

ARTIGO DE PESQUISA/RESEARCH PAPER

HP-MOCD: Um Algoritmo Evolucionário Multiobjetivo de Alto Desempenho para Detecção de Comunidades em Redes Complexas

HP-MOCD: A High-Performance Multi-Objective Evolutionary Algorithm for Community Detection in Complex Networks

Guilherme O. Santos ✉ [Universidade Federal de Ouro Preto | guilherme.os1@aluno.ufop.edu.br]

Carlos H. G. Ferreira [Universidade Federal de Ouro Preto | chgferreira@ufop.edu.br]

Gladston J. P. Moreira [Universidade Federal de Ouro Preto | gladston@ufop.edu.br]

✉ Departamento de Computação e Sistemas, Universidade Federal de Ouro Preto, João Monlevade, MG, Brasil.

Resumo. A detecção de comunidades em redes complexas é fundamental para domínios sociais, biológicos e tecnológicos. Heurísticas tradicionais de objetivo único, como Louvain e Leiden, são eficientes, porém sofrem com viés de resolução e degenerescência estrutural. Algoritmos evolucionários multiobjetivos (MOEAs) contornam tais limitações otimizando critérios conflitantes em conjunto, mas seus altos custos restringem sua aplicação a redes pequenas. Este trabalho apresenta o HP-MOCD, um MOEA de alto desempenho baseado no NSGA-II, que combina operadores genéticos cientes da topologia, paralelização completa e otimizações em nível de bits, com complexidade $O(G \cdot N_p |V|)$. Experimentos em redes sintéticas e reais demonstram superioridade em diversos cenários e forte escalabilidade: o algoritmo processa grafos com mais de um milhão de nós e supera MOEAs concorrentes em até $553\times$ no tempo de execução. O algoritmo é disponibilizado como biblioteca Python de código aberto.

Abstract. Community detection in complex networks is fundamental across social, biological, and technological domains. Traditional single-objective heuristics, such as Louvain and Leiden, are efficient but suffer from resolution bias and structural degeneracy. Multi-objective evolutionary algorithms (MOEAs) overcome these limitations by jointly optimizing conflicting criteria, yet their high computational cost has historically restricted them to small networks. This work introduces HP-MOCD, a high-performance MOEA built upon NSGA-II that combines topology-aware genetic operators, full parallelization, and bit-level optimizations, achieving $O(G \cdot N_p |V|)$ complexity. Experiments on synthetic and real-world networks show superior accuracy across multiple scenarios and strong scalability: HP-MOCD processes graphs with over one million nodes while outperforming competing MOEAs by up to $553\times$ in execution time. The algorithm is released as an open-source Python library.

Palavras-chave: Detecção de Comunidades, Algoritmos Evolucionários Multiobjetivo, NSGA-II, Redes Complexas, Computação de Alto Desempenho

Keywords: Community Detection, Multi-objective Evolutionary Algorithms, NSGA-II, Complex Networks, High-Performance Computing

Recebido/Received: 16 June 2026 • Aceito/Accepted: 16 June 2026 • Publicado/Published: 10 July 2026

1 Introdução

A análise de sistemas complexos frequentemente se fundamenta na modelagem de redes e na identificação de suas *comunidades*, definidas como grupos de nós densamente conectados internamente e esparsamente conectados entre si [Newman, 2006]. Uma vez que a detecção de comunidades constitui um problema NP-difícil [Newman and Girvan, 2004], a literatura tem amplamente adotado heurísticas rápidas de objetivo único, como os algoritmos de Louvain [Blondel *et al.*, 2008], e de Leiden [Traag *et al.*, 2019]. Contudo, tais métodos apresentam vieses estruturais significativos, particularmente o *limite de resolução* (tendência à fusão de comunidades pequenas) [Fortunato and Barthélemy, 2007] e a *degenerescência* (existência de múltiplas partições distintas com pontuações semelhantes) [Traag *et al.*, 2019; Azevedo *et al.*, 2024].

Para superar tais limitações, os Algoritmos Evolucionários Multiobjetivo (MOEAs) surgem como uma alternativa robusta, otimizando métricas conflitantes de forma simultânea

para produzir uma *Frenteira de Pareto*, um conjunto diversificado de partições de alta qualidade [Moreira and Paquete, 2019]. Apesar do avanço de propostas recentes [Pizzuti, 2009; Shi *et al.*, 2010; Shaik *et al.*, 2021; Dabaghi-Zarandi *et al.*, 2025], o elevado custo computacional dessas abordagens restringe sua aplicação a redes com menos de 10.000 nós, o que limita a adoção de MOEAs em cenários reais [Fu and Wang, 2024; Liu *et al.*, 2024].

Preenchendo não apenas a lacuna de escalabilidade, mas também oferecendo maior robustez a ruídos estruturais e desempenho superior na qualidade das partições, e a possibilidade de seleção de diferentes partições para cada tipo de problema, este trabalho apresenta o **HP-MOCD**, (*High-Performance Multi-Objective Community Detection*), um MOEA escalável baseado no NSGA-II [Deb *et al.*, 2002]. As contribuições centrais do algoritmo incluem: (i) o desenvolvimento de operadores genéticos cientes da topologia local; (ii) a redução da complexidade por geração para $O(N_p |V|)$ em

grafos esparsos; e (iii) uma arquitetura *multithreaded* visando alto desempenho. Este artigo submetido ao CTIC sintetiza os principais resultados da pesquisa desenvolvida no âmbito de um projeto de iniciação científica com duração de um ano, conduzido pelo aluno de graduação Guilherme Oliveira Santos, sob orientação do Prof. Dr. Gladston Moreira e coorientação do Prof. Dr. Carlos H. G. Ferreira. Como principais artefatos resultantes desta iniciação científica, destacam-se:

