Aplicação do Golden Ball para o Problema do Corte Máximo

Jailon W. B. Oliveira da Silva[1], **Carlos V. Dantas Araújo**[1], **Pablo L. Braga Soares**[1]

[1]Universidade Federal do Ceará – Campus Russas
Avenida Felipe Santiago – Nº 411 62.900-624, Russas/CE, Brasil

{williambrunocc, carlosvinicio}@alu.ufc.br, pablo.soares@ufc.br

Abstract. *This work applies the Golden Ball metaheuristic to the Maximum Cut Problem in graphs, modeling solutions as players organized into teams that evolve to identify vertex partitions that maximize the sum of the weights of the cut edges.A total of 120 instances from the Biq Mac Library were evaluated, and the experiments showed that Golden Ball produces satisfactory solutions, with an average deviation of 37% on the Be set in 10 seconds and only 1.4% in 1200 seconds. As an evolutionary method, its results improve with more execution time. The approach was compared to the Biq Mac Library solver, a benchmark in the field, which ran for up to 10800 seconds.*

Resumo. *Este trabalho aplica a meta-heurística Golden Ball ao Problema do Corte Máximo em grafos, modelando soluções como jogadores organizados em times que evoluem, visando identificar partições dos vértices que maximizem a soma dos pesos das arestas cortadas. Foram avaliadas 120 instâncias da Biq Mac Library, e os experimentos mostraram que o Golden Ball gera soluções satisfatórias, com desvio médio de 37% no conjunto Be em 10 segundos e apenas 1.4% em 1200 segundos. Como método evolutivo, seus resultados melhoram com mais tempo de execução. A abordagem foi comparada ao solver Biq Mac Library, referência na área, que operou por até 10800 segundos.*

1. Introdução

Os problemas de otimização combinatória abrangem uma ampla gama de desafios voltados à buscam por soluções que maximizam a eficiência e o aproveitamento dos recursos, reduzindo o tempo de execução das operações e, ao mesmo tempo, equilibrando ganhos e perdas [Miyazawa and de Souza 2015]. Entre esses problemas, o Problema do Corte Máximo se destaca por ser NP-difícil [Karp 1972], com uma ampla gama de aplicações em domínios como redes sociais [Agrawal et al. 2003], agrupamento de dados [Otterbach et al. 2017], segmentação de imagens [de Sousa et al. 2013] e projeto de Chips VLSI (*Very Large Scale Integrated*) [Liers et al. 2011].

Dado que problemas NP-difíceis não possuem, até o momento, algoritmos exatos eficientes que os resolvam em tempo polinomial, pois o crescimento exponencial do espaço de soluções inviabiliza métodos exatos em instâncias de grande porte. Nesse cenário, algoritmos aproximados surgem como alternativa viável, ainda que sem garantia de encontrar a solução ótima [Blum and Roli 2003]. Neste trabalho, emprega-se a meta-heurística *Golden Ball*, inspirada em conceitos do futebol, na qual soluções são modeladas como jogadores e times, evoluindo por meio de treinamento, disputas e transferência [Osaba et al. 2014], para abordar o Problema do Corte Máximo.

2. Definição formal do problema e abordagem proposta

O Problema do Corte Máximo busca por um subconjunto $S \subseteq V$ de um grafo $G = (V, E)$, de modo que o corte definido pela separação dos vértices V em dois subconjuntos disjuntos S e $\overline{S}$ maximize a soma dos pesos das arestas que têm exatamente um extremo em S e o outro em $\overline{S}$. Assim, o corte é o conjunto de arestas $\{(u, v) \in E \mid u \in S, v \in \overline{S}\}$, sendo o objetivo principal, identificar o subconjunto S tal que a soma dos pesos dessas arestas seja a maior possível, ou seja, maximizar $\sum_{(u,v)\in E} w(u, v)$, onde $w(u, v)$ é o peso da aresta (u, v) que conecta $u \in S$ e $v \in \overline{S}$ [Boros and Hammer 1991].

2.1. Meta-Heurística *Golden Ball*: conceitos Originais e adaptação

A meta-heurística *Golden Ball*, proposta originalmente por [Osaba et al. 2014], inspira-se em conceitos do futebol para guiar a evolução de soluções em abordagens genéticas. No modelo original, as soluções são interpretadas como jogadores organizados em times (subpopulações), formando conjunto de soluções, e o algoritmo baseia-se em três mecanismos principais:

- **Treinamento Local:** cada jogador aprimora sua solução por meio de novas soluções geradas a partir do próprio indivíduo, adotando a mudança se esta apresentar desempenho superior;
- **Disputas (*Matchdays*):** dois times competem em partidas, nas quais os jogadores são confrontados; ou seja, são comparadas as qualidades das soluções dos times, e pontos são acumulados conforme os resultados;
- **Transferência:** ao final de um ciclo (temporada), os jogadores com pior desempenho, nas equipes que não lideram, são substituídos por novos jogadores gerados por operadores de cruzamento, os quais combinam características de soluções de alto desempenho (por exemplo, a melhor solução da equipe campeã).

