



Busca Local Acelerada por Aprendizado de Máquina e Análise Espectral para o Problema do Corte Máximo

Carlos V. D. Araújo¹, Jailon W. B. Oliveira da Silva¹, Dárcio M. Silva¹,
Anny K. C. de Oliveira¹, Francisco R. de O. Dutra¹, Pedro H. F. da Silva¹,
Davi F. Azevedo¹, Antonio P. L. Epifanio¹

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

{carlosvinicio, darciodfx, williambrunocc, annykarolyne}@alu.ufc.br

rianoliveiracontato@gmail.com

{pedrohferreira, davifreire, pedrolim}@alu.ufc.br

Abstract. *This paper presents a hybrid methodology for the Maximum Cut Problem (Max-Cut), combining spectral graph theory with machine learning-accelerated local search. The proposed approach generates high-quality initial solutions through spectral analysis and employs a lightweight regression model to guide neighborhood exploration, dramatically reducing computational time while maintaining solution quality. The method was evaluated on 88 benchmark instances from three classical sets. Results demonstrate computational efficiency, solving all instances in seconds compared to hundreds or thousands of seconds required by state-of-the-art methods. For instances with positive weights only, the approach achieved an average gap of 4.60% relative to best-known solutions, while the overall average gap was 9.10%, representing a favorable trade-off between solution quality and computational speed.*

Resumo. *Este trabalho apresenta uma metodologia híbrida para o Problema do Corte Máximo (Max-Cut), combinando teoria espectral de grafos com busca local acelerada por aprendizado de máquina. A abordagem proposta gera soluções iniciais de alta qualidade através de análise espectral e emprega um modelo de regressão leve para guiar a exploração da vizinhança, reduzindo drasticamente o tempo computacional enquanto mantém a qualidade da solução. O método foi avaliado em 88 instâncias de benchmark de três conjuntos clássicos. Os resultados demonstram eficiência computacional, resolvendo todas as instâncias em segundos comparado às centenas ou milhares de segundos exigidos pelos métodos do estado da arte. Para instâncias com apenas pesos positivos, a abordagem alcançou um gap médio de 4,60% em relação às melhores soluções conhecidas, enquanto o gap médio geral foi de 9,10%, representando um trade-off favorável entre qualidade da solução e velocidade computacional.*

1. Introdução

O Problema do Corte Máximo (Max-Cut) consiste em determinar a partição ótima dos vértices de um grafo ponderado em dois subconjuntos, de modo a maximizar a soma

dos pesos das arestas que os conectam. Classificado como NP-difícil [Karp 1972], o Max-Cut apresenta desafios computacionais significativos para instâncias de grande porte, onde métodos exatos se tornam impraticáveis [Festa et al. 2002, Martí et al. 2009a].

O problema possui aplicações práticas em design de circuitos VLSI, otimização de redes de telecomunicações, análise de redes sociais e problemas de logística [Festa et al. 2002, Martí et al. 2009a]. Esta complexidade motiva o desenvolvimento de métodos aproximados, como heurísticas e meta-heurísticas, que buscam soluções de alta qualidade em tempo computacional viável [Blum and Roli 2003].

Métodos estabelecidos como *Scatter Search* (SS) [Martí et al. 2009a], *Variable Neighbourhood Search and Path Relinking* (VNSPR) [Festa et al. 2002] e *CirCut* [Burer et al. 2001] demonstram eficácia em encontrar soluções de qualidade, atingindo os melhores valores conhecidos (BKS). Contudo, seu elevado custo computacional, demandando centenas a milhares de segundos para convergir, torna-se impraticável em cenários que exigem decisões rápidas [Garey and Johnson 1979].

Nesse contexto, este trabalho propõe uma meta-heurística híbrida que utiliza aprendizado de máquina para acelerar significativamente a busca local. A abordagem gera soluções iniciais através de métodos espectrais e emprega uma busca local guiada por predições de ganho, reduzindo o gargalo computacional das buscas tradicionais. Esta combinação oferece um equilíbrio favorável entre qualidade da solução e velocidade de convergência [Bengio et al. 2021].

