Parametrização de um Algoritmo Genético para o Problema do Corte Máximo

Francisco Davi Gomes de Oliveira
Universidade Federal do Ceará (UFC)
Laboratório de Pesquisa do NEMO
Russas, Ceará, Brasil
davigomes@alu.ufc.br

Márcio Costa Santos
Universidade Federal do Ceará (UFC)
Laboratório de Pesquisa do NEMO
Russas, Ceará, Brasil
marciocs@ufc.br

Pablo Luiz Braga Soares
Universidade Federal do Ceará (UFC)
Laboratório de Pesquisa do NEMO
Russas, Ceará, Brasil
pablo.soares@ufc.br

***Resumo*—Este trabalho tem como finalidade a busca por boas soluções para o problema do corte máximo (Max-Cut) em um grafo, através da utilização de uma metaheurística baseada em algoritmo genético que por sua vez é inspirada no princípio teórico da evolução natural proposto por Charles Darwin. A ideia principal consiste na busca por uma configuração de parâmetros de refinamento do algoritmo genético que venha a apresentar bons resultados nas instâncias da literatura. O algoritmo genético foi implementado em linguagem C sendo centralizado em fornecer a otimização ao problema do corte máximo para grafos. Após a implementação do algoritmo, foram testadas 36 configurações diferentes de parâmetros de ajuste do algoritmo. Foram obtidos bons resultados, com o prevalecimento da seguinte configuração: "Com ilha, Mutação em 1 gene, Crossover que prevalece a igualdade entre os genes, e busca local a partir de 5 gerações sem convergências do melhor resultado(CIM1I5)".**

***Index Terms*—Problema do Corte Máximo, Algoritmo Genético, Configuração de Parâmetros.**

I. Introdução

Um grafo G é definido como um conjunto de vértices ou nós (V), com ligações conhecidas como arestas ou arcos (E), onde cada aresta estará associada a no máximo um par de vértices. Ou seja, com os vértices u e v $(u, v \in V)$, denota-se a aresta como o par (u, v) ou (v, u), sendo ambos iguais quando se trata de um grafo não direcionado, tal que (u, v), $(v, u) \in E$. Quando todo par $(u, v) \in E$ recebe um peso $p(u, v)$, então o grafo é conhecido como sendo ponderado [1].

Os grafos podem ser usados na representação de alguns problemas, como mapeamento, caminho mínimo ou máximo, fluxo máximo, dentre outros [2]. Dentre essas representações de problemas em grafo, existe o problema do corte máximo em um grafo ponderado, na qual o problema consiste na procura pela melhor configuração/partição dos vértices de V em dois subconjuntos, S e S^*, onde a soma do peso das arestas que cruzam os dois subconjuntos venha a obter a maior soma possível [3] ou corte máximo. Sendo classificado como um problema NP-Difícil [4], torna-se um desafio computacional obter o valor do corte máximo até mesmo para instâncias de tamanho intermediárias, grafos com aproximadamente 150 vértices. Portanto, para instâncias maiores, é necessário utilizar abordagens que forneçam soluções com boa qualidade em tempo aceitável.

Dessa forma, a busca no espaço de combinações de subconjuntos para encontrar o maior valor de corte possível será por meio da utilização de algoritmo genético, que por sua vez é um algoritmo de otimização baseado nos princípios da seleção natural e da evolução de Charles Darwin. Por ser uma meta-heurística, o algoritmo genético tem em sua natureza a aplicação de operadores genéticos, tais como, mutação e cruzamento para gerar novos indivíduos/soluções para o problema sem precisar gerar todas as possíveis soluções para o problema.

O restante desse trabalho está assim descrito: na seção II serão apresentadas as definições do problema de corte máximo, assim como a definição de algoritmo genético, as metodologias utilizadas no desenvolvimento do algoritmo genético otimizado para o problema do corte máximo, e o funcionamento do algoritmo construído; já na seção III, serão apresentados as configurações e parâmetros utilizados no algoritmo genético para a geração dos resultados obtidos nas diversas instâncias da literatura; Finalmente na seção IV, as conclusões e direcionamentos futuros sobre a pesquisa serão comentados.

