



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

Trabalho Prático 2*

Model - Pratical Exercise - ICEI - PUC Minas

Edmar de Oliveira Melandes Junior¹
Leon Júnio Martins Ferreira²
Vinicius Gabriel dos Santos Teixeira³

Resumo

Este artigo contém a documentação do Trabalho Prático II que aborda o Problema dos K-Centros, um desafio essencial na área da otimização combinatória, com aplicações amplas em setores como transporte, logística e telecomunicações. O objetivo é minimizar a distância máxima entre os vértices de um grafo ponderado e seus centros correspondentes. Neste estudo, comparamos algoritmos de força bruta, aproximações gulosas e técnicas de travessia mais distante, utilizando instâncias da OR-Library de grafos para avaliar o desempenho e a qualidade das soluções obtidas. Espera-se que essa análise contribua para a seleção adequada de abordagens em diferentes cenários, visando uma solução eficiente e eficaz para o Problema dos K-Centros. Os códigos do trabalho prático foram implementados em um mesmo projeto utilizando a linguagem C++.

Palavras-chave: Problema dos K-Centros. C++. Farthest First Traversal. Teoria dos Grafos.

* Artigo apresentado como documentação ao Instituto de Ciências Exatas e Informática da Pontifícia Universidade Católica de Minas Gerais para disciplina Teoria dos Grafos e Computabilidade.

¹ Bacharelado em Ciência da Computação da PUC Minas, Brasil– edmar.junior@sga.pucminas.br

² Bacharelado em Ciência da Computação da PUC Minas, Brasil– leon.ferreira@sga.pucminas.br

³ Bacharelado em Ciência da Computação da PUC Minas, Brasil– vgsteixeira@sga.pucminas.br

1 INTRODUÇÃO

O Problema dos K-Centros é um desafio fundamental na área da otimização combinatória, com ampla aplicação em diversas áreas, como transporte, logística e telecomunicações. Neste trabalho prático, exploramos diferentes abordagens para resolver o Problema dos K-Centros e analisamos o desempenho de algoritmos de força bruta, aproximações gulosas e técnicas baseadas em travessia mais distante.

O objetivo principal do Problema dos K-Centros é selecionar K centros em um grafo ponderado, de forma a minimizar a distância máxima entre cada vértice do grafo e seu centro mais próximo. Em outras palavras, buscamos identificar os K pontos que minimizam a maior distância entre qualquer vértice e seu centro correspondente. Essa problemática é NP-difícil, o que significa que não há algoritmos eficientes conhecidos para resolver o problema de forma exata em tempo polinomial.

Para avaliar o desempenho dos algoritmos, utilizamos instâncias da *OR-Library*, (OR-LIBRARY, 1990), de grafos, uma coleção amplamente utilizada de problemas de otimização em pesquisa operacional. Essas instâncias apresentam características diversas, como número de vértices e arestas, distribuição de pesos e estrutura topológica, permitindo uma análise abrangente e representativa dos algoritmos.

2 IMPLEMENTAÇÃO

Para o desenvolvimento dos algoritmos estudados neste trabalho, foi escolhida a linguagem C++ por ser uma linguagem multi-paradigma e de alta performance. Foi feita uma classe para manter a estrutura de dados do grafo, no qual foi representado utilizando uma lista de adjacência, bem como método de identificação do menor caminho entre cada vértice, além dos algoritmos a serem analisados, desenvolvidos para serem executados de forma sequencial.

2.1 Estrutura de Dados

As instâncias testadas obtidas através da *OR-Library*, são grafos simples não direcionados e conexos, as arestas são ponderadas porém não possuem pesos negativos. Foi escolhido a lista de adjacência como forma de representação do grafo, por consumir menos espaço em memória que a matriz de adjacência e maior flexibilidade.

2.2 Algoritmos

Foram implementadas três soluções para o problema dos K centros, a primeira abordagem garante solução ótima por meio da força bruta, já as outras duas são aproximações, a primeira é um algoritmo guloso baseado na travessia mais distante, a última abordagem se dá por um algoritmo puramente guloso escolhendo a melhor opção local.