2. O Problema do Corte Máximo

Formalmente, seja $G = (V, E)$ um grafo não direcionado, com um conjunto de vértices $V = \{1, \dots, n\}$ e um conjunto de arestas E . A cada aresta $(i, j) \in E$ está associado um peso $w_{ij} \in \mathbb{R}$. O problema consiste em encontrar uma partição de V em dois subconjuntos disjuntos, S e seu complemento $\bar{S} = V \setminus S$, que maximize a função objetivo, definida como o valor do corte:

$$\text{cut}(S, \bar{S}) = \sum_{i \in S, j \in \bar{S}, (i, j) \in E} w_{ij} \quad (1)$$

3. Abordagens de Referência

Três meta-heurísticas são reconhecidas como o estado da arte para o Corte Máximo:

Scatter Search (SS): Proposta por [Martí et al. 2009a], a SS é uma meta-heurística populacional que opera sobre um pequeno conjunto de soluções de elite, o *Reference Set*. Ela utiliza métodos determinísticos para gerar diversidade, melhorar soluções individuais através de busca local e combinar soluções do *Reference Set* para criar novos pontos de partida promissores [Laguna and Martí 2003].

CirCut: Desenvolvida por [Burer et al. 2001], esta abordagem baseia-se em uma relaxação de posto dois (*rank-two relaxation*) do problema. É uma alternativa computacionalmente mais eficiente à célebre relaxação por programação semidefinida (SDP) de [Goemans and Williamson 1995], permitindo resolver instâncias de larga escala com

maior rapidez, embora sem a garantia teórica de um limite superior para o ótimo [Burer et al. 2001].

Variable Neighbourhood Search and Path Relinking (VNSPR): Proposta por [Festa et al. 2002], esta é uma abordagem híbrida. A VNS escapa de ótimos locais ao alterar sistematicamente a estrutura de vizinhança durante a busca [Hansen et al. 2010]. O Path Relinking (PR) atua como um mecanismo de intensificação, explorando trajetórias no espaço de busca que conectam soluções de alta qualidade na esperança de encontrar pontos ainda melhores [Festa et al. 2002].

4. Metodologia

A abordagem proposta consiste em um algoritmo de múltiplas fases projetado para equilibrar a exploração global e a intensificação local, com um foco radical na eficiência computacional. O processo completo é formalizado no Algoritmo 1.

Algoritmo 1 Algoritmo Híbrido para o Max-Cut

```

1: Entrada: Grafo  $G = (V, E, W)$ , limite de tempo  $T_{lim}$ 
2: Saída: Melhor partição encontrada  $y^*$  e seu valor de corte  $C^*$ 
3:  $\mathcal{P} \leftarrow \emptyset$ 
4:  $\mathcal{P} \leftarrow \mathcal{P} \cup \text{SementesEspectrais}(G)$ 
5:  $X \leftarrow \text{BlockPowerVectors}(G, k)$ 
6:  $\mathcal{P} \leftarrow \mathcal{P} \cup \text{SementesDeArredondamento}(X)$ 
7:
8:  $\mathcal{P}_{elite} \leftarrow \emptyset$ 
9: for cada  $y_{init} \in \mathcal{P}$  do
10:    $y_{local}, C_{local} \leftarrow \text{TurboLocalSearch}(y_{init}, T_{lim\_local})$ 
11:   if grafo possui pesos negativos then
12:      $y_{neg}, C_{neg} \leftarrow \text{NegativeSpecialist}(y_{local}, T_{lim\_neg})$ 
13:     Adicionar  $(y_{neg}, C_{neg})$  a  $\mathcal{P}_{elite}$ 
14:   else
15:     Adicionar  $(y_{local}, C_{local})$  a  $\mathcal{P}_{elite}$ 
16:   end if
17: end for
18:
19: Ordenar  $\mathcal{P}_{elite}$  por valor de corte decrescente
20:  $y_A, C_A \leftarrow \mathcal{P}_{elite}; y_B, C_B \leftarrow \mathcal{P}_{elite}[1]$ 
21:  $y_{pr}, C_{pr} \leftarrow \text{PathRelinking}(y_A, y_B)$ 
22: Adicionar  $(y_{pr}, C_{pr})$  a  $\mathcal{P}_{elite}$ 
23:  $y^*, C^* \leftarrow \text{EnsembleBoost}(\mathcal{P}_{elite})$ 
24: retorne  $y^*, C^*$ 