Na adaptação para o Problema do Corte Máximo, cada solução é representada por um vetor binário $\mathbf{x} \in \{0, 1\}^n$ e a função objetivo $f(\mathbf{x})$ corresponde à soma dos pesos das arestas que cortam a partição definida por $\mathbf{x}$. Assim, o treinamento local visa melhorar $f(\mathbf{x})$ explorando a vizinhança da solução, enquanto os processos de disputa e transferência organizam e refinam as soluções em equipes, direcionando a busca para regiões promissoras do espaço de soluções.

2.2. Aplicação da Meta-Heurística *Golden Ball* ao Problema do Corte Máximo

Inicialmente, uma população de soluções é gerada de forma aleatória, onde cada solução é um vetor binário $\mathbf{x} \in \{0, 1\}^n$ que representa uma possível partição dos vértices do grafo $G = (V, E)$. Essas soluções são distribuídas entre um número predefinido de equipes, simulando a formação de times. Assim, cada jogador (solução), passa por um processo iterativo de treinamento local. Neste procedimento, uma mutação ocorre na solução, isto é, a inversão de um bit aleatório é aplicada para gerar uma solução candidata $\mathbf{x}^{(i)}$. Desta forma, se $f(\mathbf{x}^{(i)}) > f(\mathbf{x})$, onde

$$f(\mathbf{x}) = \sum_{\substack{(u,v)\in E \\ u\in S,\ v\in\overline{S}}} w(u, v),$$

a solução atual é substituída pela candidata. Esse processo se repete até que um critério de término, como um número fixo de iterações sem melhoria seja alcançado.

Em seguida, as equipes disputam entre si em formato round-robin. Desta forma em cada partida, os jogadores de dois times opostos são separados em pares e são comparados, o jogador que possuir um valor de corte maior que o outro marca um gol para sua equipe. Assim, o time que possuir mais gol ganha 3 pontos e no caso de empate, ambos os times recebem 1 ponto. Esse mecanismo competitivo não só avalia a eficácia das soluções de forma relativa, mas também incentiva a diversificação e a evolução das subpopulações. No fim, com base nesses pontos, as equipes serão ranqueadas.

O fim da disputa entre as equipes marca o término de uma temporada e dá início ao período de transferência. Durante essa fase, os jogadores com o pior desempenho das equipes que não lideram o ranking são substituídos por novos jogadores gerados por operadores de cruzamento. Este cruzamento combina partes das soluções dos jogadores de baixo desempenho com a melhor solução da equipe vencedora, promovendo a disseminação de características favoráveis.

Para exemplificar esse cruzamento, considere duas soluções $\mathbf{x}$ e $\mathbf{y}$ em $\{0,1\}^n$. O operador de cruzamento implementado divide as soluções na posição $\left\lfloor \frac{n}{2} \right\rfloor$, conforme descrito a seguir:

$$\mathbf{z} = \left(x_1, \ldots, x_{\left\lfloor \frac{n}{2} \right\rfloor}, y_{\left\lfloor \frac{n}{2} \right\rfloor + 1}, \ldots, y_n \right).$$

Essa operação visa combinar características promissoras de ambas as soluções, permitindo a propagação de componentes de alta qualidade entre as equipes.

De modo geral, o algoritmo itera por diversas temporadas, onde cada temporada contêm ao todo, dois dias de jogos, ambos seguidos pela fase de transferência. O processo global é interrompido quando são atendidos critérios preestabelecidos, como um limite máximo de tempo de execução ou a ausência de melhorias após um determinado número de temporadas. Ao final, a melhor solução obtida é considerada uma aproximação eficaz para o Problema do Corte Máximo.

3. Resultados Computacionais

Os experimentos foram conduzidos em uma máquina equipada com Intel(R) Core(TM) i5-12450H, 16 GB de RAM e Windows 11 Home. O algoritmo foi implementado em Python, utilizando as bibliotecas *random*, *pandas* e *time*, e desenvolvido em Jupyter Notebooks. Além disso, foram adotados os seguintes critérios de parada: tempo limite e execução de 200 temporadas sem melhoria. Para o tempo limite, foi realizado teste com 10 segundos e 1200 segundos.

Os testes foram realizados com instâncias da *Biq Mac Library* [Wiegele 2007], amplamente utilizadas na literatura para avaliação de algoritmos aplicados ao Problema do Corte Máximo. As instâncias selecionadas pertencem aos conjuntos *Be*, *Beasly* e *gka*, totalizando 120 instâncias, com grafos variando de 20 a 500 vértices. Adicionalmente, a fim de comparação dos valores de corte obtidos, essas instâncias foram submetidas ao *solver Biq Mac*, com um limite de tempo de até 3 horas. Esse algoritmo exato se baseia no método *Branch & Bound* e utiliza Programação Semi-Definida (SDP) [Rendl et al. 2010].

A meta-heurística *Golden Ball* apresentou resultados promissores no contexto do Problema do Corte Máximo, especialmente quando comparada ao *solver Biq Mac*, destacando-se principalmente pela redução significativa no tempo de execução em relação ao solver exato. Na Tabela 1, são apresentados de forma resumida os resultados de desempenho, com base na média de tempo para ambos os métodos.