(i) **Artigo em periódico (Estrato A1)**: Os resultados completos da pesquisa foram publicados na revista *Social Network Analysis and Mining* (Springer, 2025) [Santos et al., 2025], indexado no SCImago Journal Rank (SJR) com classificação Q1 na categoria Media Technology. (ii) **Biblioteca de código aberto**: O algoritmo HP-MOCD foi disponibilizado como uma biblioteca em Python, com o objetivo de facilitar a adoção pela comunidade científica. O código-fonte se encontra no Github. A parte experimental está parcial, visto que a disponibilidade pública dos outros métodos não foi autorizada. (iii) **Software Registrado**: Registramos o software `pymocd` no Instituto Nacional da Propriedade Industrial (INPI), para oferecer este método — e futuros —, tornando-a particularmente fácil para a análise de grafos de grande escala, que, apesar de bem recente, conta com $\approx 43K$ downloads Clickpy e com mais de $\approx 4K$ downloads no último mês PyPIstats.

2 Fundamentos e Trabalhos Relacionados

Em redes complexas, observa-se frequentemente a presença de *comunidades*, isto é, grupos de nós que apresentam maior densidade de conexões internas do que de conexões com o restante da rede [Radicchi et al., 2004]. Formalmente, considere um grafo não direcionado e não ponderado $G = (V, E)$, em que V representa o conjunto de nós e $E \subseteq \{\{u, v\} \mid u, v \in V, u \neq v\}$ o conjunto de arestas. Uma partição comunitária é definida como um conjunto de subconjuntos $\mathcal{C} = \{C_1, C_2, \dots, C_k\}$ tal que:

$$\bigcup_{i=1}^k C_i = V \quad \text{e} \quad C_i \cap C_j = \emptyset \quad \forall i \neq j \quad (1)$$

Neste trabalho, adota-se o modelo de *particionamento rígido*, no qual cada nó pertence exatamente a uma comunidade. A detecção dessas estruturas pode ser tratada como um problema de otimização multiobjetivo (MOP), que busca otimizar simultaneamente duas ou mais funções objetivo, frequentemente conflitantes entre si. Formalmente, busca-se [Ehrgott, 2005]:

$$\text{Minimizar } \mathbf{f}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \mathbf{x} \in \Omega. \quad (2)$$

O conceito de *dominância de Pareto* é utilizado para comparar as soluções geradas: um vetor \mathbf{a} domina \mathbf{b} ($\mathbf{a} \prec \mathbf{b}$) se e somente se $a_i \leq b_i$ para todo i e $a_j < b_j$ para pelo menos um j . O conjunto de todos os vetores objetivo não-dominados forma a *Fronteira de Pareto* (PF), que representa as soluções de compromisso ótimo, nas quais nenhum objetivo pode ser melhorado sem degradar outro [Moreira and Paquete, 2019; Coello et al., 2007].

Nesse contexto, trabalhos pioneiros como MOGA-Net [Pizzuti, 2009] e Shi-MOCD [Shi et al., 2010] demonstraram

o potencial dos MOEAs para superar limitações estruturais de heurísticas clássicas. Nos últimos anos, propostas buscaram aprimorar essa detecção por meio de diferentes estratégias evolucionárias e de aprendizado [Ghoshal et al., 2019, 2021; Yusupov et al., 2021]. Algoritmos recentes incluem KRM e CCM [Shaik et al., 2021], métodos com codificação contínua [Fu and Wang, 2024], estratégias focadas em aprendizado [Liu et al., 2024] e o CDRME [Dabaghi-Zarandi et al., 2025], que incorpora passeios aleatórios. Contudo, apesar desses avanços substanciais, grande parte dos métodos citados falha em demonstrar aplicabilidade prática em redes de grande escala (acima de 10.000 nós), o que motivou o desenvolvimento do HP-MOCD.

3 O Algoritmo HP-MOCD

O HP-MOCD integra o *framework* de otimização NSGA-II a uma arquitetura *multithreaded* e a operadores genéticos cientes da topologia. Para garantir a máxima eficiência computacional, o núcleo foi desenvolvido em Rust e exposto para Python (via PyO3). O fluxo geral de execução do algoritmo é ilustrado na Figura 1 (detalhes melhores descritos no artigo original [Santos et al., 2025]).

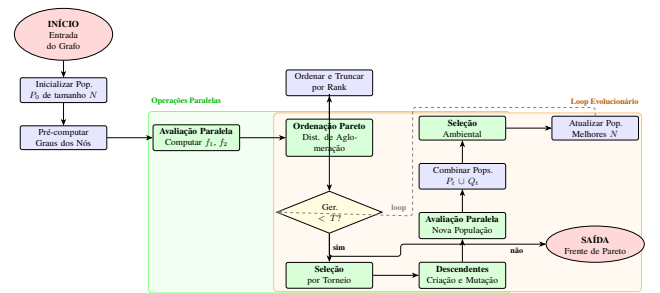


Figura 1. Fluxograma do HP-MOCD. Caixas em verde indicam operações paralelas.

