CLUSTERIZAÇÃO POR MÍNIMA DISTÂNCIA: UMA ABORDAGEM UTILIZANDO ALGORITMO GENÉTICO

Roza Maria Zoellner Lopes (GTAO / UFPR) roza.lopes@ufpr.br
Cassius Tadeu Scarpin (GTAO / UFPR)) cassiusts@ufpr.br
José Eduardo Pécora Júnior (GTAO / UFPR) pecora@ufpr.br
Tarek Nasser Sati (GTAO / UFPR) tarek.sati@ufpr.br

Resumo

O problema de clusterização consiste no agrupamento de itens baseado em alguma característica em comum. No presente estudo realizou-se o agrupamento em clusters de pontos gerados aleatoriamente com o objetivo de obter a mínima distância entre eles. Utilizou-se a metaheurística Algoritmo Genético (AG) com soluções iniciais obtidas por heurísticas construtivas e pela metodologia *K-means*, sendo esta última melhorada pelo algoritmo *Local Search*. A fim de comprovar a eficiência da abordagem proposta comparou-se os dados obtidos com os resultados da solução exata do problema. Os resultados mostram que o Algoritmo Genético obtém soluções para instâncias onde não é possível obter o resultado exato.

Palavras-Chaves: Clusterização, mínima distância, Algoritmo Genético.

1. Introdução

A clusterização é utilizada para o agrupamento de dados com características semelhantes. Possui diversas aplicações: subdivisão de problemas grandes, onde resolvê-los por completo seria inviável; a determinação da abertura de centros de distribuição com o intuito de minimizar as distâncias percorridas; agrupamento de itens por categoria, entre outros (FUENTES; CADARSO; MARÍN, 2019; GEREMEW *et al*., 2018).

Com a facilidade de acesso à internet os clientes estão consumindo cada vez mais produtos diversificados e exigindo rápida entrega (WANG *et al*., 2020). Portanto, a clusterização é importante na área de transporte, pois agrupa itens visando minimizar a distância percorrida, os custos envolvidos e o tempo de entrega.

Segundo Talbi (2009) a clusterização é um problema considerado de difícil resolução. Portanto, para casos com grandes números de pontos pode ser inviável a sua solução exata, fazendo-se necessário o uso de outras metodologias como heurísticas e metaheurísticas.

Diante disso, o presente estudo tem o objetivo de resolver o problema de clusterização por distância utilizando a metaheurística Algoritmo Genético. Para testar a eficiência do método

os resultados obtidos foram comparados com a resolução do modelo exato resolvido com o Gurobi 8.0.

Com os resultados é possível constatar que a o Algoritmo Genético obtém a solução ótima para pequenas instâncias e consegue resolver problemas de maior escala em um tempo computacional baixo.

O artigo está estruturado em quatro seções. A seção 2 apresenta a revisão de literatura sobre a clusterização e o Algoritmo Genético. Na seção 3 são apresentados o problema e a metodologia proposta. A seção 4 contém a análise dos resultados obtidos. E por fim, na seção 5 apresentam-se as considerações finais do estudo.

2. Clusterização

Os algoritmos de cluster podem ser classificados em hierárquicos e de particionamento. O hierárquico obtém uma rede com uma série de divisões e o de particionamento agrupa os dados simultaneamente para otimizar de acordo com um critério ou função objetivo (NASCIMENTO; TOLEDO; CARVALHO, 2010). Os algoritmos de hierarquização mais utilizados são o link único e o link completo e o de particionamento mais utilizado é o *k-means* (JAIN, 2010).

A maioria dos algoritmos de clusterização são convertidos em problemas de otimização combinatória com o intuito de agrupar dados de acordo com uma função objetivo, porém para problemas de grande porte é difícil de se resolver computacionalmente devido à sua complexidade, portanto é necessário buscar meios de solução que demandem menor esforço computacional (JAIN, 2010).