Tabela 1. Resumo de desempenho do Golden Ball

Conjunto de Instância	Tempo do solver (s)	Golden Ball (10s)		Golden Ball (1200s)	
		Desvio Médio (%)	Tempo (s)	Desvio Médio (%)	Tempo (s)
Be	9181.61	37.0%	10.0	1.4%	1044.62
Beasly	5422.26	28.7%	10.0	4.2%	620.56
Gka	3005.21	15.6%	10.0	3.3%	412.33

A primeira observação consiste na eficiência do *Golden Ball*, no que diz respeito ao tempo de execução. Note que ao ser executado com um limite máximo de 10 segundos, apesar da margem de desvio médio ter sido relativamente alta, especialmente no conjunto *Be* (37.0%), o que é esperado dado o tempo reduzido de execução, o algoritmo ainda conseguiu encontrar soluções dentro de desvio médio tolerável. Isso evidencia a capacidade dessa meta-heurística de fornecer soluções rápidas, embora não ótimas, em problemas de grande escala.

Note que à medida que o tempo de execução é aumentado para 1200 segundos, a margem de erro diminui consideravelmente. No conjunto *Be* por exemplo, a margem de erro cai para 1.4%, implicando que, com mais tempo, o *Golden Ball* se aproxima das soluções ótimas ou de alta qualidade, se comparado com o *solver Biq Mac*. No geral, esse comportamento é consistente com a natureza das meta-heurísticas, que exploram o espaço de soluções de forma progressiva e podem melhorar seu desempenho ao decorrer das iterações.

Ao comparar o *Golden Ball* com o *solver Biq Mac*, surge um ponto positivo para o primeiro, que destacou-se por sua capacidade de fornecer resultados rápidos e satisfatórios, mesmo com limitações de tempo. O *solver Biq Mac*, embora seja preciso, levou consideravelmente mais tempo para processar as mesmas instâncias, com alguns casos ultrapassando 9000 segundos de execução, o que demonstra a vantagem de soluções aproximadas quando o tempo de execução é um fator crítico.

4. Conclusão

Este trabalho demonstra a viabilidade e eficácia da meta-heurística *Golden Ball* para o Problema do Corte Máximo em grafos. A abordagem mostrou-se eficaz na geração de soluções, e os testes realizados com tempos de execução variando de 10 e 1200 segundos indicam que, à medida que o tempo de execução aumenta, o método é capaz de fornecer soluções competitivas em relação ao *solver Biq Mac*, especialmente para grandes instâncias. A implementação considerou apenas uma abordagem pura do Golden Ball, porém, espera-se uma melhoria considerável do algoritmo com a incorporação de técnicas híbridas, capazes de explorar o espaço de soluções de forma mais eficaz.

Ademais, trabalhos futuros poderão explorar a integração dessa meta-heurística com propriedades intrínsecas do grafo no qual se busca determinar o valor de corte ótimo. Características estruturais, como coeficiente de aglomeração, centralidade do grau e intermediação dos vértices, têm o potencial de guiar o método na varredura do espaço de soluções de maneira mais assertiva e eficiente.

Referências

Agrawal, R., Rajagopalan, S., Srikant, R., and Xu, Y. (2003). Mining newsgroups using networks arising from social behavior. In *Proceedings of the 12th international conference on World Wide Web*, pages 529–535.

Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM computing surveys (CSUR)*, 35(3):268–308.

Boros, E. and Hammer, P. L. (1991). The max-cut problem and quadratic 0–1 optimization; polyhedral aspects, relaxations and bounds. *Annals of Operations Research*, 33(3):151–180.

de Sousa, S., Haxhimusa, Y., and Kropatsch, W. G. (2013). Estimation of distribution algorithm for the max-cut problem. In *Graph-Based Representations in Pattern Recognition: 9th IAPR-TC-15 International Workshop, GbRPR 2013, Vienna, Austria, May 15-17, 2013. Proceedings 9*, pages 244–253. Springer.

Karp, R. M. (1972). Reducibility among combinatorial problems. complexity of computer computations (the ibm research symposia series).

Liers, F., Nieberg, T., and Pardella, G. (2011). Via minimization in vlsi chip design-application of a planar max-cut algorithm.

Miyazawa, F. K. and de Souza, C. C. (2015). Introdução à otimização combinatória. *Sociedade Brasileira de Computação*.

Osaba, E., Diaz, F., and Onieva, E. (2014). Golden ball: a novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. *Applied intelligence*, 41:145–166.

Otterbach, J. S., Manenti, R., Alidoust, N., Bestwick, A., Block, M., Bloom, B., Caldwell, S., Didier, N., Fried, E. S., Hong, S., et al. (2017). Unsupervised machine learning on a hybrid quantum computer. *arXiv preprint arXiv:1712.05771*.

Rendl, F., Rinaldi, G., and Wiegele, A. (2010). Solving Max-Cut to optimality by intersecting semidefinite and polyhedral relaxations. *Math. Programming*, 121(2):307.

Wiegele, A. (2007). Biq mac library—a collection of max-cut and quadratic 0-1 programming instances of medium size. *Preprint*, 51.