3.1 Formulação e Operadores Evolucionários

A detecção de comunidades é formulada como a minimização simultânea de dois objetivos conflitantes [Shi et al., 2010]: maximizar a coesão interna (f_1) e evitar comunidades excessivamente grandes (f_2). Para uma partição \mathcal{C} do grafo $G(V, E)$ com m arestas, temos:

$$f_1(\mathcal{C}) = 1 - \sum_{c \in \mathcal{C}} \frac{|E(c)|}{m}, \quad f_2(\mathcal{C}) = \sum_{c \in \mathcal{C}} \left(\frac{\sum_{v \in c} \deg(v)}{2m} \right)^2 \quad (3)$$

onde $|E(c)|$ é o número de arestas internas na comunidade c , e $\deg(v)$ é o grau do nó v . Cada indivíduo da população é mapeado de forma eficiente associando cada nó $v \in V$ a um identificador de comunidade em tempo $O(1)$, por meio de um vetor de rótulos indexado pelo identificador inteiro de cada vértice. Esse *layout* compacto reduz a pegada de memória por indivíduo a $O(|V|)$ e habilita consultas e atualizações em tempo constante durante a aplicação dos operadores genéticos.

Para evoluir essas soluções, o HP-MOCD introduz dois operadores customizados que exploram explicitamente a topologia local do grafo, em substituição aos clássicos operadores de ponto-de-corte cego usados em representações vetoriais genéricas:

Cruzamento de Votação (Topology-aware crossover). Em vez de pontos de corte cegos, o descendente herda, de um conjunto de N_p pais, o rótulo da comunidade mais frequente atribuído a cada nó. Formalmente, seja $P = (P_1, \dots, P_{N_p})$ uma coleção de partições parentais e $C_R \in [0, 1]$ um limiar de cruzamento; sorteia-se $x \sim \mathcal{U}[0, 1]$. Se $x > C_R$, o operador é pulado e um dos pais é retornado integralmente, preservando soluções já promissoras. Caso contrário, para cada $v \in V$ contam-se as ocorrências de cada rótulo c entre os pais por $\text{count}(c, v) = \sum_{i=1}^{N_p} 1\{P_i(v) = c\}$, e atribui-se $P_{\text{child}}(v) = \arg \max_c \text{count}(c, v)$, com desempate aleatório uniforme. Como contagens são mantidas em *hash maps* reusados, o custo do operador é $O(|V|)$ esperado por filho, preservando padrões topológicos dominantes sem suprimir a variabilidade necessária à exploração.

Mutação por Vizinhaça (Topology-aware mutation). Com probabilidade M_R , a comunidade de um nó v é trocada pelo rótulo predominante entre seus vizinhos diretos $N(v)$. Cada nó é avaliado de forma independente: amostra-se $r_v \sim \mathcal{U}[0, 1]$ e define-se o conjunto $V_m = \{v \in V : r_v < M_R\}$ de nós alvo. Para cada $v \in V_m$, percorrem-se as arestas incidentes em batches, agregam-se os rótulos dos vizinhos em uma tabela de frequências reaproveitada, e atribui-se a v o rótulo majoritário. Em grafos esparsos ($|E| = O(|V|)$) e com M_R constante, o custo esperado é $O(|V| + M_R \cdot |E|) = O(|V|)$. Diferentemente de mutações puramente aleatórias, esse operador refina as fronteiras das comunidades guiando-se pela conectividade real, reduzindo perturbações destrutivas e acelerando a convergência.

Os indivíduos gerados por esses operadores são submetidos à **ordenação não-dominada** e ao cálculo de **distância de aglomeração** do NSGA-II, garantindo seleção elitista com diversidade preservada na Fronteira de Pareto.

Ao final do processo, o algoritmo retorna uma Fronteira de Pareto. Caso o usuário deseje uma única solução recomendada, aplica-se uma função de agregação escalar que seleciona a partição que maximiza $Q(C) = 1 - f_1 - f_2$. O comportamento desse mecanismo de seleção e a distribuição das soluções obtidas podem ser visualizados na Figura 2. $Q(C)$ é aplicada estritamente *a posteriori*, apenas para fins de recomendação de uma única partição quando o usuário não deseja inspecionar a Fronteira completa: durante todo o processo evolutivo, f_1 e f_2 permanecem como objetivos independentes e a dominância de Pareto é preservada. Embora Q guarde semelhança formal com a modularidade de Newman, ele não induz o conhecido limite de resolução [Fortunato and Barthélemy, 2007]. Sob a formulação de Shi *et al.* [2010] o uso de Q apenas como critério de desempate sobre soluções já não-dominadas atenua esse viés, pois a Fronteira contém partições com diferentes granularidades.

3.2 Análise de Complexidade

O grande diferencial do HP-MOCD para viabilizar a análise de redes em larga escala é a sua arquitetura algorítmica. O uso de pré-computação de graus e de estruturas hash garante que os operadores genéticos operem em $O(|V|)$ por indivíduo. Para os grafos esparsos tipicamente encontrados no mundo real ($|E| \approx O(|V|)$), a complexidade assintótica por geração é dominada pela ordenação não-dominada do NSGA-II e

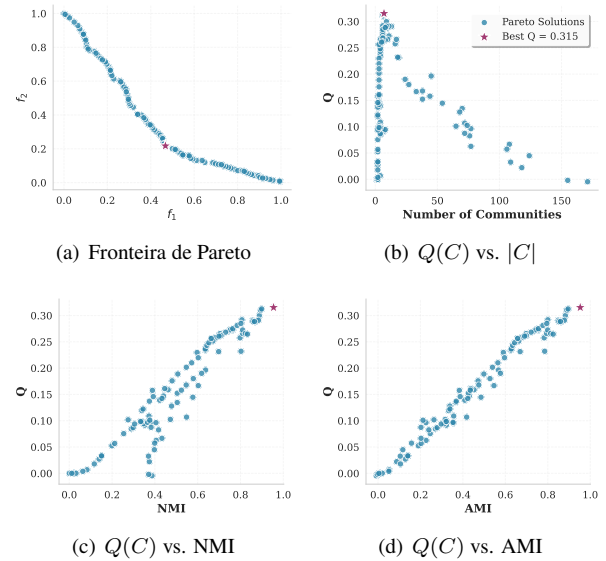


Figura 2. Análise do método de seleção baseado na escalarização.

