Problemas em Grafos: Problema de Máxima Cobertura

Marco A. E. Santo[1], Mateus R. Gonçalves[2], Matheus M. Aguiar[3], Rafael A. Batista[4]

Resumo

O Problema da Máxima Cobertura consiste em localizar em um grafo os nodos que selecionados cubram a maior área possível, podendo ser representada como a maior quantidade possível de vértices, através da definição de k subconjuntos, maximizando a união entre os mesmos, ou seja, maximizar a área máxima coberta. É considerado um problema NP-completo clássico da teoria da complexidade computacional e possui diversas aplicações em problemas reais, como a procura da melhor localização para infraestruturas ou qualquer tipo de problema que exija uma cobertura máxima de uma área/conjunto. O objetivo deste artigo é apresentar um algoritmo heurístico baseado em Algoritmos Genéticos para a resolução deste problema. É esperado através deste estudo ser capaz de determinar de forma eficaz possíveis melhores locais com a intenção de maximizar a cobertura sobre um conjunto de sub-conjuntos. Será abordado neste artigo uma aplicação real e prática deste clássico problema da otimização, seu uso na localização de postos de ajuda humanitária em um assentamento de refugiados. No decorrer do artigo será abordado com maiores detalhes as escolhas tomadas e os pormenores do desenvolvimento. Foram obtidos resultados satisfatórios na solução do problema, tendo um grande potencial para problemas de larga escala. Este trabalho contribui com uma abordagem própria de resolução do problema da Máxima Cobertura, sendo consideravelmente confiável.

Palavras-chave: Máxima Cobertura. Problema de Localização da Máxima Cobertura. Otimização. Grafo. Algoritmo Genético.

1 – Introdução

O Problema de Localização, conforme citado por Fuller (1997), tem por objetivo escolher um conjunto de pontos de facilidades que "cobrem" um dado conjunto de pontos de demanda. Um ponto de facilidade cobre um ponto de demanda se este ponto de demanda está dentro de uma dada métrica de um ponto de facilidade (SANTOS; MÜLLER, 2006). O Problema que será abordado neste estudo é o da Máxima Cobertura, que consiste em localizar em um grafo os nodos que selecionados cubram a maior área possível através da definição de k subconjuntos, maximizando a união entre os mesmos, ou seja, maximizar a área máxima coberta, que neste caso pode ser vista como uma maior quantidade de vértices do grafo. É considerado um problema NP-completo clássico da teoria da complexidade computacional.

O problema da Máxima cobertura possui diversas aplicações no mundo real, tendo uma grande importância no campo de estudo da Otimização. Desde a escolha de locais mais adequados para a instalação de facilidades, até escolha sobre conjuntos, sempre tendo como objetivo a cobertura máxima possível. Será abordado o seu emprego no problema dos assentamentos de ajuda humanitária. Um problema já recorrente que tem ganhado muita ênfase

[3]Autor correspondente: matheus.mmag@gmail.com
[1]Centro Federal de Educação Tecnológica de Minas Gerais
[2]Centro Federal de Educação Tecnológica de Minas Gerais
[3]Centro Federal de Educação Tecnológica de Minas Gerais
[4]Centro Federal de Educação Tecnológica de Minas Gerais

na atualidade pelo grande aumento do número de assentamentos ao redor do mundo. A localização dos postos de ajuda humanitária nestes assentamentos é crucial para garantir que uma maior quantidade de pessoas sejam atendidas.

Um mundo pacífico e próspero é aquele no qual as pessoas podem se sentir seguras e protegidas em suas casas, com suas famílias e em suas comunidades. É um mundo no qual elas podem se sentir confiantes em seu país, com sua cultura e na família das nações e dos povos do nosso planeta. Às vezes, por razões econômicas ou outras razões pessoais, as pessoas optam por deixar as suas casas e começar uma nova vida em um novo local. Para melhor ou pior, essas decisões são tomadas por uma questão de escolha consciente. Quando catástrofes naturais acontecem, casas são destruídas, deslocando comunidades inteiras(ONU, 2017).

Quando a guerra ou a agitação civil devastam uma comunidade, pessoas são deslocadas à força para proteger a vida e a integridade física. Elas têm apenas duas opções: a morte por privação, assaltos ou genocídios, ou a vida no exílio. Quando as pessoas escolhem viver em exílio, deslocando-se para um novo local, ocorre a formação de assentamentos de refugiados. Grande parte deste apoio é prestado através da Ação Humanitária das Nações Unidas. Abrigos de emergência são criados em cooperação com a Federação Internacional da Cruz Vermelha e do Crescente Vermelho. É de extrema importância a localização dos assentamentos de ajuda humanitária. Problemas como este apresentado, podem ser tratados utilizando meios computacionais (ONU, 2017).

