UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ

LEONARDO BRUNNO SINK LOPES

FRAMEWORK DE BENCHMARK PARA SELEÇÃO DE ALGORITMOS DE PARTICIONAMENTO DE GRAFOS

PATO BRANCO

2026

LEONARDO BRUNNO SINK LOPES

FRAMEWORK DE BENCHMARK PARA SELEÇÃO DE ALGORITMOS DE PARTICIONAMENTO DE GRAFOS

Benchmark Framework for Graph Partitioning Algorithm Selection

Trabalho de Conclusão de Curso de Graduação como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.
Orientador: Prof. Dr. Marco Antonio de Castro Barbosa
Coorientador: Prof. Dr. Gustavo Weber Denardin

PATO BRANCO

2026

LEONARDO BRUNNO SINK LOPES

FRAMEWORK DE BENCHMARK PARA SELEÇÃO DE ALGORITMOS DE PARTICIONAMENTO DE GRAFOS

Trabalho de Conclusão de Curso de Graduação como requisito para obtenção do título de Bacharel em Engenharia de Computação do Curso de Engenharia de Computação da Universidade Tecnológica Federal do Paraná.

Data de aprovação: 24 / junho / 2026

Marco Antonio de Castro Barbosa
Doutorado em Informática
Universidade Tecnológica Federal do Paraná

Viviane Dal Molin
Doutorado em Informática
Universidade Tecnológica Federal do Paraná

Dalcimar Casanova
Doutorado em Física Computacional
Universidade Tecnológica Federal do Paraná

PATO BRANCO

2026

Dedico este trabalho à minha família.
Aos meus pais, à minha irmã e aos meus avós que já partiram, por terem sido a base que me sustentou ao longo desta caminhada. Pelo apoio, pela confiança, pelo incentivo e pelos sacrifícios, mesmo quando o percurso parecia incerto. Se chego ao fim desta etapa, é porque nunca caminhei sozinho.

AGRADECIMENTOS

Agradeço ao meu orientador, Prof. Dr. Marco Antonio de Castro Barbosa, pela orientação, pela confiança e pelo acompanhamento durante o desenvolvimento deste trabalho. A oportunidade de participar da iniciação científica, em um momento importante da minha trajetória, teve papel decisivo na continuidade da minha formação acadêmica e na construção do caminho que tornou este trabalho possível. Sou grato pelas contribuições e pela confiança ao longo desse percurso.

Agradeço também ao meu coorientador, Prof. Dr. Gustavo Weber Denardin, pelo apoio ao longo do processo.

Aos colegas que fizeram parte da minha rotina na universidade, nos projetos, nos espaços de estudo e no ambiente de desenvolvimento acadêmico e profissional, deixo meu sincero agradecimento. Com muitos deles compartilhei disciplinas, laboratórios, conversas, dúvidas, noites de trabalho e momentos difíceis. A presença, a ajuda e a parceria desses colegas foram importantes para que eu seguisse em frente em diferentes etapas do curso. Mais do que dividir espaços, dividimos dificuldades, aprendizados e pequenas conquistas que marcaram minha graduação.

A todos que, de alguma forma, fizeram parte desta caminhada, meu agradecimento.

RESUMO

Redes de sensores, redes sociais, sistemas computacionais, malhas viárias, mapas de rotas e bases científicas podem ser representados como grafos, ou seja, conjuntos de elementos conectados por relações. Em sistemas desse tipo, particionar um grafo significa dividir a rede em partes equilibradas, reduzindo as conexões cortadas entre elas. Essa tarefa pertence à otimização combinatória e aparece em problemas práticos de organização de dados, distribuição de carga computacional, análise de redes e redução de comunicação entre componentes de um sistema. Como diferentes estruturas de grafo, números de partes e limites de tempo podem favorecer estratégias distintas, a escolha do algoritmo de particionamento não é universal. Este trabalho investiga essa variação por meio de um *framework* de *benchmarking* experimental para o problema de particionamento de grafos, combinando otimização em grafos, experimentação controlada, seleção de algoritmos e aprendizado de máquina. Foram comparados os particionadores *multilevel* METIS e KaHIP e quatro meta-heurísticas implementadas em Rust: *simulated annealing*, *tabu search*, *iterated local search* e GRASP. A campanha avaliou 60 grafos reais e sintéticos, 298 combinações entre grafo e número de partes e sete limites de tempo, formando 2.071 comparações completas. Os resultados mostram um cenário heterogêneo, no qual a melhor escolha depende da instância e do limite de tempo. As meta-heurísticas obtiveram 950 vitórias estritas em 45 grafos, com crescimento de 30,3 por cento em 1 segundo para 52,0 por cento a partir de 60 segundos, enquanto os métodos *multilevel* mantiveram vantagem em outras regiões avaliadas. Em uma comparação pareada com os seis algoritmos, a inclusão de *iterated local search* e GRASP produziu 135 mudanças de algoritmo vencedor, incluindo 74 casos em que a melhor família deixou de ser *multilevel* e passou a ser meta-heurística. Para transformar os resultados em apoio interpretável à tomada de decisão, foram treinadas árvores de classificação e regressão, uma técnica de aprendizado de máquina que produz regras interpretáveis. Esses modelos alcançaram acurácia balanceada entre 0,7958 e 0,8810 nos alvos binários de referência, indicando que parte da variação observada pode ser descrita por padrões associados às propriedades dos grafos, ao número de partes e ao limite de tempo. O trabalho entrega uma base experimental, um protocolo de comparação temporal entre famílias distintas de algoritmos e uma análise interpretável para apoiar a seleção de algoritmos por instância. As conclusões se restringem ao painel, às implementações e ao ambiente avaliados.

Palavras-chave: particionamento de grafos; seleção de algoritmos; benchmarking rastreável; meta-heurísticas anytime; aprendizado de máquina.

ABSTRACT

Sensor networks, social networks, computer systems, road networks, route maps, and scientific datasets can be represented as graphs, that is, sets of elements connected by relationships. In such systems, graph partitioning means dividing the network into balanced parts while reducing the connections cut between them. This task belongs to combinatorial optimization and appears in practical problems involving data organization, computational load distribution, network analysis, and communication reduction between components of a system. Because different graph structures, numbers of parts, and time limits may favor different strategies, the choice of a partitioning algorithm is not universal. This work investigates this variation through an experimental benchmarking framework for the graph partitioning problem, combining graph optimization, controlled experimentation, algorithm selection, and machine learning. The study compared the multilevel partitioners METIS and KaHIP with four metaheuristics implemented in Rust: simulated annealing, tabu search, iterated local search, and GRASP. The campaign evaluated 60 real and synthetic graphs, 298 combinations between graph and number of parts, and seven time limits, resulting in 2,071 complete comparisons. The results show a heterogeneous scenario in which the best choice depends on the instance and the time limit. The metaheuristics obtained 950 strict wins across 45 graphs, with an increase from 30.3 percent at 1 second to 52.0 percent from 60 seconds onward, while the multilevel methods maintained an advantage in other evaluated regions. In a paired comparison with all six algorithms, the inclusion of iterated local search and GRASP produced 135 changes in the winning algorithm, including 74 cases in which the best family changed from multilevel to metaheuristic. To turn these results into interpretable support for decision-making, classification and regression trees were trained as a machine learning technique that produces interpretable rules. These models achieved balanced accuracy between 0.7958 and 0.8810 on the reference binary targets, indicating that part of the observed variation can be described by patterns associated with graph properties, the number of parts, and the time limit. The work delivers an experimental dataset, a temporal comparison protocol between distinct algorithm families, and an interpretable analysis to support instance-specific algorithm selection. The conclusions are limited to the evaluated panel, implementations, and execution environment.

Keywords: graph partitioning; algorithm selection; traceable benchmarking; anytime metaheuristics; machine learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE QUADROS

LISTA DE ABREVIATURAS E SIGLAS

ASP	Algorithm Selection Problem
AutoML	Automated Machine Learning
C1	Primeiro recorte de cobertura usado como verificação de sensibilidade
C2	Segundo recorte de cobertura usado na análise completa com seis algoritmos
CART	Classification and Regression Trees
CC0	Creative Commons Zero
CPU	Central Processing Unit
CSV	Comma-Separated Values
ECDF	Empirical Cumulative Distribution Function
ERT	Expected Running Time
FM	Fiduccia–Mattheyses
GPP	Graph Partitioning Problem
GPU	Graphics Processing Unit
GRASP	Greedy Randomized Adaptive Search Procedure
HEM	Heavy-Edge Matching
HPC	High-Performance Computing
IA	Inteligência artificial
IC	Intervalo de confiança
ILS	Iterated Local Search
JSON	JavaScript Object Notation
KaHIP	Karlsruhe High Quality Partitioning
KL	Kernighan–Lin
LFR	Lancichinetti–Fortunato–Radicchi
METIS	Particionador *multilevel* METIS
MKL	Math Kernel Library
MPI	Message Passing Interface
NFE	Number of Function Evaluations
NUMEXPR	Biblioteca NumExpr
ODbL	Open Database License
ODC-BY	Open Data Commons Attribution License
OGB	Open Graph Benchmark
OMP	Open Multi-Processing
OPENBLAS	Biblioteca OpenBLAS
OSM	OpenStreetMap
PQ	Pergunta de pesquisa
RAM	Random Access Memory

RCL	Restricted Candidate List
SA	Simulated Annealing
SBS	Single Best Solver
SNAP	Stanford Network Analysis Platform
TIGER/Line	Topologically Integrated Geographic Encoding and Referencing Line
TS	Tabu Search
TTT	Time To Target
VBS	Virtual Best Solver
WSL2	Windows Subsystem for Linux 2

LISTA DE SÍMBOLOS

$G = (V,E)$	Grafo simples, não direcionado e não ponderado
V	Conjunto de vértices
E	Conjunto de arestas
$n = \|V\|$	Número de vértices
$m = \|E\|$	Número de arestas
u,v	Vértices que definem uma aresta (u,v)
k	Número de blocos do particionamento
P	Particionamento dos vértices em k blocos
B_j	Bloco j do particionamento
χ	Função que associa cada vértice a um bloco, $\chi : V \to \{1,\ldots,k\}$
$\mathbf{1}[\cdot]$	Função indicadora, igual a 1 se a condição é verdadeira e a 0 caso contrário
$\mathrm{cut}(P)$	Número de arestas com extremos em blocos diferentes
ε	Tolerância de balanceamento por cardinalidade
$L(n,k,\varepsilon)$	Maior tamanho permitido para um bloco
$\delta(P)$	Desequilíbrio efetivo de P em relação à média n/k
C_{ml}	Menor corte elegível entre os métodos *multilevel* do recorte
C_{meta}	Menor corte elegível entre as meta-heurísticas do recorte
g	Diferença relativa entre C_{meta} e C_{ml}
BA	Acurácia balanceada usada na avaliação dos classificadores
F_1	Medida F_1 usada na avaliação dos classificadores
B	Número de permutações no teste empírico de permutação
b	Número de permutações com pontuação igual ou superior à observada
p	Valor empírico do teste de permutação

SUMÁRIO

1 INTRODUÇÃO

Grafos representam entidades por vértices e suas relações por arestas em aplicações como redes de computadores, malhas de transporte, circuitos eletrônicos e redes sociais (NEWMAN, 2010). Quando o grafo cresce, dividi-lo em partes menores pode reduzir os recursos exigidos por análises e processamento. Essa divisão constitui o problema de particionamento de grafos (*graph partitioning problem*, GPP).

Um particionamento atribui os vértices a blocos e concilia duas exigências. Os blocos devem manter tamanhos próximos para distribuir a carga, e o número de arestas entre blocos deve ser reduzido. Essa segunda medida é o corte de arestas, ou *edge cut* (KARYPIS; KUMAR, 1998).

1.1 Motivação aplicada e delimitação

A motivação aplicada do trabalho vem de sistemas distribuídos que precisam dividir uma rede em grupos com baixo acoplamento entre si. Em uma rede de sensores, os vértices podem representar dispositivos e as arestas podem representar comunicação, proximidade ou dependência de medição (ELBHIRI *et al.*, 2013; CHIKARADDI *et al.*, 2020). Em uma frota de robôs ou veículos autônomos, os vértices podem representar unidades, regiões de atuação ou tarefas, enquanto as arestas podem indicar proximidade espacial, necessidade de comunicação ou dependência operacional. Nesses casos, particionar o grafo significa formar grupos de elementos, buscando reduzir as interações entre grupos diferentes sem concentrar elementos demais em um único bloco.

Nesse tipo de cenário, a questão prática é escolher, dado o estado atual da rede, o número de grupos desejado e o limite de tempo disponível, qual algoritmo tende a produzir uma partição adequada. Essa pergunta motiva a comparação por orçamento de tempo adotada neste trabalho, em que orçamento designa o limite de tempo de parede concedido a uma execução ou comparação. Diferentes algoritmos podem produzir resultados distintos conforme a estrutura do grafo, o valor de k e o tempo disponível para execução.

Esta monografia, porém, não implementa o controle de uma frota, o roteamento de sensores nem uma aplicação distribuída em produção. Também não trata reparticionamento dinâmico ou atualização contínua do grafo. O estudo permanece no nível algorítmico e experimental. Ele constrói e avalia um *framework* de *benchmarking* para comparar algoritmos de particionamento sob condições controladas, registrar resultados elegíveis em diferentes orçamentos de tempo e examinar padrões associados à seleção de algoritmos no painel avaliado.

O desempenho dos algoritmos varia entre instâncias e configurações do problema (RICE, 1976; KOTTHOFF, 2014). Particionadores *multilevel* reduzem o grafo, constroem uma partição em escala menor e refinam a solução durante o retorno ao grafo original (BULUC *et al.*, 2016). Meta-heurísticas *anytime*, como *simulated annealing* (SA), *tabu search* (TS), *iterated local search* (ILS) e *greedy randomized adaptive search procedure* (GRASP), percorrem solu-

ções candidatas e podem reduzir o menor corte registrado ao longo do tempo (BLUM; ROLI, 2003). Diferenças de tamanho, densidade, distribuição de graus e conectividade tornam a escolha dependente do grafo e da configuração analisada.

Essa variação impede que a escolha seja reduzida a uma regra única, como associar grafos grandes a uma ferramenta específica. O mesmo algoritmo pode produzir cortes distintos quando mudam a estrutura do grafo, o número de blocos ou o tempo disponível. A comparação precisa registrar essas condições junto ao resultado, em vez de tratar cada ferramenta como superior ou inferior de forma geral (HOOKER, 1995).

O problema de seleção de algoritmos (ASP, *algorithm selection problem*) formaliza essa escolha ao relacionar características da instância ao participante associado ao melhor resultado segundo um critério definido (RICE, 1976). A seleção exige dados obtidos sob condições comparáveis, pois o rótulo depende da função de qualidade, do orçamento de tempo e do conjunto de participantes. Esses dados vêm de um painel de grafos reais e sintéticos avaliado por grafo, valor de k e orçamento de tempo. Árvores de decisão CART (*classification and regression trees*) associam atributos conhecidos antes da execução aos rótulos produzidos pelo protocolo experimental (BREIMAN *et al.*, 1984). As divisões da árvore permitem examinar quais atributos aparecem nas regras dentro do painel estudado. Essa interpretação permanece condicionada ao portfólio e às instâncias avaliadas.

A comparação entre famílias exige uma medida de esforço disponível para todos os participantes. Tempo de parede é o tempo externo decorrido durante a execução, medido pelo *runner*. Sob o ambiente descrito na seção 3.2, ele fornece a referência comum de esforço temporal (MCGEOCH, 2012). O número de avaliações da função objetivo (NFE) permanece como diagnóstico das meta-heurísticas instrumentadas, pois METIS e KaHIP são acionados como executáveis externos, sem exigir os mesmos contadores internos (AUGER; BROCKHOFF; HANSEN, 2021; KARYPIS; KUMAR, 1998; SANDERS; SCHULZ, 2013). As regras de elegibilidade temporal, reconstrução das fatias e agregação são definidas no capítulo 3.

1.2 Problema e formulação

Adota-se o GPP não ponderado, k-*way*, com restrição de balanceamento por cardinalidade. Considera-se um grafo simples e não direcionado $G = (V,E)$, com $n = |V|$, em que vértices e arestas têm peso unitário. O objetivo é dividir V em $k \geq 2$ blocos, respeitando uma tolerância de balanceamento $\varepsilon \in [0,1)$, e minimizar o corte de arestas entre blocos distintos.

O corte de arestas é a função de qualidade, o balanceamento define a viabilidade e a disponibilidade dentro do orçamento é analisada separadamente. A seção 2.2 relaciona essas escolhas à literatura.

Um particionamento k-*way* divide V em blocos disjuntos $P = \{B_1, \ldots, B_k\}$, cuja união é V. A função $\chi : V \to \{1, \ldots, k\}$ indica o bloco de cada vértice. O corte adotado é definido em (1).

$$\mathrm{cut}(P) = \sum_{(u,v)\in E} \mathbf{1}[\chi(u) \neq \chi(v)]. \tag{1}$$

A função indicadora $\mathbf{1}[\cdot]$ vale 1 quando seus argumentos são verdadeiros e 0 nos demais casos. Cada aresta contribui para o corte somente quando seus extremos pertencem a blocos diferentes.

O balanceamento limita a cardinalidade dos blocos e impede que uma solução reduza o corte concentrando a maior parte dos vértices em poucas partes. Como $|V|/k$ pode não ser inteiro, aplica-se a tolerância ε antes do arredondamento por teto. O limite $L(n,k,\varepsilon)$ representa o maior tamanho permitido para qualquer bloco, e a restrição é dada em (2).

$$L(n,k,\varepsilon) = \left\lceil (1+\varepsilon)\frac{n}{k} \right\rceil, \qquad \max_{1\leq j\leq k} |B_j| \leq L(n,k,\varepsilon). \tag{2}$$

O maior bloco pode exceder a média ideal somente dentro da folga definida por ε e pelo arredondamento inteiro. Quanto menor ε, mais próximos da média os blocos devem ficar. Valores maiores admitem diferenças maiores de cardinalidade.

Além do teste de admissibilidade, o desequilíbrio efetivo informa quanto o maior bloco se afasta da média ideal. Essa medida é definida em (3).

$$\delta(P) \;=\; \max_{j} \frac{|B_j|}{|V|/k} - 1. \tag{3}$$

O valor $\delta(P) = 0$ indica que o maior bloco coincide com a média ideal. Valores positivos medem a fração pela qual o maior bloco a ultrapassa. A medida descreve o resultado obtido, enquanto $L(n,k,\varepsilon)$ define o teste inteiro de aceitação. A condição $\delta(P) \leq \varepsilon$ expressa a mesma ideia de balanceamento de (2), ressalvado o arredondamento por teto.

O problema adotado é formulado em (4).

$$\min_{\{B_1,\ldots,B_k\}} \mathrm{cut}(P) \quad \text{sujeito à restrição de balanceamento em (2).} \tag{4}$$

A tolerância foi fixada em $\varepsilon = 0{,}03$ e convertida para as interfaces das ferramentas avaliadas. A validação posterior aplica novamente o limite matemático comum, evitando que diferenças de parametrização das interfaces alterem a condição de aceitação. O corte permaneceu como função de qualidade sujeita ao balanceamento.

1.3 Objetivo geral

Desenvolver e avaliar um *framework* de *benchmarking* para apoiar a seleção explicável de algoritmos de particionamento de grafos sob orçamento de tempo.

1.4 Objetivos específicos

- Definir um protocolo de comparação por orçamento de tempo para particionadores *multilevel* e meta-heurísticas sob a mesma função objetivo, tolerância de balanceamento e validação das partições.
- Organizar um painel de grafos reais e sintéticos, caracterizado por atributos topológicos calculados antes da execução dos algoritmos.
- Executar e validar a campanha de *benchmarking* para METIS, KaHIP, SA, TS, ILS e GRASP, registrando estados, tempos, sementes, partições e metadados de execução.
- Construir bases analíticas para comparar qualidade, disponibilidade, família do algoritmo selecionado, vitórias com menor corte meta-heurístico e efeito do portfólio nos recortes definidos.
- Treinar e avaliar CARTs por validação cruzada agrupada por grafo, examinando regras associadas aos atributos das instâncias, ao valor de k e ao orçamento de tempo.
- Delimitar o alcance dos resultados em relação ao painel, às implementações, ao ambiente e à avaliação interna dos modelos.

1.5 Perguntas de pesquisa

As perguntas são respondidas sob os recortes e as regras definidos no capítulo 3.

PQ1 Como a qualidade relativa varia entre os orçamentos nos recortes de qualidade, e como a disponibilidade dos resultados varia na base operacional definida para esse fim?

PQ2 Em quais condições topológicas e experimentais as meta-heurísticas vencem, empatam ou permanecem competitivas em relação aos particionadores *multilevel*?

PQ3 Em que medida os atributos das instâncias e as variáveis da configuração experimental permitem classificar os rótulos de qualidade dentro do painel estudado, e quais regras são produzidas pelos CARTs?

1.6 Escopo, premissas e não-objetivos

O escopo delimita o problema, as instâncias, o portfólio e o regime de execução aos quais as conclusões se aplicam. As premissas registram decisões mantidas durante a campanha, e os não-objetivos separam o estudo de extensões que exigiriam outro protocolo.

1.6.1 Escopo

- Problema. GPP não ponderado, k-*way*, com balanceamento por cardinalidade e tolerância ε comum às ferramentas.
- Instâncias. Painel sintético e real avaliado sob a mesma função-objetivo e a mesma tolerância de desequilíbrio.
- Regime de execução. Avaliação *offline*, em CPU e modo *mono-thread*, sem GPU.
- Portfólio. SA, TS, ILS e GRASP como meta-heurísticas *anytime*, com METIS e KaHIP como particionadores *multilevel* acionados como executáveis externos.