pelas avaliações locais. Isso reduz drasticamente o gargalo tradicional dos MOEAs, alcançando um custo de $O(N_p|V|)$ por geração, onde N_p é o tamanho da população. Para G gerações, a complexidade total restringe-se a $O(G \cdot N_p|V|)$, o que, aliado à paralelização *multithreaded*, viabiliza a execução em grafos com milhões de nós.

3.3 Impacto da Paralelização Multi-Thread no Desempenho

A natureza intrinsecamente paralela do HP-MOCD motiva uma investigação dedicada sobre como o desempenho escala com o número de *threads* lógicas e qual é o ponto operacional eficiente. Para quantificar esse efeito, executamos o algoritmo em um sistema de alto desempenho equipado com processador AMD Ryzen Threadripper 3960X (24 núcleos físicos, 48 *threads* lógicas, @ 3,70 GHz) e 128 GB de RAM DDR4. O número de *threads* foi variado de 1 a 48 sobre os mesmos grafos sintéticos LFR utilizados na análise de escalabilidade, com $n \in \{10\,000, 30\,000, 50\,000\}$. Cada configuração foi repetida cinco vezes; o desvio-padrão observado foi consistentemente baixo.

A Figura 3 mostra que o aumento do número de *threads* produz reduções substanciais de tempo nas primeiras adições, seguidas por um **platô** alcançado, aproximadamente, com 8 *threads* para 10.000 nós, 15 *threads* para 30.000 nós e 20 *threads* para 50.000 nós. Além desses pontos, ganhos marginais são compensados pelo *overhead* de coordenação entre *threads*, comportamento típico de cargas com dependência de dados.

A evidência mais contundente do papel da paralelização aparece no contraste com a execução estritamente sequencial: comparando o melhor tempo paralelo com o tempo *single-threaded* para o mesmo problema, observam-se *speedups* de **10,27**× para 10.000 nós, **16,45**× para 30.000 nós e **17,83**× para 50.000 nós.

4 Protocolo Experimental

Esta seção descreve o protocolo experimental adotado para avaliar o desempenho do algoritmo proposto.

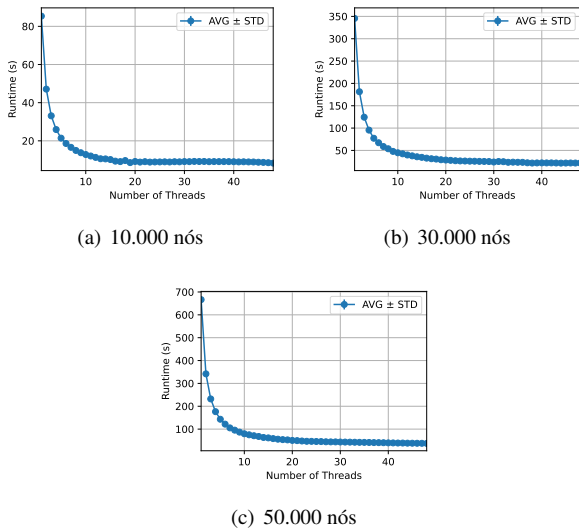


Figura 3. Tempo médio de execução (média $\pm \sigma$) em função do número de *threads* lógicas, para grafos sintéticos LFR com 10.000 (a), 30.000 (b) e 50.000 (c) vértices. Comparações conduzidas no mesmo *hardware* descrito no texto.

4.1 Algoritmos de Comparação

O HP-MOCD foi comparado a um amplo conjunto de heurísticas clássicas e MOEAs, cujas configurações detalhadas são apresentadas na Tabela 1. Os algoritmos incluem: MOCD [Shi *et al.*, 2010], MOGA-Net [Pizzuti, 2009], Louvain [Blondel *et al.*, 2008], Leiden [Traag *et al.*, 2019], ASYN-LPA [Raghavan *et al.*, 2007], KRM e CCM [Shaik *et al.*, 2021] e CDRME [Dabaghi-Zarandi *et al.*, 2025].

4.2 Métricas de Avaliação

Para avaliação, utilizam-se três métricas externas de validação: *Normalized Mutual Information* (NMI), *Adjusted Mutual Information* (AMI) e *F1-score* pareado, que comparam as partições obtidas com a partição de referência (*ground truth*), assumindo valores no intervalo $[0, 1]$, em que valores maiores indicam maior concordância [Vinh *et al.*, 2009, 2010]. Adicionalmente, a modularidade é empregada para avaliar a estrutura interna das comunidades [Newman, 2006]. Testes estatísticos pareados (teste-*t* com $\alpha = 0,05$ e correção de Bonferroni) são aplicados para determinar a significância.

4.3 Redes Sintéticas (LFR)

Emprega-se o gerador de *benchmarks* LFR [Lancichinetti *et al.*, 2008] em duas configurações experimentais distintas, a fim de avaliar os algoritmos em cenários sintéticos:

- **Robustez:** $n = 1.000$ nós fixos, variando o parâmetro de mistura $\mu \in [0,1; 0,8]$ em incrementos de 0,1, que controla a fração de arestas inter-comunitárias;
- **Escalabilidade:** $n \in \{500, 1.000, \dots, 20.000\}$, grau $\in [20, \dots, 50]$ e $\mu = 0,3$.