Através deles que é prestado grande parte do auxílio às pessoas residentes destes assentamentos. Este artigo possui como propósito propor um algoritmo que busque a localização dos melhores pontos para a instalação dos postos de ajuda humanitária nestes assentamentos, de modo que permitam a máxima cobertura desses locais. Além do contexto desse problema apresentado, o desenvolvimento deste método de resolução do problema da Máxima Cobertura pode ser aplicado nos mais diversos contextos, sendo ampla a sua utilização e alta a sua eficácia.

No decorrer deste artigo será apresentado características referentes ao algoritmo desenvolvido, e o seu método de elaboração. Todo o processo de desenvolvimento do algoritmo e as principais características da heurística utilizada. Foram realizados diversos testes para avaliar a sua eficácia em comparação com o programa Cplex, que também tinha como objetivo resolver o mesmo problema. Os testes realizados, assim como a análise deles pode ser vista mais à frente deste trabalho.

2 – Desenvolvimento

O Problema de Localização de Máxima Cobertura (MCLP) foi introduzido por Church e Reville em 1974 possuindo a seguinte representação matemática (PRATA, 2017):

$$\text{Maximizar z} = \sum_{i=1}^{m} w_i y_i$$

Sujeito a:

$$\sum_{j=1}^{n} a_{ij} x_j \geq y_i, \forall i = 1, ..., m$$

$$\sum_{j=1}^{n} x_j = d$$

Onde:

$$x_j \in \{0, 1\}, \forall j = 1, ..., n$$

$$0 \leq y_i \leq 1, \forall i = 1, ..., m$$

$$x_j = \text{Recursos, 1 se coberto e 0 se não}$$

$$y_i = \text{Demanda, 1 se atendida e 0 se não}$$

$$w_i = \text{Bonificação pela cobertura do ponto i}$$

$$d = \text{Número máximo de recursos disponíveis}$$

$$a_{ij} = \text{Se o recurso j atende a demanda de i}$$

$$m = \text{Número de demandas}$$

$$n = \text{Número de recursos}$$

2.1 – O algoritmo

Para o desenvolvimento deste trabalho optamos por usar a heurística de Algoritmos Genéticos baseada na apresentada por Gomes (2017). O algoritmo genético é um tipo de algoritmo evolutivo que parte de uma população inicial buscando a evolução da mesma, através da aplicação de operações como **Seleção**, **Cruzamento** e **Mutação** até que se atinja algum dos critérios de parada definidos, os quais podem ser desde um número máximo de gerações (ciclos) ou quando já não se obtém mais uma diferença significativa entre as gerações.

2.1.1 Função de aptidão

A função de aptidão é aquela que transforma o valor da função objetivo em uma medida de aptidão relativa, desta forma de acordo com a função objetiva estabelecida no começo desta seção, definimos como função de aptidão a quantidade de demandas atendidas por aquela solução.

2.1.2 População Inicial

A população inicial é obtida de forma completamente aleatória, sendo vetores constituídos por campos binários, os quais definirão quais recursos são disponibilizados para aquela solução (1 se é disponibilizado, 0 se não), a partir destes recursos conseguimos calcular a quantidade de pontos de demanda atendidos, uma vez que possuímos esta relação definida em uma matriz.

2.1.3 Seleção

Nesta etapa é simulado o processo de seleção natural, onde os indivíduos com uma maior taxa de aptidão serão privilegiados em decorrência dos que tem uma taxa menor, para a realização deste procedimento foi utilizado a seleção por roleta. Esta seleção simula o funcionamento de uma roleta, onde cada indivíduo ocupará um campo da mesma e a sua área de ocupação será proporcional a sua taxa de aptidão. O processo pode ser melhor compreendido conforme o algoritmo 1.

Algorithm 1 Seleção

```
for i de 0 a POPULACAO do
    sum_fitness ← Somar_Fitness()
    sum_fitness ← rand
    selec[i] ← pegaInd(sum_fitness)
end for
```

2.1.4 Cruzamento

O Cruzamento tem como princípio básico a troca de genes entre dois indivíduos para a criação de novos indivíduos. Para a realização deste procedimento foi escolhido o cruzamento de ponto único, no qual é selecionado um ponto aleatório do cromossomo, o primeiro filho herdará a primeira parte do primeiro pai e a segunda parte do segundo pai, o segundo filho herdará o oposto. Caso não haja o cruzamento os dois pais serão copiados os genes dos pais gerando dois indivíduos idênticos a eles. O processo pode ser melhor compreendido conforme o algoritmo 2.