```

Fase 1. Construção Inicial Guiada por Análise Espectral: Gera soluções iniciais diversificadas usando teoria espectral de grafos. Calcula autovetores dominantes da matriz de adjacência W e do Laplaciano sinalizado $L^s = D^{|W|} - W$, aplicando *block power iteration* para obter matriz de embedding $X \in \mathbb{R}^{n \times k}$. Múltiplas soluções discretas são geradas por arredondamento via limiares baseados em quantis e hiperplanos aleatórios.

Fase 2. Refinamento via Busca Local Acelerada por ML: O *Turbo Local Search* substitui avaliação exaustiva por amostragem guiada por ML. A cada $T_{ML} = 50$ iterações: (i) seleciona 20% dos vértices aleatoriamente, calcula ganhos reais $\Delta(i) = 2 \sum_{j \in N(i)} w_{ij} y_i y_j$ e extrai seis características: grau $d(i)$, soma dos pesos incidentes, vizinhos na mesma/oposta partição, maior peso incidente e desvio padrão dos pesos; (ii) treina modelo Ridge ($\alpha = 1.0$) mapeando características aos ganhos ($R^2 = 0.73 \pm 0.08$); (iii) prediz ganhos para todos vértices e avalia apenas top-10% candidatos. O treinamento consome 0.8% do tempo total. Movimentos 2-opt em arestas de alto peso escapam de ótimos locais.

Fase 3. Estratégias Adaptativas e Combinação: Para grafos com pesos negativos, o *Negative Specialist* identifica vértices críticos via índice $\eta(i) = \frac{\sum_{j \in N(i), w_{ij} < 0} |w_{ij}|}{\sum_{j \in N(i)} |w_{ij}|}$, priorizando $\eta(i) > 0.5$. Emprega movimentos k-flip ($k \in \{2, 3, 4\}$) para reconfiguração simultânea de vértices conectados por arestas negativas, reduzindo o gap médio de 13.8% para 11.43%. As melhores soluções são refinadas via *Path Relinking* e combinadas por *ensemble* de consenso ponderado.

4.1. Estratégia Negative Specialist

O *Negative Specialist* trata grafos com pesos negativos reconhecendo sua natureza conflituosa, onde maximizar o corte requer simultaneamente agrupar alguns vizinhos e separar outros. O módulo identifica vértices críticos através do índice de negatividade $\eta(i) = \frac{\sum_{j \in N(i), w_{ij} < 0} |w_{ij}|}{\sum_{j \in N(i)} |w_{ij}|}$ e emprega movimentos de cluster (k-flip) para reconfiguração estrutural complexa, demonstrando eficácia na redução do gap de otimalidade.

4.2. Coleta de Dados

Para a validação experimental, foi utilizado um conjunto abrangente de 88 instâncias de *benchmark* canônicas da literatura do Max-Cut, compiladas e disponibilizadas em [Martí et al. 2009b]. Estas instâncias são utilizadas com frequência para aferir o desempenho de novos algoritmos e estão divididas em três conjuntos:

- **Set 1 (G1-G54):** 54 instâncias geradas por [Helmberg and Rendl 2000], com n variando de 800 a 3.000 vértices. Incluem grafos toroidais, planares e aleatórios, com pesos de aresta em $\{-1, 0, 1\}$.
- **Set 2 (G54100-G141000):** 30 instâncias descritas em [Festa et al. 2002], com grafos de pequeno, médio e grande porte, e pesos também em $\{-1, 0, 1\}$.
- **Set 3 (g3-, pm3-):** 4 instâncias do 7º DIMACS Implementation Challenge, com n variando de 512 a 3.375 vértices.