II. Materiais e Métodos

A. Definição do problema de Corte Máximo

Seja $G(V, E)$ um grafo não direcionado, onde V representa o conjunto de vértices, com $|V| = n$ vértices, e E representa o conjunto das arestas, com $E = m$ arestas. O problema do corte máximo (Max-Cut) ponderado consiste em obter o peso com valor máximo para o somatório das arestas que liguem os subconjuntos S e S^*, de tal maneira que parte dos vértices de V estejam contidos no subconjunto S enquanto no subconjunto S^* estejam os $V \setminus S$ vértices restantes. A equação (1) mostra o somatório dos pesos $p(u, v)$ das arestas que cruzam os subconjuntos S e S^*.

$$CM(G) = \sum_{\substack{\forall u \in S \\ \forall v \in S^*}} p_{(u,v)} \qquad (1)$$

B. Algoritmo Genético

O Algoritmo genético é um método meta-heurístico de otimização que se baseia na teoria evolucionária de Charles Darwin, ou seja, os indivíduos de uma população irão sofrer mutações no decorrer das gerações, e a partir dessas transformações os melhores irão sobreviver com tendência a gerar herdeiros com característica(s) do(s) pai(s) [3]. A evolução das espécies pode ser interpretada como um processo de otimização, ao passo que, no decorrer do tempo os seres vivos vão se adaptando a um ambiente que evolui com constância [5]. Portanto, o algoritmo genético tem uma estrutura em que as informações de cada sistema podem ser processadas analogamente aos cromossomos biológicos [6].

O mapeamento dos conceitos, gene, cromossomo, população, mutação e *crossover*, de algoritmos genéticos com o problema do corte máximo são definidos a seguir da seguinte forma:

- ***Gene***: É uma posição do vetor que representa cada vértice do grafo. E nesta, pode está preenchida com a valoração de 0 ou 1, onde 0 representa que o vértice está contido no subconjunto S e 1 representa que o vértice está contido no subconjunto S^*;
- ***Cromossomo***: É um vetor com n genes. Ou seja, é uma sequência binária que representa uma possível configuração/solução de como estão particionados os vértices entre os subconjuntos S e S^*;
- ***População***: É uma matriz contendo em cada linha um cromossomo, ou ainda, um ambiente com configurações de possíveis soluções que é evolutiva ao decorrer das transformações feitas em cima de cada cromossomos;
- ***Mutação***: Consiste na ação de alterar o valor binário de um gene, em um cromossomo da população, para o inverso do estado atual, ou seja, alterar o vértice de um subconjunto para o outro, a fim de ampliar o seu espaço de busca e verificar se aquele novo cromossomo obtido seja melhor do que algum cromossomo da população atual;
- ***Crossover***: Consiste na ação de escolher dois pais/cromossomos dentro da população, e fazer a troca de genes entre esses dois cromossomos, gerando assim dois filhos com características dos dois pais, que terão tendências a sobreviver nessa população e tendendo a serem alocados na mesma, aquele que seja melhor dentre o espaço de soluções.

C. Aplicação do algoritmo genético para tratar o problema de Corte máximo de um grafo

O desenvolvimento do Algoritmo Genético, com representação da solução em binário, consiste em um algoritmo simples e de fácil implementação, foi realizada em linguagem C no ambiente de desenvolvimento Visual Studio. desenvolvimento se deu através dos seguintes passos:

1) Carregar um conjunto de arestas de um grafo G ponderado contidos de um arquivo de texto que contém as seguintes informações: quantidade de vértices e quantidade de arestas do grafo na primeira linha; e nas demais linhas informações referentes as arestas, sendo o primeiro dado informando qual o vértice u, o segundo dado para o vértice v, e o terceiro dado da linha está disposto para o peso da aresta (u, v);
2) Ao carregar os dados do grafo G, é realizado um cálculo para obter a quantidade de soluções possíveis que este determinado grafo pode ter, onde o número total de soluções será:

$$S(G) = 2^n \tag{2}$$

onde, n é a quantidade de vértices do grafo e para cada vértice têm-se 2 valorações que podem ser atribuídas (0 ou 1). Após esse cálculo, será criado a população inicial um tamanho limitado t, definido na função de criação de população, pois trata-se de um problema que pode conter um número exponencial de soluções a depender de n. Simultaneamente será criado um vetor contendo os valores de corte para cada cromossomo da população, e em seguida, o vetor é utilizado para ordenar a população, através do algortimo de *Insertion Sort*, de acordo com o maior valor de corte. Para auxiliar os próximos passos, é criado uma nova matriz de tamanho $t + \frac{t}{2}$ para armazenar as novas populações;
3) Na sequência, essa população gerada inicialmente passa pelo processo de transformações por uma quantidade pré-definida de gerações, estando sujeita a passar pelas operações de *Crossover* ou Mutação, dependendo de qual variável de probabilidade de cada transformação esteja prevalecendo para buscar convergir a população. E também, caso as transformações anteriores não encontrem um cromossomo de corte maior que as gerações anteriores, dentro de um período determinado a ficar sem convergência, será ativado a transformação de uma busca local nos seguintes cromossomos: posição inicial da população, segunda posição da população, posição intermediária da população e posição final da população. A busca local é uma transformação na qual consiste em passar por todo o cromossomo, verificando em cada gene se ao alternar o valor que está atribuído no próprio, será encontrado valores de corte maior que os atuais. A cada geração e criação de cromossomos com as transformações, estes cromossomos serão alocados no final da matriz auxiliar designado para a nova população, e seus respectivos valores de corte no final do vetor de cortes para nova população. A matriz será ordenada após todos os processos.
4) Para finalizar, o algoritmo calcula a diferença entre o maior valor de corte máximo encontrado na população inicial e o maior valor de corte máximo encontrado após passar por todas as gerações. E em seguida, retorna a diferença encontrada e o cromossomo com melhor valor de corte máximo final.

III. Experimentos Computacionais e Resultados

Em busca de obter resultados sobre o funcionamento do Algoritmo, foram realizados testes em 9 instâncias, sendo 3 instâncias diferentes para 3 quantidades de vértices diferentes, com n assumindo os valores de 60, 80 e 100.

Pensando em obter soluções melhores, foi implementado o conceito de ilha, que basicamente faz a busca por um cromossomo ótimo dentro da população que tende a ser uma boa solução. De posse do melhor cromossomo, é aplicado um conjunto de restrições válidas que atestam se determinado gene deve ser 0 ou 1, e em caso contrário o valor do gene é modificado para atender a restrição.

Para os experimentos, foram utilizadas algumas configurações de algoritmo genético que combinam quatro tipos de parâmetros que serão mostrados a seguir:

1)
   - **SI**: Sem ilha;
   - **CI**: Com ilha.
2)
   - **M1**: Mutação em 1 gene dos cromossomos escolhidos;
   - **M2**: Mutação em 2 genes dos cromossomos escolhidos.
3)
   - **P**: *Crossover* que faz a troca percentual, variando o tamanho escolhido, dos cromossomos;
   - **T2**: *Crossover* que faz a troca de 2 genes entre os cromossomos;
   - **I**: *Crossover* que prevalece a igualdade, ou seja, dos dois cromossomos escolhidos na vez, para o novo cromossomo será mantido o gene quando na mesma posição nos pais tiverem ambos valores iguais, e quando não, será sorteado entre os dois valores para o gene.
4)
   - **5**: Busca local a partir de um período de 5 gerações sem convergências;
   - **10**: Busca local a partir de um período de 10 gerações sem convergências;
   - **20**: Busca local a partir de um período de 20 gerações sem convergências.

Ao todo, essas configurações possibilitaram ter 36 combinações diferentes para execuções dos testes, e ao aplicar todas essas combinações nas 9 instâncias, foram obtidos os resultados para as instâncias com n = 60, n = 80 e n = 100, apresentados nas tabelas I, II e III, respectivamente.