2.2.1 Método I - Força Bruta

A primeira abordagem considerada é o algoritmo de força bruta, que examina todas as combinações possíveis de K-centros para determinar a solução ótima. Embora seja um método intuitivo, de fácil implementação e capaz de fornecer resultados exatos, sua complexidade computacional pode tornar inviável de ser aplicada em problemas reais. A comparação de todos os possíveis conjuntos de K-centros é um passo muito custoso, que deverá ser refletido nos resultados dos testes executados, aos quais serão discutidos na seção 3 deste relatório.

2.2.2 Método II - Travessia mais distante gulosa

O segundo algoritmo desenvolvido explora técnicas baseadas em travessia mais distante, que ao selecionar um primeiro vértice arbitrariamente (foi escolhido o vértice solução para $K = 1$), para os próximos $K - 1$ vértices busca identificar centros que maximizam a distância em relação aos vértices já selecionados. Essa abordagem é uma heurística que visa obter soluções aproximadas com melhor qualidade.

2.2.3 Método III - Algoritmo Guloso Puro

O último método implementado utiliza de aproximação puramente gulosa, que fornecem soluções sub-ótimas, mas de forma mais eficiente. Esses algoritmos fazem escolhas locais baseadas em critérios heurísticos, selecionando gradualmente os centros mais promissores. A análise comparativa desses algoritmos permite avaliar sua capacidade de encontrar soluções próximas à ótima, em um tempo razoável.

3 ANÁLISE DE RESULTADOS

A seguir será analisado os resultados de tempos obtidos por cada algoritmo, bem como sua eficiência ao comparar o raio de cada aproximação com os raios das soluções ótimas já previamente disponíveis. Os testes foram todos executados na mesma máquina (AMD Ryzen™ 5 5600G @ 3.9GHz) e no mesmo compilador com os mesmos parâmetros de otimização (GCC 12.2.0 -O3). Durante o planejamento dos testes definiu-se um limite superior de tempo (*TIME-OUT*), de 30 minutos para execução de cada algoritmo para cada instância.

Os resultados dos tempos de execução dos métodos aproximados estão graficamente representados na **Figura 1**.

Os resultados de eficácia do raio da solução dos métodos aproximados estão graficamente representados na **Figura 2**.

Todos os dados obtidos pelos testes estão disponíveis na **Tabela 1**.

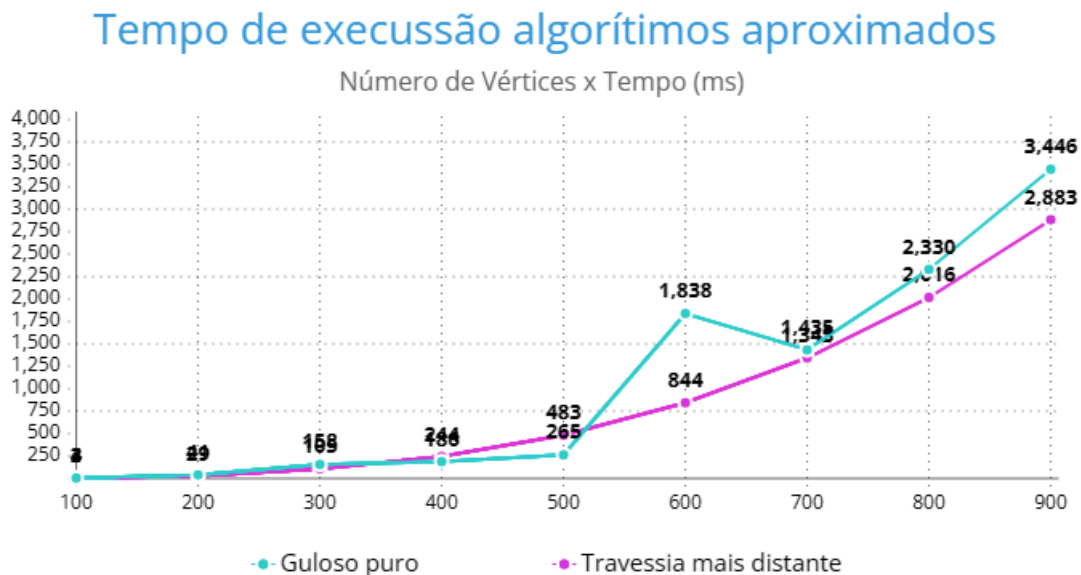


Figura 1 - Comparação de tempo algoritmos aproximados

3.1 Análise Comparativa

A imagem acima denota a ausência do método I (2.2.1), visto que este obteve *TIMEOUT* em quase todas as instâncias do teste, exceto as instâncias 1 e 6. Acredita-se que a falta de testes completos pelo algoritmo de força bruta se dá por que este testa todos os conjuntos de K-Centros possíveis entre cada par de vértices.