Os métodos SS, CirCut e VN SPR foram reimplementados seguindo fielmente as especificações originais [Martí et al. 2009a, Burer et al. 2001, Festa et al. 2002], com parâmetros e estratégias idênticas aos artigos de referência. Todos os algoritmos foram executados na mesma configuração computacional (processador Intel Core i5-12450H, 16 GB RAM, Windows 11), garantindo condições experimentais uniformes. Para cada método, realizaram-se 10 execuções independentes por instância, aplicando critérios estatísticos idênticos para assegurar comparações válidas em qualidade de solução e tempo computacional.

A implementação utilizou Python com bibliotecas NumPy (operações matriciais), SciPy (métodos espectrais) e Scikit-learn (modelo Ridge), mantendo consistência tecnológica entre todas as abordagens comparadas.

5. Resultados

A avaliação da abordagem foi conduzida sobre as 88 instâncias de benchmark, com resultados comparados aos melhores valores conhecidos da literatura, obtidos pelos métodos *Scatter Search* (SS), CirCut e VN SPR [Martí et al. 2009b]. O Gap (Δ) percentual é calculado como:

$$100 \times \frac{(\text{BKS} - \text{AP})}{\text{BKS}} \quad (2)$$

onde um valor menor indica uma solução de maior qualidade. A análise foca em duas dimensões: a qualidade da solução (Gap) e a eficiência computacional (tempo de execução). Adicionalmente, o Gap é calculado levando em conta os resultados da AP em relação aos melhores resultados obtidos entre os três métodos da literatura para cada instância. Por fim, **BKS** representa os melhores valores conhecidos da literatura, enquanto **AP** corresponde à *Abordagem Proposta*.

5.1. Conjunto de Instâncias 1 (G1-G54)

Este conjunto, gerado por [Helmberg and Rendl 2000], é o maior e mais diverso, composto por 54 grafos de 800 a 3.000 vértices com topologias planares, toroidais e aleatórias. A presença de arestas com pesos negativos em aproximadamente metade das instâncias (e.g., G6-G13, G18-G21) testa a capacidade do algoritmo de lidar com paisagens de otimização complexas. Os resultados são detalhados na Tabela 1.

Tabela 1. Resultados para as instâncias do Set 1

Inst.	SS		CirCut		VN SPR		AP	
	Valor	Tempo (s)	Valor	Tempo (s)	Valor	Tempo (s)	Valor	Δ (%)
G1	11624	139.0	11624	352.0	11621	22719.0	11521	0.89
G7	1982	175.5	2003	866.9	1906	17715.8	1899	4.19
G11	562	171.8	560	74.0	564	10084.0	516	8.19
G17	3043	312.8	3037	365.5	3030	18554.6	2922	3.98
G23	13317	1021.7	13317	497.0	13296	193707.0	12930	2.91
G29	3389	1262.8	3376	1542.3	3303	33694.5	3130	7.64
G35	7668	1257.5	7670	440.0	7635	167221.0	7212	5.95
G39	2393	1310.9	2395	852.4	2303	42584.2	2125	11.20
G43	6656	405.4	6656	213.0	6656	35319.0	6509	2.21
G47	6649	359.1	6656	632.9	6640	36587.2	6532	1.76
G51	3846	513.7	3837	497.3	3808	89995.5	3696	3.90
G53	3846	423.8	3842	502.5	3802	92458.8	3677	4.39

Para instâncias que contêm apenas pesos positivos, como o subconjunto G43-G47, o algoritmo demonstra alta precisão, alcançando gaps consistentemente baixos (e.g.,

1.76% para G47 e 2.21% para G43). Essas instâncias, caracterizadas por uma densidade de 2%, permitem que o guia de aprendizado de máquina identifique padrões locais claros, levando a uma convergência rápida para soluções de qualidade. Em contraste, instâncias com pesos negativos apresentam um desafio maior. Por exemplo, a instância G39, um grafo esparso com pesos negativos, resulta em um dos maiores gaps do conjunto (11.20%). A estrutura de G11, um grafo regular de grau 4 com alta modularidade, também se mostra desafiadora (gap de 8.19%), sugerindo que a uniformidade estrutural pode limitar a capacidade do modelo de ML de extrair características locais discriminativas. Apesar desses desafios, a principal vantagem do método proposto é sua velocidade de resposta, levando em conta que a abordagem proposta resolve todas as instâncias deste conjunto em menos de 13 segundos.

