



Trabalho Prático II*

Model - Magazine Abakós - ICEI - PUC Minas

Fernanda Mendes Gomes¹
Pedro Olyntho Carvalho Alves²

Resumo

Este artigo apresenta a documentação do Segundo Trabalho Prático da disciplina Teoria dos Grafos e Computabilidade, referente ao problema dos k -centros. Este problema lida com a questão de se particionar elementos em conjuntos (clusters) de acordo com suas (dis)similaridades, que são representadas pelas distâncias entre vértices. Dado um grafo completo com custos nas arestas e um inteiro positivo k , o problema em questão objetiva encontrar um conjunto de k vértices (chamados centros) que minimize a maior distância de um vértice qualquer do grafo ao conjunto de centros

Neste trabalho, foram desenvolvidos dois métodos para a solução do problema. O primeiro consiste em um método de força bruta. Já o segundo, é uma implementação do método de Gon, introduzido em 1985. Para comparar as duas execuções, foram realizados experimentos em grafos disponíveis para OR-Library (pmed).

Palavras-chave: Grafos. Grafos Não Direcionados. Algoritmo de Gon. Força Bruta. Teoria dos Grafos e Computabilidade. K -centros. Gonzalez. Problema dos k -centros. Clustering.

*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, E-mail:fernandamendesgomes@gmail.com

Instituto de Ciências Exatas e de Informática da PUC Minas, Brasil.

², E-mail:pedro.olynto@sga.pucminas.br

Instituto de Ciências Exatas e de Informática da PUC Minas, Brasil.

1 INTRODUÇÃO

O problema dos K-centros em grafos é um problema clássico de clustering do tipo NP-difícil que objetiva selecionar k centros em um grafo ponderado de forma que a maior distância entre um vértice do grafo e seu centro mais próximo seja mínima. Apresenta diversas aplicações como logísticas de roteamento, localização de instalações e divisão em categorias.

Formalmente, pode ser definido como:

Problema dos k-centros: Dada uma coleção de V pontos (ou vértices) com distâncias definidas para cada par de pontos pela função $d : V \times V \rightarrow R^+$, encontrar um conjunto $C \subseteq V$, com $|C| \leq k$, cujos elementos são chamados de centros, de forma que a maior distância de um ponto (ou vértice) $v \in V$ ao centro mais próximo seja mínima.

No presente trabalho, serão exploradas duas maneiras de encontrar os centros de um grafo: a primeira encontra a solução exata utilizando força bruta, enquanto a segunda, busca soluções aproximadas. As implementações foram testadas utilizando as quarenta instâncias disponíveis na OR-Library para o problema das p-medianas.

2 IMPLEMENTAÇÃO

Esta seção apresenta a implementação do trabalho prático: descrição da classe Grafo utilizada no projeto e dos algoritmos desenvolvidos para encontrar o raio da solução do problema dos k-centros. Ademais, as classes Gon e ForçaBruta complementam a classe Grafo na resolução.

Cumprе ressaltar que durante o desenvolvimento, o ambiente Visual Studio Code foi empregado em conjunto com a linguagem Java. As operações implementadas no projeto foram implementadas de forma a serem executadas em uma única Thread de execução.

2.1 Grafo: Representação e Considerações Gerais

As soluções utilizando Força Bruta e o algoritmo de Gon têm em comum a classe Grafo, a qual é a principal do projeto. Na representação do grafo, optou-se pela utilização da estrutura de matriz de custo, onde são armazenadas as arestas e seus respectivos pesos. Os custos são armazenados no atributo *matrizCusto* da classe Matriz e seu tipo é *int[][]*.

A leitura dos arquivos da OR-Library e posterior população da matriz é implementada no método *gerarMatrizCusto*. Nesse método, o algoritmo de Floyd Warshall foi utilizado de forma a encontrar os caminhos mais curtos entre todos os pares de vértices do grafo.