São diversas as aplicações da clusterização na literatura, o Problema de Roteamento de Veículo Verde (GVRP) foi resolvido por Tiwari e Change (2015) utilizando uma abordagem de cluster para cada cidade do problema. Foi aplicada a recombinação de blocos com o intuito de diminuir a emissão de dióxido de carbono.

Defryn *et al*. (2016) resolveram o problema de roteamento colaborativo com três clientes, foi utilizada a clusterização para verificar o impacto da proximidade de clientes que pertencem a cada parceiro.

Geremew *et al*. (2018) solucionaram o problema de determinação de centros de distribuição para os clientes e também um centro principal pra abastecer os demais, sendo definido como

um problema de cluster hierárquico em dois níveis. Foi proposto dois algoritmos implementáveis baseados em DCA para resolver o problema.

Gatica *et al*. (2018) utilizaram *K-means* para determinar o centro do cluster para abastecimento de veículos elétricos visando encontrar o melhor local de abertura, minimizando os custos.

Fuentes *et al*. (2019) usaram a clusterização para o problema de agendamento. O método consiste em primeiramente realizar uma decomposição matemática *ad-hoc* com base na clusterização de tempo-pessoal. Após, foi utilizado o *Fix and Relax* para resolver o problema resultante.

Chao, Zhihui e Baozhen (2019) resolveram o problema de roteamento de localização e inventário com janela de tempo em uma rede de distribuição de alimentos em duas etapas, a primeira para o problema de localização e a segunda para o problema de transporte. Utilizaram uma abordagem de cluster baseado em distância para aumentar a eficiência da metodologia proposta, melhorando a velocidade de convergência da solução.

Cao *et al*. (2020) utilizaram um algoritmo de cluster baseado na metodologia hierárquica para agrupar regiões funcionais da trajetória do eletrodo na cirurgia de estimulação cerebral profunda.

Algumas das metaheurísticas mais utilizadas para a clusterização de dados são: *Tabu Search, Simulated Annealing, Greedy Randomized Adaptive Search Proce- dure (GRASP)* e o Algoritmo Genético (AG) (Nascimento *et al*., 2010).

2.1 Algoritmo Genético

Os Algoritmos Evolutivos (EA) obtém soluções aproximadas em um tempo computacional baixo se comparado com os métodos exatos e conseguem ótimos resultados para problemas grandes e complexos. Um dos algoritmos EA mais utilizados para problemas de otimização é o Algoritmo Genético (Cao *et al*., 2020).

Em um estudo realizado por Coello em 2017 que considerou os anos de 1988 à 2017 foram encontrados 4987 artigos que utilizaram algoritmos EA para resolver problemas de otimização multiobjetivo, sendo que em 56% dos artigos foi utilizado o Algoritmo Genético.

O AG foi proposto inicialmente por Holland em 1975 e encontra soluções com ótima qualidade para problemas NP difíceis realizando uma busca global, podendo assim, escapar da solução local (Wu *et al*., 2019).

O Algoritmo Genético é uma metaheurística baseada na teoria de Charles Darwin onde os seres mais aptos sobrevivem (CASTRO, 2001). A partir de soluções iniciais é realizada uma seleção aleatória de indivíduos para cruzar seus cromossomos e obter novos indivíduos. A nova população sofre alguma mutação com o intuito de diversificar as soluções encontradas (TALBI, 2009). Para obter a nova geração de indivíduos existem várias técnicas, entre elas, inserir o novo indivíduo na população e descartar o de pior função objetivo (TALBI, 2009; BARBOZA, 2010)

Segundo Lopes, Rodrigues e Arns (2013) os passos para a implementação do AG são: Obtenção da população inicial; Seleção dos indivíduos para reprodução; Realização do cruzamento entre os indivíduos pais; Mutação da nova população e verificação dos critérios de parada.