Portanto, com o total de 324 testes realizados, notou-se que os resultados obtidos se aproximam ou encontram os valores ótimos disponíveis no artigo do BiqMac [8]. Sendo a combinação "CIM1I5" a prevalecente para os melhores resultados obtidos, com 55,56% dos testes para essa combinação de parâmetros retornando o cromossomo com valor ótimo ou próximo ao valor ótimo, e a combinação "CIM2T220" como a prevalecente para retornar os resultados mais distantes das melhores soluções, com 44,4% dos testes para essa combinação de parâmetros retornando cromossomos com valores mais distantes do valor ótimo.

Tabela I
Testes em instâncias com $n = 60$ vértices

Instância	g05_60.0	g05_60.1	g05_60.2
Corte máximo	**536**	**532**	**529**
SIM1P5	535	531	528
SIM1P10	535	526	518
SIM1P20	536	522	528
SIM1I5	536	532	517
SIM1I10	536	532	528
SIM1I20	534	519	524
SIM1T25	535	526	529
SIM1T210	536	528	524
SIM1T220	534	532	528
SIM2P5	531	526	522
SIM2P10	529	526	527
SIM2P20	535	521	523
SIM2I5	523	517	528
SIM2I10	526	531	528
SIM2I20	536	529	527
SIM2T25	526	526	527
SIM2T210	534	520	521
SIM2T220	528	526	526
CIM1P5	527	531	527
CIM1P10	535	531	515
CIM1P20	536	532	521
CIM1I5	536	532	526
CIM1I10	536	528	526
CIM1I20	534	532	528
CIM1T25	536	531	523
CIM1T210	536	531	529
CIM1T220	535	532	522
CIM2P5	535	518	523
CIM2P10	532	522	523
CIM2P20	534	530	521
CIM2I5	534	527	523
CIM2I10	535	523	528
CIM2I20	533	529	525
CIM2T25	518	521	524
CIM2T210	536	517	529
CIM2T220	535	532	529

Além disso, pode-se notar a influência individual dessas configurações para os resultados dos testes, como citadas a seguir:

- Para os testes com instâncias de n = 60 (Tabela I), os resultados retornaram com melhores valores para soluções quando foram influenciados pelos seguintes parâmetros: **CI**; **M1**; **P, T2 ou I**; e **20**.
- Para os testes com instâncias de n = 80 (Tabela II), os resultados retornaram com melhores valores para soluções quando foram influenciados pelos seguintes parâmetros: **CI**; **M1**; **P**; e **10**.
- Para os testes com instâncias de n = 100 (Tabela III), os resultados retornaram com melhores valores para soluções quando foram influenciados pelos seguintes parâmetros: **CI**; **M1**; **T2**; e **5, 10 ou 20**.

IV. Conclusões

Este trabalho trouxe a abordagem de configuração de parâmetros de um algoritmo genético a serem aplicadas no problema de corte máximo de um grafo, com a finalidade de obter bons resultados com o máximo possível de

Tabela II
Testes em instâncias com n = 80 vértices

Instâncias	g05_80.0	g05_80.1	g05_80.2
Corte máximo	**929**	**941**	**934**
SIM1P5	915	941	920
SIM1P10	927	941	929
SIM1P20	927	941	919
SIM1I5	919	941	926
SIM1I10	927	941	923
SIM1I20	924	940	926
SIM1T25	911	941	920
SIM1T210	917	909	915
SIM1T220	919	937	921
SIM2P5	912	918	922
SIM2P10	913	941	927
SIM2P20	918	908	917
SIM2I5	914	921	910
SIM2I10	914	915	923
SIM2I20	925	927	926
SIM2T25	913	940	921
SIM2T210	910	930	922
SIM2T220	907	905	916
CIM1P5	914	941	920
CIM1P10	913	941	927
CIM1P20	929	941	919
CIM1I5	929	941	923
CIM1I10	913	940	932
CIM1I20	924	937	921
CIM1T25	923	941	924
CIM1T210	913	933	922
CIM1T220	906	924	917
CIM2P5	915	913	926
CIM2P10	923	913	918
CIM2P20	911	937	934
CIM2I5	905	938	929
CIM2I10	915	931	917
CIM2I20	907	941	914
CIM2T25	903	921	920
CIM2T210	919	913	922
CIM2T220	904	908	914