1.6.2 Premissas operacionais da campanha

- Repetições dos métodos estocásticos. Foram previstas cinco sementes por combinação de grafo e valor de k. Cada meta-heurística é representada pelo menor corte temporalmente elegível entre as repetições disponíveis.
- Configurações anteriores à campanha. Orçamentos, mapeamentos da tolerância e perfis de hiperparâmetros foram fixados antes da campanha final. Temperatura, resfriamento, perturbação, duração tabu e demais parâmetros não foram ajustados por instância.
- Ambiente rastreável. Os registros documentam código, dados, configurações e artefatos, com as limitações de versionamento descritas na metodologia.

1.6.3 Não-objetivos

- Tratar pesos de vértice ou aresta, capacidades de bloco, múltiplas restrições ou hipergrafos.
- Adotar outra função-objetivo principal, como condutância, *normalized cut*, expansão ou formulações multiobjetivo.
- Avaliar MPI, GPU ou reparticionamento dinâmico.
- Realizar *tuning* sistemático, *AutoML* ou ablação de inicialização.
- Usar métricas de *restarts*, como *expected running time* (ERT).
- Desenvolver novos particionadores. O estudo compara as implementações disponíveis sob o protocolo definido.

1.7 Protocolo de comparação e rastreabilidade

O protocolo de comparação por tempo de parede compara resultados elegíveis sob o mesmo orçamento de tempo em cada fatia, definida por grafo, valor de k e limite temporal. O uso do mesmo limite temporal em cada fatia estabelece uma condição externa comum, sem equiparar operações internas, esforço acumulado ou limites físicos das chamadas. O capítulo 3 especifica a reconstrução temporal e os critérios de inclusão.

Todos os métodos usam a mesma função-objetivo e tolerância de balanceamento, com configurações fixadas antes da campanha. As partições materializadas são verificadas quanto à integridade, ao balanceamento e ao corte. Qualidade e disponibilidade permanecem em bases distintas, pois cada análise usa seu próprio conjunto de fatias válidas como denominador. Os registros vinculam sementes, orçamentos, configurações, estados e saídas às chamadas que os produziram.

1.8 Contribuições

As contribuições do trabalho concentram-se no protocolo de comparação, na infraestrutura experimental e nos CARTs avaliados no painel.

- Definição de um protocolo para comparar particionadores *multilevel* e meta-heurísticas sob orçamento de tempo de parede, execução mono-*thread*, controle de aleatoriedade quando aplicável e validação independente das partições produzidas.
- Implementação de uma infraestrutura rastreável e organização de uma base com instâncias sintéticas e reais, atributos estruturais, métricas de qualidade, tempos e metadados de execução e verificação.
- Construção e avaliação interna de CARTs que relacionam atributos das instâncias e variáveis da configuração experimental aos rótulos definidos pelo *benchmarking*, com validação agrupada por grafo.

1.9 Convenções e notação

Usa-se $G = (V,E)$ para um grafo não direcionado, com $n = |V|$ vértices e $m = |E|$ arestas. Um particionamento k-*way* é denotado por $P = \{B_1, \ldots, B_k\}$, e $\chi : V \to \{1, \ldots, k\}$ indica o bloco de cada vértice. Vértices e arestas têm peso unitário.

O corte é denotado por $\mathrm{cut}(P)$. O termo *cutsize* aparece somente quando reproduz a nomenclatura de ferramentas ou artefatos. O desequilíbrio efetivo é

$$\delta(P) = \max_j \frac{|B_j|}{n/k} - 1.$$

A tolerância ε segue a regra de arredondamento de (2). NFE designa o número de avaliações da função objetivo registrado pelas meta-heurísticas instrumentadas e tem função diagnóstica. O tempo de parede é a medida externa usada entre famílias.

Uma fatia analítica é definida por grafo, valor de k e orçamento de tempo. O algoritmo e a semente não criam novas fatias. Eles são participantes ou repetições dentro da mesma combinação analítica. Resultado temporal elegível designa o valor aceito pela reconstrução dentro desse orçamento. A mesma execução física pode contribuir para mais de uma fatia, o que torna os orçamentos derivados da mesma chamada dependentes entre si. Salvo indicação contrária, os tempos foram medidos por relógio monotônico em ambiente controlado e execução *mono-thread*.

1.10 Organização do trabalho

A monografia está organizada em mais quatro capítulos. O capítulo 2 discute o problema de particionamento de grafos, métricas de avaliação, métodos de solução e seleção de algoritmos por instância. O capítulo 3 apresenta o ambiente experimental, o painel de grafos, o protocolo de comparação, a construção das bases analíticas e os modelos CART. O capítulo 4 apresenta a disponibilidade dos dados, as comparações entre algoritmos, os resultados dos modelos e as análises complementares auditadas. O capítulo 5 responde às perguntas de pesquisa, sintetiza as contribuições e delimita o alcance dos achados.

2 TRABALHOS RELACIONADOS

2.1 Escopo e tese da revisão

A estrutura do grafo pode estar associada ao desempenho dos algoritmos de particionamento. Densidade, distribuição de graus, pseudo-diâmetro, componentes e conexões entre regiões pouco integradas descrevem aspectos que variam entre instâncias (NEWMAN, 2010; BULUC *et al.*, 2016).

Comparar famílias com instrumentação diferente requer uma função de qualidade comum, uma regra temporal e critérios explícitos de inclusão. O NFE descreve o esforço interno das meta-heurísticas instrumentadas, mas não está disponível de forma compatível em METIS e KaHIP (AUGER; BROCKHOFF; HANSEN, 2021). Esse recorte prioriza os métodos avaliados, os atributos usados para descrever as instâncias e os controles necessários para relacionar essas duas dimensões (HOOKER, 1995; MCGEOCH, 2012).

Em redes de sensores, robôs ou sistemas distribuídos, a aresta cortada pode representar uma interação entre grupos que exige comunicação ou coordenação. O particionamento só é útil quando reduz esse acoplamento sem concentrar carga demais em um bloco. Essa relação entre aplicação e formulação justifica tratar corte e balanceamento como critérios simultâneos, e não como medidas independentes.

2.2 Métricas de qualidade e esforço

O trabalho minimiza o corte de arestas entre partições que atendem ao balanceamento. Essa escolha trata o corte como medida de qualidade e o balanceamento como condição de aceitação, em vez de combinar os dois valores em uma função multiobjetivo. No protocolo, o tempo de parede define a referência externa de comparação. A disponibilidade registra se há resultado no limite analisado, enquanto o NFE descreve a busca instrumentada quando essa medida está disponível.

2.2.1 Corte de arestas

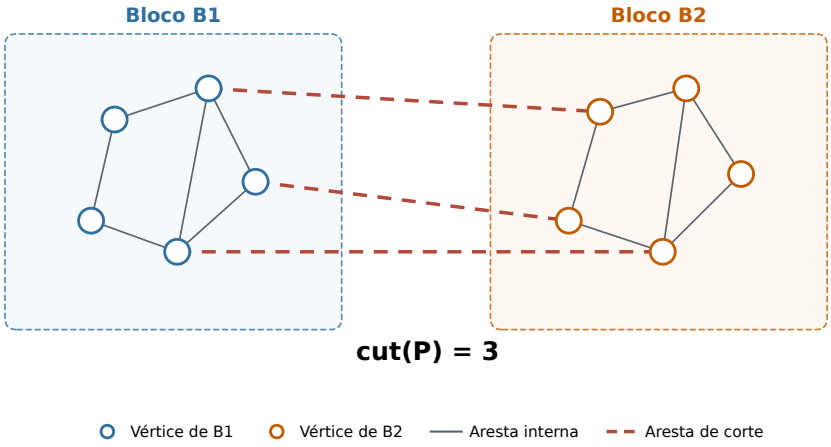
Para a partição $P = \{B_1, \ldots, B_k\}$, o corte conta as arestas cujos extremos pertencem a blocos diferentes. A formulação do capítulo 1 pode ser escrita como

$$\mathrm{cut}(P) = \sum_{\{u,v\} \in E} \mathbf{1}[\chi(u) \neq \chi(v)].$$

O estudo usa grafos não ponderados e minimiza essa medida (HENDRICKSON; KOLDA, 2000; KARYPIS; KUMAR, 1998). O corte funciona como aproximação do volume de comunicação em aplicações distribuídas. A medida não descreve, por si só, a topologia da rede,

o tamanho das mensagens ou o padrão de troca entre blocos, o que restringe sua interpretação. A figura 1 ilustra um particionamento didático em dois blocos. As arestas internas ligam vértices do mesmo bloco, enquanto as arestas de corte ligam vértices de blocos diferentes. Nesse exemplo, há três arestas de corte, de modo que $\mathrm{cut}(P) = 3$. O objetivo do particionamento é reduzir esse valor sem violar a restrição de balanceamento.

Figura 1 – Exemplo didático de particionamento e arestas de corte



Fonte: Autoria própria (2026).

2.2.2 Balanceamento como restrição

Para n vértices, k blocos e tolerância ε, o limite inteiro por bloco é

$$L(n,k,\varepsilon) = \left\lceil (1+\varepsilon)\frac{n}{k} \right\rceil .$$

Uma partição é admissível quando

$$\max_{1 \le j \le k} |B_j| \le L(n,k,\varepsilon).$$

O balanceamento atua como restrição, e a qualidade é comparada somente entre resultados aceitos pelos critérios experimentais de validação.

2.2.3 Tempo de parede e elegibilidade temporal

O tempo de parede mede externamente o intervalo da chamada e permite comparar participantes com instrumentação interna diferente (MCGEOCH, 2012). Neste trabalho, um resultado é temporalmente elegível quando está disponível dentro do orçamento da fatia analisada. A metodologia define como saídas pontuais e históricos instrumentados entram nessa regra.

2.2.4 Disponibilidade e estado operacional

A disponibilidade informa se existe resultado elegível no orçamento. O estado de término informa como a chamada terminou. Uma execução pode terminar por limite temporal e ainda possuir um resultado elegível registrado antes desse limite. Também pode terminar sem fornecer o participante exigido pela comparação. Uma ausência permanece sem imputação de corte.

2.2.5 NFE

O NFE contabiliza avaliações da função objetivo registradas durante a busca das meta-heurísticas instrumentadas. Como os particionadores externos não expõem contador equivalente, essa medida permanece diagnóstica (AUGER; BROCKHOFF; HANSEN, 2021).

2.3 Métodos de solução e seleção de algoritmos por instância

O portfólio reúne particionadores *multilevel* e meta-heurísticas. Cada família produz resultados por mecanismos diferentes. Os CARTs atuam depois do *benchmarking*, relacionando atributos e configurações aos rótulos obtidos.

2.3.1 O paradigma *multilevel*

Métodos *multilevel* contraem o grafo, constroem uma partição no nível reduzido e projetam essa solução durante a expansão, refinando-a em níveis progressivamente mais finos (KARYPIS; KUMAR, 1998; BULUC *et al.*, 2016). A contração reduz o número de decisões no nível inicial. A projeção devolve a solução ao grafo original, e o refinamento corrige fronteiras que se tornam visíveis nos níveis mais detalhados. O desempenho depende de quanto da estrutura associada ao corte é preservado durante esse percurso.

A contração, ou *coarsening*, agrupa vértices em supervértices para formar grafos menores. Cada supervértice representa um conjunto do grafo anterior, e arestas entre esses conjuntos são combinadas no nível reduzido. O *heavy-edge matching* (HEM), associado ao METIS, seleciona pares sem reutilizar vértices e prioriza arestas de maior peso quando elas existem (KARYPIS; KUMAR, 1998). Em grafos não ponderados, o procedimento busca preservar adjacências locais durante a redução. A propagação de rótulos com limite de tamanho, usada no KaHIP, forma agregados a partir da vizinhança e impede que um único grupo comprometa o balanceamento dos níveis seguintes (SANDERS; SCHULZ, 2013). Agregados maiores produzem menos níveis, mas podem ocultar separadores locais. Agregados menores preservam mais detalhe e prolongam a hierarquia.

No grafo mais contraído, uma heurística produz a partição inicial que será projetada aos níveis seguintes. Crescimento de regiões constrói blocos de forma incremental e verifica o

balanceamento durante a expansão de cada parte. Métodos espectrais usam o Laplaciano para identificar direções de separação global e, na bisseção, podem orientar a divisão pelo vetor de Fiedler (CHUNG, 1997). Diferentes sementes ou estratégias de inicialização podem levar a pontos de partida distintos. A solução inicial não encerra o processo, mas condiciona os ajustes disponíveis no refinamento.

Durante o *uncoarsening*, a partição é projetada e ajustada sem violar o balanceamento. Kernighan–Lin (KL) e Fiduccia–Mattheyses (FM) usam trocas ou movimentos de vértices entre blocos (KERNIGHAN; LIN, 1970; FIDUCCIA; MATTHEYSES, 1982). Estruturas de ganho atualizam o efeito de cada movimento sem recomputar todas as alternativas. O refinamento pode tratar pares de blocos ou permitir movimentos entre várias partes em uma formulação k-*way*. O KaHIP também combina reinícios locais *multi-try* e subproblemas de *max-flow/min-cut* próximos às fronteiras (SANDERS; SCHULZ, 2013). Esses subproblemas reorganizam conjuntos de vértices e ampliam o alcance em relação a movimentos individuais.

A bisseção recursiva divide partes sucessivamente até obter k blocos. O particionamento k-*way* trata os blocos de forma conjunta. A primeira estratégia herda decisões de cortes anteriores, enquanto a segunda permite ajustar simultaneamente fronteiras entre várias partes (KARYPIS; KUMAR, 1998).

Na bisseção recursiva, uma separação feita nos níveis iniciais restringe as subdivisões posteriores, pois cada novo bloco é formado dentro de uma parte já definida. A formulação k-*way* evita essa sequência de decisões binárias e permite considerar várias fronteiras durante o refinamento. Essa flexibilidade amplia as alternativas locais e aumenta a quantidade de blocos que podem receber um vértice em cada movimento.

O METIS implementa particionamento *multilevel* k-*way*, associa a contração ao HEM e usa refinamentos locais relacionados ao método FM (KARYPIS; KUMAR, 1998). Seu uso no portfólio fornece um particionador externo com saída pontual para comparação com as metaheurísticas.

O KaHIP combina contração por propagação de rótulos com limite de tamanho, busca local, refinamento *multi-try* e procedimentos baseados em fluxo (SANDERS; SCHULZ, 2013). Variantes como KaFFPa e KaFFPaE configuram esses mecanismos de formas distintas.

METIS e KaHIP seguem a mesma sequência geral de redução, partição inicial e refinamento, mas usam políticas diferentes em cada fase. O HEM do METIS forma agregados por emparelhamentos, enquanto a propagação de rótulos do KaHIP pode reunir grupos com mais de dois vértices. No refinamento, os procedimentos de fluxo do KaHIP reorganizam conjuntos próximos à fronteira, ao passo que movimentos locais tratam alterações mais restritas. Essas diferenças justificam avaliá-los separadamente no portfólio.

SCOTCH e PT-Scotch combinam particionamento, mapeamento e ordenação, com versões para execução distribuída (CHEVALIER; PELLEGRINI, 2008). ParMETIS aplica o paradigma a ambientes baseados em MPI e a aplicações com matrizes esparsas (KARYPIS; SCHLOEGEL; KUMAR, 2003). KaMinPar explora paralelismo em memória compartilhada (HEUER;

SANDERS; SCHLAG, 2021). Seus mecanismos de paralelismo não integram o recorte mono-*thread* da campanha.

Implementações *multilevel* podem combinar HEM, propagação de rótulos, múltiplas inicializações, movimentos FM ou KL e refinamentos por fluxo (BULUC *et al.*, 2016; KARYPIS; KUMAR, 1998; SANDERS; SCHULZ, 2013; CHUNG, 1997). Ciclos adicionais e múltiplas partidas ampliam o esforço, enquanto execuções distribuídas ou em memória compartilhada introduzem custos de sincronização e comunicação interna (MCGEOCH, 2012; CHEVALIER; PELLEGRINI, 2008; HEUER; SANDERS; SCHLAG, 2021; HENDRICKSON; KOLDA, 2000). Essas extensões permanecem fora do protocolo mono-*thread*.

As implementações diferem nas políticas de contração, refinamento e execução. O estudo avalia METIS e KaHIP nas configurações registradas, sem atribuir à família inteira o comportamento de uma ferramenta específica.

Malhas e grafos quase planares apresentam localidade, caminhos associados à geometria e, em muitos casos, menor variação de grau que redes complexas (KARYPIS; KUMAR, 1998; CHUNG, 1997). Separadores locais podem permanecer visíveis durante a contração e orientar o refinamento. Essa propriedade motiva sua presença em painéis heterogêneos, sem antecipar qual ferramenta produzirá o menor corte.

Redes de pequeno mundo ou livres de escala podem combinar aglomeração local, caminhos curtos, comunidades e vértices de alto grau. A contração precisa decidir se preserva esses concentradores ou os incorpora a agregados maiores. Propagação de rótulos, refinamentos locais e fluxo interagem com aspectos diferentes dessas estruturas (SANDERS; SCHULZ, 2013). A diversidade morfológica fornece condições para examinar associações entre atributos e desempenho sem assumir uma vantagem prévia de qualquer método.

A solução depende da política de contração, da tolerância de balanceamento e do refinamento. Estruturas eliminadas nos níveis contraídos podem não ser recuperadas durante a expansão, e procedimentos de maior alcance também consomem mais tempo.

2.3.2 Meta-heurísticas de propósito geral

Meta-heurísticas percorrem o espaço de atribuições de vértices sem garantir a solução ótima. SA, TS, ILS e GRASP representam mecanismos não populacionais baseados em aceitação probabilística, memória, perturbação e construção gulosa aleatorizada (KIRKPATRICK, 1984; GLOVER, 1989; LOURENÇO; MARTIN; STÜTZLE, 2018; FEO; RESENDE, 1995; BLUM; ROLI, 2003; GLOVER; KOCHENBERGER, 2003).

Uma vizinhança define as soluções alcançáveis por um movimento. No GPP, um movimento pode transferir um vértice entre blocos quando o balanceamento continua válido. O ganho mede a alteração do corte causada por essa transferência. Quando nenhum movimento admissível reduz a função objetivo, a solução é um ótimo local para a vizinhança adotada, não necessariamente para o problema inteiro. As meta-heurísticas diferem no mecanismo usado para deixar essa região, controlar retornos e escolher a próxima solução corrente.

O SA aceita movimentos de melhora e pode aceitar pioras com probabilidade dependente da temperatura. Temperaturas maiores permitem atravessar regiões de corte mais alto e reduzir a dependência do ótimo local corrente. O resfriamento diminui essa probabilidade e concentra a busca em melhorias locais. Temperatura inicial, taxa de resfriamento, temperatura mínima e quantidade de avaliações por nível determinam a duração de cada regime (KIRKPATRICK, 1984; BLUM; ROLI, 2003).

A TS restringe temporariamente movimentos recentes para reduzir ciclos (GLOVER, 1989). A lista tabu registra atributos dos movimentos e impede o retorno imediato a configurações anteriores. A duração tabu determina o período da restrição. Um critério de aspiração libera um movimento proibido quando ele melhora o melhor resultado já encontrado. Algumas variantes acrescentam frequência de uso para penalizar movimentos repetidos e favorecer outras regiões.

A ILS alterna busca local e perturbação (LOURENÇO; MARTIN; STÜTZLE, 2018). A busca local leva a solução a um ótimo da vizinhança. A perturbação modifica parte da atribuição para iniciar nova descida. Alterações pequenas tendem a retornar à mesma região, enquanto alterações maiores aproximam o passo de uma reinicialização. Uma regra de aceitação decide se a solução perturbada substitui a corrente e controla o equilíbrio entre intensificação e diversificação.

O GRASP repete uma construção gulosa aleatorizada seguida de busca local (FEO; RESENDE, 1995). Em cada etapa da construção, a lista de candidatos restrita (RCL, *restricted candidate list*) reúne alternativas que atendem ao limiar guloso. O parâmetro α controla a amplitude da lista conforme a convenção implementada. Listas estreitas aproximam a construção de uma decisão gulosa, enquanto listas amplas aumentam a variedade das soluções iniciais. A busca local refina cada construção antes da iteração seguinte.

SA, TS e ILS desenvolvem trajetórias a partir de soluções correntes, enquanto GRASP inicia novas construções. SA usa probabilidade para aceitar pioras, TS restringe movimentos por memória, ILS desloca a solução por perturbações e GRASP diversifica a etapa construtiva. Esses mecanismos podem produzir históricos de melhoria com ritmos diferentes, o que justifica observar a qualidade ao longo dos orçamentos em vez de apenas no término da maior execução.

Exploração e intensificação indicam, respectivamente, deslocamentos para novas regiões do espaço de soluções e concentração em torno de soluções já favoráveis. No SA, temperatura e resfriamento controlam essa alternância. Na TS, esse papel recai sobre a duração tabu e a aspiração. Na ILS, recai sobre a intensidade da perturbação. No GRASP, sobre o parâmetro da RCL. Esses parâmetros precisam ser fixados no experimento para que as diferenças observadas não resultem de ajustes específicos por instância.

O portfólio reúne quatro mecanismos de busca não populacional sob a mesma função de qualidade. Os históricos registram o melhor corte ao longo da execução e sustentam a reconstrução temporal, mas não contêm necessariamente uma partição materializada para cada instante.

2.3.3 Seleção de algoritmos orientada por aprendizado de máquina

Na formulação de Rice, a seleção relaciona características da instância ao algoritmo indicado pelo critério de avaliação (RICE, 1976). O problema exige um espaço de instâncias, um conjunto de atributos, um portfólio e uma medida de desempenho. No GPP, atributos topológicos, valor de k e orçamento descrevem cada observação. O rótulo indica o resultado sob o recorte e a política de agregação adotados. A relação aprendida permanece condicionada ao portfólio, aos rótulos e ao painel (KOTTHOFF, 2014).