4.4 Redes do Mundo Real

Para avaliação em cenários reais, utilizam-se quatorze redes bem conhecidas na literatura provenientes de diferentes domínios, incluindo mídias sociais, e-commerce, literatura científica, infraestruturas tecnológicas e sistemas biológicos. As redes variam desde grafos pequenos, como *Karate* (34 nós) e *Dolphins* (62 nós), até grafos de grande escala, como *Youtube* (1.134.890 nós) e *Amazon* (334.863 nós). Todos

os detalhes topológicos dessas redes são descritos no artigo original [Santos *et al.*, 2025].

5 Resultados

Esta seção apresenta os principais resultados experimentais obtidos.

5.1 Redes Sintéticas

Esses experimentos avaliaram a robustez ao ruído e a escalabilidade do algoritmo em redes sintéticas LFR. A Figura 4 detalha o impacto da mistura comunitária (μ). O HP-MOCD apresentou alta acurácia (NMI e AMI estáveis) e tempos de execução constantes até $\mu = 0,4$. Em cenários de alto ruído ($\mu \geq 0,4$), o desempenho superou expressivamente os *baselines* multiobjetivos (alcançando também a maior modularidade entre si) e rivalizou estatisticamente com heurísticas como Louvain e Leiden ($p < 0,01$ para $\mu \leq 0,3$).

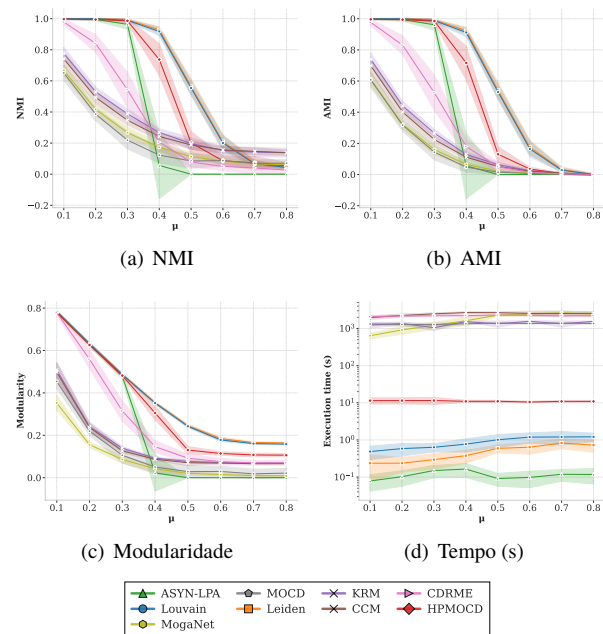


Figura 4. Robustez estrutural (μ) em redes LFR (1.000 nós). Médias de 20 execuções com IC de 95%.

A Figura 5 demonstra a escalabilidade do HP-MOCD com o aumento do tamanho da rede (n). Enquanto outros MOEAs apresentaram degradação severa de qualidade e estouro de tempo (limite de 32h), o HP-MOCD manteve NMI $\approx 1,0$ e processou grafos de 20.000 nós em apenas 148 ± 13 segundos. O ganho de desempenho foi drástico: o algoritmo foi de $132\times$ a $259\times$ mais rápido que KRM, CCM e MOCD em redes de 13k–15,5k nós, e impressionantes $553\times$ superior ao CDRME em 2.000 nós.

Justiça da comparação: Reconhecemos abertamente que parte expressiva dos ganhos de tempo do HP-MOCD frente a CCM, KRM, MOCD e CDRME, não decorre exclusivamente do projeto algorítmico, mas também de escolhas de **engenharia de baixo nível**: o núcleo do HP-MOCD foi escrito em *Rust*, uma linguagem compilada, sem *garbage collector* e com modelo de memória sem alocações ocultas, e explora paralelismo nativo a fio de execução. Mesmo aplicando *baselines* em modo *single-threaded* para reduzir esse viés, a

Tabela 1. Parâmetros dos algoritmos comparados. Ger. (gerações), Pop. (população), C_P (cruzamento), M_P (mutação) e E_S (número de pais).

Algoritmo	Tipo	Ger.	Pop.	C_P	M_P	E_S	Referência
ASYN-LPA	Prop.	–	–	–	–	–	[Raghavan et al., 2007]
Louvain	Guloso	–	–	–	–	–	[Blondel et al., 2008]
Leiden	Guloso	–	–	–	–	–	[Traag et al., 2019]
MOGA-Net	MOEA	100	100	0,9	0,1	–	[Pizzuti, 2009]
MOCD	MOEA	100	100	0,9	0,1	–	[Shi et al., 2010]
KRM	MOEA	200	100	0,9	0,1	–	[Shaik et al., 2021]
CCM	MOEA	200	100	0,9	0,1	–	[Shaik et al., 2021]
CDRME	MOEA	100	100	0,9	0,2	–	[Dabaghi-Zarandi et al., 2025]
HP-MOCD	MOEA	100	100	0,8	0,2	4	Este trabalho