Ao analisar o método guloso puro, observou-se que ele apresentou tempos maiores em instâncias com menos arestas, isso se dá pois estas instâncias possuem um valor de K maior, com isso necessitando de mais comparações locais.

Finalmente no algoritmo de Travessia mais distante gulosa, é visível que este apresenta crescimento constate, visto que com o aumento do valor de K o número de comparações aumenta linearmente.

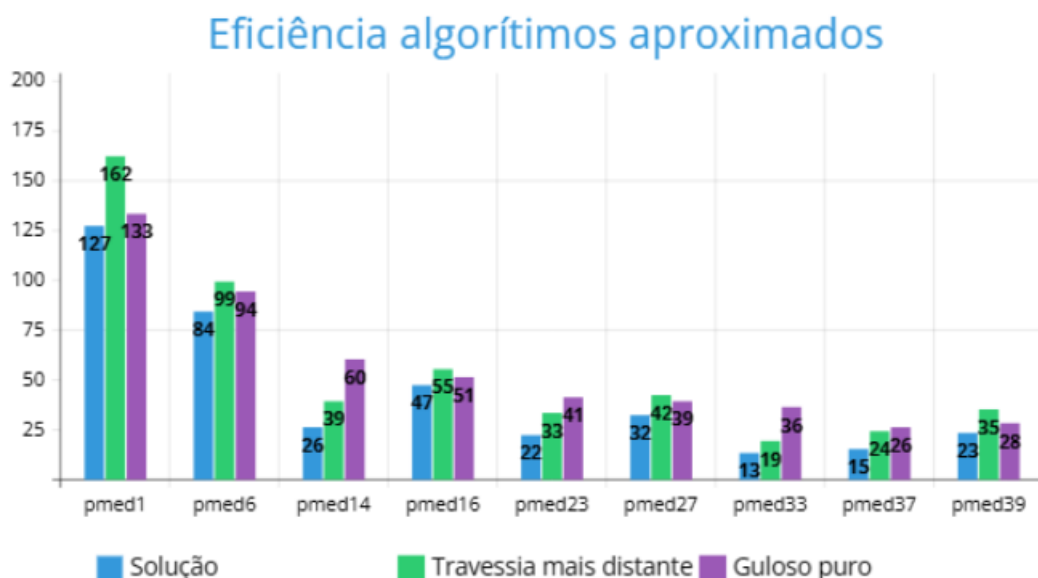


Figura 2 - Comparação de eficiência algoritmos aproximados

A eficiência dos raios obtidos nas instâncias denotadas na **Figura 2** no mostra que, no geral, o algoritmo de travessia mais distante demonstrou um desempenho superior em termos de eficiência em relação ao algoritmo guloso. Apresentou tempos médios de execução mais baixos e obteve soluções mais próximas da ótima na maioria das instâncias testadas. No entanto, é importante destacar que em algumas instâncias específicas, o algoritmo guloso se mostrou mais eficiente. Portanto, a escolha do algoritmo mais adequado depende do contexto específico, considerando a natureza do problema, as características das instâncias e as restrições de tempo disponíveis.

4 CONCLUSÃO

Neste artigo, abordamos o Problema dos K-Centros, um desafio fundamental na otimização combinatória, com amplas aplicações em diversos setores. Através da análise comparativa de algoritmos de força bruta, aproximações gulosas e técnicas de travessia mais distante, utilizando instâncias da *OR-Library*, pudemos obter *insights* valiosos sobre o desempenho e a qualidade das soluções obtidas.