2.2 Algoritmos

Esta subseção introduz os algoritmos desenvolvidos para a resolução do problema dos k-centros. Na primeira solução, foi utilizada força bruta. Já a segunda, parte de uma heurística aproximada introduzida por Gonzalez, Dyer e Frieze em 1985 (GONZALEZ, 1985)(DYER; FRIEZE, 1985).

2.2.1 Força Bruta

Na solução de força bruta todas as possíveis combinações de centros são testadas. Sendo assim, o resultado encontrado será sempre exato. No entanto, como o problema dos k-centros pode envolver muitas combinações, essa abordagem funciona apenas para instâncias pequenas.

A classe *ForcaBruta* possui três atributos: o primeiro uma instância da classe *Matriz*, o segundo o raio da solução e, por fim, o terceiro é o número de centros da solução desejada.

O método *gerarCombinacoes* inicia o processo de geração de combinações chamando a função *gerarCombinacoesRecursivamente*. Nesta, as combinações são geradas recursivamente e, para cada uma delas, é calculado o valor do raio. Se esse valor for menor do que o raio da solução encontrado até o momento, o atributo é atualizado.

2.2.2 Algoritmo de Gon

O algoritmo de Gon é uma solução aproximada para o problema dos k-centros que garante uma taxa de aproximação igual a dois. Isso quer dizer que o raio encontrado será, no máximo, o dobro da solução exata.

Nessa solução, o método *GonAlgo* é responsável por executar o algoritmo Gon. Ele inicia selecionando um centro aleatório entre os vértices do grafo e adiciona esse centro ao conjunto centros. Em seguida, para selecionar os outros centros, um loop é realizado *numCentros - 1* vezes, chamando a função *executar*. Nessa função, o novo centro é determinado a partir da seleção do vértice que está a maior mínima distância em relação aos centros previamente definidos. Ao final da seleção de centros, o raio da solução é calculado em *calcularRaio*.

3 ANÁLISE DOS RESULTADOS

Nesta seção são descritos os testes realizados utilizando cada uma das abordagens e seus resultados. Além disso, o desempenho das soluções propostas será discutido e comparado, levando em consideração sua eficiência e eficácia.

3.1 Testes Realizados

Para testar os algoritmos, foram utilizadas as quarenta instâncias da biblioteca OR-
Library para o problema das p-medianas. Os grafos definidos na biblioteca possuem entre
100 e 900 vértices e o número de centros definidos varia entre 5 e 200.

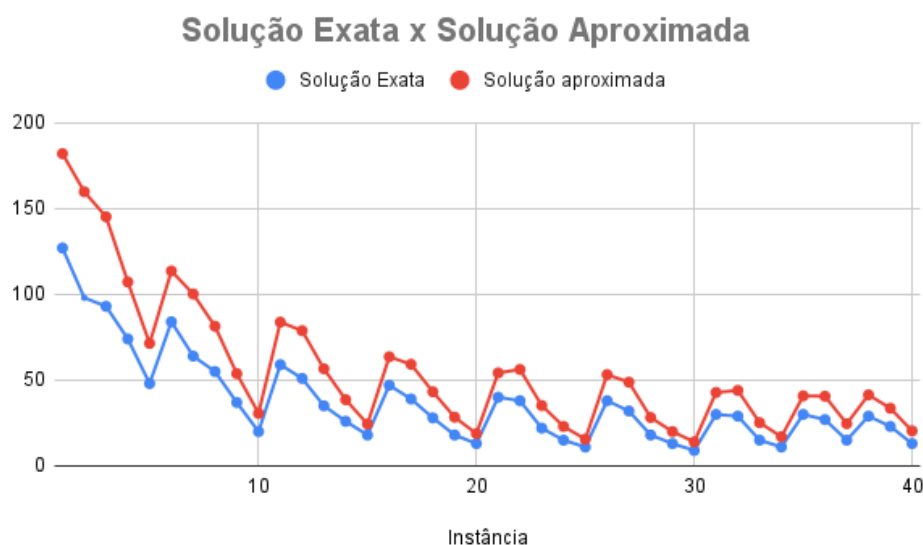
Os testes foram realizados em um processador intel i7 de sétima geração (2.8 GHz). No
caso de algum dos testes não ser concluído em até uma hora, a execução é abortada e o resultado
do teste é definido como *timeout* (*t.out*).