Os preditores devem estar disponíveis antes da escolha do algoritmo e não podem incorporar cortes, vencedores ou outros resultados do próprio *benchmarking*. Número de vértices e arestas descrevem a escala. Densidade e estatísticas de graus descrevem a concentração relativa de conexões. Pseudo-diâmetro, componentes e vértices isolados descrevem conectividade global. O custo de extração deve ser considerado em uma avaliação operacional da política de seleção, pois o cálculo dos atributos pode consumir parte do tempo que a recomendação pretende economizar.

Cada observação combina um vetor de preditores $\mathbf{x}_i$ com um rótulo $y_i^\star$. O rótulo deriva da comparação na fatia correspondente e depende do recorte, dos participantes, da agregação e do desempate. Duas fatias do mesmo grafo podem receber rótulos diferentes quando mudam k ou orçamento, embora compartilhem os atributos topológicos. A metodologia separa algoritmo nominal, família nominal e condições baseadas na diferença relativa entre famílias.

O CART divide sucessivamente o espaço dos atributos e associa cada folha a uma classe (BREIMAN *et al.*, 1984). Cada divisão escolhe um atributo e um limiar ou categoria que reduz a impureza do nó. Os caminhos da raiz às folhas formam regras condicionais sobre densidade, pseudo-diâmetro, k, orçamento ou outros preditores. Profundidade e mínimo de observações por folha limitam a complexidade da árvore. A inspeção das regras depende desses controles e do procedimento usado para avaliar o modelo (MOLNAR, 2020; BOAS *et al.*, 2021).

Várias fatias podem derivar do mesmo grafo. Por exemplo, uma instância avaliada em cinco valores de k e sete orçamentos produz observações que compartilham os mesmos atributos topológicos. Se parte delas estiver no treino e parte no teste, o modelo recebe informações sobre o mesmo grafo nas duas etapas. A validação agrupada mantém todas as fatias de cada grafo no mesmo *fold*, de modo que a avaliação ocorra em instâncias retiradas do ajuste.

Campos derivados da execução permanecem fora dos preditores, pois sua inclusão revelaria ao modelo parte do resultado que ele deveria prever. As métricas obtidas desse modo estimam o desempenho interno ao painel. A validação agrupada reduz o vazamento entre configurações do mesmo grafo, mas não substitui um conjunto externo. A ausência de validação aninhada também permite que a escolha de hiperparâmetros influencie a estimativa principal.

A seleção interpretável examina associações entre propriedades dos grafos, configurações e rótulos. Ela não substitui o protocolo de comparação, pois os rótulos dependem das mesmas regras de qualidade, elegibilidade e agregação usadas no *benchmarking*. Os modelos de referência usam o painel conservador com METIS, KaHIP, SA e TS. ILS e GRASP apare-

cem na análise secundária do recorte C2, definido na metodologia e calculado sobre as fatias completas desse recorte. As regras produzidas descrevem o painel avaliado e não constituem recomendação geral para grafos externos.

2.4 Engenharia experimental para *benchmarking* em GPP

Comparações de GPP dependem do portfólio, das instâncias, das métricas e dos critérios de inclusão. A literatura recomenda variedade de instâncias, controle do ambiente e documentação das condições observadas (HOOKER, 1995; MCGEOCH, 2012; ZIEMANN; POULAIN; BORA, 2023). O painel precisa distinguir diversidade de origem de representatividade estatística. Um conjunto curado pode cobrir morfologias diferentes sem constituir amostra aleatória da população de grafos. A caracterização topológica permite examinar se diferenças de escala, densidade, graus e conectividade acompanham mudanças de desempenho.

2.4.1 Fontes de instâncias e tipos de estrutura dos grafos

Coleções reais e geradores sintéticos cumprem funções complementares. Walshaw reúne problemas de particionamento, em especial malhas e grafos quase planares. SNAP contém redes sociais, tecnológicas e de informação. SuiteSparse reúne matrizes cuja estrutura pode ser convertida em grafos (WALSHAW, 2023; LESKOVEC; KREVL, 2014; DAVIS; HU, 2011). Essas fontes aproximam estruturas observadas em aplicações, mas não controlam diretamente suas propriedades e podem concentrar instâncias em certos domínios.

Erdős–Rényi permite controlar a probabilidade de aresta. Watts–Strogatz combina aglomeração e caminhos curtos. Barabási–Albert produz distribuições de grau concentradas em poucos vértices. Grafos geométricos introduzem localidade espacial, e LFR controla comunidades e heterogeneidade de graus (ERDŐS; RÉNYI, 1959; WATTS; STROGATZ, 1998; BARABÁSI; ALBERT, 1999; PENROSE, 2003; LANCICHINETTI; FORTUNATO; RADICCHI, 2008). Um painel heterogêneo não representa todos os grafos possíveis, mas evita que as conclusões dependam de uma única morfologia.

2.4.2 Caracterização por atributos da instância

A caracterização registra propriedades calculadas antes da execução. Medidas de escala, densidade, graus e conectividade apoiam a interpretação dos resultados e formam os preditores dos modelos descritos no capítulo 3. Esses atributos não medem diretamente a dificuldade do particionamento. Eles fornecem descrições observáveis que podem ser relacionadas aos rótulos dentro do painel. A interpretação deve considerar correlações entre medidas, custo de extração e cobertura das diferentes morfologias.

2.5 Síntese e desafios metodológicos

A revisão justifica o uso do corte sob restrição de balanceamento, a comparação por elegibilidade temporal e a validação agrupada dos CARTs. Os atributos topológicos entram como descrições disponíveis antes da execução, usadas para examinar associações com os rótulos. O quadro 1 relaciona esses desafios às decisões do protocolo.

Quadro 1 – Desafios de avaliação e encaminhamentos metodológicos

Desafio metodológico	**Encaminhamento adotado**
Métricas de esforço incompatíveis entre famílias	Comparação por elegibilidade temporal em fatias definidas por grafo, valor de k e orçamento, usando tempo de parede como referência comum e NFE apenas como diagnóstico interno (AUGER; BROCKHOFF; HANSEN, 2021; MCGEOCH, 2012).
Instrumentação temporal assimétrica	Reconstrução dos cortes históricos das meta-heurísticas por *checkpoints* e uso das saídas pontuais de METIS e KaHIP apenas nas fatias em que são temporalmente elegíveis.
Heterogeneidade na validação das saídas	Aplicação de uma verificação independente comum à integridade, ao balanceamento e ao corte das partições materializadas, sem apresentar os cortes históricos como partições validadas em cada instante.
Agregação dos métodos estocásticos	Representação de cada algoritmo pelo menor corte temporalmente elegível entre as repetições disponíveis, com a mediana restrita a artefatos diagnósticos.
Painéis concentrados em poucas estruturas	Combinação de grafos reais e sintéticos, descritos por atributos de tamanho, densidade, distribuição de graus e conectividade.
Cobertura desigual dos participantes	Separação entre o painel conservador, usado nos CARTs de referência, o recorte C2 com os seis algoritmos e o painel operacional de disponibilidade.
Relação entre atributos e rótulos de desempenho	Construção de rótulos por fatia e treinamento de CARTs com atributos das instâncias e variáveis da configuração experimental conhecidos antes da execução (RICE, 1976; BREIMAN *et al.*, 1984).
Risco de vazamento entre configurações do mesmo grafo	Validação cruzada agrupada pelo identificador do grafo e exclusão de campos derivados do *benchmark* entre os preditores.
Documentação incompleta das condições experimentais	Registro das condições recuperadas do ambiente, das configurações, dos estados operacionais, dos resultados e das verificações necessárias para conferir o protocolo (ZIEMANN; POULAIN; BORA, 2023; MCGEOCH, 2012).

Fonte: Autoria própria (2026).

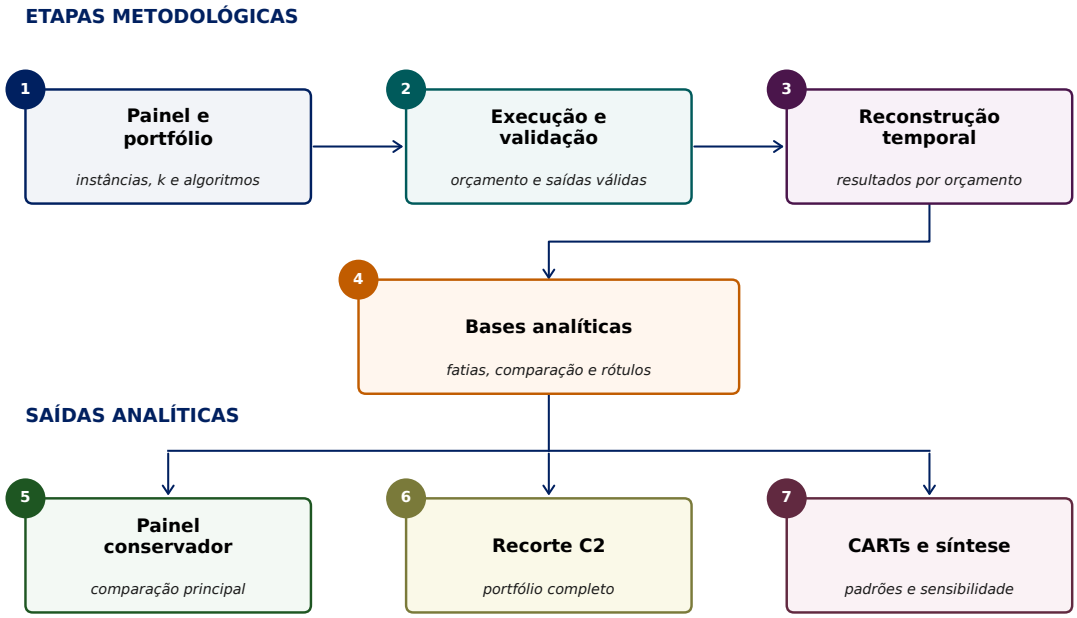
3 METODOLOGIA

O protocolo de *benchmarking* compara particionadores *multilevel* e meta-heurísticas e organiza as bases usadas na seleção por instância. Como os métodos expõem instrumentação diferente, a comparação usa tempo de parede como referência externa, mantendo a mesma definição do problema, a mesma tolerância de balanceamento, a mesma elegibilidade temporal e a mesma validação das partições materializadas. O tempo de parede é observado pelo *runner* e não torna equivalentes as operações internas dos algoritmos. Os recortes separam qualidade, análises complementares e disponibilidade operacional.

3.1 Visão geral do desenho experimental

As chamadas dos algoritmos, a validação das partições materializadas e a formação das observações foram tratadas em etapas separadas. Cada chamada gerou um registro operacional. Depois, esses registros foram organizados em fatias definidas por grafo, valor de k e orçamento de tempo, conforme a elegibilidade temporal e os critérios de cada recorte. As bases de qualidade e disponibilidade foram mantidas separadas. A figura 2 resume essa sequência.

Figura 2 – Etapas metodológicas e saídas analíticas



Fonte: Autoria própria (2026).

3.1.1 Unidade física de execução e unidade analítica

A unidade física corresponde a uma chamada individual de um algoritmo para uma combinação de grafo, valor de k e semente, quando aplicável. Cada chamada constitui um registro

operacional individual, independentemente do estado de término ou da existência de outras tentativas para a mesma combinação.

A unidade analítica é a fatia formada por grafo, valor de k e orçamento de tempo. Algoritmo e semente não definem a fatia. Os algoritmos são participantes dentro dela, e as sementes são repetições processadas para representar cada algoritmo. Uma mesma execução física pode fornecer resultados para várias fatias. Nas meta-heurísticas, a reconstrução temporal considera, em cada execução, o menor corte histórico instrumentado até o limite correspondente. Quando não havia *checkpoint* elegível, a saída final materializada podia representar a execução se tivesse sido aprovada pela validação e seu tempo não excedesse o orçamento. As repetições disponíveis eram posteriormente agregadas conforme a regra definida na subseção 3.5.2. Em METIS e KaHIP, a saída pontual aprovada é considerada nas fatias cujo orçamento é igual ou superior ao tempo observado da execução.

Uma partição materializada aprovada é uma saída cuja integridade, balanceamento e corte foram verificados independentemente. Um corte histórico instrumentado é um valor registrado em *checkpoint*, sem verificação independente da partição naquele instante. A expressão resultado temporal elegível abrange tanto o corte histórico admitido pela reconstrução temporal quanto a saída materializada aprovada cujo tempo não excedeu o orçamento. Para fins de comparação, a fatia só entra na análise de qualidade quando os participantes exigidos pelo recorte possuem resultado elegível.

A distinção entre execução física e fatia impede que cada orçamento seja interpretado como uma chamada independente. O registro operacional e a comparação de qualidade constituem níveis distintos da análise.

3.1.2 Recortes analíticos e critérios de inclusão

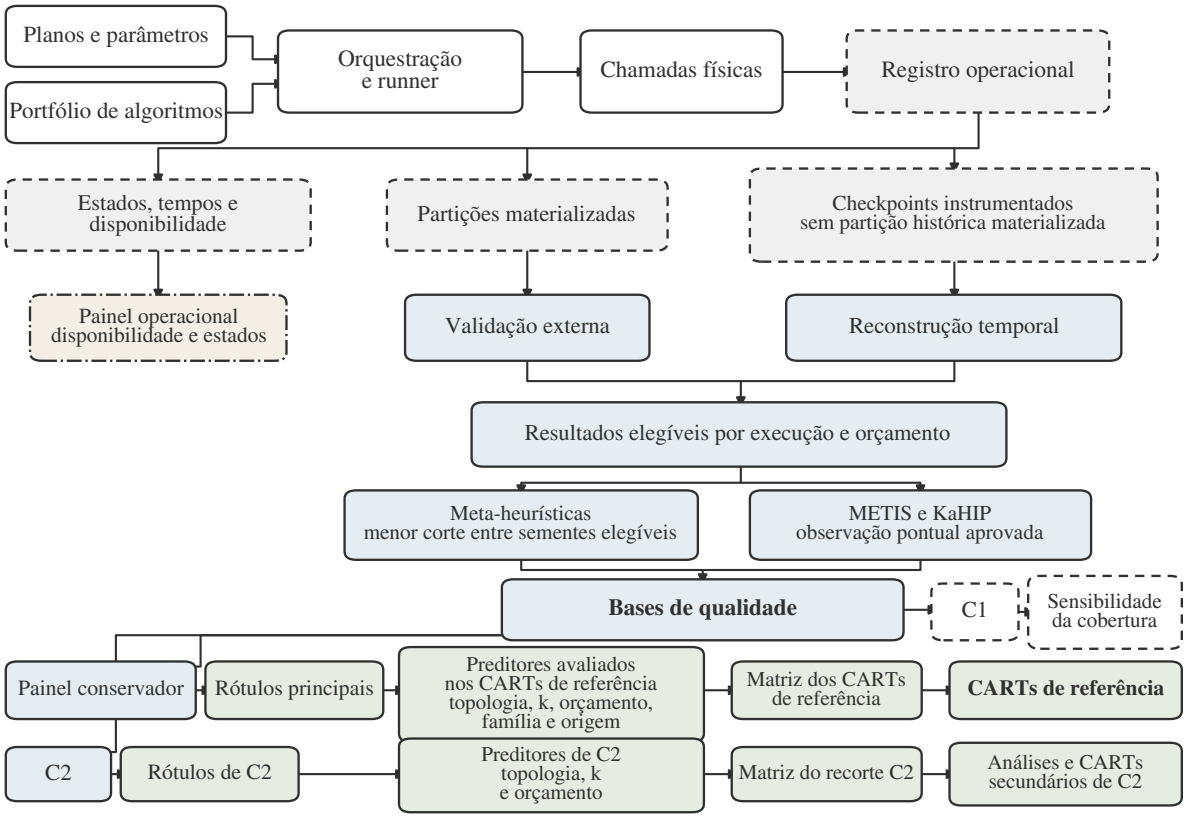
Os registros foram separados em recortes analíticos. O painel conservador reúne as fatias completas com METIS, KaHIP, SA e TS e foi usado nas comparações principais e nos CARTs de referência. O recorte C1 incorpora ILS e GRASP quando há ao menos uma partição aprovada desses métodos. O recorte C2 incorpora ILS e GRASP quando há cinco sementes aprovadas para cada um deles. O painel operacional reúne estados de término e disponibilidade, sem exigir que todos os participantes estejam disponíveis em todas as fatias.

Os critérios completos, as contagens e os denominadores são definidos na seção 3.6.

A figura 3 detalha o percurso entre as chamadas executadas e as bases analíticas. Registros operacionais alimentam estados e disponibilidade, partições materializadas seguem para validação independente, e *checkpoints* das meta-heurísticas entram apenas na reconstrução temporal. Na figura, a expressão validação externa deve ser lida como verificação independente das partições materializadas, não como validação externa dos modelos CART.

Depois da reconstrução e da agregação, os resultados elegíveis formam as bases de qualidade e seus rótulos. As matrizes de modelagem combinam esses rótulos com atributos

Figura 3 – Fluxo de execução, validação e construção das bases analíticas



Fonte: Autoria própria (2026).

topológicos, k e orçamento. O painel operacional mantém os casos incompletos sem imputar qualidade.

3.2 Ambiente de execução e validação experimental

As campanhas experimentais foram executadas no servidor com uma única *thread* por chamada, controle do paralelismo implícito das bibliotecas numéricas e medição do tempo de parede pelo *runner*. Essas condições foram mantidas durante a produção dos resultados, e os tempos reportados são interpretados no ambiente computacional em que foram medidos.

3.2.1 Ambientes computacionais

O WSL2 foi usado no desenvolvimento, na preparação de artefatos e nas verificações locais. As campanhas usadas nos resultados foram executadas no servidor, onde os tempos, registros e partições foram produzidos.

Quadro 2 – Ambientes computacionais usados no projeto

Ambiente	Papel no projeto	Configuração registrada
WSL2 local	Desenvolvimento, preparação de artefatos e verificações locais.	Linux em WSL2, *Rust*/*Cargo* 1.75.0 e KaHIP 3.17.
Servidor `srv-noctua`	Execução das campanhas experimentais usadas nos resultados.	Ubuntu 22.04.5, AMD Phenom II X4 B95 com 4 CPUs, 7,3 GiB de RAM, *Python* 3.10.12 e KaHIP 3.17.

Fonte: Autoria própria (2026).

O *runner* configurou uma única *thread* e, quando aplicável, definiu `OMP_NUM_THREADS=1`, `OPENBLAS_NUM_THREADS=1`, `MKL_NUM_THREADS=1` e `NUMEXPR_NUM_THREADS=1`. Não houve concorrência deliberada entre algoritmos. Os registros não confirmam exclusividade administrativa da máquina durante todo o período, o que limita a interpretação dos tempos de parede.

Os registros não permitiram confirmar a versão de METIS/`gpmetis`, o compilador *Rust* dos binários do servidor nem a versão de *scikit-learn* dos modelos históricos. O retreinamento focalizado do CART de vitória estrita ocorreu em ambiente *Poetry* local com *Python* 3.11.15 e *scikit-learn* 1.8.0. Essas versões não são atribuídas aos modelos históricos.

3.2.2 Pilha de ferramentas

O fluxo experimental combinou componentes em *Python*, binários em *Rust* e executáveis externos. Os componentes em *Python* coordenaram os planos, acionaram o *runner*, validaram as partições e consolidaram os resultados. As meta-heurísticas foram compiladas e executadas em *Rust*. METIS e KaHIP foram integrados ao mesmo fluxo por meio de adaptadores.

O repositório registra dependências do desenvolvimento e contém um *Dockerfile*. Os registros da campanha final não atribuem as execuções a esse contêiner.

Quadro 3 – Ferramentas usadas no desenvolvimento e na execução experimental

Ferramenta	Papel no projeto	Uso no protocolo
Python	Orquestração, validação, consolidação e análise.	Leitura dos planos, acionamento do *runner*, verificação das partições e construção dos artefatos analíticos.
Poetry	Gerenciamento do ambiente *Python*.	Instalação e registro das dependências usadas no desenvolvimento local.
Rust e *Cargo*	Implementação e compilação das meta-heurísticas.	Produção dos binários de SA, TS, ILS e GRASP usados nas campanhas.
METIS	Particionador *multilevel* externo.	Execução por adaptador que aciona `gpmetis` e recolhe a partição produzida.
KaHIP	Particionador *multilevel* externo.	Execução por adaptador que aciona `kaffpa` e recolhe a partição produzida.
Git e `gh`	Controle de versão e apoio às operações no repositório remoto.	Registro das alterações do código e organização das etapas de desenvolvimento e verificação.
Dockerfile	Especificação versionada de um ambiente para execução em contêiner.	Registro das dependências declaradas nesse ambiente, sem atribuição da campanha final à execução em contêiner.

Fonte: Autoria própria (2026).

3.2.3 Planos, formatos e metadados

Os planos fixaram grafos, algoritmos, valores de k, sementes, orçamentos e identificadores antes das chamadas. Arquivos `CSV` apoiaram a inspeção e a contagem, enquanto arquivos `JSON` registraram orçamentos e parâmetros consumidos pelo *runner*. Cada chamada produziu identificador, algoritmo, grafo, k, semente, tempo observado, estado e caminhos das saídas.

Os resultados foram organizados em fatias somente após a validação das partições e a reconstrução temporal. Manifestos e relatórios distinguiram as tabelas aceitas dos registros intermediários. Caminhos completos e nomes locais permanecem no material de reprodutibilidade.

3.2.4 Orquestração e fluxo de execução

A orquestração percorreu as chamadas dos planos e acionou o *runner*. Cada execução física foi identificada por grafo, algoritmo, k, semente e identificador da chamada. As fatias de orçamento foram construídas posteriormente a partir dos registros produzidos.