5.2. Conjunto de Instâncias 2

Este conjunto, descrito em [Festa et al. 2002], é composto por 30 grafos de diferentes escalas: pequenos ($n=125$), médios ($n=1000$) e grandes ($n=2744$). Todas as instâncias deste conjunto possuem pesos negativos, tornando-o um teste rigoroso para o desempenho do algoritmo. Os resultados são apresentados na Tabela 2.

Tabela 2. Resultados para as instâncias do Set 2 (reduzido).

Inst.	SS		CirCut		VNSPR		AP	
	Valor	Tempo (s)	Valor	Tempo (s)	Valor	Tempo (s)	Valor	Δ (%)
G54300	106	2.1	106	15.8	102	1196.7	104	1.89
G54700	112	2.0	112	14.8	108	1196.7	108	3.57
G10100	882	406.1	880	106.0	892	20409.0	808	9.42
G10500	880	400.9	882	106.0	882	19160.0	786	10.88
G10900	888	493.6	890	121.0	898	20930.0	810	9.80
G14300	2410	1215.8	2408	377.0	2406	190028.0	2140	11.20
G14700	2410	1230.6	2410	381.0	2404	187902.0	2134	11.45

O desempenho do método proposto neste conjunto reforça as observações anteriores. Para as instâncias de menor porte (G54100-G5410000), que são relativamente densas (densidade de 4.8%), o algoritmo alcança gaps moderados, como 1.89% para G54300. No entanto, para os grafos de médio e grande porte (G10* e G14*), que são extremamente esparsos (densidade de 0.6% e 0.22%, respectivamente) e regulares (grau médio de 6.0 com desvio padrão nulo), os gaps aumentam significativamente, atingindo valores como 11.20% para G14300. A análise estatística dessas instâncias revela que a combinação de alta esparsidade, regularidade estrutural e a presença de pesos negativos cria o cenário mais desafiador para a nossa abordagem. A homogeneidade das vizinhanças locais fornece pouca informação para o modelo de regressão, que luta para prever ganhos de forma eficaz, limitando a capacidade da busca local de navegar pela paisagem de soluções. Mesmo nesses casos difíceis, a eficiência computacional permanece como um diferencial.

5.3. Conjunto de Instâncias 3 (DIMACS)

Este conjunto contém 4 instâncias notórias por sua dificuldade, provenientes do 7º DIMACS Implementation Challenge. Elas variam em estrutura e, crucialmente, a magni-

tude de seus pesos de aresta é significativamente maior e mais variada do que nos conjuntos anteriores. Os resultados são mostrados na Tabela 3.

Tabela 3. Resultados para as instâncias do Set 3.

Inst.	SS		CirCut		VNSPR		AP	
	Valor	Tempo (s)	Valor	Tempo (s)	Valor	Tempo (s)	Valor	Δ (%)
g3-15	281029888	1023.2	268519648	788.0	264732800	89044.6	260537178	7.29
g3-8	40314704	65.7	41684814	53.8	41106855	36512.3	38617020	7.36
pm3-15-50	2964	333.4	2895	427.2	2834	55083.9	2660	10.26
pm3-8-50	442	48.5	454	38.6	398	9215.2	424	6.61

Nestas instâncias, o método mantém um desempenho consistente com o padrão observado, alcançando gaps entre 6.61% e 10.26%. As instâncias ‘pm3-*’ são grafos regulares de grau 6 com pesos negativos, confirmando que esta topologia é inerentemente desafiadora para a heurística. As instâncias ‘g3-*’ possuem uma distribuição de pesos muito ampla, com valores positivos e negativos de alta magnitude. Apesar dessa complexidade adicional, o algoritmo consegue encontrar soluções de qualidade, como demonstrado pelo gap de 7.29% para a instância g3-15, que possui 3375 vértices. Este resultado evidencia a capacidade de generalização do framework proposto, que se adapta a diferentes distribuições de peso sem requerer recalibração.

