Quando uma combinação completa é formada, seu raio é calculado e comparado com o melhor raio global encontrado até o momento. Apesar de garantir uma solução ótima, por testar todas as combinações possíveis, a abordagem de força bruta sofre com o problema da explosão combinatória. O número de combinações a serem testadas, cresce muito com o aumento do número de vértices e de centros. Isso torna o algoritmo extremamente lento e computacionalmente inviável para todas as instâncias, exceto as menores.

No caso deste trabalho, o algoritmo só conseguiu gerar um resultado em tempo hábil para o arquivo "pmed1.txt", que era o único pequeno o suficiente.

Vértices	**K**	**Tempo de execução**	**Raio**	**Centros**
100	5	90,473 segundos	127,00	[4, 8, 56, 62, 77]

Tabela 1 – Resultado dos testes com solução exata

2.3 Solução Aproximada

Para lidar com as instâncias maiores do problema, onde a abordagem de força bruta se torna computacionalmente inviável, foi implementada uma solução aproximada. O método escolhido foi o algoritmo de Gonzalez.

O Algoritmo inicia escolhendo um vértice qualquer do grafo para ser o primeiro centro. Escolhemos o vértice 1 por padrão. A partir daí, os k-1 centros restantes são escolhidos um a um, selecionando-se sempre o vértice que possui a maior distância mínima para qualquer um dos centros já escolhidos.

Embora não garanta a solução ótima, o algoritmo de Gonzalez possui uma garantia de que o raio da solução encontrada nunca será maior que o dobro do raio ótimo. Sua baixa complexidade computacional o torna extremamente rápido e eficaz para grafos grandes.

```
//encontrar k centros
        for(int i = 1; i < k; i++){
            // Encontrar o vertice mais distante dos centros atuais
            int proxC = -1;
            double maior = -1.0;

            for(int j = 0; j < grafo.getnVertices(); j++){
                if(distanciasMinimas[j] > maior){
                    maior = distanciasMinimas[j];
```

```
                    proxC = j;
                }
            }
            centros[i] = proxC;

            // Atualizar as distancias minimas com o novo centro
               encontrado
            for(int l = 0; l < grafo.getnVertices(); l++){
                double disNovoC = grafo.getDist(l, centros[i]);
                if(disNovoC < distanciasMinimas[l]){
                    distanciasMinimas[l] = disNovoC;
                }
            }
        }
```

Arquivo	Vértices	K	Tempo de execução	Raio	Centros
pmed1.txt	100	5	0,2971 milissegundos	186,00	[1, 77, 63, 47, 16]
pmed5.txt	100	33	1,1617 milissegundos	71,00	
pmed6.txt	200	5	0,4218 milissegundos	138,00	[1, 118, 33, 108, 63]
pmed16.txt	400	5	0,9222 milissegundos	84,00	[1, 332, 118, 138, 291]

Tabela 2 – Resultado dos testes com solução aproximada

3 CONCLUSÃO

A implementação e a análise dos algoritmos exato e aproximado para o problema dos k-centros permitiram uma observação das suas características e limitações. Os testes práticos confirmaram as expectativas teóricas sobre o desempenho de cada abordagem, mostrando que quanto maior a qualidade da solução, maior o custo computacional para obtê-la.

O método exato, baseado em força bruta, cumpriu seu objetivo ao encontrar a solução ótima para a instância "pmed1.txt", com raio 127. No entanto, sua grande desvantagem, a explosão combinatória, foi a razão pela qual ele não conseguiu executar em tempo hábil para as demais instâncias. O número de combinações de centros a serem testadas cresce de forma astronômica com o aumento do número de vértices ou de centros (k), tornando a busca exaustiva computacionalmente inviável para problemas de escala maior, como o do arquivo "pmed6.txt"ou "pmed16.txt".

Para "pmed1.txt", com 100 vértices e 5 centros a serem encontrados, O algoritmo precisou avaliar em aproximadamente 75 milhões de possibilidades. Esta é uma tarefa computacionalmente pesada, mas ainda realizável, como demonstrado pelo tempo de execução de 90 segundos. Já para o arquivo "pmed6.txt", com 200 vértices e 5 centros a serem encontrados, o número de combinações salta para aproximadamente 2,5 bilhões. Este aumento de mais de 33 vezes no número de operações torna a execução em tempo hábil impraticável.

Em contrapartida, o algoritmo de aproximação de Gonzalez mostrou-se extremamente eficiente, gerando soluções para todas as instâncias em frações de segundo. Por realizar uma busca gulosa, permite diminuir a complexidade do problema, oferecendo uma solução de boa qualidade rapidamente. Nos testes, os raios encontrados foram consistentemente maiores que os ótimos (por exemplo, 84,00 para o arquivo "pmed16.txt", onde o ótimo é 47). Contudo, para um problema onde a alternativa exata é inviável, obter uma resposta razoável de forma quase instantânea representa um ganho imenso.