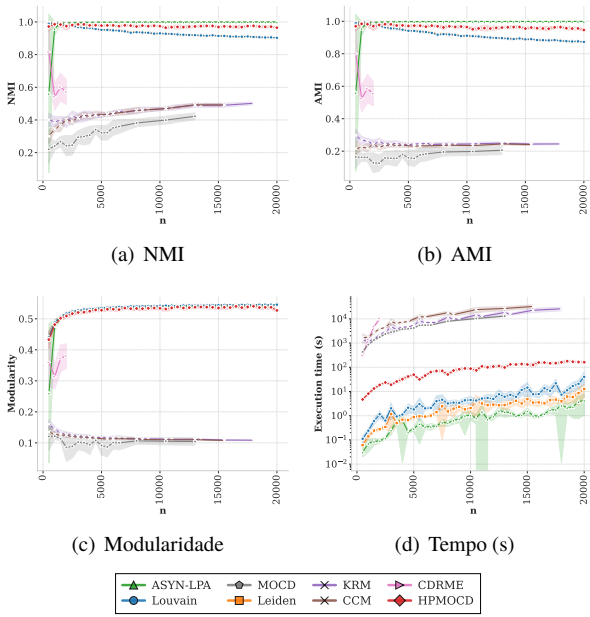


Figura 5. Escalabilidade em redes LFR ($\mu = 0,3$). Lacunas indicam estouro do tempo limite (32h).

vantagem persiste em ordens de magnitude (Veja a Figura 3). Argumentamos que esse acoplamento entre algoritmo e implementação é parte deliberada da contribuição: o objetivo declarado do HP-MOCD é **viabilizar o uso prático de MOEAs em redes massivas**, e isso só é possível casando uma complexidade assintótica favorável ($O(G \cdot N_p|V|)$) a uma implementação que efetivamente concretize esse limite no *hardware* disponível. A redução de complexidade discutida na Seção 3.2 já é suficiente para explicar parte substancial dos ganhos em redes maiores; o restante reflete a eliminação de sobrecargas que historicamente impediam MOEAs de competir com heurísticas gulosas em escalas industriais.

5.2 Redes Reais

A avaliação em redes do mundo real foi conduzida em quatorze bases de dados de domínios variados, desde redes sociais clássicas como *Karate* (34 nós) até grafos massivos de e-commerce e de interações web, como *Amazon* (334.863 nós) e *Youtube* (1.134.890 nós) [Yang and Leskovec, 2012]. O HP-MOCD foi comparado com MOEAs do estado da arte (CCM, KRM, CDRME) e com heurísticas clássicas de objetivo único (Louvain, Leiden, ASYN-LPA).

5.2.1 Escalabilidade em Redes Massivas

O avanço mais notável desta pesquisa reside na quebra da barreira de escalabilidade. O HP-MOCD foi o **único** algoritmo baseado em MOEA capaz de concluir a execução nas

seis maiores redes do *benchmark* no tempo limite de 15 horas estipulado para os experimentos. A Tabela 2 detalha esses resultados.

O HP-MOCD demonstrou escalabilidade em grandes redes, como *Youtube* e *Amazon*, produzindo partições estruturalmente coesas, com modularidade de 0,660 e 0,762, respectivamente. Chama atenção o contraste, na Tabela 2, entre a modularidade elevada e o F1-score quase nulo nessas duas redes massivas (0,033 e 0,007, respectivamente). Esse fenômeno não indica falha do algoritmo, mas sim uma característica conhecida dos *ground truths* dessas bases [Yang and Leskovec, 2012]: as comunidades de referência da *Amazon* correspondem a **categorias de produtos co-comprados** e as da *Youtube* a **grupos de interesse declarados pelos usuários**.

Ambas são rotulagens *funcionais*/baseadas em metadados, frequentemente *sobrepostas*, hierárquicas e nem sempre alinhadas com a estrutura topológica densa que algoritmos de otimização de modularidade — de qualquer família — recuperam. Como o F1-score pareado pune pesadamente toda divergência de pares de nós, qualquer partição topologicamente coerente que não reproduza exatamente a granularidade dos rótulos funcionais é penalizada de forma severa. Esse comportamento é observado em todos os *baselines* avaliados sobre essas redes, incluindo Louvain e Leiden. Por isso, modularidade e AMI/NMI permanecem indicadores mais informativos do que o F1-score pareado em redes de larga escala com *ground truths* desse tipo.

5.3 Síntese Global de Desempenho

A Tabela 3 compila os ganhos percentuais médios do HP-MOCD em relação aos *baselines* multiobjetivo nos diferentes cenários avaliados, consolidando o avanço proporcionado pelo algoritmo.

À primeira vista, a baixa taxa de vitória do HP-MOCD em modularidade pura no cenário de escalabilidade (34,1%, Tabela 3) poderia sugerir limitação do método. Tal leitura, porém, ignora a natureza **multiobjetivo** do algoritmo: o HP-MOCD não otimiza modularidade diretamente, e sim a Fronteira de Pareto sobre (f_1, f_2) definida em (3). Métodos como Louvain e Leiden são, por construção, heurísticas *específicas* para maximizar a modularidade de Newman e, portanto, é natural que a vençam em métrica única. O ganho oferecido pelo HP-MOCD aparece nas demais colunas da mesma tabela: 74,3% de superioridade tanto em NMI quanto em AMI no cenário de escalabilidade, com melhorias médias de 50,86% e 113,24%, respectivamente. Em outras palavras, o HP-MOCD abre mão de uma fração de modularidade “ótima de papel” — frequentemente vulnerável ao limite de resolução [Fortunato and Barthélemy, 2007] e à degenerescência [Azevedo et al.,

Tabela 2. HP-MOCD em todas as redes reais. Redes com † são aquelas que nenhum outro MOEA completou a execução dentro de 15 horas.