5.4. Análise Agregada e Discussão

A análise dos resultados consolida a principal contribuição deste trabalho: o método proposto estabelece um novo ponto de operação no espectro de desempenho, priorizando a velocidade computacional de forma extrema, enquanto mantém uma qualidade de solução razoável. Enquanto os métodos do estado da arte, como SS e VNSPR, demandam de centenas a dezenas de milhares de segundos para convergir, a abordagem consistentemente resolve as mesmas instâncias em segundos, uma redução de tempo de ordens de magnitude.

O custo associado a essa velocidade é um gap médio de 9.10% em relação às melhores soluções conhecidas. No entanto, este valor é fortemente influenciado pela estrutura do grafo. A análise estatística revela que a presença de pesos negativos é o fator mais determinante: o gap médio para instâncias com apenas pesos positivos é de **4.60%**, um resultado altamente competitivo. Em contrapartida, para instâncias com pesos negativos, o gap médio sobe para **11.43%**. Isso ocorre porque os pesos negativos criam uma paisagem de otimização mais complexa, com objetivos conflitantes (agrupar alguns vizinhos enquanto separa outros), o que é mais difícil de ser capturado por um modelo de ML baseado em características puramente locais.

Adicionalmente, a combinação de alta esparsidade e regularidade estrutural (baixo desvio padrão do grau dos vértices), como visto nas instâncias G10* e G14*, também representa um desafio. A uniformidade das vizinhanças locais fornece menos sinal informativo para o modelo de regressão, limitando a eficácia da sua orientação. O desempenho do método, avaliada através de múltiplas execuções, mostra uma variabilidade controlada. Dada a velocidade do algoritmo, a estratégia de realizar múltiplos arranques (*multi-start*) e reter a melhor solução é não apenas viável, mas recomendada, representando uma abor-

dagem prática que mitiga a estocasticidade e ainda assim permanece ordens de magnitude mais rápida que uma única execução dos métodos estado da arte.

6. Conclusão

Este trabalho introduziu uma abordagem híbrida para o Problema do Corte Máximo que prioriza a eficiência computacional. A contribuição central reside na sinergia entre métodos espectrais e uma busca local acelerada por aprendizado de máquina, que utiliza um modelo de regressão leve para guiar a exploração da vizinhança de forma estruturada e rápida.

Referências

- Bengio, Y., Lodi, A., and Prouvost, A. (2021). Machine learning for combinatorial optimization: a methodological tour d’horizon. *European Journal of Operational Research*, 290(2):405–421.
- Blum, C. and Roli, A. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys (CSUR)*, 35(3):268–308.
- Burer, S., Monteiro, R. D. C., and Zhang, Y. (2001). Rank-two relaxation heuristics for max-cut and other binary quadratic programs. *SIAM Journal on Optimization*, 12(2):503–521.
- Festa, P., Pardalos, P. M., Resende, M. G. C., and Ribeiro, C. C. (2002). Randomized heuristics for the max-cut problem. *Optimization Methods and Software*, 7(1-4):1033–1058.
- Garey, M. R. and Johnson, D. S. (1979). *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, New York.
- Goemans, M. X. and Williamson, D. P. (1995). Improved approximation algorithms for maximum cut and satisfiability problems using semidefinite programming. *Journal of the ACM*, 42(6):1115–1145.
- Hansen, P., Mladenović, N., and Pérez, J. A. M. (2010). Variable neighbourhood search: Methods and applications. *Annals of Operations Research*, 175(1):367–407.
- Helmberg, C. and Rendl, F. (2000). A spectral bundle method for semidefinite programming. *SIAM Journal on Optimization*, 10(3):673–696.
- Karp, R. M. (1972). Reducibility among combinatorial problems. In Miller, R. and Tatchers, J., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York.
- Laguna, M. and Martí, R. (2003). *Scatter Search: Methodology and Implementations in C*. Kluwer Academic Publishers.
- Martí, R., Duarte, A., and Laguna, M. (2009a). Advanced scatter search for the max-cut problem. *INFORMS Journal on Computing*, 21(1):26–38.
- Martí, R., Duarte, A., and Laguna, M. (2009b). Max-cut problem instances and computational results. <https://grafo.etsii.urjc.es/optsiacom/maxcut.html>. Acessado em 08 de Setembro de 2025.