3.2 Análise Comparativa dos Resultados

Ao analisar a Tabela 01, é possível notar que a solução exata foi capaz de ser executada
apenas na primeira instância, que apresenta o menor tamanho de todas. Para todas as demais, o
tempo limite de uma hora foi excedido. Isso ocorre pois o algoritmo de Força Bruta itera sobre
todas as combinações de centros possíveis do grafo, o que gera números muito grandes. Para
a instância 01, a combinação $C(100, 5)!$ alcança 75.287.520 possíveis combinações, enquanto
para a segunda, $C(100, 10)!$, o resultado é 17.310.309.456.440. Por esse motivo, o cálculo da
solução exata, utilizando Força Bruta, torna-se inviável.

Já o Algoritmo Gon foi capaz de encontrar todas as soluções em tempo hábil, respeitando
a taxa de aproximação equivalente a dois. Ademais, examinando a Tabela 02, percebe-se
que apesar de haver alta variação média em relação ao raio em alguns casos, não houveram
resultados discrepantes para nenhuma das instâncias. Por fim, cabe destacar que a eficácia do
algoritmo não foi impactada pelo aumento do número de vértices, arestas ou centros.

**Figura 1 – Comparação da solução exata versus o resultado da solução aproximada para
o raio de cada instância**



4 CONCLUSÃO

Considerando a relevância do problema dos k-centros, o presente estudo examinou duas abordagens: exata, realizando um algoritmo de força bruta e aproximada, implementando o Algoritmo de Gon, todas utilizando uma única Thread de execução.

Com base nos testes realizados em grafos de tamanhos e número de centros variados, conclui-se que o cálculo da solução exata usando o método de Força Bruta não pode ser executado em grafos de grandes dimensões, devido ao elevado número de combinações possíveis. Já a solução aproximada introduzida por Gonzalez, Dyer e Frieze em 1985, com taxa de aproximação igual a 2, além de ter rápida execução, encontrou resultados aceitáveis.

Tabela 1 – Resultado dos raios encontrados para as instâncias da OR-Library utilizando Algoritmo de Força Bruta e Gon. Para o último, cada instância foi testada cinco vezes e a mediana dos resultados foi exibida

Instância	V	k	Raio	Força Bruta	Algoritmo Gon
1	100	5	127	127	188
2	100	10	98	T.OUT	166
3	100	10	93	T.OUT	147
4	100	20	74	T.OUT	112
5	100	33	48	T.OUT	71
6	200	5	84	T.OUT	114
7	200	10	64	T.OUT	102
8	200	20	55	T.OUT	81
9	200	40	37	T.OUT	54
10	200	67	20	T.OUT	31
11	300	5	59	T.OUT	85
12	300	10	51	T.OUT	77
13	300	30	35	T.OUT	57
14	300	60	26	T.OUT	38
15	300	100	18	T.OUT	24
16	400	5	47	T.OUT	62
17	400	10	39	T.OUT	61
18	400	40	28	T.OUT	43
19	400	80	18	T.OUT	28
20	400	133	13	T.OUT	19
21	500	5	40	T.OUT	53
22	500	10	38	T.OUT	56
23	500	50	22	T.OUT	35
24	500	100	15	T.OUT	23
25	500	167	11	T.OUT	16
26	600	5	38	T.OUT	52
27	600	10	32	T.OUT	47
28	600	60	18	T.OUT	28
29	600	120	13	T.OUT	20
30	600	200	9	T.OUT	14
31	700	5	30	T.OUT	42
32	700	10	29	T.OUT	44
33	700	70	15	T.OUT	26
34	700	140	11	T.OUT	17
35	800	5	30	T.OUT	41
36	800	10	27	T.OUT	41
37	800	80	15	T.OUT	25
38	900	5	29	T.OUT	41
39	900	10	23	T.OUT	33
40	900	90	13	T.OUT	20