3. Descrição do problema

O problema do presente artigo consiste em determinar em qual cluster cada ponto será alocado com o objetivo de obter a soma mínima das distâncias entre os pontos de cada cluster. Cada ponto pode pertencer a apenas um cluster e a distância utilizada é a euclidiana.

O problema será resolvido utilizando o Algoritmo Genético. Para testar a eficiência do método proposto foi realizada a comparação dos resultados obtidos com as soluções do modelo exato.

3.1. Modelo matemático

O modelo exato utilizado neste artigo foi formulado por Nascimento, Todelo e Carvalho (2010) que modificaram o modelo proposto por RAO (1971):

Tabela 1 - Índices do modelo

Índices	
i	nó de partida
j	nó de chegada
k	cluster

Tabela 2 - Parâmetros do modelo

Parâmetros	
n	número total de vértices
m	número total de clusters
$d_{i,j}$	distância entre os vértices i e j

As variáveis de decisão são definidas como:

$$x_{i,k} = \begin{cases} 1 \text{ se o vértice i pertence ao cluster } k. \\ 0, \text{caso contrário.} \end{cases}$$

$$y_{i,j} = \begin{cases} 1 \text{ se os vértices } i \text{ e } j \text{ pertencem ao mesmo cluster.} \\ 0, \text{caso contrário.} \end{cases}$$

Função objetivo:

$$Min \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} d_{ij}\, y_{ij}$$

Sujeito a

$$\sum_{k=1}^{M} x_{ik} = 1, \qquad i = 1, \ldots, N \tag{1}$$

$$\sum_{i=1}^{N} x_{ik} \geq 1, \qquad k = 1, \ldots, M \tag{2}$$

$$x_{ik} \in \{0, 1\}, \qquad i = 1, \ldots, N \qquad k = 1, \ldots, M \tag{3}$$

$$y_{ij} \geq x_{ik} + x_{jk} - 1, \qquad i = 1, \ldots N \qquad j = i + 1, \ldots, N \qquad k = i, \ldots, M \tag{4}$$

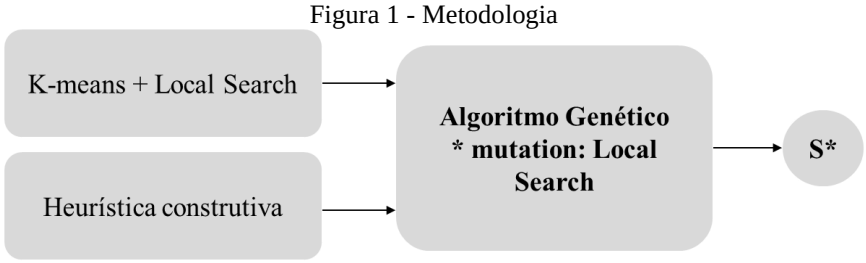
$$y_{ij} \geq 0, \qquad i = 1, \ldots, N \qquad j = i + 1, \ldots, N \tag{5}$$

A função objetivo visa obter a mínima distância entre os vértices do mesmo cluster. A restrição (1) determina que um vértice pode pertencer a apenas um cluster. A restrição (2) garante que o cluster k terá pelo menos um vértice . As restrições (3) e (5) determinam que as variáveis x_{ik} e y_{ij} são binárias. A restrição (4) garante que y_{ij} será 1 se tanto x_{ik} quanto x_{jk} forem 1.

4. Metodologia proposta

A figura 1 apresenta o método proposto para a resolução do problema de clusterização. São obtidas soluções iniciais a partir de heurísticas construtivas e do *k-means* que tem sua solução melhorada por meio de uma busca em sua vizinhança utilizando o *Local Search*. Essas soluções são utilizadas para alimentar o Algoritmo Genético que realiza uma busca global para obter a solução final.

Figura 1 - Metodologia



Fonte: os autores (2020)

A população inicial é formada por dez indivíduos, o primeiro é obtido por meio do *k-means* melhorado pelo *Local Search* e a outros nove por uma heurística construtiva.