Rede	$ V $	AMI	NMI	Modularidade	F1-Score
Youtube†	1.134.890	0,301 ± 0,016	0,538 ± 0,037	0,660 ± 0,009	0,033 ± 0,004
Amazon†	334.863	0,402 ± 0,006	0,667 ± 0,001	0,762 ± 0,012	0,007 ± 0,001
AS†	23.752	0,345 ± 0,010	0,419 ± 0,008	0,503 ± 0,016	0,070 ± 0,018
Cora†	23.166	0,441 ± 0,007	0,524 ± 0,004	0,660 ± 0,023	0,145 ± 0,021
CiteSeer†	3.327	0,199 ± 0,004	0,318 ± 0,003	0,792 ± 0,013	0,033 ± 0,009
Prog.†	3.109	0,342 ± 0,008	0,391 ± 0,007	0,669 ± 0,019	0,150 ± 0,039
OS	2.176	0,282 ± 0,011	0,308 ± 0,010	0,580 ± 0,014	0,195 ± 0,063
Network	1.249	0,254 ± 0,014	0,285 ± 0,012	0,633 ± 0,020	0,176 ± 0,040
Databases	1.046	0,313 ± 0,007	0,369 ± 0,006	0,683 ± 0,023	0,171 ± 0,032
Eu-core	1.005	0,494 ± 0,026	0,563 ± 0,025	0,399 ± 0,009	0,313 ± 0,047
Football	115	0,881 ± 0,013	0,912 ± 0,010	0,584 ± 0,009	0,864 ± 0,023
Polbooks	105	0,502 ± 0,018	0,521 ± 0,016	0,525 ± 0,002	0,743 ± 0,024
Dolphins	62	0,532 ± 0,074	0,550 ± 0,068	0,518 ± 0,006	0,544 ± 0,077
Karate	34	0,658 ± 0,046	0,676 ± 0,041	0,417 ± 0,004	0,698 ± 0,033

Tabela 3. Casos em que HP-MOCD obteve desempenho superior aos *baselines* multiobjetivo e a melhoria média observada.

Cenário	Métrica	Superioridade (%)	Melhoria Média (%)
Escalabilidade	Modularidade	34,1	111,91
	NMI	74,3	50,86
	AMI	74,3	113,24
Robustez (μ)	Modularidade	69,0	227,93
	NMI	63,8	144,88
	AMI	67,2	230,94
Redes Reais	Modularidade	55,0	58,35
	NMI	51,7	13,30
	AMI	55,0	65,60

2024], para entregar partições mais aderentes ao *ground truth* e, simultaneamente, um espectro de soluções de compromisso para o tomador de decisão, vantagem ausente em métodos de objetivo único.

Além dos experimentos de escalabilidade, robustez e análise em redes do mundo real aqui discutidos, investigações adicionais foram conduzidas. Tais experimentos incluíram a avaliação do tempo de convergência por geração, a análise estrutural da diversidade da Fronteira de Pareto gerada e o comportamento da modularidade ao longo do processo evolutivo. Detalhes completos sobre essas análises complementares, juntamente com o aprofundamento estatístico, podem ser consultados no artigo original de referência [Santos *et al.*, 2025].

6 Conclusão

Este trabalho resumiu o HP-MOCD, um algoritmo evolucionário multiobjetivo de alto desempenho para detecção de comunidades em redes complexas de grande escala, publicado na revista *Social Network Analysis and Mining* (Springer, 2025, vol. 15, artigo 110) [Santos *et al.*, 2025]. HP-MOCD incorpora operadores genéticos cientes da topologia, uma arquitetura paralela e uma implementação otimizada em Rust, alcançando complexidade $O(G \cdot N_p |V|)$ em grafos esparsos.

Experimentos extensivos demonstraram que o HP-MOCD supera consistentemente os MOEAs da literatura em tempo de execução e robustez, sendo o único MOEA capaz de processar redes com mais de um milhão de nós dentro de limites práticos de tempo. Em relação aos métodos de objetivo único, sua principal vantagem reside na geração de uma Fronteira de Pareto, que permite estabelecer compromissos específicos para cada aplicação e fornece aos tomadores de decisão um espectro de estruturas comunitárias de alta qualidade.

Declarações complementares

Agradecimentos

Este trabalho foi apoiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq - bolsa 307151/2022-0). Agradecemos também à UFOP pelo apoio.

Financiamento

Financiamento concedido pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), bolsa 307151/2022-0.

Contribuições dos autores

GOS contribuiu para a concepção do estudo, desenvolvimento do algoritmo e dos experimentos, e redação do manuscrito. GJPM e CHGF realizaram a orientação e coordenação do projeto, revisão crítica do conteúdo e aprovação da versão final. Todos os autores leram e aprovaram o manuscrito final.

Conflitos de interesse

Os autores declaram que não têm nenhum conflito de interesses.

Disponibilidade de dados e materiais

O código-fonte do algoritmo HP-MOCD está disponível em <https://github.com/oliveira-sh/pymocd>. Os conjuntos de dados utilizados nos experimentos são públicos e referenciados no artigo original [Santos *et al.*, 2025].