Tabela 2 – Resultado dos raios encontrados para as instâncias da OR-Library utilizando Algoritmo Gon. Cada instância foi testada cinco vezes e foram calculadas as médias dos resultados, o limite inferior e superior e a variação para cada instância.

Inst.	Raio	Raio - Algoritmo Gon					Média	MIN	MAX	Variação Média
1	127	188	173	190	193	166	182	166	193	43,3
2	98	166	151	177	166	139	159,8	139	177	63,1
3	93	158	155	130	136	147	145,2	130	158	56,1
4	74	112	112	100	112	100	107,2	100	112	44,9
5	48	69	69	73	71	75	71,4	69	75	48,8
6	84	115	118	112	114	109	113,6	109	118	35,2
7	64	93	103	102	101	102	100,2	93	103	56,6
8	55	82	81	82	81	81	81,4	81	82	48,0
9	37	54	55	53	51	56	53,8	51	56	45,4
10	20	31	29	31	31	31	30,6	29	31	53,0
11	59	85	77	85	77	95	83,8	77	95	42,0
12	51	77	78	92	73	74	78,8	73	92	54,5
13	35	54	57	56	57	59	56,6	54	59	61,7
14	26	41	37	40	38	37	38,6	37	41	48,5
15	18	25	24	25	24	24	24,4	24	25	35,6
16	47	62	57	59	73	67	63,6	57	73	35,3
17	39	61	63	62	55	55	59,2	55	63	51,8
18	28	41	44	43	43	45	43,2	41	45	54,3
19	18	29	29	28	28	28	28,4	28	29	57,8
20	13	19	18	19	19	19	18,8	18	19	44,6
21	40	54	53	52	61	51	54,2	51	61	35,5
22	38	56	53	52	57	63	56,2	52	63	47,9
23	22	35	36	35	36	34	35,2	34	36	60,0
24	15	23	23	23	23	23	23	23	23	53,3
25	11	16	15	16	15	16	15,6	15	16	41,8
26	38	52	55	49	50	60	53,2	49	60	40,0
27	32	45	56	47	45	51	48,8	45	56	52,5
28	18	30	26	28	29	28	28,2	26	30	56,7
29	13	20	19	21	20	20	20	19	21	53,8
30	9	14	14	14	14	14	14	14	14	55,6
31	30	42	52	36	43	41	42,8	36	52	42,7
32	29	43	44	39	46	48	44	39	48	51,7
33	15	26	26	24	24	26	25,2	24	26	68,0
34	11	17	17	17	17	17	17	17	17	54,5
35	30	41	42	39	43	39	40,8	39	43	36,0
36	27	43	41	37	43	39	40,6	37	43	50,4
37	15	24	25	25	24	25	24,6	24	25	64,0
38	29	46	38	39	43	41	41,4	38	46	42,8
39	23	31	36	32	36	33	33,6	31	36	46,1
40	13	21	21	20	20	20	20,4	20	21	56,9

Referências

DYER, M.E; FRIEZE, A.M. A simple heuristic for the p-centre problem. **Operations Research Letters**, v. 3, n. 6, p. 285–288, 1985. ISSN 0167-6377. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0167637785900021>>.

GONZALEZ, Teofilo F. Clustering to minimize the maximum intercluster distance. **Theoretical Computer Science**, v. 38, p. 293–306, 1985. ISSN 0304-3975. Disponível em: <<https://www.sciencedirect.com/science/article/pii/0304397585902245>>.