No *k-means* os pontos centrais dos clusters são escolhidos aleatoriamente. Após, é realizado o cálculo da distância entre todos os pontos e o ponto central (centroide) de cada cluster. Os pontos são designados para o cluster mais próximo até que todos tenham sido alocados. Os centroides são recalculados a partir da média de todos os elementos pertencentes ao cluster k. O algoritmo continua a atribuição dos clusters e cálculo dos centroides até que não haja diferença entre a solução atual e a anterior. Meilă (2006) destaca que o *k-means* é um algoritmo que converge rapidamente para um ótimo local. A partir da solução inicial obtida pelo *k-means* é realizada uma busca em sua vizinhança com a aplicação do *Local Search*. A cada iteração a melhor solução é guardada. O critério de parada estabelecido é de 20 iterações sem melhoria.

É utilizada uma heurística construtiva para obtenção de soluções iniciais variadas com o intuito aumentar a diversidade dos cruzamentos. Na heurística construtiva o ponto central de cada cluster é escolhido aleatoriamente e os outros pontos são designados para o cluster mais próximo. Diferente do *k-means*, o processo é feito apenas uma vez, sem recalcular os centroides.

Três indivíduos são selecionados aleatoriamente, os dois com menor distância são escolhidos para a realização do cruzamento (crossover), o ponto de troca é selecionado de forma aleatória. São obtidas duas novas soluções a partir do crossover das soluções selecionadas. As duas piores soluções da população anterior são removidas e as duas novas são adicionadas. Para cada um dos 10 indivíduos da população é sorteado um número aleatório entre 0 e 1, os indivíduos que obtiverem valor menor ou igual a 0,2 sofrerão uma mutação em sua estrutura através da aplicação do *Local Search*. O método continua até que sejam obtidas dez iterações sem melhoria.

Tabela 3 – Pseudocódigo

Pseudocódigo - Algoritmo Genético
1 Obter 9 soluções iniciais a partir da heurística construtiva
2 Obter 1 solução inicial utilizando *k-means+local Search*
3 **Enquanto** não houver 10 iterações sem melhoria, faça:
4 Escolher aleatoriamente 3 indivíduos da população de soluções iniciais
5 Selecionar os dois com menor distancia (M1 e M2)
6 Selecionar aleatoriamente o ponto t para realizar a troca
7 **Para** i < t
8 Nova Solução 1[i] =M1[i]
9 Nova Solução 2[i] = M2[i]
10 **end**
11 Para i ≥ t
12 Nova Solução 1[i] = M2[i]
13 Nova Solução 2 [i] = M1[i]
14 **end**
15 Realizar o cálculo da Função objetivo de cada indivíduo da solução inicial
16 Retirar as duas piores soluções
17 Colocar na população inicial a Nova Solução 1 e a Nova Solução 2
18 Sortear um número n entre 0 e 1 para cada elemento da população
19 **Para** os elementos com n ≤ 0,20, faça:
20 Aplicar o *Local Search*
21 **end**
22 Guardar o melhor indivíduo da população gerada
23 **end**

Fonte: os autores (2020)

5. Resultados computacionais

Foram resolvidas 47 instâncias geradas aleatoriamente com 2,4,7 e 10 clusters e 20, 50 e 200 vértices. Foi utilizado o Visual Basic 2017 na linguagem de programação C# em um computador Intel core i5-8250U, com 8Gb de memória ram. O modelo exato foi resolvido com o software Gurobi 8.0 em suas configurações padrões.

Como critério de tempo computacional para a obtenção da solução foi determinado o tempo máximo de seis horas para cada instância.