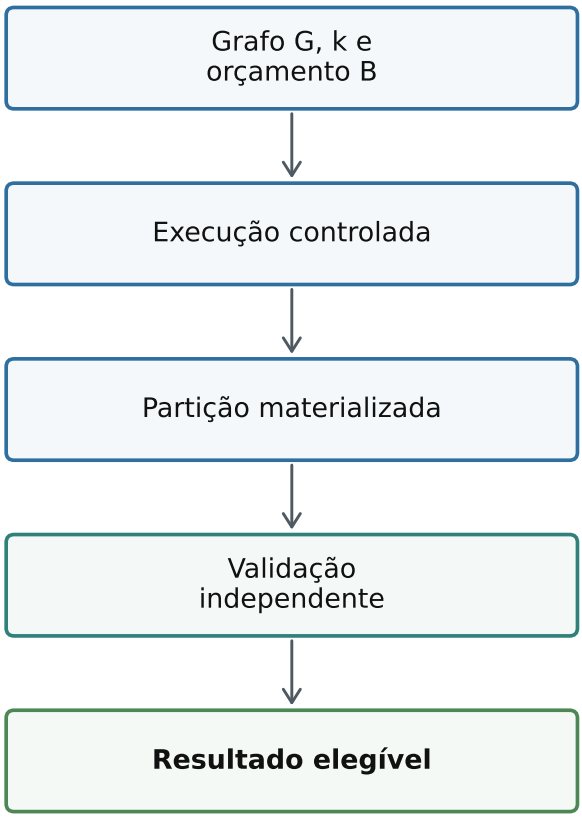
O *runner* preparou entradas e parâmetros, acionou os binários em *Rust* ou os adaptadores de `gpmetis` e `kaffpa` e recolheu as saídas. Ao término, registrou estado, tempo

de parede, arquivos e, nas meta-heurísticas, *checkpoints*. As partições materializadas foram associadas à validação, e os demais registros seguiram para reconstrução temporal e para os recortes correspondentes.

3.2.5 Validação das partições

A validação incide sobre as partições materializadas produzidas pelos algoritmos. O procedimento verifica a integridade da partição, o atendimento ao balanceamento e a consistência do corte reportado. A verificação confere a atribuição de todos os vértices, o intervalo dos rótulos de bloco, a compatibilidade com o valor de k, o atendimento ao balanceamento e a coincidência entre o corte reportado e o corte recomputado a partir do grafo e da partição. A figura 4 resume o caminho entre execução e resultado elegível.

Figura 4 – Fluxo de validação até o resultado elegível



Fonte: Autoria própria (2026).

A validação separa o estado operacional da aceitação da partição. O alcance da aprovação e a distinção em relação aos cortes históricos seguem as subseções 3.1.1 e 3.5.1.

Configurações com $k > n$ foram identificadas antes da execução ou da comparação. Erros, violações de balanceamento e saídas inconsistentes foram tratados antes dos rótulos. Reexecuções pontuais de METIS ou KaHIP só substituíram um registro depois de satisfazer os mesmos critérios de validação.

3.2.6 Disponibilidade, *timeout* e estados operacionais

O estado de término informa como a chamada terminou. A disponibilidade informa se existe resultado admitido dentro do orçamento. Uma fatia entra na qualidade somente quando contém os participantes exigidos pelo recorte. Caso contrário, o registro permanece no painel operacional sem corte imputado nem vencedor atribuído.

3.3 Painel de instâncias e configurações experimentais

O painel reúne grafos reais e sintéticos avaliados nos valores de k e orçamentos definidos. Sua composição delimita o domínio empírico das conclusões.

3.3.1 Composição do painel

O painel final reuniu 60 grafos, sendo 24 reais e 36 sintéticos. Os grafos reais foram obtidos de cinco grupos de fontes e incluem redes de coautoria, malhas viárias e grafos derivados de matrizes esparsas e de coleções públicas. Os grafos sintéticos foram distribuídos entre as famílias morfológicas F01–F08 e gerados segundo configurações predefinidas.

Quadro 4 – Composição do painel de grafos

Fonte ou grupo	Tipo	Quantidade	Descrição
Metadados arXiv (CC0)	Real	7	Redes de coautoria derivadas de metadados com condições documentadas de uso.
TIGER/Line	Real	4	Grafos viários derivados de dados geográficos oficiais.
SuiteSparse	Real	11	Grafos construídos a partir da estrutura de matrizes esparsas selecionadas para o painel.
OGB	Real	1	Instância proveniente de uma coleção pública de grafos.
OpenStreetMap (OSM) via Geofabrik	Real	1	Grafo viário preparado a partir de dados geográficos abertos.
Famílias F01–F08	Sintético	36	Instâncias geradas segundo famílias morfológicas e configurações predefinidas.

Fonte: Autoria própria (2026).

O painel é curado. As conclusões se limitam às 60 instâncias, às transformações aplicadas e às configurações experimentais.

3.3.2 Geração dos grafos sintéticos

O gerador sintético produziu oito famílias que variam em estrutura comunitária, gargalos, concentração de grau, quantidade relativa de arestas e condições ligadas ao balanceamento.

Foram produzidos 120 candidatos com família, semente, configuração, métricas estruturais e registros de integridade. O painel reteve 36.

Quadro 5 – Famílias usadas na geração dos grafos sintéticos

Família	Ênfase morfológica	Candidatos	Selecionados
F01	Variação modular com perturbação da estrutura comunitária.	7	3
F02	Módulos densos organizados em cadeia ou anel.	15	3
F03	Estruturas do tipo *barbell* com gargalo entre regiões densas.	7	4
F04	Estruturas com vértices concentradores e distribuição de graus assimétrica.	23	6
F05	Núcleo denso combinado com extensões de estrutura arbórea.	15	2
F06	Grafos esparsos com organização próxima à de malhas viárias.	7	4
F07	Grafos densos com menor contraste entre comunidades.	23	6
F08	Estruturas plantadas com propriedades associadas à dificuldade de balanceamento.	23	8

Fonte: Autoria própria (2026).

A seleção preservou as oito famílias e não usou resultados dos algoritmos. Como os registros não definem uma regra determinística ou probabilística para a passagem de 120 candidatos a 36 instâncias, o conjunto é tratado como painel curado.

Configurações, sementes, métricas e verificações dos 36 grafos constam do material de reprodutibilidade.

3.3.3 Extração e curadoria dos grafos reais

A curadoria selecionou fontes com origem identificada, condições de uso documentadas e processo de preparação recuperável. Dados sem condição de redistribuição confirmada não foram incluídos nos artefatos públicos.

As sete redes de coautoria foram reconstruídas a partir de metadados descritivos do arXiv publicados entre 2020 e 2024. Os termos das interfaces consultadas permitem o uso desses metadados sob CC0 1.0, sem estender essa condição aos textos integrais dos trabalhos (arXiv, s.d.b; arXiv, s.d.a).

Os quatro grafos do TIGER/Line foram obtidos dos arquivos de estradas de 2025 para os condados indicados no quadro 6. A origem e a versão dos arquivos foram registradas com base na documentação e nos pacotes oficiais do U.S. *Census Bureau* (U.S. Census Bureau,

Quadro 6 – Fontes e instâncias reais do painel

Fonte	Instâncias	Total
arXiv	astro-ph, cond-mat, cs, hep-th, math, quant-ph e stat	7
TIGER/Line	Los Angeles County, Kings County, New York County e Dallas County	4
OSM e Geofabrik	Rhode Island	1
OGB	ogbn-arxiv	1
SuiteSparse Matrix Collection	AG-Monien/3elt, DIMACS10/data, DIMACS10/delaunay_n10, DIMACS10/delaunay_n11, DIMACS10/delaunay_n12, DIMACS10/delaunay_n13, DIMACS10/fe_4elt2, DIMACS10/wing_nodal, Hamm/add32, HB/bcsstk33 e Nasa/nasa1824	11

Fonte: Autoria própria (2026).

2025e; U.S. Census Bureau, 2025c; U.S. Census Bureau, 2025b; U.S. Census Bureau, 2025d; U.S. Census Bureau, 2025a). O grafo de *Rhode Island* foi preparado a partir de dados do OSM distribuídos pela Geofabrik. Esses dados estão sujeitos à *Open Database License* (ODbL), que estabelece condições de atribuição e redistribuição de bases derivadas (OpenStreetMap contributors, s.d.; Open Data Commons, s.d.b; Geofabrik GmbH, s.d.).

A instância ogbn-arxiv representa uma rede direcionada de citações entre artigos de ciência da computação. Para o experimento, foi construída uma projeção simples e não direcionada de sua maior componente conexa. A coleção e a instância são documentadas por Hu et al. e pela página oficial da OGB, que registra a licença ODC-BY (HU *et al.*, 2020; Open Graph Benchmark, s.d.; Open Data Commons, s.d.a).

Os onze grafos derivados da *SuiteSparse Matrix Collection* seguiram as condições de atribuição, preservação dos metadados e registro das transformações estabelecidas pela coleção (DAVIS; HU, 2011; KOLODZIEJ *et al.*, 2019; SuiteSparse Matrix Collection, s.d.a).[1] As quatro triangulações de Delaunay também têm como referência o trabalho de Holtgrewe, Sanders e Schulz (HOLTGREWE; SANDERS; SCHULZ, 2010).

A preparação converteu as fontes para grafos simples, não direcionados e não ponderados. Quando necessário, foram descartados pesos e tipos de aresta, convertidas as relações direcionadas, removidos laços e duplicatas, selecionada a maior componente conexa e remapeados os vértices.

O material de reprodutibilidade relaciona as 24 instâncias reais, suas fontes, condições de uso, transformações e registros de integridade.

[1] As páginas específicas das onze matrizes estão registradas em (SuiteSparse Matrix Collection, s.d.b; SuiteSparse Matrix Collection, s.d.c; SuiteSparse Matrix Collection, s.d.d; SuiteSparse Matrix Collection, s.d.e; SuiteSparse Matrix Collection, s.d.f; SuiteSparse Matrix Collection, s.d.g; SuiteSparse Matrix Collection, s.d.h; SuiteSparse Matrix Collection, s.d.i; SuiteSparse Matrix Collection, s.d.j; SuiteSparse Matrix Collection, s.d.k; SuiteSparse Matrix Collection, s.d.l).

3.3.4 Caracterização topológica das instâncias

Os atributos topológicos foram calculados após a geração e a curadoria dos grafos, antes da incorporação dos resultados dos algoritmos. Foram registrados o número de vértices, o número de arestas, a densidade, o pseudo-diâmetro e as medidas de conectividade.

A distribuição de graus foi caracterizada pela média, pelo valor máximo, pelo desvio-padrão, pelo coeficiente de variação, pelos quantis de 50%, 90%, 95% e 99% e pelo coeficiente de Gini. A conectividade foi descrita pelo número de componentes, pelo tamanho e pela fração da maior componente e pelo número e pela fração de vértices isolados. Esse conjunto representa diferenças de escala, densidade, heterogeneidade dos graus e conectividade entre as instâncias.

Nos CARTs, os atributos topológicos numéricos foram combinados com o valor de k e o orçamento. Algumas variantes também incluíram a família morfológica e a coleção de origem. A seleção dos preditores foi realizada separadamente para cada alvo. O CART de vitória estrita de meta-heurística selecionou o conjunto numérico e categórico completo. Os CARTs de família nominal selecionada, vantagem meta-heurística acima de 1% e competitividade na margem de 5% selecionaram a variante numérica formada pelos atributos topológicos, pelo valor de k e pelo orçamento.

3.3.5 Configurações de particionamento e orçamentos

Cada grafo foi avaliado para os valores de k iguais a 2, 4, 8, 16 e 32. Das 300 combinações possíveis entre 60 grafos e cinco valores de k, duas foram excluídas por apresentarem $k > n$, restando 298 combinações elegíveis.

A tolerância matemática de balanceamento foi fixada em $\varepsilon = 0{,}03$. Para um grafo com n vértices e uma partição $P = \{B_1, \ldots, B_k\}$, o limite inteiro de tamanho por bloco foi calculado por

$$L(n,k,\varepsilon) = \left\lceil (1+\varepsilon)\frac{n}{k} \right\rceil .$$

Uma partição satisfazia a restrição de balanceamento quando

$$\max_{1 \le j \le k} |B_j| \le L(n,k,\varepsilon).$$

O teto é aplicado depois da tolerância. Partições só entram na qualidade quando satisfazem esse limite e os critérios de validação. O campo operacional `beta` representa a mesma tolerância denotada por ε.

A grade temporal reuniu os orçamentos de tempo de 1 s, 5 s, 15 s, 30 s, 60 s, 120 s e 300 s. Nos artefatos da campanha, esses valores foram registrados em milissegundos. A avaliação das 298 combinações nos sete orçamentos formou um universo de 2.086 fatias candidatas.

O ponto de 300 s é apenas o maior orçamento avaliado. O orçamento também integra os preditores dos CARTs.

3.4 Portfólio de algoritmos avaliado

O portfólio reúne METIS e KaHIP como executáveis *multilevel* e SA, TS, ILS e GRASP como meta-heurísticas em *Rust*.

METIS, KaHIP, SA e TS formam o painel conservador. ILS e GRASP integram as análises com seis algoritmos quando a cobertura requerida está presente. Os critérios de inclusão dos recortes são definidos na subseção 3.6.2.

3.4.1 METIS e KaHIP

METIS e KaHIP foram usados como particionadores *multilevel* do portfólio (KARYPIS; KUMAR, 1998; SANDERS; SCHULZ, 2013). Cada chamada produziu uma partição pontual, submetida aos critérios de validação.

METIS foi acionado pelo executável `gpmetis`, enquanto KaHIP foi acionado pelo executável `kaffpa`. A integração ocorreu por adaptadores que prepararam o formato de entrada, transmitiram o valor de k e os parâmetros exigidos por cada ferramenta, executaram o comando e recolheram a partição produzida. A verificação da integridade da partição, do balanceamento e do corte foi executada fora dos adaptadores.

Como não registraram uma trajetória comparável aos *checkpoints*, cada saída aprovada tornou-se elegível nos orçamentos iguais ou superiores ao tempo observado. Não foram reconstruídas soluções intermediárias.

3.4.2 Infraestrutura em *Python* e meta-heurísticas em *Rust*

A infraestrutura em *Python* leu os planos, preparou as chamadas, acionou o *runner*, validou as saídas e consolidou os registros. SA, TS, ILS e GRASP foram compiladas em *Rust*, com a orquestração separada da lógica de busca.

Versões anteriores em *Python* apoiaram o desenvolvimento, e a versão de TS também verificou invariantes da implementação em *Rust*. Essas versões não integraram as bases finais. Os tempos descrevem conjuntamente implementação, configuração e ambiente, pois o desenho não isolou o efeito da linguagem ou da maturidade de cada código.

3.4.3 Estrutura comum das meta-heurísticas

As quatro meta-heurísticas em *Rust* representaram cada partição por um vetor que associa vértices a blocos e mantiveram explicitamente o tamanho de cada bloco. SA, ILS e GRASP

usaram um núcleo compartilhado. TS adotou estruturas e rotinas locais com a mesma representação lógica e as mesmas regras de cálculo do corte e balanceamento.

A função objetivo corresponde ao número de arestas cujas extremidades pertencem a blocos diferentes. As quatro implementações dispunham de rotinas para recomputar o corte e calcular incrementalmente a variação produzida pela transferência de um vértice entre blocos. Antes da aplicação de cada movimento, sua compatibilidade com a restrição de balanceamento era verificada.

Cada método manteve o menor corte encontrado durante a execução e a partição correspondente. A representação lógica do problema, a função objetivo e a regra de balanceamento foram comuns, enquanto a inicialização, a geração dos candidatos, os mecanismos de exploração e os critérios internos variaram entre os algoritmos.

O alcance dos *checkpoints* e da validação das saídas segue as definições das subseções 3.1.1 e 3.5.1. Os registros históricos não armazenaram a partição correspondente a cada corte, enquanto as partições materializadas ao final das chamadas foram verificadas pela infraestrutura experimental.

3.4.4 Funcionamento das meta-heurísticas

SA manteve uma solução corrente e avaliou transferências de vértices entre blocos. Movimentos que reduziam o corte eram aceitos, enquanto movimentos de piora podiam ser aceitos com probabilidade determinada pela temperatura. A redução progressiva da temperatura diminuiu essa probabilidade ao longo da execução.

TS avaliou movimentos admissíveis a partir da solução corrente e usou uma memória temporária para restringir movimentos recentes. Um movimento classificado como tabu podia ser aceito quando satisfazia o critério de aspiração por melhorar o menor corte encontrado. A avaliação dos candidatos também incorporou uma penalidade de frequência para reduzir a repetição de movimentos.

ILS alternou busca local e perturbação. A busca local examinou transferências factíveis de vértices e aplicou movimentos de melhora segundo o critério definido pela implementação. Após o encerramento dessa etapa, uma sequência de perturbações modificou a solução, que serviu como ponto de partida para uma nova busca local.

Cada iteração do GRASP combinou uma construção gulosa aleatorizada com busca local. Durante a construção, os candidatos foram selecionados de uma lista restrita definida com auxílio do parâmetro α. A solução construída foi então submetida à busca local antes da iteração seguinte.

Nos quatro métodos, movimentos incompatíveis com a restrição de balanceamento foram descartados. As execuções foram encerradas pelo orçamento de tempo ou pelos critérios internos definidos para cada implementação. Os valores usados nesses controles são apresentados na subseção 3.4.6.

3.4.5 Cobertura e disponibilidade operacional

Neste trabalho, cobertura indica a presença das execuções aprovadas exigidas no nível da combinação de grafo e k. Disponibilidade indica se há resultado elegível dentro de um orçamento específico. Completude indica se todos os participantes requeridos pelo recorte estão disponíveis na fatia de grafo, k e orçamento. Na campanha ampliada usada para examinar os seis algoritmos, SA e TS apresentaram cinco sementes com partições materializadas aprovadas nas 298 combinações elegíveis de grafo e valor de k. METIS apresentou partição materializada aprovada em 297 combinações, enquanto KaHIP apresentou partição materializada aprovada em 296.

ILS apresentou cinco sementes com partições materializadas aprovadas em 279 das 298 combinações. GRASP atingiu essa cobertura em 290 combinações. Os dois métodos apresentaram simultaneamente cobertura $5/5$ em 279 combinações. Uma dessas combinações não possuía partição materializada aprovada de METIS, enquanto duas não possuíam partição materializada aprovada de KaHIP. Por isso, o recorte C2 reteve 276 combinações com cobertura completa dos seis algoritmos no nível da combinação.

A cobertura no nível da combinação não assegurou disponibilidade em todos os orçamentos. Uma combinação podia apresentar execuções aprovadas sem possuir corte elegível em todos os pontos da grade temporal. A completude da comparação entre os seis algoritmos foi verificada novamente em cada fatia definida por grafo, valor de k e orçamento.

Quando faltava corte elegível de um participante exigido no orçamento considerado, a fatia era incompleta para a comparação de qualidade entre os seis algoritmos. O registro integrava o painel operacional, sem imputação de corte ou atribuição de vencedor. A cobertura por combinação registra a presença dos resultados exigidos de cada participante, enquanto a disponibilidade por orçamento indica se esses resultados estavam disponíveis no limite temporal analisado.

3.4.6 Parâmetros, sementes e configuração

Os hiperparâmetros foram definidos antes da campanha final a partir de testes preliminares de sanidade operacional. Esses testes verificaram a executabilidade das meta-heurísticas, a estabilidade das chamadas, a produção de registros compatíveis com a reconstrução temporal e a cobertura das combinações planejadas. Após essa etapa, os valores foram mantidos fixos para todos os grafos, valores de k e orçamentos. Com isso, cada meta-heurística foi avaliada sob uma configuração única e rastreável, preservando a comparabilidade entre instâncias e evitando ajustes específicos por caso.

Os valores adotados são apresentados no quadro 7. Eles representam a configuração experimental usada na campanha e delimitam o escopo dos resultados obtidos para SA, TS, ILS e GRASP.

Quadro 7 – Hiperparâmetros das meta-heurísticas

Algoritmo	Configuração usada na campanha
SA	Temperatura inicial $1{,}0$, fator de resfriamento $0{,}997$, temperatura mínima $0{,}001$, limite de $100\,000$ passos e registro de *checkpoint* a cada 100 avaliações da função objetivo.
TS	Limite de $10\,000$ passos, permanência tabu mínima 7, fator de escala $1{,}0$, variação aleatória da permanência 4, penalidade de frequência $0{,}01$ e registro de *checkpoint* a cada 100 avaliações da função objetivo.
ILS	Limite de 100 iterações, perturbação composta por 4 movimentos e registro de *checkpoint* a cada iteração.
GRASP	Parâmetro $\alpha = 0{,}30$, limite de 100 iterações e registro de *checkpoint* a cada iteração.

Fonte: Autoria própria (2026).

Cada chamada terminou ao atingir o limite interno do método ou o limite externo de tempo, o que ocorresse primeiro.

Nos adaptadores, o valor $\varepsilon = 0{,}03$ foi convertido para as opções específicas das interfaces externas. METIS foi acionado com `-ufactor=30`, em unidades de milésimos, enquanto KaHIP recebeu `-imbalance=3.000`, em percentual. Independentemente dessa conversão, as partições produzidas somente foram aceitas nas análises quando satisfizeram, na verificação independente, o limite $L(n,k,\varepsilon)$ definido na subseção 3.3.5.

3.4.7 Custo computacional e comportamento operacional

O custo das meta-heurísticas depende dos movimentos avaliados, da vizinhança e das iterações. O cálculo incremental evita recomputar todo o corte para cada candidato. SA e TS percorrem movimentos sucessivos, ILS alterna busca local e perturbação, e GRASP repete construção e busca local.