Observamos que o algoritmo de força bruta, embora forneça soluções ótimas, é impraticável para problemas de maior escala, devido à sua complexidade computacional. Por outro lado, os algoritmos de aproximação gulosa mostraram-se razoavelmente eficientes, fornecendo soluções sub-ótimas.

Além disso, as técnicas baseadas em travessia mais distante demonstraram ser promissoras na busca por soluções aproximadas de alta qualidade. A escolha adequada da estratégia de travessia desempenhou um papel crucial no desempenho dessas abordagens.

Espera-se que este estudo contribua para uma compreensão abrangente do Problema dos K-Centros e das diferentes abordagens para resolvê-lo.

Tabela 1 – Tempos de execução e raios obtidos

Instância	 V 	K	Raio	Força Bruta	Travessia mais distante	Guloso puro
1	100	5	127	127 (21s)	162 (3ms)	133 (3ms)
2	100	10	98	TIMEOUT	124 (2ms)	117 (3ms)
3	100	10	93	TIMEOUT	133 (3ms)	116 (3ms)
4	100	20	74	TIMEOUT	99 (2ms)	127 (3ms)
5	100	33	48	TIMEOUT	64 (2ms)	87 (5ms)
6	200	5	84	84 (22min)	99 (28ms)	94 (29ms)
7	200	10	64	TIMEOUT	87 (29ms)	79 (31ms)
8	200	20	55	TIMEOUT	72 (29ms)	72 (35ms)
9	200	40	37	TIMEOUT	51 (29ms)	73 (45ms)
10	200	67	20	TIMEOUT	29 (30ms)	44 (64ms)
11	300	5	59	TIMEOUT	72 (106ms)	67 (107ms)
12	300	10	51	TIMEOUT	70 (106ms)	72 (110ms)
13	300	30	35	TIMEOUT	51 (105ms)	64 (126ms)
14	300	60	26	TIMEOUT	39 (104ms)	60 (173ms)
15	300	100	18	TIMEOUT	25 (104ms)	42 (273ms)
16	400	5	47	TIMEOUT	55 (247ms)	51 (248ms)
17	400	10	39	TIMEOUT	51 (246ms)	50 (250ms)
18	400	40	28	TIMEOUT	41 (249ms)	50 (308ms)
19	400	80	18	TIMEOUT	28 (248ms)	40 (458ms)
20	400	133	13	TIMEOUT	19 (249ms)	32 (786ms)
21	500	5	40	TIMEOUT	51 (495ms)	48 (513ms)
22	500	10	38	TIMEOUT	53 (476ms)	49 (481ms)
23	500	50	22	TIMEOUT	33 (474ms)	41 (617ms)
24	500	100	15	TIMEOUT	23 (480ms)	35 (1000ms)
25	500	167	11	TIMEOUT	15 (491ms)	27 (1993ms)
26	600	5	38	TIMEOUT	47 (841ms)	43 (843ms)
27	600	10	32	TIMEOUT	42 (841ms)	39 (849ms)
28	600	60	18	TIMEOUT	28 (838ms)	33 (1126ms)
29	600	120	13	TIMEOUT	19 (859ms)	36 (2025ms)
30	600	200	9	TIMEOUT	14 (840ms)	29 (4169ms)
31	700	5	30	TIMEOUT	38 (1337ms)	34 (1363ms)
32	700	10	29	TIMEOUT	43 (1338ms)	35 (1360ms)
33	700	70	15	TIMEOUT	25 (1346ms)	26 (1864ms)
34	700	140	11	TIMEOUT	17 (1350ms)	30 (3822ms)
35	800	5	30	TIMEOUT	37 (2013ms)	32 (2026ms)
36	800	10	27	TIMEOUT	41 (2014ms)	34 (2034ms)
37	800	80	15	TIMEOUT	24 (2021ms)	26 (2934ms)
38	900	5	29	TIMEOUT	38 (2890ms)	35 (2881ms)
39	900	10	23	TIMEOUT	35 (2872ms)	28 (2909ms)
40	900	90	22	TIMEOUT	22 (2887ms)	22 (4547ms)

Fonte: Elaborado pelos Autores

Referências

OR-LIBRARY. 1990. <<http://people.brunel.ac.uk/~mastjjb/jeb/info.html>>.