Referências

- Azevedo, B. F., Rocha, A. M. A. C., Fernandes, F. P., Pacheco, M. F., and Pereira, A. I. (2024). Comparison between single and multi-objective clustering algorithms: Mathe case study. In *Optimization, Learning Algorithms and Applications*.
- Blondel, V. D., Guillaume, J.-L., Lambiotte, R., and Lefebvre, E. (2008). Fast unfolding of communities in large networks. *Journal of Statistical Mechanics: Theory and Experiment*, 2008(10):P10008. DOI: <https://doi.org/10.1088/1742-5468/2008/10/P10008>.
- Coello, C. A. C., Lamont, G. B., and Veldhuizen, D. A. V. (2007). *Evolutionary Algorithms for Solving Multi-Objective Problems*. Last edition.
- Dabaghi-Zarandi, F., Afkhami, M. M., and Ashoori, M. H. (2025). Community detection method based on random walk and multi objective evolutionary algorithm in complex networks. *Journal of Network and Computer Applications*.
- Deb, K., Pratap, A., Agarwal, S., and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii.

- IEEE Transactions on Evolutionary Computation*, 6(2):182–197. DOI: <https://doi.org/10.1109/4235.996017>.
- Ehrgott, M. (2005). *Multicriteria Optimization*, volume 491 of *Lecture Notes in Economics and Mathematical Systems*. Springer, Berlin, Heidelberg, 2nd edition.
- Fortunato, S. and Barthélemy, M. (2007). Resolution limit in community detection. *Proceedings of the National Academy of Sciences*, 104(1). DOI: <https://doi.org/10.1073/pnas.0605965104>.
- Fu, J. and Wang, Y. (2024). An improved continuous-encoding-based multiobjective evolutionary algorithm for community detection in complex networks. *IEEE Transactions on Artificial Intelligence*, 5(11):5815–5827.
- Ghoshal, A. K., Das, N., Bhattacharjee, S., and Chakraborty, G. (2019). A fast parallel genetic algorithm based approach for community detection in large networks. In *COMSNETS*, pages 95–101.
- Ghoshal, A. K., Das, N., and Das, S. (2021). Disjoint and overlapping community detection in small-world networks leveraging mean path length. *IEEE Transactions on Computational Social Systems*, 9(2):406–418.
- Lancichinetti, A., Fortunato, S., and Radicchi, F. (2008). Benchmark graphs for testing community detection algorithms. *Phys. Rev. E*, 78:046110. DOI: <https://doi.org/10.1103/PhysRevE.78.046110>.
- Liu, B., Wang, D., and Gao, J. (2024). A multi-objective community detection algorithm with a learning-based strategy. *International Journal of Computational Intelligence Systems*, 17(1):311.
- Moreira, G. and Paquete, L. (2019). Guiding under uniformity measure in the decision space. In *2019 IEEE Latin American Conference on Computational Intelligence (LA-CCI)*, pages 1–6. DOI: <https://doi.org/10.1109/LA-CCI47412.2019.9037034>.
- Newman, M. E. (2006). Modularity and community structure in networks. *Proceedings of the national academy of sciences*, 103(23):8577–8582.
- Newman, M. E. and Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical review E*, 69(2):026113.
- Pizzuti, C. (2009). A multi-objective genetic algorithm for community detection in networks. In *21st IEEE International Conference on Tools with Artificial Intelligence*, pages 379–386. DOI: <https://doi.org/10.1109/ICTAI.2009.58>.
- Radicchi, F., Castellano, C., Cecconi, F., Loreto, V., and Parisi, D. (2004). Defining and identifying communities in networks. *Proceedings of the National Academy of Sciences*, 101(9):2658–2663. DOI: <https://doi.org/10.1073/pnas.0400054101>.
- Raghavan, U. N., Albert, R., and Kumara, S. (2007). Near linear time algorithm to detect community structures in large-scale networks. *Physical Review E—Statistical, Non-linear, and Soft Matter Physics*, 76(3):036106.
- Santos, G. O., Vieira, L. S., Rossetti, G., Ferreira, C. H. G., and Moreira, G. J. P. (2025). A high-performance evolutionary multiobjective community detection algorithm. *Social Network Analysis and Mining*, 15(1):110. DOI: [10.1007/s13278-025-01519-7](https://doi.org/10.1007/s13278-025-01519-7).
- Shaik, T., Ravi, V., and Deb, K. (2021). Evolutionary multi-objective optimization algorithm for community detection in complex social networks. *SN Computer Science*, 2(1):1.
- Shi, C., Zhong, C., Yan, Z., Cai, Y., and Wu, B. (2010). A multi-objective approach for community detection in complex network. In *IEEE Congress on Evolutionary Computation*, pages 1–8. DOI: <https://doi.org/10.1109/CEC.2010.5585987>.
- Traag, V. A., Waltman, L., and van Eck, N. J. (2019). From Louvain to Leiden: guaranteeing well-connected communities. *Scientific Reports*, 9(1):5233. DOI: [10.1038/s41598-019-41695-z](https://doi.org/10.1038/s41598-019-41695-z).
- Vinh, N. X., Epps, J., and Bailey, J. (2009). Information theoretic measures for clusterings comparison: is a correction for chance necessary? In *Proceedings of the 26th annual international conference on machine learning*, pages 1073–1080.
- Vinh, N. X., Epps, J., and Bailey, J. (2010). Information theoretic measures for clusterings comparison: Variants, properties, normalization and correction for chance. *Journal of Machine Learning Research*, 11:2837–2854.
- Yang, J. and Leskovec, J. (2012). Defining and evaluating network communities based on ground-truth. In *Proceedings of the ACM SIGKDD workshop on mining data semantics*, pages 1–8.
- Yusupov, J., Palakonda, V., Mallipeddi, R., and Veluvolu, K. C. (2021). Multi-objective evolutionary algorithm based on ensemble of initializations for overlapping community detection. In *International Conference on Electronics, Information, and Communication (ICEIC)*, pages 1–7.