Esses mecanismos e os limites do quadro 7 determinam quando a saída se torna disponível. Os critérios internos coexistem com o limite externo de tempo. Como METIS e KaHIP não expõem instrumentação equivalente, o tempo de parede medido externamente é a referência comum (KARYPIS; KUMAR, 1998; SANDERS; SCHULZ, 2013).

3.5 Protocolo de comparação por tempo de parede

A comparação foi organizada por tempo de parede. Em cada fatia analítica, os participantes foram comparados sob o mesmo orçamento de tempo, a mesma definição do problema, a mesma tolerância de balanceamento e os critérios de inclusão definidos para o recorte correspondente. Essa equivalência é definida no nível da fatia e não pressupõe igualdade no número de iterações, movimentos avaliados ou operações internas realizadas por cada método (HOOKER, 1995; MCGEOCH, 2012).

O tempo de parede foi adotado como referência comum porque os participantes apresentam níveis diferentes de instrumentação. Ele corresponde ao tempo externo decorrido du-

rante a execução, medido pelo *runner*. METIS, KaHIP e as meta-heurísticas não expõem necessariamente o mesmo contador interno de esforço. O número de avaliações da função objetivo e o número de iterações foram usados para descrever a busca das meta-heurísticas, mas não estavam disponíveis de forma equivalente para os seis algoritmos (AUGER; BROCKHOFF; HANSEN, 2021).

3.5.1 Orçamentos e reconstrução temporal

As execuções físicas foram posteriormente associadas às fatias definidas pelos sete limites temporais da grade experimental. Uma mesma chamada podia fornecer registros para mais de uma fatia, sem que o algoritmo fosse executado novamente para cada ponto da grade temporal. Os limites operacionais das chamadas físicas não foram tratados como necessariamente idênticos entre todas as implementações. A comparabilidade temporal decorre da elegibilidade dos resultados sob o orçamento considerado em cada fatia.

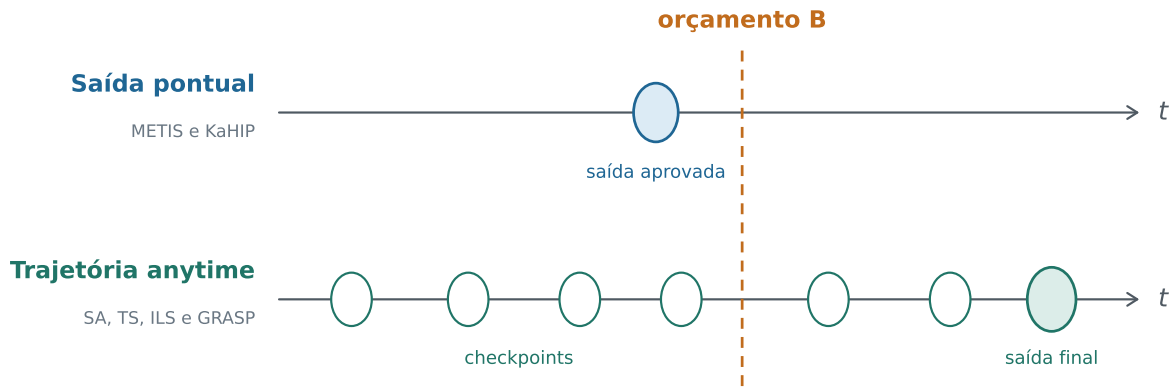
Nas meta-heurísticas, os *checkpoints* registraram o tempo decorrido, o menor corte encontrado e o número de avaliações da função objetivo. Para cada semente e orçamento, foi selecionado o menor corte registrado em um *checkpoint* cujo tempo não excedia o limite correspondente. Quando não havia *checkpoint* elegível, a saída final podia ser usada se tivesse sido aprovada pela validação e seu tempo coubesse no orçamento.

Os *checkpoints* foram tratados como registros instrumentados da trajetória de busca e não armazenavam a partição associada a cada corte histórico. A validação independente incidiu sobre as partições materializadas nas saídas das execuções.

METIS e KaHIP forneceram uma única observação quando a chamada terminou com uma partição aprovada. Essa observação foi associada às fatias cujo orçamento era igual ou superior ao tempo de parede medido. Não foram reconstruídas soluções intermediárias para esses métodos.

A figura 5 resume a relação entre a execução física e as fatias de orçamento. Nas meta-heurísticas, os registros históricos permitem selecionar, para cada limite temporal, o menor corte elegível disponível até aquele instante. Em METIS e KaHIP, a única saída aprovada da chamada torna-se elegível somente nos orçamentos iguais ou superiores ao tempo de parede observado.

Figura 5 – Relação entre execução física e reconstrução das fatias temporais



Fonte: Autoria própria (2026).

Os limites temporais de cada fatia definem quais registros estavam disponíveis em cada ponto da grade, sem gerar novas execuções físicas.

3.5.2 Repetições e agregação por algoritmo

Os métodos estocásticos foram planejados com cinco sementes por combinação de grafo e valor de k. Cada semente correspondeu a uma execução física distinta sob o mesmo perfil de hiperparâmetros.

A agregação ocorreu em duas etapas. Primeiro, para cada semente, foi escolhido o resultado temporal elegível no orçamento analisado. Em seguida, quando mais de uma semente do mesmo algoritmo estava disponível na fatia, o algoritmo foi representado pelo menor corte observado entre essas execuções. No painel conservador, empates de corte foram resolvidos pelo menor tempo observado e, depois, pela menor semente.

A mediana dos cortes foi calculada em artefatos diagnósticos, mas não definiu os vencedores, os rótulos de qualidade ou os alvos usados nos CARTs de referência.

A igualdade temporal foi aplicada às observações elegíveis dentro de cada orçamento. Como as meta-heurísticas foram repetidas e agregadas pelo menor corte, sua observação representa o melhor desempenho encontrado entre até cinco execuções com sementes distintas. METIS e KaHIP foram representados por uma execução pontual em cada combinação. A política compara resultados elegíveis por fatia, mas o esforço acumulado difere entre algoritmos e a agregação pelo menor corte não estima o desempenho esperado de uma execução escolhida ao acaso.

Os vencedores, os rótulos e as previsões dos CARTs são interpretados segundo essa política experimental de repetição e agregação.

A listagem 1 formaliza a reconstrução e a agregação.

Listagem 1 – Reconstrução temporal e agregação por algoritmo

```
Receba o algoritmo, o orçamento, os registros disponíveis e as sementes aplicáveis
Se o algoritmo for METIS ou KaHIP
  Se existir saída pontual aprovada com tempo observado não superior ao orçamento
    Retorne a saída pontual aprovada
  Retorne indisponibilidade
Considere vazio o conjunto de representantes por semente
Para cada semente disponível
  Reúna os checkpoints instrumentados cujo tempo não excede o orçamento
  Se houver ao menos um checkpoint elegível
    Selecione o menor corte histórico instrumentado da semente
  Caso contrário
    Considere a saída final somente se ela estiver aprovada e couber no orçamento
  Quando houver resultado temporal elegível, inclua seu representante no conjunto
Se o conjunto de representantes estiver vazio
  Retorne indisponibilidade
Selecione o menor corte entre os representantes disponíveis
No painel conservador, resolva empates pelo menor tempo observado e depois pela menor semente
Retorne o resultado agregado elegível para o algoritmo na fatia
```

Fonte: Autoria própria (2026).

Os orçamentos geram fatias a partir das execuções já realizadas. Ausências permanecem sem imputação.

3.5.3 Tratamento da indisponibilidade

Uma comparação de qualidade exige todos os participantes do recorte, e um resultado ausente não recebe corte penalizado. As fatias incompletas permanecem nas análises operacionais com seu estado de término. Qualidade e operação usam conjuntos diferentes de fatias válidas como denominador.

3.6 Construção das bases analíticas

A campanha gerou registros que, após validação e reconstrução temporal, formaram bases com critérios específicos para qualidade, disponibilidade e modelagem. A consolidação preservou a ausência de imputação quando faltava um resultado exigido pelo recorte.

Reexecuções pontuais de METIS e KaHIP só substituíram registros depois de passar pelos mesmos critérios de validação. O quadro 8 reúne unidade, universo, completude e finalidade de cada produto.

Quadro 8 – Síntese dos recortes e produtos analíticos

Recorte ou produto	Participantes ou origem	Unidade e dimensão	Finalidade
Painel conservador	METIS, KaHIP, SA e TS	Fatia analítica. Universo de 2.086 fatias candidatas, das quais 2.071 são completas.	Comparações principais de qualidade.
Recorte C2	METIS, KaHIP, SA, TS, ILS e GRASP	Fatia analítica. Universo de 1.932 fatias candidatas, das quais 1.317 são completas.	Comparação completa dos seis algoritmos e análise secundária com CARTs em C2.
Painel operacional	Registros dos seis participantes	Fatia da grade temporal. Abrange as 2.086 fatias e não exige completude dos seis algoritmos.	Análise de disponibilidade e estados operacionais.
Matriz dos CARTs de referência	Derivada das fatias completas do painel conservador	Observação supervisionada. Contém 2.071 observações com os preditores e rótulos exigidos.	Treinamento e avaliação interna dos CARTs de referência.

Fonte: Autoria própria (2026).

C1 e C2 aplicaram critérios diferentes à cobertura de ILS e GRASP, mas retiveram as mesmas 276 combinações na campanha final. C2 foi adotada na análise completa por exigir as cinco sementes previstas, enquanto C1 permaneceu como verificação de sensibilidade do critério de cobertura.

3.6.1 Painel conservador de qualidade

O painel conservador é o recorte principal de qualidade. Ele parte de 298 combinações de grafo e valor de k, avaliadas nos sete orçamentos, totalizando 2.086 fatias candidatas. Dessas, 2.071 apresentaram resultados temporalmente elegíveis para METIS, KaHIP, SA e TS e formaram a base completa de comparação de qualidade entre os quatro participantes.

As 15 fatias restantes não possuíam resultado temporal elegível de METIS nem de KaHIP dentro do orçamento considerado. SA e TS estavam disponíveis, mas a ausência dos dois participantes *multilevel* impediu a comparação completa do portfólio conservador e excluiu essas observações da atribuição de vencedor e da matriz dos CARTs de referência.

Em cada fatia completa, os algoritmos foram representados conforme as regras de reconstrução temporal e agregação definidas na seção 3.5. Para SA e TS, primeiro foi selecionado o resultado temporal elegível de cada semente. Em seguida, o algoritmo foi representado pelo menor corte entre as sementes disponíveis, com desempate por menor tempo observado e menor semente. METIS e KaHIP foram representados por suas saídas pontuais aprovadas quando o tempo de parede medido não excedia o orçamento da fatia.

As 2.071 fatias completas sustentam as comparações de qualidade do painel conservador e o treinamento dos CARTs de referência. Os 60 grafos e as 298 combinações de grafo e valor de k foram representados. A consolidação desse painel foi mantida separada da campanha ampliada usada para examinar ILS e GRASP.

3.6.2 Análises completas com os seis algoritmos

As definições C1 e C2 aplicaram critérios de cobertura no nível da combinação de grafo e valor de k. Ambas exigiram os resultados necessários de METIS, KaHIP, SA e TS. C1 exigiu ainda ao menos uma partição materializada aprovada para ILS e uma para GRASP. C2 adotou o critério de cinco sementes com partições materializadas aprovadas para ILS e cinco para GRASP. C2 reduz a cobertura, mas permite comparar os seis algoritmos nas mesmas fatias completas. Ele não substitui o painel conservador, pois responde à pergunta sobre efeito do portfólio ampliado.

As 276 combinações formaram 1.932 fatias candidatas nos sete orçamentos. A comparação entre os seis algoritmos foi possível em 1.317 fatias, nas quais todos os participantes apresentaram resultado elegível dentro do limite temporal considerado.

As outras 615 fatias eram incompletas no nível do orçamento e não receberam rótulo de vencedor entre os seis algoritmos. A análise e os CARTs secundários de C2 usam as 1.317 fatias completas e têm alcance específico em relação aos modelos de referência do painel conservador.

3.6.3 Painel operacional

O painel operacional abrange o universo das 298 combinações e das 2.086 fatias da grade temporal. Ele mantém os estados de término, os tempos observados e a disponibilidade dos resultados em cada orçamento.

Das 2.086 fatias, 1.317 permitiram a comparação completa entre os seis algoritmos. As 769 restantes incluem 615 fatias temporalmente incompletas nas 276 combinações retidas por C2 e 154 fatias pertencentes às 22 combinações que não satisfizeram o critério de cobertura completa no nível da combinação.

Os registros incompletos foram usados nas análises de disponibilidade e de estados operacionais. Quando faltava o resultado de algum participante exigido, nenhum corte era imputado e nenhum vencedor entre os seis algoritmos era atribuído à fatia.

3.6.4 Construção dos rótulos de qualidade

Os rótulos de qualidade foram construídos separadamente em cada recorte analítico. O conjunto de participantes considerado em uma fatia depende da base usada. No painel conser-

vador, a comparação reúne METIS, KaHIP, SA e TS. Na análise C2, a comparação reúne os seis algoritmos.

O algoritmo nominal corresponde ao participante registrado como vencedor após a reconstrução temporal e a agregação das sementes. A seleção nominal pode incluir empates resolvidos pelas regras de desempate da base final. A vitória estrita exige vantagem real da melhor meta-heurística em relação ao melhor particionador multilevel, isto é, $g < 0$. A regra documentada priorizava o menor corte, o menor tempo observado e, persistindo o empate, a ordem fixa METIS, KaHIP, SA, TS, ILS e GRASP. A regra de desempate por menor tempo não reproduziu sete rótulos algorítmicos da base final, todos pertencentes à mesma combinação de grafo e k, uma em cada orçamento. Nesses casos, a divergência ocorre entre SA e TS, e não foi inferida uma regra alternativa.

A família nominal selecionada recebe a classe *multilevel* para METIS ou KaHIP e a classe meta-heurística para SA, TS, ILS ou GRASP. A divergência entre SA e TS não altera essa família nem os rótulos derivados de g. Quando $C_{\mathrm{meta}} = C_{\mathrm{ml}}$, o alvo nominal pode indicar uma meta-heurística sem configurar vitória estrita. Nesse caso, $g = 0$, a fatia pertence ao empate na margem de 1% e à competitividade na margem de 5%, enquanto a vitória estrita e a vantagem meta-heurística acima de 1% são falsas.

Para comparar as duas famílias, foram definidos C_{ml} e C_{meta}. O termo C_{ml} corresponde ao menor corte elegível entre METIS e KaHIP. O termo C_{meta} corresponde ao menor corte elegível entre as meta-heurísticas incluídas no recorte analisado.

A diferença relativa foi calculada por

$$g = \frac{C_{\mathrm{meta}} - C_{\mathrm{ml}}}{C_{\mathrm{ml}}}.$$

Valores negativos de g indicam menor corte da melhor meta-heurística. Valores positivos indicam menor corte da referência *multilevel*. Nas bases finais usadas para construir esses rótulos, não ocorreram fatias com $C_{\mathrm{ml}} = 0$. Por isso, não foi necessária uma regra adicional para o denominador nulo.

A partir dos cortes agregados da fatia, a listagem 2 recebe o algoritmo nominal registrado na base final e resume a derivação dos rótulos usados nas análises de qualidade. O procedimento mantém separados o rótulo nominal e os rótulos derivados da diferença relativa entre as famílias.

Listagem 2 – Derivação dos rótulos de qualidade de uma fatia completa

```
Receba uma fatia completa, os participantes do recorte e os
    cortes elegíveis agregados
Receba o algoritmo nominal registrado na base final
Considere somente os participantes definidos pelo recorte
    analisado
Associe ao algoritmo nominal sua família nominal selecionada
Selecione C_ml como o menor corte elegível entre METIS e KaHIP
Considere como meta-heurísticas somente os participantes
    definidos pelo recorte
Selecione C_meta como o menor corte elegível entre essas
    meta-heurísticas
Calcule g pela expressão definida anteriormente
Defina a vitória estrita de meta-heurística pela condição g < 0
Defina a vantagem meta-heurística acima de 1% pela condição g <
    -0,01
Defina o empate na margem de 1% pela condição -0,01 <= g <= 0,01
Defina a competitividade na margem de 5% pela condição g <= 0,05
Retorne o algoritmo nominal, a família nominal selecionada e os
    rótulos derivados
```

Fonte: Autoria própria (2026).

Quando $g = 0$, não há vitória estrita. A fatia pertence às margens de 1% e 5%, e a família nominal ainda depende do desempate registrado.

Quadro 9 – Rótulos derivados das comparações de qualidade

Rótulo	Definição
Família nominal selecionada	Família do algoritmo nominal registrado na base final.
Algoritmo vencedor	Algoritmo nominal registrado na base final, sujeito à limitação na proveniência dos sete rótulos descrita no texto.
Vitória estrita de meta-heurística	Ocorre quando $C_{\text{meta}} < C_{\text{ml}}$, ou, equivalentemente, quando $g < 0$.
Vantagem meta-heurística acima de 1%	Ocorre quando $g < -0{,}01$.
Empate na margem de 1%	Ocorre quando $-0{,}01 \leq g \leq 0{,}01$.
Competitividade na margem de 5%	Rótulo não exclusivo que ocorre quando $g \leq 0{,}05$, incluindo vitórias, empates na margem de 1% e derrotas por diferença relativa de até 5%.

Fonte: Autoria própria (2026).

Somente fatias completas recebem rótulos de qualidade. As demais permanecem no painel operacional.

Os atributos e as configurações formam os preditores dos CARTs, e os rótulos desta subseção formam os alvos.

3.7 Seleção interpretável por CART

A seleção foi formulada como classificação supervisionada (RICE, 1976). Cada observação relaciona atributos conhecidos antes da execução a um rótulo da subseção 3.6.4.

O CART produz previsões por divisões sucessivas nos atributos (BREIMAN *et al.*, 1984). As regras podem envolver topologia, k e orçamento.

3.7.1 Base de treinamento e atributos dos CARTs de referência

Os CARTs de referência foram treinados com as 2.071 fatias completas do painel conservador. Essas observações apresentavam resultados temporalmente elegíveis para METIS, KaHIP, SA e TS. As 15 fatias incompletas do universo de 2.086 fatias candidatas não foram incorporadas à matriz de modelagem.

Os conjuntos candidatos de preditores reuniram o valor de k, o orçamento de tempo e os atributos topológicos numéricos descritos na subseção 3.3.4. Esses atributos incluíram medidas de escala, densidade, distribuição de graus e conectividade. Algumas variantes acrescentaram a família morfológica, quando aplicável, e a coleção de origem.

A matriz final de modelagem não apresentou valores ausentes nos preditores requeridos. A diferença entre as 2.086 fatias candidatas e as 2.071 observações modeladas decorre da exigência de completude temporal da comparação entre os quatro participantes, e não de uma exclusão posterior por ausência de preditores.

A seleção das variantes ocorreu separadamente por alvo. O CART de vitória estrita de meta-heurística selecionou a variante com atributos topológicos numéricos, k, orçamento, família morfológica e coleção de origem. Os CARTs de família nominal selecionada, vantagem meta-heurística acima de 1% e competitividade na margem de 5% mantiveram a variante numérica formada por atributos topológicos, k e orçamento.

Os cortes observados, os vencedores, as diferenças relativas, os indicadores de competitividade e os demais campos derivados do *benchmark* não foram usados como atributos de entrada. Esses campos foram usados somente na construção dos alvos e na avaliação do modelo. Informações produzidas pela própria comparação entre algoritmos não foram usadas para prever seus resultados.

3.7.2 Alvos supervisionados

Os alvos supervisionados foram construídos a partir dos rótulos de qualidade definidos na subseção 3.6.4. Cada alvo corresponde a uma pergunta sobre o resultado da comparação na fatia analisada.

Foram definidos quatro alvos binários distintos. O alvo de família nominal selecionada indica se o algoritmo nominal registrado pertence a um particionador *multilevel* ou a uma meta-

heurística. Essa formulação reúne os algoritmos em duas classes e não distingue qual participante foi selecionado dentro de cada família. Em empates exatos entre famílias, o alvo registra a família nominal resultante do desempate e não uma vitória estrita.

O alvo de vitória estrita de meta-heurística ocorre quando $C_{\mathrm{meta}} < C_{\mathrm{ml}}$, ou, equivalentemente, quando $g < 0$. $g = 0$ nunca constitui vitória estrita, ainda que o empate tenha sido nominalmente atribuído a uma meta-heurística. O alvo de vantagem meta-heurística acima de 1% identifica as fatias em que $g < -0{,}01$. O alvo de competitividade na margem de 5% é não exclusivo e ocorre quando $g \leq 0{,}05$, incluindo vitórias, empates na margem de 1% e derrotas por diferença relativa de até 5%.

O algoritmo vencedor da fatia também foi usado como alvo de classificação multiclasse. Sua interpretação depende do recorte analítico, dos participantes exigidos, da política de agregação das sementes e da completude da comparação. Por isso, seus resultados foram apresentados separadamente dos alvos por família e por margem relativa.

A disponibilidade operacional foi analisada em base separada. Os registros indicam se o resultado exigido estava disponível dentro do orçamento, sem atribuir qualidade quando não havia corte elegível.

3.7.3 Configuração e validação agrupada

Os CARTs de referência do painel conservador foram ajustados separadamente para cada alvo supervisionado por meio da implementação de árvores de classificação da biblioteca *scikit-learn*. O procedimento comparou sete variantes de preditores e diferentes configurações da árvore.