Tabela III
Testes em instâncias com n = 100 vértices

Instâncias	g05_100.0	g05_100.1	g05_100.2
Corte máximo	**1430**	**1425**	**1432**
SIM1P5	1414	1420	1404
SIM1P10	1410	1411	1409
SIM1P20	1416	1409	1407
SIM1I5	1408	1409	1404
SIM1I10	1420	1413	1421
SIM1I20	1408	1405	1405
SIM1T25	1408	1417	1427
SIM1T210	1418	1421	1421
SIM1T220	1421	1417	1431
SIM2P5	1397	1408	1396
SIM2P10	1383	1393	1414
SIM2P20	1402	1383	1393
SIM2I5	1403	1414	1409
SIM2I10	1395	1417	1393
SIM2I20	1422	1413	1416
SIM2T25	1411	1417	1422
SIM2T210	1404	1410	1404
SIM2T220	1400	1400	1405
CIM1P5	1396	1409	1411
CIM1P10	1422	1401	1402
CIM1P20	1406	1408	1416
CIM1I5	1420	1424	1404
CIM1I10	1403	1419	1407
CIM1I20	1414	1418	1427
CIM1T25	1417	1412	1405
CIM1T210	1415	1416	1420
CIM1T220	1426	1413	1423
CIM2P5	1397	1405	1414
CIM2P10	1395	1410	1404
CIM2P20	1387	1381	1409
CIM2I5	1421	1400	1425
CIM2I10	1408	1413	1412
CIM2I20	1412	1408	1408
CIM2T25	1418	1403	1414
CIM2T210	1413	1416	1411
CIM2T220	1411	1403	1378

otimização e menos custos para os resultados do problema. Ao concluir os testes, constatou-se que muitas dessas configurações de parâmetros utilizados geram valores próximos aos valores do corte ótimos que foram obtidos e tabulados por [8]. Além disso, a configuração com parâmetros "CIM1I5" se mostrou a mais eficiente dentre todas as 36 testadas. Para futuros trabalhos e continuação dessa proposta, propõe-se a implementação de mais parâmetros e operadores genéticos que venham a divergir mais ainda a população e assim aumente o espaço de busca da solução de forma menos custosa. Além disso, pretende-se estudar formas alternativas na geração da população inicial.

Agradecimentos

O trabalho foi realizado com o apoio do Programa Institucional de Bolsas de Iniciação Científica (PIBIC). Portanto, os autores agradecem à Universidade Federal do Ceará - Campus Russas.

Referências

[1] KHAN ACADEMY. Descrevendo grafos. Disponível em: https://pt.khanacademy.org/computing/computer-science/algorithms/graph-representation/a/describing-graphs. Acesso em: 06 set. 2022.

[2] JURKIEWICZ, Samuel. Grafos–uma introdução. São Paulo: OBMEP, 2009.

[3] SOARES, P. L. B., FILHO, J. A. F. C. HEURÍSTICA PROBABILÍSTICA GUIADA PELOS POTENCIAIS DOS VÉRTICES PARA O PROBLEMA DO CORTE MÁXIMO., In: LIII Simpósio Brasileiro de Pesquisa Operacional. Anais . . . , João Pessoa, SOBRAPO, 2021, p. 1-2.

[4] Karp, R. M. Reducibility among combinatorial problems. Springer. 1972

[5] LUCAS, D. C. Algoritmos genéticos: uma introdução. Universidade Federal do Rio Grande do Sul, Brasil, 2002.

[6] IKEDA, Patrícia Akemi. Introdução aos Algoritmos Genéticos. 2009.

[7] OPTSICOM PROJECT. MaxCut Problem. Disponivel em: https://grafo.etsii.urjc.es/optsicom/maxcut.html. Acesso em: 05 abr. 2022.

[8] Wiegele, Angelika. "Biq Mac Library—A collection of Max-Cut and quadratic 0-1 programming instances of medium size."Preprint 51 (2007).