2.1.5 Mutação

A mutação tem como objetivo auxiliar na manutenção da diversidade genética, assim como a explorar novas possibilidades de soluções, uma vez que ela realiza a mutação dos indivíduos que foram selecionados e passaram pelo cruzamento. Para a realização da mutação selecionamos o método de mutação inversa,

Algorithm 2 Cruzamento

```
for i de 0 a POPULACAO, 2 em 2 do
    gera chance de crossover
    if chance <= CROSSOVER then
        for j de 0 a N/2 do
            Escolhe os recursos para o filho 1
            Escolhe os recursos para o filho 2
        end for
        for j de N/2 a N do
            Escolhe os recursos para o filho 1
            Escolhe os recursos para o filho 2
        end for
        calcula fitness do filho 1
        calcula fitness do filho 2
        if recursos dos filhos igual D then
            seleciona filho 1 pra posição i
            seleciona filho 2 pra posição i+1
        end if
    end if
end for
```

aonde é definida aleatoriamente uma região do indivíduo, os elementos que se encontram nesta região serão dispostos na ordem inversa. O processo pode ser melhor compreendido conforme o algoritmo 3.

Algorithm 3 Mutação

```
for i de 0 a POPULACAO do
    gera chance de mutação
    if chance <= MUTACAO then
        posEscolhido ← 0
        posNaoEscolhido ← 0
        for j de 0 a N do
            if o recurso foi escolhido then
                próxrecursoescolhido ← j
            else
                próxrecursoNescolhido ← j
            end if
        end for
        gera pos aleatório do escolhido
        gera pos aleatório do não escolhido
        pega o valor da posição do escolhido
        recurso escolhido recebe não escolhido
        atribui valor à ao não escolhido
    end if
end for
```

2.1.6 Atualização da população

Para a atualização da população optamos pela abordagem geracional, a qual substitui toda a população atual pela nova população, a qual foi obtida através dos métodos de Seleção, Cruzamento e Mutação, os quais já foram descritos neste artigo.

2.1.7 O Algoritmo Genético desenvolvido

A partir dos operadores genéticos descritos anteriormente vemos nesta seção o algoritmo genético que sintetiza todo o processo desenvolvimento, assim como os valores das taxas que foram estipulados, informações como o tamanho da população serão mostradas na seção de resultados, uma vez que foram testadas várias instâncias diferentes. O processo geral pode ser melhor compreendido conforme o algoritmo 4.

Taxa de cruzamento: 80%.

Taxa de mutação: 1%.

Algorithm 4 Algoritmo Genético

```
Gera população inicial
while não for o max de gerações do
    Realiza a seleção
    Realiza o crossover
    Realiza a mutação
    Atualiza população e calcula fitness médio
end while
```

3 – Resultados

Foram realizados oito testes no Cplex e no Algoritmo Genético (AG) para a comparação dos dados de Gap, Tempo de execução (T) e o valor retornado pela função objetivo (FO) em três casos de execução do algoritmo.

Abaixo mostramos a tabela com os dados de entrada para cada teste. O número de nós corresponde à quantidade de nós a serem atendidos, a coluna de candidatos se refere ao número total de recursos que podem atendê-los e máx. selecionados corresponde ao número de recursos que quero utilizar.

Entradas			
Teste	Nós	Candidatos	Máx.
1	5	7	2
2	25	7	2
3	50	10	2
4	200	10	5
5	1000	10	5
6	5000	50	10
7	10000	150	50
8	100000	250	150

Abaixo estão as tabelas de cada teste com seus valores de saída para o Cplex e o algoritmo genético. Tanto o Gap do algoritmo genético quanto o Gap do Cplex, são a porcentagem da distância da solução ótima, tendo em vista que no genético ele é feito com base na média das soluções.

Teste 1				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	0.13	80
	2	0	0.14	80
	3	0	0.14	80
AG	1	15.0	2.08	68
	2	18.75	2.40	65
	3	0	2.94	80

Teste 2				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	0.13	541
	2	0	0.15	541
	3	0	0.14	541
AG	1	2.7	2.72	526
	2	2.0	2.24	530
	3	6.4	2.76	506

Teste 3				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	0.03	1.2k
	2	0	0.03	1.2k
	3	0	0.04	1.2k
AG	1	0	2.21	1.2k
	2	0	2.46	1.2k
	3	0	2.46	1.2k

Teste 4				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	0.27	4.7k
	2	0	0.26	4.7k
	3	0	0.27	4.7k
AG	1	4.2	2.40	4.5k
	2	2.1	2.64	4.6k
	3	4.2	2.48	4.5k