Para o alvo de vitória estrita de meta-heurística, a variante selecionada reuniu os atributos topológicos numéricos, o valor de k, o orçamento, a família morfológica e a coleção de origem. Os alvos de família nominal selecionada, vantagem meta-heurística acima de 1% e competitividade na margem de 5% selecionaram a variante numérica formada por atributos topológicos, k e orçamento. O alvo exploratório de algoritmo vencedor também selecionou uma variante que incluía as informações categóricas e é descrito separadamente.

A configuração das árvores foi selecionada por validação cruzada agrupada em cinco *folds*. Foram avaliados os critérios de impureza Gini, entropia e *log loss*, profundidades máximas iguais a 2, 3, 4, 5 ou 6, números mínimos de observações por folha iguais a 10, 20 ou 40 e uso opcional de ponderação balanceada das classes.

O agrupamento foi definido pelo identificador do grafo. Todas as fatias derivadas da mesma instância, incluindo diferentes valores de k e orçamentos, permaneceram no mesmo *fold*. Configurações de um mesmo grafo não apareceram simultaneamente nos dados usados para ajustar a árvore e nos dados usados para avaliá-la.

Para cada alvo e variante de preditores, a configuração foi escolhida com base no desempenho obtido na validação agrupada. A acurácia balanceada foi usada como medida principal, acompanhada pelo F_1 macro. Após a seleção, uma árvore foi reajustada sobre a base

correspondente, com a configuração escolhida, para produzir as regras examinadas no trabalho.

A escolha dos hiperparâmetros e a estimativa principal de desempenho usaram a mesma validação cruzada agrupada, sem validação aninhada entre as duas etapas. As métricas são estimativas internas ao painel e podem incorporar otimismo decorrente da seleção do modelo no mesmo conjunto de grafos.

As configurações selecionadas para os alvos de vitória estrita de meta-heurística, família nominal selecionada, vantagem meta-heurística acima de 1% e competitividade na margem de 5% são apresentadas no quadro 10.

Quadro 10 – Configurações selecionadas para os CARTs de referência

Alvo	Critério	Profundidade máxima	Mínimo por folha	Ponderação das classes
Vitória estrita de meta-heurística	Entropia	6	10	Balanceada
Família nominal selecionada	Gini	5	10	Balanceada
Vantagem meta-heurística acima de 1%	Entropia	5	20	Sem ponderação
Competitividade na margem de 5%	Entropia	6	10	Balanceada

Fonte: Autoria própria (2026).

3.7.4 Variantes de preditores e análises secundárias

A comparação considerou sete variantes de preditores.

1. somente atributos topológicos numéricos.
2. somente valor de k e orçamento.
3. atributos topológicos numéricos e valor de k, sem orçamento.
4. atributos topológicos numéricos e orçamento, sem valor de k.
5. atributos topológicos numéricos, valor de k e orçamento.
6. somente família morfológica e coleção de origem.
7. atributos topológicos numéricos, valor de k, orçamento, família morfológica e coleção de origem.

A comparação foi usada para examinar a variação do desempenho preditivo entre conjuntos formados por atributos topológicos, variáveis da configuração experimental e informações categóricas de origem. Para a vitória estrita de meta-heurística, a variante selecionada reuniu

os atributos topológicos numéricos, o valor de k, o orçamento, a família morfológica e a coleção de origem. Os CARTs de família nominal selecionada, vantagem meta-heurística acima de 1% e competitividade na margem de 5% mantiveram suas variantes numéricas efetivamente selecionadas.

O algoritmo vencedor na fatia foi tratado como alvo multiclasse exploratório. Para esse alvo, a variante selecionada também incluiu família morfológica e coleção de origem. A configuração escolhida usou o critério de entropia, profundidade máxima igual a 6, mínimo de 10 observações por folha e ponderação balanceada das classes.

A distinção entre algoritmos individuais depende da cobertura dos participantes, da frequência das classes e da política de agregação das sementes. O alvo de algoritmo vencedor é apresentado separadamente das análises por família e por margem relativa. Suas verificações posteriores usaram a variante numérica formada por atributos topológicos, valor de k e orçamento, diferente da variante selecionada, que também incluía família morfológica e coleção de origem.

A análise secundária de C2 usa as 1.317 fatias completas dos seis algoritmos e tem alcance restrito a esse recorte. A disponibilidade operacional integra base separada, sem conversão de registros sem corte elegível em classes de qualidade ou valores artificiais.

3.8 Verificações de sensibilidade e estabilidade

A estabilidade dos CARTs foi avaliada por repetições dos *folds*, permutação dos rótulos, *bootstrap* por grafo, ablação de grupos de preditores e análises *leave-one* por família ou fonte. Esses procedimentos reutilizaram as configurações selecionadas e descrevem estabilidade condicionada, não a variabilidade de uma nova seleção completa.

As 20 repetições variaram apenas a semente usada para embaralhar os grafos e formar cinco *folds*. Todas as fatias de um grafo permaneceram juntas, sem estratificação formal. A semente interna do classificador ficou fixa. Em cada repetição, o CART foi reajustado no treino e avaliado nas dobras correspondentes. Acurácia, acurácia balanceada e F_1 macro mediram a sensibilidade às divisões.

O teste de permutação usou 100 embaralhamentos por alvo. Os *folds*, os preditores e os hiperparâmetros permaneceram fixos, e o CART foi reajustado dentro de cada dobra com os rótulos permutados. A pontuação observada foi comparada à distribuição obtida após romper a relação entre atributos e classes. Os testes foram interpretados separadamente, sem correção por multiplicidade. O valor empírico de p em cauda superior foi calculado por

$$p = \frac{b + 1}{B + 1}.$$

Para $B = 100$, b representa a quantidade de resultados permutados iguais ou superiores ao observado. A correção de uma unidade evita valor nulo quando nenhuma permutação alcança a pontuação observada. O embaralhamento foi feito por linha e não preservou a depen-

dência conjunta entre valores de k e orçamentos do mesmo grafo. O valor de p corresponde ao esquema de permutação adotado, que não ajusta a dependência entre fatias do mesmo grafo.

O *bootstrap* agrupado usou 1.000 reamostragens de grafos com reposição sobre as predições fora de dobra. Cada sorteio selecionou grafos, não linhas isoladas. Quando um grafo aparecia mais de uma vez, todas as suas fatias recebiam a mesma multiplicidade. O CART não foi reajustado em cada amostra. Acurácia balanceada e F_1 macro produziram intervalos percentis de 95%, delimitados por 2,5% e 97,5%. Esses intervalos descrevem a variação associada à reamostragem dos grafos, condicionada às configurações e às predições já obtidas.

A ablação posterior considerou nove conjuntos de preditores.

1. os atributos topológicos numéricos.
2. o valor de k e o orçamento.
3. os atributos de tamanho.
4. os atributos de forma.
5. os atributos topológicos combinados com o valor de k.
6. os atributos topológicos combinados com o orçamento.
7. a variante numérica completa, formada por atributos topológicos, valor de k e orçamento.
8. as informações categóricas de família e coleção de origem.
9. o conjunto completo de variáveis numéricas e categóricas.

Cada conjunto reutilizou hiperparâmetros e *folds*, com novo ajuste do CART e sem nova busca de configuração. A comparação mediu a alteração das métricas após incluir ou retirar grupos de variáveis. A seleção de hiperparâmetros não foi refeita para cada conjunto.

As análises *leave-one-family* e *leave-one-source* omitiram, respectivamente, cada uma das 22 categorias de família e das seis coleções. O grupo omitido saiu do treino e foi usado como teste, sem nova seleção de configuração. Foram calculadas acurácia, acurácia balanceada e F_1 macro. Casos com treino ou teste contendo uma única classe foram registrados, pois métricas como a acurácia balanceada podem deixar de representar discriminação entre classes nessas condições.

Quando família ou origem integrava os preditores, a coluna categórica permaneceu na entrada. Como a categoria omitida não aparecia no treino, o codificador a tratava como desconhecida no teste. As análises mediram transferência interna para grupos do próprio painel e sensibilidade à sua composição. Elas não constituíram validação em famílias ou fontes externas.

As verificações cobriram quatro alvos binários e o alvo multiclasse. Para este último, a variante numérica usada difere da variante completa selecionada, o que limita a leitura de estabilidade.

C2 empregou três repetições agrupadas, oito permutações e 100 reamostragens por *bootstrap*. A árvore binária específica de C2 usou Gini, profundidade máxima 2, mínimo de 30 observações por folha e nenhuma ponderação de classes. Esse protocolo pertence ao recorte C2 e não substitui as verificações dos CARTs de referência.

As métricas resultantes descrevem a estabilidade interna das configurações selecionadas nos respectivos painéis.

3.9 Rastreabilidade dos artefatos

Cada chamada foi registrada com grafo, algoritmo, k, semente, parâmetros, tempo, estado, saídas e, quando aplicável, validação da partição. Os *checkpoints* foram associados às chamadas, e as etapas de validação, reconstrução e agregação ligaram esses registros às observações do painel conservador, de C1, de C2 e do painel operacional. Esse encadeamento permite recuperar a chamada, a semente e o orçamento que originaram cada representante analítico e separar as bases de análise dos registros de verificação.

Para cada CART, registraram-se base, preditores, alvo, variante, configuração e métricas. Repetições, permutações, *bootstrap*, ablação e omissões de grupos permaneceram separadas do procedimento de seleção. Reexecuções de METIS e KaHIP só entraram nas bases depois da validação. Identificadores, caminhos, *hashes* e relações detalhadas permanecem no material de reprodutibilidade.

3.10 Ameaças à validade e delimitações metodológicas

As conclusões dependem das implementações, configurações, máquina, painel e regras analíticas adotadas.

A execução mono-*thread*, o controle do paralelismo implícito, a medição externa e a validação comum limitam fontes conhecidas de variação dentro da campanha. Os tempos continuam vinculados à máquina e às implementações usadas. Permanecem diferenças de linguagem, estruturas de dados, maturidade e interfaces, que o desenho não isola como variáveis independentes. O mesmo orçamento por fatia também não iguala operações ou esforço acumulado, sobretudo diante do melhor corte entre sementes e das saídas pontuais de METIS e KaHIP.

Os *checkpoints* registram tempo, corte e NFE sem a partição de cada instante. A validação independente incide nas partições materializadas, e os cortes históricos representam a trajetória informada pela implementação. Os rótulos descrevem o resultado sob a política de repetição e agregação adotada. Eles não estimam o desempenho esperado de uma execução estocástica escolhida ao acaso.

O alcance externo se limita às 60 instâncias, às transformações, aos valores de k, aos sete orçamentos e ao ambiente medido. O painel reúne grafos reais e sintéticos, mas foi curado e não constitui amostra aleatória. O ponto de 300 s é apenas o maior orçamento avaliado. ILS e

GRASP são comparados nas fatias completas de C2, enquanto os CARTs de referência usam o painel conservador. Nesse painel, 15 das 2.086 fatias candidatas ficaram fora da qualidade e da modelagem.

Os CARTs foram selecionados e avaliados internamente. A mesma validação agrupada participou da escolha da configuração e da estimação das métricas, sem validação aninhada ou conjunto externo. As verificações posteriores reutilizaram configurações selecionadas no painel. A permutação por linha não preservou a dependência entre fatias, o *bootstrap* reutilizou predições e as análises *leave-one* omitiram grupos do próprio painel. No alvo multiclasse, as verificações usaram uma variante diferente da selecionada.

Os resultados descrevem o painel, o portfólio, o ambiente e o protocolo avaliados.

4 RESULTADOS E DISCUSSÃO

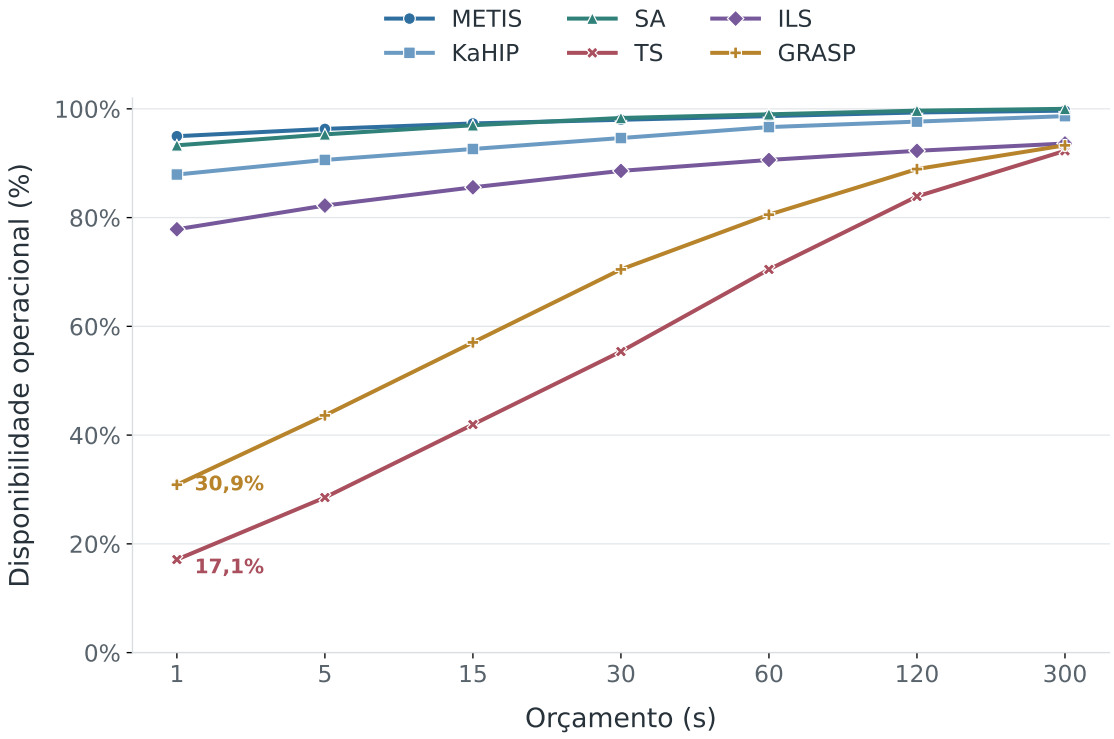
O painel conservador reúne 2.071 fatias completas para METIS, KaHIP, SA e TS e também sustenta os CARTs de referência. O recorte C2 contém 1.317 fatias completas para os seis algoritmos. O painel operacional abrange as 2.086 fatias da grade e registra disponibilidade e estados de término.

4.1 Visão geral e disponibilidade dos resultados

A disponibilidade operacional indica se o algoritmo forneceu ao menos um resultado elegível dentro do orçamento de tempo da fatia. Ela difere da elegibilidade temporal usada nas comparações de qualidade, definida na seção 3.5.1.

A figura 6 mostra a disponibilidade acumulada nos sete orçamentos. Em 1 s, METIS apresentou resultado em 283 das 298 combinações de grafo e k, KaHIP em 262, SA em 278, ILS em 232, GRASP em 92 e TS em 51. Em 300 s, as contagens foram 297, 294, 298, 279, 278 e 275, respectivamente. GRASP e TS tiveram crescimento mais gradual. Esses valores descrevem disponibilidade, não qualidade de corte.

Figura 6 – Disponibilidade operacional por algoritmo e orçamento



Denominador: 298 combinações elegíveis de grafo e k.

Fonte: Autoria própria (2026).

A cobertura integral das cinco sementes é mais restrita que a disponibilidade de um resultado. Em 1 s, ela ocorreu em 278 fatias para SA, 219 para ILS, 92 para GRASP e 44 para TS. Em 300 s, ocorreu em 298, 279, 278 e 273 fatias, respectivamente. Essas contagens

caracterizam as repetições, mas não substituem a agregação pelo menor corte elegível entre as sementes disponíveis.

No painel operacional, 1.317 das 2.086 fatias tinham os seis participantes disponíveis e 769 eram incompletas. No painel conservador, faltou resultado temporal elegível de METIS e KaHIP em 15 fatias. Foram 14 no orçamento de 1 s e uma no de 5 s. Os denominadores distintos decorrem dos critérios de inclusão, não da qualidade dos cortes.

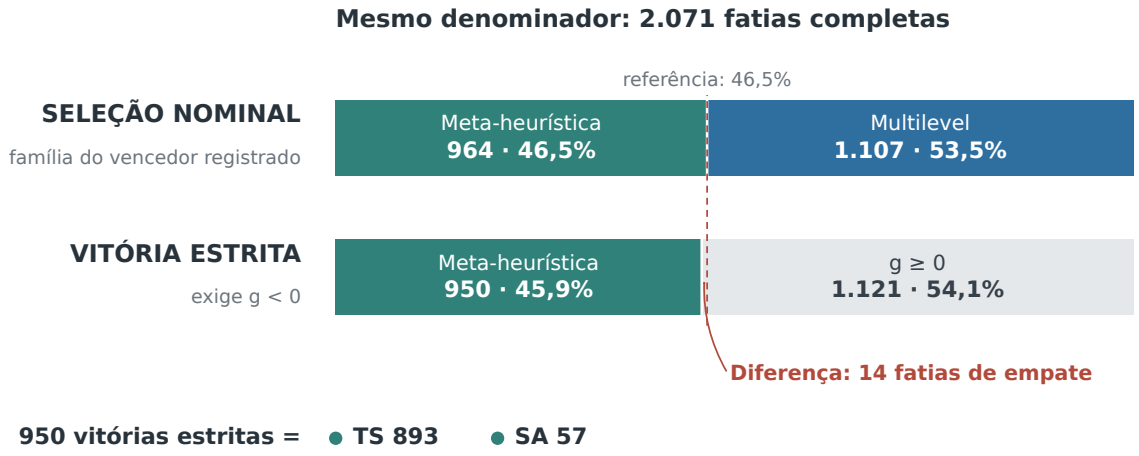
4.2 Qualidade no painel conservador

As 2.071 fatias completas do painel conservador sustentam as comparações de qualidade entre METIS, KaHIP, SA e TS.

4.2.1 Vencedores e diferenças relativas

O vencedor nominal é o algoritmo registrado como melhor na fatia após reconstrução, agregação e desempate. A família desse vencedor pode diferir do rótulo de vitória estrita, que exige $g < 0$, conforme definido na metodologia. Quando $g < 0$, a melhor meta-heurística teve corte menor que o melhor multilevel. Quando $g = 0$, há empate em corte entre as famílias. Quando $g > 0$, a melhor referência multilevel teve corte menor. A figura 7 resume a diferença entre família nominal e vitória estrita no painel conservador.

Figura 7 – Família nominal e vitória estrita no painel conservador



Fonte: Autoria própria (2026).

Entre as 2.071 fatias, TS foi o vencedor nominal em 900, METIS em 699, KaHIP em 408 e SA em 64. A tabela 1 mostra também o suporte em combinações de grafo e k e em grafos distintos.

A família nominal foi meta-heurística em 964 fatias (46,5%) e *multilevel* em 1.107. A vitória estrita de meta-heurística ocorreu em 950 fatias (45,9%) e não ocorreu em 1.121. A

Tabela 1 – Vencedores nominais no painel conservador

Algoritmo	Família	Fatias	Proporção	Combinações	Grafos
KaHIP	*multilevel*	408	19,7%	77	40
METIS	*multilevel*	699	33,8%	148	46
SA	meta-heurística	64	3,1%	17	10
TS	meta-heurística	900	43,5%	151	45

Fonte: Autoria própria (2026).

família nominal corresponde à família do algoritmo registrado na fatia, enquanto a vitória estrita exige $g < 0$. Os dois rótulos podem divergir em empates exatos entre famílias.

Entre os 149 empates exatos no menor corte, 88 envolveram algoritmos de famílias distintas e produziram $g = 0$. Esses casos não constituem vitória estrita, embora o desempate possa selecionar nominalmente uma meta-heurística. Dos 88 empates entre famílias no painel conservador, 14 foram nominalmente atribuídos a uma meta-heurística e 74 a um método *multilevel*. Por isso, a família nominal e a vitória estrita divergiram em 14 fatias desse painel. Em C2, a mesma diferença entre os dois rótulos ocorreu em 74 fatias. Os empates com $g = 0$ também pertencem às margens de 1% e 5%.

A tabela 2 separa indicadores que podem se sobrepor de faixas exclusivas de g. As quatro primeiras linhas devem ser lidas individualmente. As quatro últimas formam faixas disjuntas sobre o mesmo conjunto de fatias.

Tabela 2 – Rótulos não exclusivos e faixas exclusivas da diferença relativa

Natureza	Condição	Fatias	Proporção	Interpretação
Não exclusivo	$g < 0$	950	45,9%	Vitória estrita de meta-heurística.
Não exclusivo	$g < -0{,}01$	852	41,1%	Vantagem meta-heurística acima de 1%.
Não exclusivo	$-0{,}01 \leq g \leq 0{,}01$	233	11,3%	Cortes dentro da margem definida no protocolo, incluindo $g = 0$.
Não exclusivo	$g \leq 0{,}05$	1.211	58,5%	O melhor corte meta-heurístico não excede em mais de 5% o melhor *multilevel*.
Exclusiva	$g < -0{,}01$	852	41,1%	Vantagem meta-heurística acima de 1%.
Exclusiva	$-0{,}01 \leq g \leq 0{,}01$	233	11,3%	Faixa central de empate para visualização.
Exclusiva	$0{,}01 < g \leq 0{,}05$	126	6,1%	Meta-heurística até 5% acima do melhor *multilevel*.
Exclusiva	$g > 0{,}05$	860	41,5%	Vantagem *multilevel* acima de 5%.

Fonte: Autoria própria (2026).