Os resultados computacionais são apresentados na tabela 4. Como medida de comparação foi utilizado o tempo computacional para obter a solução e o GAP entre a metodologia proposta e o modelo exato que é determinado pela fórmula: $Sol_m - Sol_e/Sol_m$, onde Sol_m é a solução obtida com a metaheurística e Sol_e a solução do modelo exato. É possível observar que a metodologia proposta alcança o resultado ótimo em 86,21% das instâncias com até 50 pontos. Para as instâncias onde não foi possível obter a solução ótima com a metaheurística o maior GAP encontrado foi de 3,63%. Para 50 pontos e 4 clusters o modelo exato aumenta consideravelmente o tempo de resolução, até que em instâncias com 200 pontos não é possível obter a resposta no tempo máximo de 6 horas. O AG resolve instâncias com 200 vértices em um tempo computacional baixo, não ultrapassando 7 minutos.

Tabela 3 – Resultados

Instância	Pontos	Clusters	FO Metodologia Proposta	Tempo(s)	FO do modelo exato	Tempo(s)	GAP
1	20	2	8348,68	0,67	8348,68	0,35	0,00%
2	20	2	8009,9	0,57	8009,9	0,14	0,00%
3	20	2	8201,6	0,48	8201,6	0,05	0,00%
4	20	2	6891,74	0,48	6891,74	0,03	0,00%
5	20	2	8410,65	0,51	8410,65	0,07	0,00%
6	20	2	9365,53	0,5	9365,53	0,09	0,00%
7	20	2	7845,59	0,51	7845,59	0,07	0,00%
8	20	2	6390,47	0,49	6390,47	0,19	0,00%
9	20	2	9667,85	0,48	9667,85	0,17	0,00%
10	20	2	8991,27	0,45	8991,27	0,07	0,00%
11	20	2	12523,03	0,55	12523,02	0,25	0,00%
12	20	2	12014,85	0,55	12014,85	0,11	0,00%
13	20	2	10866,39	0,51	10866,39	4,76	0,00%
14	20	2	10337,61	0,49	10337,61	0,06	0,00%
15	20	2	12615,98	0,49	12615,98	0,06	0,00%
16	20	2	14048,29	0,51	14048,29	0,07	0,00%

17	20	2	10896,24	0,52	10896,24	0,4	0,00%
18	20	2	9585,71	0,54	9585,71	0,06	0,00%
19	20	2	14293,55	0,5	14293,55	0,25	0,00%
20	20	2	13486,91	0,49	13486,91	0,35	0,00%
21	50	2	102967,1	8,4	102967,1	4,12	0,00%
22	50	4	27353,42	3,76	27175,58	8381,23	0,65%
23	50	4	32614,42	3,77	32359,9	19952,63	0,78%
24	50	4	27483,05	4,31	27483,05	2072,32	0,00%
25	50	4	29114,41	3,12	29114,41	179,19	0,00%
26	50	4	30562,3	4,31	30562,3	4836,12	0,00%
27	50	4	33863,75	3,55	32634,85	9443,27	3,63%
28	50	4	30824,74	3,61	30824,74	273,26	0,00%
29	50	4	29154,91	3,49	28923,1	1568,17	0,80%
30	200	4	593911,3	414,77	-	-	-
31	200	4	780548,61	403,37	-	-	-
32	200	4	865205,03	936	-	-	-
33	200	4	926090,71	385,95	-	-	-
34	200	4	940394,92	405,56	-	-	-
35	200	4	657299,91	409,97	-	-	-
36	200	4	724547,7	469,26	-	-	-
37	200	7	245336,09	318,82	-	-	-
38	200	10	266096,31	193,457	-	-	-
39	200	10	211931,55	239,47	-	-	-
40	200	10	208012,1	165,43	-	-	-
41	200	10	238713,72	225,12	-	-	-
42	200	10	194576,94	217,39	-	-	-
43	200	10	188248,07	174,42	-	-	-
44	200	10	238591,99	178,04	-	-	-
45	200	10	202627,39	191,94	-	-	-
46	200	10	225438,37	183,7	-	-	-
47	200	10	173109,61	251,66	-	-	-

fonte: Os autores (2020)

6. Considerações finais

O objetivo do presente estudo foi propor uma metodologia utilizando metaheurísticas para a resolução do problema de mínima distância entre vértices pertencentes a um cluster para a resolução de problemas onde não é possível obter a solução exata em um tempo computacional baixo.