Teste 5				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	0.14	23.9k
	2	0.4	0.16	23.8k
	3	0	0.15	23.9k
AG	1	0	2.69	23.9k
	2	0.8	3.01	23.7k
	3	0.8	2.98	23.7k

Teste 6				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	0.53	121.7k
	2	0	0.59	121.7k
	3	0	0.48	121.7k
AG	1	0.08	6.31	121.6k
	2	0	6.90	121.7k
	3	0	6.86	121.7k

Teste 7				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	0	5.16	247.3k
	2	0	6.01	247.3k
	3	0	5.39	247.3k
AG	1	0	44.86	247.3k
	2	0	45.27	247.3k
	3	0	45.69	247.3k

Teste 8				
Local	Caso	Gap(%)	T(s)	FO
CPLEX	1	-	-	-
	2	-	-	-
	3	-	-	-
AG	1	-	1286.67	1.5M
	2	-	1294.42	1.5M
	3	-	1289.36	1.5M

Na tabela 8, o Cplex não conseguiu atingir uma solução em tempo hábil(30 min). Por isso, não constam seus dados.

4 – Conclusão

O trabalho desenvolvido apresentou uma nova perspectiva para a solução do problema da máxima cobertura, abordando uma heurística robusta e confiável. É relevante para esta área de pesquisa, onde novos métodos de resoluções são sempre bem vindos, em especial, métodos como o que foi abordado que apresentam resultados positivos favoráveis.

Devido à simplicidade do modelo, o Cplex demonstrou um resultado mais eficiente que o algoritmo genético na grande maioria dos testes. Porém, para problemas de larga escala, o algoritmo genético apresenta um potencial para se tornar superior ao Cplex. Este que é o principal ogjetivo deste trabalho. O Cplex, até um certo momento, apresentou um comportamento adequado. Entretanto, à partir de um determinado ponto, houve um overflow na memória. Concluímos que foi satisfatório os resultados obtidos, assim como a contribuição deste trabalho para este importante ramo de pesquisa computacional.

Este trabalho pode ser estendido com um estudo mais profundo da implicância que os métodos utilizados de mutação e de cruzamento possuem sobre o resultado obtido, o tamanho da população e o número de interações. Um aprofundamento buscando conclusões a respeito de possíveis formas mais eficazes de realizar estes métodos. O aprimoramento deste trabalho pode vir grandemente de possíveis variações mais eficazes de realizar a mutação e o cruzamento.

Agradecimentos

Agradecemos aos professores João Fernando Machry Sarubbi e Rogério Martins Gomes pelos ensinos referentes ás matérias abordadas no artigo, assim como ao Centro Federal de Educação Tecnológica de Minas Gerais local onde atualmente estudam os integrantes deste grupo.

Abstract

The Maximum Cover Problem consists in locating on a graph the nodes that, if selected, cover the maximum possible area, which can be the number of vertices, through the definition of k subsets, maximizing the union between them - in other words, maximizing the cover area. It is considered as a classic NP-complete problem of the computational complexity theory and have a wide variety of applications on real problems, such as a search for the best location for infrastructures or any kind of problem that requires the maximum cover of an area or set. The objective of this article is to introduce a heuristic algorithm based on genetic algorithm to solve this problem. It is expected from this study to be able to determine possible maxima points efficiently, with the intention of maximizing the cover of a set or subset. This article will approach a real application of this classic problem of optimization, the location of humanitarian stations on a refugee camp. This article will approach the details of the decisions made during the development. The results obtained were satisfactory for solving the problem, containing high potential for large scale problems. This assignment contributes with a unique approach for solving the maximum cover problem, being considerably reliable.

Referências

GOMES, R. M. *Algoritmos Genéticos*. 2017. <https://ava.cefetmg.br/pluginfile.php/12787/mod_resource/content/1/Aula%2017_AlgoritmosGeneticos_1%C2%AAparte.pdf>. Acesso em: 29 de maio de 2017. 3

ONU. *A ONU e os refugiados*. 2017. <https://nacoesunidas.org/acao/refugiados/>. Acesso em: 07 de junho de 2017. 2

PRATA, P. B. de A. *Modelo Matemático*. 2017. <https://www.youtube.com/watch?v=rZhhe5vmnyI>. Acesso em: 12 de junho de 2017. 2

SANTOS, R. P.; MÜLLER, C. Problema de localização de máxima cobertura aplicado à localização de esquadrões de aeronaves de interceptação na região amazônica. XXXVIII Simpósio Brasileiro de Pesquisa Operacional, 2006. 1