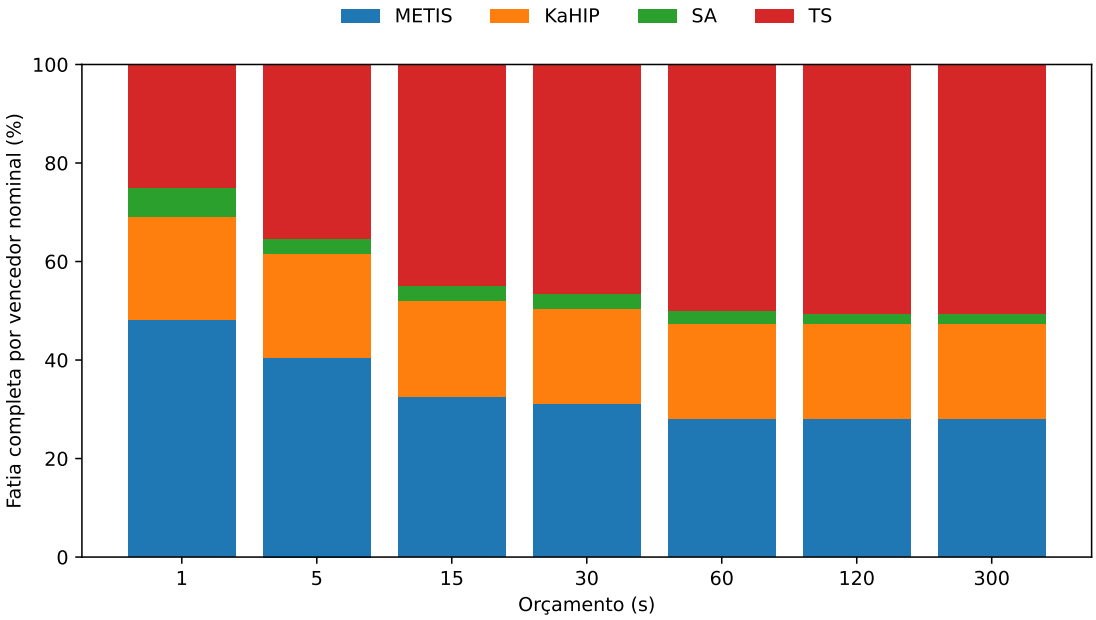
A mediana global de g foi zero, com quartis de $-0{,}0370$ e $2{,}0349$. A dispersão entre grafos justifica priorizar proporções, medianas, quantis e suporte, em vez de médias de cortes brutos entre instâncias.

4.2.2 Efeito do orçamento e do número de blocos

Os pontos de orçamento pertencem a trajetórias reconstruídas e não constituem amostras independentes.

A tabela 3 e a Figura 8 mostram a evolução temporal. A proporção de fatias com $g < 0$ passou de 30,3% em 1 s para 52,0% em 60 s e permaneceu nesse nível em 120 s e 300 s. A mediana passou de 0,2332 para $-0{,}0061$ entre 1 s e 120 s, valor mantido em 300 s. O platô após 60 s indica estabilidade na contagem de vitórias estritas, embora os melhores cortes anteriores permaneçam elegíveis nos orçamentos seguintes.

Figura 8 – Distribuição dos vencedores nominais por orçamento no painel conservador



Fonte: Autoria própria (2026).

Tabela 3 – Diferença relativa por orçamento no painel conservador

Orçamento	Fatias	$g < 0$	$g = 0$	$g > 0$	Mediana de g	Grafos
1 s	284	86 (30,3%)	9	189	0,2332	58
5 s	297	112 (37,7%)	14	171	0,0268	60
15 s	298	141 (47,3%)	13	144	0	60
30 s	298	146 (49,0%)	13	139	0	60
60 s	298	155 (52,0%)	13	130	$-0{,}0048$	60
120 s	298	155 (52,0%)	13	130	$-0{,}0061$	60
300 s	298	155 (52,0%)	13	130	$-0{,}0061$	60

Fonte: Autoria própria (2026).

Não houve ordenação monotônica por número de blocos. A maior proporção de vitórias estritas ocorreu em $k = 2$, com 231 de 418 fatias, e a menor em $k = 16$, com 158 de 410. A mediana de g foi negativa em $k = 2$, nula em $k = 4$ e positiva em $k = 8$, 16 e 32. Como k varia junto com grafos e orçamentos, a variação observada não pode ser atribuída exclusivamente ao número de blocos.

Tabela 4 – Diferença relativa por número de blocos

k	**Fatias**	$g < 0$	$g = 0$	$g > 0$	**Mediana de g**	**Grafos**
2	418	231 (55,3%)	35	152	$-0{,}0227$	60
4	416	167 (40,1%)	52	197	0	60
8	417	202 (48,4%)	1	214	0,0474	60
16	410	158 (38,5%)	0	252	0,0178	59
32	410	192 (46,8%)	0	218	0,0077	59

Fonte: Autoria própria (2026).

4.2.3 Condições associadas às vitórias estritas

As 950 vitórias estritas abrangeram 160 combinações de grafo e k e 45 grafos. TS respondeu por 893 fatias, 150 combinações e 44 grafos. SA respondeu por 57 fatias, 16 combinações e oito grafos. As contagens medem recorrência, não estabilidade entre sementes, pois o rótulo usa o menor corte elegível entre as repetições disponíveis.

A frequência variou entre as fontes. Nos 1.260 casos sintéticos, 791 tiveram $g < 0$, com mediana de $-0{,}0167$. As vitórias também ocorreram em arXiv (76 de 231) e SuiteSparse (83 de 385), mas não nas 30 fatias OGB, nas 30 da OSM Geofabrik nem nas 135 da TIGER/Line. A tabela 5 apresenta as medianas e o suporte completo. A origem da instância deve ser lida como uma categoria descritiva do painel. Ela pode estar confundida com propriedades estruturais dos grafos e não isola um efeito causal.

Tabela 5 – Diferença relativa por fonte das instâncias

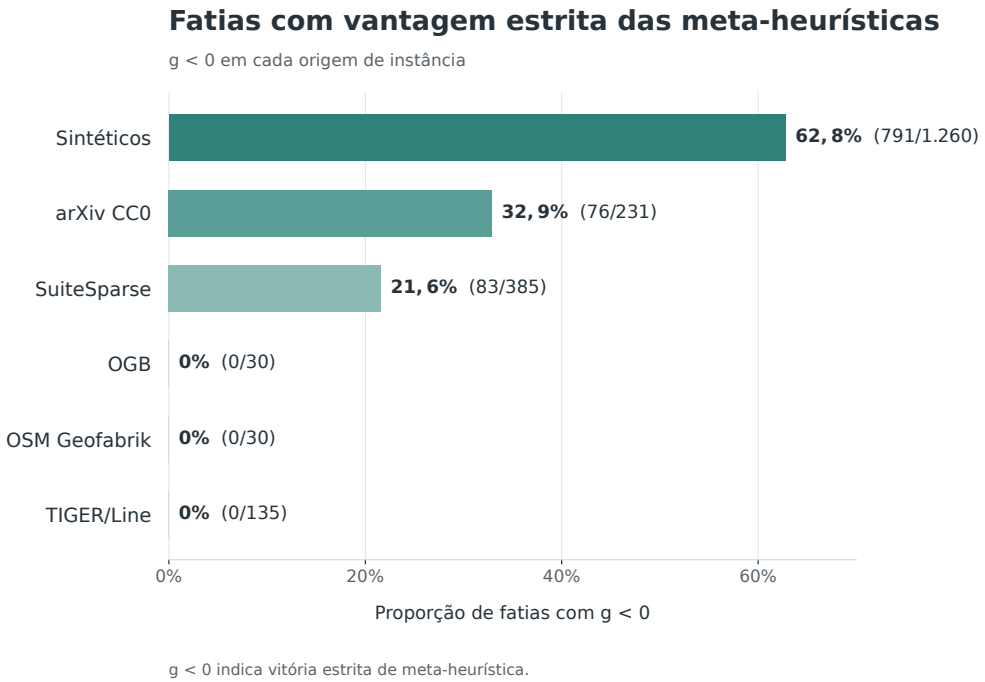
Fonte	**Fatias**	$g < 0$	**Proporção**	**Mediana de g**	**Combinações**	**Grafos**
OGB	30	0	0%	3,4839	5	1
OSM Geofabrik	30	0	0%	4.771,06	5	1
SuiteSparse	385	83	21,6%	1,6563	55	11
TIGER/Line	135	0	0%	197	20	4
arXiv CC0	231	76	32,9%	0,4216	33	7
Sintéticos	1.260	791	62,8%	$-0{,}0167$	180	36

Fonte: Autoria própria (2026).

A figura 9 resume a proporção de fatias com $g < 0$ por origem da instância. No painel avaliado, as fatias sintéticas concentraram a maior proporção de vitórias estritas de meta-heurísticas, com 791 de 1.260 fatias. As fontes arXiv CC0 e SuiteSparse apresentaram proporções intermediárias, enquanto OGB, OSM Geofabrik e TIGER/Line não apresentaram casos com $g < 0$ nesse recorte. Essa leitura descreve a composição observada no painel e não isola causalmente o efeito da origem da instância, pois origem, escala e propriedades estruturais dos grafos podem estar correlacionadas.

As maiores vantagens relativas ocorreram em três grafos sintéticos F02, formados por módulos densos em cadeia ou anel e avaliados com $k = 32$. O menor g foi $-0{,}6107$, de TS contra KaHIP, e reapareceu na mesma combinação entre 15 s e 300 s. Como os orçamentos reutilizam a mesma trajetória, o achado se limita às três instâncias.

Figura 9 – Vitórias estritas por origem da instância



Fonte: Autoria própria (2026).

4.3 Comparação dos seis algoritmos no recorte C2

C2 compara os portfólios de quatro e seis algoritmos nas mesmas 1.317 fatias, mantendo fixos grafo, k e orçamento. O pareamento isola as mudanças nominais associadas à inclusão de ILS e GRASP nesse recorte.

O vencedor foi mantido em 1.182 fatias e mudou em 135 (10,3%), distribuídas por 21 combinações de grafo e k e 12 grafos. GRASP passou a vencer 121 fatias e ILS, 14. A tabela 6 detalha as transições. O pareamento não deve ser confundido com a comparação entre os denominadores de 2.071 e 1.317.

Tabela 6 – Transição pareada dos vencedores no recorte C2

Vencedor com quatro	Vencedor com seis	Fatias
KaHIP	KaHIP	265
KaHIP	GRASP	67
METIS	METIS	185
METIS	GRASP	7
SA	SA	14
SA	GRASP	21
TS	TS	718
TS	ILS	14
TS	GRASP	26

Fonte: Autoria própria (2026).

Das 135 mudanças de algoritmo, 74 passaram de um vencedor *multilevel* para uma meta-heurística e 61 ocorreram dentro da família meta-heurística. Nenhuma seguiu a direção

inversa. Após a ampliação, a família nominal foi meta-heurística em 867 fatias e *multilevel* em 450.

A vitória estrita ocorreu em 793 fatias e não ocorreu em 524. A diferença de 74 entre 867 e 793 corresponde a empates entre famílias com $g = 0$ atribuídos nominalmente a uma meta-heurística. Esse total de 74 descreve empates e não é o mesmo conjunto das 74 mudanças de família.

4.4 Resultados dos CARTs de referência

Os quatro CARTs de referência foram avaliados internamente nas 2.071 observações do painel conservador.

4.4.1 Distribuição dos alvos e desempenho preditivo

Os quatro alvos têm suportes distintos, condição considerada na leitura das métricas.

Os alvos binários representam vitória estrita, família nominal selecionada, vantagem meta-heurística acima de 1% e competitividade na margem de 5%. A tabela 7 mostra seus suportes nas 2.071 observações de 60 grafos.

Tabela 7 – Distribuição dos alvos binários dos CARTs de referência

Alvo	Classe positiva	Positivos	Negativos
Vitória estrita de meta-heurística	$g < 0$	950	1.121
Família nominal selecionada	Algoritmo nominal pertence à família meta-heurística	964	1.107
Vantagem meta-heurística acima de 1%	$g < -0{,}01$	852	1.219
Competitividade na margem de 5%	$g \leq 0{,}05$	1.211	860

Fonte: Autoria própria (2026).

O CART de vitória estrita selecionou a variante numérica e categórica. Os demais selecionaram topologia, k e orçamento.

Nas predições fora de dobra, a acurácia balanceada variou de $0{,}7958$, no alvo de vantagem meta-heurística acima de 1%, a $0{,}8810$, na competitividade em 5%, acima da referência de $0{,}5000$. A tabela 8 apresenta também acurácia, F_1 macro e intervalos. As estimativas são internas porque a mesma validação agrupada foi usada para selecionar e avaliar os modelos, sem validação aninhada nem grafos externos.

Nenhuma das 100 permutações igualou ou superou a acurácia balanceada ou o F_1 macro observados. Com a correção de $+1$, o valor empírico foi $p = 1/101 = 0{,}0099$, o mínimo possível com 100 permutações. Os testes foram interpretados separadamente, sem correção por multiplicidade. O embaralhamento por linha não preservou a dependência entre as fatias do mesmo grafo.

Tabela 8 – Desempenho interno dos CARTs binários de referência

Alvo	Acurácia	Acurácia balanceada	F_1 macro	IC 95% da acurácia balanceada	p
Vitória estrita de meta-heurística	0,7866	0,7965	0,7859	$0{,}7380$ a $0{,}8480$	0,0099
Família nominal selecionada	0,8030	0,8086	0,8029	$0{,}7668$ a $0{,}8480$	0,0099
Vantagem meta-heurística acima de 1%	0,7842	0,7958	0,7829	$0{,}7490$ a $0{,}8384$	0,0099
Competitividade na margem de 5%	0,8885	0,8810	0,8841	$0{,}8350$ a $0{,}9203$	0,0099

Fonte: Autoria própria (2026).

O *bootstrap* estimou a variação das métricas por meio de 1.000 reamostragens agrupadas por grafo. O cálculo usou as predições fora de dobra, sem reajustar o CART em cada reamostragem, e produziu os intervalos percentis de 95% da tabela 8. Esses intervalos descrevem a incerteza condicionada às configurações selecionadas e às predições já obtidas.

O alvo nominal e o estrito diferem em 14 empates e selecionaram variantes e configurações distintas. A diferença entre as acurácias balanceadas de $0{,}8086$ e $0{,}7965$ não pode ser atribuída apenas às 14 observações nem interpretada como ablação controlada. A versão da biblioteca dos modelos históricos também não foi recuperada.

Nas 20 repetições agrupadas, as médias da acurácia balanceada foram $0{,}7884$, $0{,}7941$, $0{,}7573$ e $0{,}8620$, respectivamente. Os intervalos mínimo–máximo foram $0{,}7486$–$0{,}8095$, $0{,}7547$–$0{,}8259$, $0{,}6983$–$0{,}7991$ e $0{,}8125$–$0{,}9008$. A variação indica sensibilidade à composição das dobras, mas todos os valores permaneceram acima de $0{,}5000$.

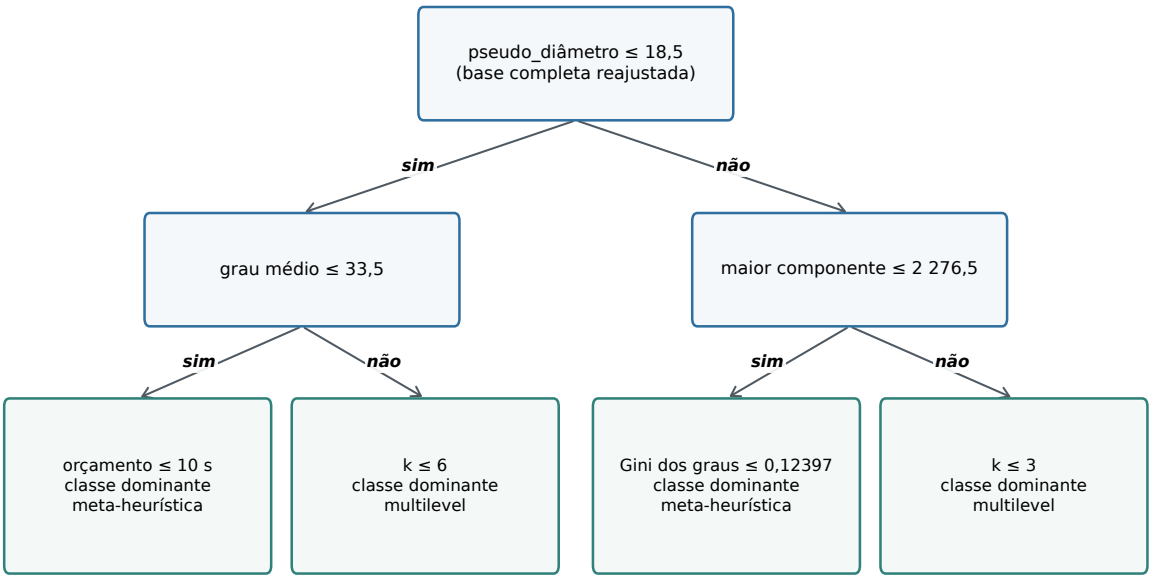
4.4.2 Regras produzidas pelos modelos

Após a seleção das configurações, as árvores foram reajustadas na base completa para examinar suas regras. Os limiares descrevem associações do modelo, não relações causais.

As árvores atingiram profundidade seis nos alvos de vitória estrita e competitividade e cinco nos de família nominal e vantagem meta-heurística acima de 1%. A raiz do CART estrito usou densidade. Os níveis seguintes incluíram grau máximo, mediana e Gini dos graus, tamanho da maior componente, k, orçamento e família morfológica. Nos outros modelos, as raízes usaram pseudo-diâmetro ou tamanho da maior componente. A figura 10 mostra os primeiros níveis do CART de família nominal.

Na figura 10, a árvore nominal, reajustada nas 2.071 observações, separa o pseudo-diâmetro em 18,5. Os ramos seguintes usam grau médio, tamanho da maior componente, orçamento, k e desigualdade dos graus. A árvore completa tem profundidade cinco e 27 folhas. A figura exibe apenas os primeiros níveis. As caixas inferiores resumem subárvores, e as métricas da tabela 8 continuam sendo de validação agrupada, não de ressubstituição.

Figura 10 – Primeiros níveis do CART reajustado para a família nominal selecionada



Fonte: Autoria própria (2026).

4.4.3 Variantes de preditores e estabilidade

As verificações seguintes examinam a variação do desempenho entre conjuntos de preditores e a estabilidade interna das configurações selecionadas.

Na vitória estrita, a variante completa obteve acurácia balanceada de $0{,}7965$, contra $0{,}7933$ na variante numérica, diferença de $0{,}0032$ sem comparação inferencial específica. Entropia e *log loss* empataram, e a entropia foi retida. As categorias não melhoraram família nominal nem competitividade e empataram no alvo de vantagem meta-heurística acima de 1%. No alvo exploratório de algoritmo, acrescentaram 0,0036.

Na ablação, k e orçamento isolados não reproduziram as configurações completas. No alvo estrito, a acurácia balanceada foi $0{,}5717$, contra $0{,}7933$ na variante numérica e $0{,}7965$ na selecionada. Nos demais alvos, os valores foram 0,5912, 0,5062 e 0,5875, contra 0,8086, 0,7958 e 0,8810. A comparação é pós-seleção.

As análises *leave-one* examinaram a transferência interna para grupos omitidos. Os resultados foram avaliados separadamente para cada família ou fonte retirada. No alvo estrito, 10 das 22 famílias e três das seis fontes produziram testes com uma única classe. Nesses testes, a acurácia balanceada igual a $1{,}0$ não mede discriminação entre classes. Quando a família ou a fonte omitida integrava os preditores, a categoria ausente do treino foi tratada como desconhecida no teste. O desempenho variou nos testes com duas classes, mas o procedimento não equivale a uma avaliação em famílias ou fontes independentes externas ao painel.

4.5 Análises exploratórias

As análises exploratórias tratam o algoritmo exato, os CARTs secundários de C2 e a instrumentação interna das meta-heurísticas. Seus resultados não são diretamente equivalentes aos CARTs de referência.

4.5.1 Alvo de algoritmo vencedor

A classificação do algoritmo vencedor depende dos rótulos nominais e inclui apenas 64 observações de SA.

O alvo continha 699 observações de METIS, 408 de KaHIP, 900 de TS e 64 de SA. A melhor variante, com topologia, k, orçamento, família e fonte, obteve acurácia de 0,6282, acurácia balanceada de 0,6103 e F_1 macro de 0,5650. Nas 20 repetições da variante numérica, a média foi 0,5399, entre 0,4601 e 0,6049.

A acurácia balanceada de $0{,}6103$ foi inferior às dos alvos binários. Sua interpretação deve considerar as quatro classes e o suporte de apenas 64 observações para SA. As sete divergências pertencem à mesma combinação de um grafo arXiv e valor de $k = 4$, com uma fatia em cada orçamento. A diferença nominal ocorre entre SA e TS e não altera a família nem os rótulos derivados de g. O alvo permanece exploratório.

4.5.2 CARTs secundários do recorte C2

Os CARTs de C2 foram avaliados nas 1.317 fatias completas por um procedimento distinto do usado nos modelos de referência.

As 1.317 observações abrangem 56 grafos. A família nominal foi meta-heurística em 867 casos e *multilevel* em 450. A vitória estrita ocorreu em 793 e não em 524. Houve ainda 365 casos de vantagem meta-heurística acima de 1%, 991 casos competitivos em 5% e seis classes no alvo de algoritmo, incluindo 121 seleções de GRASP e 14 de ILS.

O CART estrito usou topologia, k e orçamento em uma árvore específica de C2, com profundidade máxima 2, mínimo de 30 observações por folha, critério Gini e sem ponderação. Sua acurácia balanceada foi $0{,}6694$. A tabela 9 reúne os cinco alvos.