A metodologia proposta consiste na utilização do Algoritmo Genético a partir de soluções iniciais obtidas por uma heurística construtiva e pelo *k-means* melhorado pelo *Local Search*.

O Algoritmo Genético é utilizado pois o *k-means* converge rapidamente para um ótimo local enquanto o AG realiza uma busca global.

Para testar a eficiência da metodologia o modelo exato proposto por Nascimento, Todelo e Carvalho (2010) foi resolvido de forma exata e os resultados da função objetivo e tempo computacional foram comparados com o AG.

Como resultado da pesquisa obteve-se que a metodologia proposta é eficiente pois obtém a solução ótima em pequenas instâncias e nas maiores consegue uma solução em tempo computacional baixo enquanto o modelo exato não obtém resposta.

REFERÊNCIAS

BARBOZA, A. O. Simulação e técnicas da computação evolucionária aplicadas a problemas de programação linear inteira mista. 2010.

CAO, L. et al. Automatic feature group combination selection method based on GA for the functional regions clustering in DBS. **Computer Methods and Programs in Biomedicine**, v. 183, 2020.

CASTRO, R. E. DE. Otimização de estruturas com multi-objetivos via algoritmos genéticos. **PhD**, p. 206, 2001.

CHAO, C.; ZHIHUI, T.; BAOZHEN, Y. Optimization of two-stage location–routing–inventory problem with time-windows in food distribution network. **Annals of Operations Research**, v. 273, n. 1–2, p. 111–134, 2019.

DEFRYN, C.; SÖRENSEN, K.; CORNELISSENS, T. The selective vehicle routing problem in a collaborative environment. **European Journal of Operational Research**, 2016.

FUENTES, M.; CADARSO, L.; MARÍN, Á. A hybrid model for crew scheduling in rail rapid transit networks. **Transportation Research Part B: Methodological**, 2019.

GATICA, G. et al. Efficient heuristic algorithms for location of charging stations in electric vehicle routing problems. **Studies in Informatics and Control**, v. 27, n. 1, p. 73–82, 2018.

GEREMEW, W. et al. A DC programming approach for solving multicast network design problems via the Nesterov smoothing technique. **Journal of Global Optimization**, v. 72, n. 4, p. 705–729, 2018.

JAIN, A. K. Data clustering: 50 years beyond K-means. **Pattern Recognition Letters**, v. 31, n. 8, p. 651–666, 2010.

LOPES, H. S.; RODRIGUES, L. C. DE A.; ARNS, M. T. S. **Meta-Heurísticas em Pesquisa Operacional**. Editora Omnipax. 2013.

MEILĂ, M. The uniqueness of a good optimum for K-means. **ACM International Conference Proceeding Series**, v. 148, p. 625–632, 2006.

NASCIMENTO, M. C. V.; TOLEDO, F. M. B.; DE CARVALHO, A. C. P. L. F. Investigation of a new GRASP-based clustering algorithm applied to biological data. **Computers and Operations Research**, v. 37, n. 8, p. 1381–1388, 2010.

TALBI, M. E. **Metaheuristics : from Design to Implementation Single solution-based metaheuristics**. John Wiley & Sons. 2009.

TIWARI, A.; CHANG, P. C. A block recombination approach to solve green vehicle routing problem. **International Journal of Production Economics**, 2015.

WANG, Y. et al. Collaborative multi-depot logistics network design with time window assignment. **Expert Systems with Applications**, v. 140, 2020.

WU, Z.; ZHAO, C.; LIU, B. Polygonal Approximation based on Coarse-grained Parallel Genetic Algorithm. **Journal of Visual Communication and Image Representation**, p. 102717, 2019.