Tabela 9 – Desempenho interno dos CARTs secundários em C2

Alvo	Acurácia	Acurácia balanceada	F_1 macro	Média em três repetições
Vitória estrita de meta-heurística	0,6841	0,6694	0,6697	0,6836
Família nominal selecionada	0,7775	0,7637	0,7578	0,6976
Vantagem meta-heurística acima de 1%	0,7122	0,5459	0,5357	0,5260
Competitividade na margem de 5%	0,7244	0,6450	0,6405	0,6399
Algoritmo vencedor	0,5421	0,2563	0,2225	0,2288

Fonte: Autoria própria (2026).

O procedimento usou três repetições agrupadas, oito permutações e 100 reamostragens por *bootstrap*. Com oito permutações, o menor valor empírico possível foi $1/9 = 0{,}1111$. O procedimento não permite distinguir valores inferiores a esse limite.

Os intervalos de 95% da acurácia balanceada foram $0{,}5737$–$0{,}7514$ para vitória estrita e $0{,}6704$–$0{,}8246$ para família nominal. Para vantagem meta-heurística acima de 1% e competitividade em 5%, foram $0{,}4805$–$0{,}6104$ e $0{,}5606$–$0{,}7617$, respectivamente. O alvo de algoritmo vencedor apresentou o intervalo $0{,}2035$–$0{,}3334$. Não houve correção por multiplicidade. As métricas pertencem ao protocolo específico de C2 e não são diretamente comparáveis às dos CARTs de referência.

4.5.3 Diagnósticos das meta-heurísticas

Os registros operacionais caracterizam execuções, estados e instrumentação, sem fornecer uma medida comum de esforço.

Foram observadas 1.490 execuções de SA e 1.490 de TS. ILS teve 1.395 execuções ok e 95 encerramentos por limite de segurança. GRASP teve 1.450 execuções ok e 40 encerramentos pelo mesmo limite. Nas execuções ok, registraram-se 151.341 *checkpoints* para SA, 159.803 para TS, 140.895 para ILS e 139.146 para GRASP. Essas contagens descrevem instrumentação, não trabalho comparável.

Como as tabelas consolidadas não contêm NFE comparável, não foi feita análise quantitativa desse contador entre as meta-heurísticas ou em relação a METIS e KaHIP.

4.5.4 Análises complementares auditadas

Durante o desenvolvimento, também foram produzidos artefatos de tempo para alvo (TTT), curvas de distribuição acumulada empírica (ECDF), políticas retrospectivas de referência e ablações. Esses materiais foram mantidos como análises complementares porque registram verificações feitas durante o projeto, sem substituir os resultados principais. O inventário completo ficou preservado no pacote técnico de auditoria.

As análises de TTT e ECDF foram usadas para examinar quando certos níveis de qualidade eram atingidos ao longo do orçamento. Quando usam *checkpoints*, elas descrevem trajetórias instrumentadas de corte e tempo, não partições intermediárias validadas. Por isso, são tratadas como análises complementares.

As políticas *single best solver* (SBS) e *virtual best solver* (VBS) foram usadas como referências retrospectivas internas. O SBS representa, após a campanha, a melhor escolha fixa no painel. O VBS representa, também após a campanha, um limite oracular que seleciona o melhor participante observado em cada fatia. Essas referências servem apenas como comparação retrospectiva dentro do painel.

As ablações de atributos dos CARTs indicam como grupos de preditores afetam os resultados de classificação. As comparações Rust–Python foram tratadas como diagnóstico técnico

de implementação. Elas documentam decisões de desenvolvimento das meta-heurísticas e não sustentam conclusões gerais sobre linguagem ou desempenho.

Quadro 11 – Estatuto das análises complementares auditadas

Análise	Estatuto	Uso na monografia	Ressalva
Tempo para alvo (TTT)	Análise complementar auditada	Descreve quando níveis de qualidade são alcançados ao longo do orçamento	Quando usa *checkpoints*, refere-se a trajetórias instrumentadas de corte e tempo
ECDF	Análise complementar auditada	Resume a distribuição empírica de alcance dos alvos em recortes temporais	Tem função complementar aos rótulos principais
SBS e VBS	Referências retrospectivas internas	Comparam a melhor escolha fixa do painel e o melhor participante observado em cada fatia	Não constituem métodos prospectivos de seleção
Ablação de atributos dos CARTs	Sensibilidade interna dos modelos	Indica como grupos de preditores afetam os resultados de classificação	Não substitui a validação dos CARTs
Comparações Rust–Python	Diagnóstico técnico de implementação	Documentam decisões de desenvolvimento das meta-heurísticas	Não sustentam conclusão geral sobre superioridade de linguagem

Fonte: Autoria própria (2026).

4.6 Síntese das perguntas de pesquisa

As respostas abaixo respeitam o conjunto de fatias válido para cada análise de qualidade, disponibilidade e modelagem.

PQ1 — qualidade e disponibilidade ao longo dos orçamentos.

A disponibilidade cresceu com o orçamento, mais gradualmente para TS e GRASP. No painel conservador, a vitória estrita passou de 30,3% em 1 s para 52,0% em 60 s e permaneceu nesse nível até 300 s. A mediana de g tornou-se ligeiramente negativa. Os pontos pertencem a trajetórias aninhadas.

PQ2 — condições associadas a vitórias, empates e competitividade.

A família nominal foi meta-heurística em 964 fatias e a vitória estrita ocorreu em 950, distribuídas por 45 grafos. As ocorrências concentraram-se nos sintéticos e apareceram também

em arXiv e SuiteSparse. Orçamento, k, conectividade e estatísticas de grau estiveram associados aos rótulos dentro do painel, sem evidência causal ou de estabilidade entre sementes.

PQ3 — capacidade classificatória e regras dos CARTs.

Os CARTs binários alcançaram acurácia balanceada entre $0{,}7958$ e $0{,}8810$. Para vitória estrita, o valor foi $0{,}7965$, com $p = 0{,}0099$ e intervalo de 95% de $0{,}7380$ a $0{,}8480$. As regras usaram topologia, k, orçamento e, no modelo estrito, categorias de origem. A avaliação é interna e não aninhada. O alvo multiclasse ($0{,}6103$, 64 casos de SA) e o CART estrito de C2 ($0{,}6694$, $p = 0{,}1111$) permanecem exploratórios.

5 CONCLUSÃO

O trabalho desenvolveu e avaliou um *framework* de *benchmarking* para comparar algoritmos de particionamento de grafos e examinar a seleção interpretável de algoritmos por instância. A avaliação combinou resultados temporalmente elegíveis, validação das partições materializadas e recortes com conjuntos distintos de fatias válidas. Os CARTs foram treinados a partir de atributos dos grafos e das configurações experimentais.

A primeira pergunta de pesquisa exigiu separar qualidade e disponibilidade. Nas 2.071 fatias completas do painel conservador, a proporção de vitórias estritas de meta-heurísticas passou de 30,3% em 1 s para 52,0% em 60 s e permaneceu nesse nível até 300 s. A disponibilidade foi analisada no painel operacional, pois produzir um resultado dentro do orçamento não determina a qualidade relativa do corte. Os pontos temporais são aninhados e não constituem experimentos independentes.

Para a segunda pergunta, a família nominal selecionada e a vitória estrita mostraram aspectos diferentes da comparação. Empates podem atribuir nominalmente a fatia a uma meta-heurística sem produzir $g < 0$. No painel conservador, as 950 vitórias estritas abrangeram 45 grafos. As maiores vantagens relativas, porém, concentraram-se em três grafos sintéticos F02 com $k = 32$, sem se estender automaticamente à família ou a outros grafos modulares. No recorte C2, a inclusão de ILS e GRASP alterou 135 vencedores nominais em 1.317 fatias pareadas. Dessas mudanças, 74 trocaram a família de *multilevel* para meta-heurística. As outras 61 mudanças ocorreram entre meta-heurísticas, sem troca da família nominal.

Na terceira pergunta, os CARTs binários de referência alcançaram acurácia balanceada entre $0{,}7958$ e $0{,}8810$ na validação agrupada por grafo. As árvores produziram regras baseadas em topologia, k, orçamento e, em um alvo, categorias de origem. Esses valores medem capacidade classificatória interna ao painel. A seleção e a estimação usaram a mesma validação agrupada, sem validação aninhada, conjunto externo ou avaliação como política operacional. As divisões das árvores representam associações, não efeitos causais. O alvo multiclasse e os CARTs de C2 permaneceram exploratórios. Sua interpretação está condicionada ao suporte das classes e ao protocolo específico desse recorte.

O protocolo comparou famílias com instrumentação diferente usando o tempo de parede como referência comum e manteve separadas as análises de qualidade e disponibilidade. As bases analíticas distinguem família nominal, vitória estrita, margens relativas e o recorte pareado C2 sem imputar cortes ausentes. Os CARTs produziram regras inspecionáveis sobre parte da variação observada, condicionadas ao painel e ao procedimento de avaliação.

O alcance dos resultados é limitado pelo painel curado de 60 grafos, pelo ambiente e pelas implementações avaliadas. As meta-heurísticas foram representadas pelo melhor corte de até cinco sementes, enquanto METIS e KaHIP forneceram saídas pontuais. Os *checkpoints* não armazenaram partições intermediárias materializadas, e os modelos não foram validados externamente. A reprodução exata também é limitada pela falta de confirmação de versões de alguns componentes históricos.

Trabalhos futuros podem avaliar a seleção em grafos externos com validação aninhada, repetir o protocolo em outros ambientes e comparar políticas de agregação e escolha operacional. Estudos com painel e portfólio ampliados, além do armazenamento e da validação das partições intermediárias, poderão verificar se os padrões observados também ocorrem fora do recorte atual.

Dentro do escopo avaliado, nenhuma família apresentou superioridade uniforme. As meta-heurísticas obtiveram vitórias estritas e permaneceram competitivas em parte do painel, enquanto os métodos *multilevel* foram superiores em outras configurações. Os CARTs produziram regras interpretáveis sobre essa variação, mas sua conversão em uma política aplicável a grafos externos ainda depende de validação independente e de avaliação operacional.

5.1 Declaração de Uso de IA Generativa

Neste trabalho, ferramentas de IA generativa, incluindo o ChatGPT, da OpenAI, foram utilizadas como apoio à revisão linguística, à padronização textual, à identificação de inconsistências de redação e à preparação de ajustes editoriais e visuais. Essas ferramentas não foram empregadas para gerar dados experimentais, recalcular métricas, produzir resultados científicos, definir decisões metodológicas ou substituir a interpretação crítica da autoria.

As análises, os resultados, as referências, as decisões metodológicas e as conclusões foram conferidos e assumidos pela autoria. A responsabilidade pelo conteúdo final, pela verificação das referências, pela consistência dos artefatos e pela integridade acadêmica do trabalho permanece integralmente sob responsabilidade do autor.

REFERÊNCIAS

arXiv. **Open Archives Initiative (OAI)**. s.d. Disponível em: https://info.arxiv.org/help/oa/index.html. Acesso em: 10 jun. 2026.

arXiv. **Terms of Use for arXiv APIs**. s.d. Disponível em: https://info.arxiv.org/help/api/tou.html. Acesso em: 10 jun. 2026.

AUGER, A.; BROCKHOFF, D.; HANSEN, N. Coco: The experimental procedure. **ACM Transactions on Evolutionary Learning and Optimization**, v. 1, n. 4, 2021.

BARABÁSI, A.-L.; ALBERT, R. Emergence of scaling in random networks. **Science**, v. 286, n. 5439, p. 509–512, 1999.

BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, v. 35, n. 3, p. 268–308, 2003.

BOAS, M. G. V. *et al.* Optimal decision trees for the algorithm selection problem: integer programming based approaches. **International Transactions in Operational Research**, v. 28, n. 5, p. 2759–2781, 2021.

BREIMAN, L. *et al.* **Classification and Regression Trees**. Belmont, CA: Wadsworth, 1984. ISBN 9780412048418.

BULUC, A. *et al.* Recent advances in graph partitioning. In: KLIEMANN, L.; SANDERS, P. (Ed.). **Algorithm Engineering: Selected Results and Surveys**. Cham: Springer, 2016, (Lecture Notes in Computer Science, v. 9220). p. 117–158.

CHEVALIER, C.; PELLEGRINI, F. Pt-scotch: A tool for efficient parallel graph ordering. **Parallel Computing**, v. 34, n. 6–8, p. 318–331, 2008.

CHIKARADDI, A. K. *et al.* Graph partitioning link quality energy aware routing (gplqear) hybrid protocol in wireless multimedia sensor networks. In: **2020 International Conference on Inventive Computation Technologies (ICICT)**. Piscataway, NJ, USA: IEEE, 2020. p. 241–246.

CHUNG, F. R. K. **Spectral Graph Theory**. Providence, RI, USA: American Mathematical Society, 1997. v. 92. (CBMS Regional Conference Series in Mathematics, v. 92).

DAVIS, T. A.; HU, Y. The university of florida sparse matrix collection. **ACM Transactions on Mathematical Software**, v. 38, n. 1, p. 1:1–1:25, 2011. Atualmente conhecido como SuiteSparse Matrix Collection.

ELBHIRI, B. *et al.* A new spectral classification for robust clustering in wireless sensor networks. In: **6th Joint IFIP Wireless and Mobile Networking Conference (WMNC)**. Piscataway, NJ, USA: IEEE, 2013. p. 1–10.

ERDŐS, P.; RÉNYI, A. On random graphs i. **Publicationes Mathematicae**, v. 6, p. 290–297, 1959.

FEO, T. A.; RESENDE, M. G. C. Greedy randomized adaptive search procedures. **Journal of Global Optimization**, v. 6, n. 2, p. 109–133, 1995.

FIDUCCIA, C. M.; MATTHEYSES, R. M. A linear-time heuristic for improving network partitions. In: **Proceedings of the 19th Design Automation Conference**. New York, NY, USA: ACM, 1982. p. 175–181.

Geofabrik GmbH. **Download OpenStreetMap Data for Rhode Island**. s.d. Disponível em: https://download.geofabrik.de/north-america/us/rhode-island.html. Acesso em: 10 jun. 2026.

GLOVER, F. Tabu search—part i. **ORSA Journal on Computing**, v. 1, n. 3, p. 190–206, 1989.

GLOVER, F.; KOCHENBERGER, G. A. (Ed.). **Handbook of Metaheuristics**. Boston: Kluwer Academic Publishers, 2003. v. 57. (International Series in Operations Research and Management Science, v. 57). ISBN 9781402072635.

HENDRICKSON, B.; KOLDA, T. G. Graph partitioning models for parallel computing. **Parallel Computing**, v. 26, n. 12, p. 1519–1534, 2000.

HEUER, T.; SANDERS, P.; SCHLAG, S. Kaminpar: High-quality graph partitioning in shared-memory. In: **Proceedings of the SIAM Symposium on Algorithm Engineering and Experiments (ALENEX)**. Philadelphia, PA, USA: SIAM, 2021. p. 1–13.

HOLTGREWE, M.; SANDERS, P.; SCHULZ, C. **Engineering a Scalable High Quality Graph Partitioner**. 2010. Disponível em: https://arxiv.org/abs/0910.2004.

HOOKER, J. N. Testing heuristics: We have it all wrong. **Journal of Heuristics**, v. 1, p. 33–42, 1995.

HU, W. *et al.* Open graph benchmark: Datasets for machine learning on graphs. In: **Advances in Neural Information Processing Systems**. [s.n.], 2020. v. 33, p. 22118–22133. Disponível em: https://proceedings.neurips.cc/paper/2020/hash/fb60d411a5c5b72b2e7d3527cfc84fd0-Abstract.html.

KARYPIS, G.; KUMAR, V. A fast and high quality multilevel scheme for partitioning irregular graphs. **SIAM Journal on Scientific Computing**, v. 20, n. 1, p. 359–392, 1998.

KARYPIS, G.; SCHLOEGEL, K.; KUMAR, V. **ParMETIS: Parallel Graph Partitioning and Sparse Matrix Ordering Library, Version 3.1**. Minneapolis, MN, USA, 2003.

KERNIGHAN, B. W.; LIN, S. An efficient heuristic procedure for partitioning graphs. **The Bell System Technical Journal**, v. 49, n. 2, p. 291–307, 1970.

KIRKPATRICK, S. Optimization by simulated annealing: Quantitative studies. **Journal of statistical physics**, Springer, v. 34, n. 5, p. 975–986, 1984.

KOLODZIEJ, S. P. *et al.* The suitesparse matrix collection website interface. **Journal of Open Source Software**, v. 4, n. 35, p. 1244–1248, 2019. Disponível em: https://doi.org/10.21105/joss.01244.

KOTTHOFF, L. Algorithm selection for combinatorial search problems: A survey. **AI Magazine**, v. 35, n. 3, p. 48–60, 2014.

LANCICHINETTI, A.; FORTUNATO, S.; RADICCHI, F. Benchmark graphs for testing community detection algorithms. **Physical Review E**, v. 78, n. 4, p. 046110, 2008.

LESKOVEC, J.; KREVL, A. **SNAP Datasets: Stanford Large Network Dataset Collection**. 2014. http://snap.stanford.edu/data.

LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In: **Handbook of metaheuristics**. Cham: Springer, 2018. p. 129–168.

MCGEOCH, C. C. **A Guide to Experimental Algorithmics**. Cambridge: Cambridge University Press, 2012. ISBN 9781107039871.

MOLNAR, C. **Interpretable Machine Learning**. Morrisville, NC, USA: Lulu.com, 2020. https://christophm.github.io/interpretable-ml-book/.

NEWMAN, M. E. J. **Networks: An Introduction**. Oxford: Oxford University Press, 2010. ISBN 9780199206650.

Open Data Commons. **Open Data Commons Attribution License (ODC-By) v1.0**. s.d. Disponível em: https://opendatacommons.org/licenses/by/1-0/. Acesso em: 10 jun. 2026.

Open Data Commons. **Open Database License (ODbL) v1.0**. s.d. Disponível em: https://opendatacommons.org/licenses/odbl/1-0/. Acesso em: 10 jun. 2026.

Open Graph Benchmark. **Node Property Prediction: ogbn-arxiv**. s.d. Disponível em: https://ogb.stanford.edu/docs/nodeprop/\#ogbn-arxiv. Acesso em: 10 jun. 2026.

OpenStreetMap contributors. **Copyright and License**. s.d. Disponível em: https://www.openstreetmap.org/copyright. Acesso em: 10 jun. 2026.

PENROSE, M. D. **Random Geometric Graphs**. Oxford, UK: Oxford University Press, 2003. (Oxford Studies in Probability).

RICE, J. R. The algorithm selection problem. In: **Advances in Computers**. New York, NY, USA: Academic Press, 1976. v. 15, p. 65–118.

SANDERS, P.; SCHULZ, C. Think locally, act globally: Highly-balanced graph partitioning. In: **Proceedings of the 12th International Symposium on Experimental Algorithms**. Berlin, Heidelberg: Springer, 2013. p. 164–175.

SuiteSparse Matrix Collection. **About the SuiteSparse Matrix Collection**. s.d. Disponível em: https://sparse.tamu.edu/about. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **AG-Monien/3elt**. s.d. Disponível em: https://sparse.tamu.edu/AG-Monien/3elt. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/data**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/data. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/delaunay_n10**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/delaunay_n10. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/delaunay_n11**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/delaunay_n11. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/delaunay_n12**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/delaunay_n12. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/delaunay_n13**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/delaunay_n13. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/fe_4elt2**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/fe_4elt2. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **DIMACS10/wing_nodal**. s.d. Disponível em: https://sparse.tamu.edu/DIMACS10/wing_nodal. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **Hamm/add32**. s.d. Disponível em: https://sparse.tamu.edu/Hamm/add32. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **HB/bcsstk33**. s.d. Disponível em: https://sparse.tamu.edu/HB/bcsstk33. Acesso em: 10 jun. 2026.

SuiteSparse Matrix Collection. **Nasa/nasa1824**. s.d. Disponível em: https://sparse.tamu.edu/Nasa/nasa1824. Acesso em: 10 jun. 2026.

U.S. Census Bureau. **2025 TIGER/Line Shapefile: Roads, Dallas County, Texas (FIPS 48113)**. 2025. Disponível em: https://www2.census.gov/geo/tiger/TIGER2025/ROADS/tl_2025_48113_roads.zip. Acesso em: 10 jun. 2026.

U.S. Census Bureau. **2025 TIGER/Line Shapefile: Roads, Kings County, New York (FIPS 36047)**. 2025. Disponível em: https://www2.census.gov/geo/tiger/TIGER2025/ROADS/tl_2025_36047_roads.zip. Acesso em: 10 jun. 2026.

U.S. Census Bureau. **2025 TIGER/Line Shapefile: Roads, Los Angeles County, California (FIPS 06037)**. 2025. Disponível em: https://www2.census.gov/geo/tiger/TIGER2025/ROADS/tl_2025_06037_roads.zip. Acesso em: 10 jun. 2026.

U.S. Census Bureau. **2025 TIGER/Line Shapefile: Roads, New York County, New York (FIPS 36061)**. 2025. Disponível em: https://www2.census.gov/geo/tiger/TIGER2025/ROADS/tl_2025_36061_roads.zip. Acesso em: 10 jun. 2026.

U.S. Census Bureau. **2025 TIGER/Line Shapefiles**. 2025. Disponível em: https://www.census.gov/geographies/mapping-files/time-series/geo/tiger-line-file.2025.html. Acesso em: 10 jun. 2026.

WALSHAW, C. **The Graph Partitioning Archive**. 2023. https://chriswalshaw.co.uk/partition/. Originally hosted at the University of Greenwich; last updated 2023-03-27. Accessed 2025-10-30.

WATTS, D. J.; STROGATZ, S. H. Collective dynamics of small-world networks. **Nature**, v. 393, p. 440–442, 1998.

ZIEMANN, M.; POULAIN, P.; BORA, A. The five pillars of computational reproducibility: bioinformatics and beyond. **Briefings in Bioinformatics**, Oxford University Press, v. 24, n. 6, p. bbad375, 2023.