ISMAEL MARTINS SILVA

APLICAÇÃO DE DECOMPOSIÇÕES SOBRE O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA, ENTREGA E JANELAS DE TEMPO

LAVRAS – MG
2023

ISMAEL MARTINS SILVA

APLICAÇÃO DE DECOMPOSIÇÕES SOBRE O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA, ENTREGA E JANELAS DE TEMPO

TCC apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, área de concentração em Pesquisa Operacional, para a obtenção do título de Bacharel.

Prof. Dr. Mayron César de Oliveira Moreira
Orientador
Prof. Dr. Dilson Lucas Pereira
Coorientador

LAVRAS – MG
2023

Ficha Catalográfica preparada pelo Sistema de Geração de Ficha Catalográfica da Biblioteca Universitária da UFLA, com dados informados pelo(a) próprio(a) autor(a).

Silva, Ismael Martins
Aplicação de Decomposições Sobre o Problema de Roteamento de Veículos com Coleta, Entrega e Janelas de Tempo / Ismael Martins Silva. - 2023.
50 p. : il.

Orientador(a): Mayron César de Oliveira Moreira.
Coorientador(a): Dilson Lucas Pereira.
TCC (graduação) - Universidade Federal de Lavras, 2023.
Bibliografia.

1. PRV. 2. PRVCEJT. 3. Decomposições. 4. Execução Paralela. 5. Redução de Tempo. I. Moreira, Mayron César de Oliveira. II. Pereira, Dilson Lucas. III. Título.

ISMAEL MARTINS SILVA

APLICAÇÃO DE DECOMPOSIÇÕES SOBRE O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM COLETA, ENTREGA E JANELAS DE TEMPO

APPLICATION OF DECOMPOSITIONS TO THE VEHICLE ROUTING PROBLEM WITH PICKUP, DELIVERY AND TIME WINDOWS

TCC apresentado à Universidade Federal de Lavras, como parte das exigências do Curso de Ciência da Computação, área de concentração em Pesquisa Operacional, para a obtenção do título de Bacharel.

APROVADA em 07 de Dezembro de 2023.

Prof. Dr. Dilson Lucas Pereira	UFLA
Prof. Dr. Julio Cesar Alves	UFLA
Prof. Dr. Mayron César de Oliveira Moreira	UFLA

gov.br Documento assinado digitalmente
MAYRON CESAR DE OLIVEIRA MOREIRA
Data: 21/12/2023 08:46:17-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. Mayron César de Oliveira Moreira
Orientador
Prof. Dr. Dilson Lucas Pereira
Coorientador

gov.br Documento assinado digitalmente
DILSON LUCAS PEREIRA
Data: 20/12/2023 22:50:49-0300
Verifique em https://validar.iti.gov.br

LAVRAS – MG
2023

Nossas virtudes e nossos defeitos são inseparáveis, assim como a força e a matéria, quando se separam, o homem deixa de existir.

(Nikola Tesla)

AGRADECIMENTOS

Em primeiro lugar, agradeço a toda a minha família, principalmente à minha mãe e ao meu pai. Sem o auxílio deles eu não teria superado todos os percalços que a vida coloca em nossos caminhos, e não teria chegado onde estou hoje. Deixo também um grande agradecimento aos meus colegas Caio de Oliveira Lopes e Layse Cristina Silva Garcia, pelo cotidiano descontraído que vocês proporcionaram e por todo o suporte nos momentos de dificuldades. Por fim, mas não menos importante, agradeço aos meus orientadores Dilson Lucas Pereira e Mayron César de Oliveira Moreira, por todo o apoio e ensinamentos que dedicaram a mim nesta árdua etapa da vida, que é a graduação.

A sorte favorece a mente bem preparada.

(Louis Pasteur)

RESUMO

Este artigo estuda a utilização de técnicas de decomposição sobre a resolução do Problema de Roteamento de Veículos com Coleta, Entrega e Janelas de Tempo (PRVCEJT). A questão levantada por este trabalho é: "Apesar da economia de tempo dos métodos de decomposição na resolução do PRVCEJT, a perda de qualidade se sobressai?". Nós apresentamos sete adaptações das decomposições elaborados por Santini et al. (2023). Os experimentos foram realizados utilizando tanto as instâncias, quanto o solver estado da arte proposto por Sartori e Buriol (2020). Com os resultados prontos, realizamos comparações com os obtidos por Sartori e Buriol (2020), além de trazer à tona as características das decomposições. Nossas análises mostram que as adaptações **Agrupamento de Baricentro 2** e **Agrupamento de Baricentro 4**, possuem o melhor desempenho dentre as que apresentamos.

Palavras-chave: PRV. PRVCEJT. Decomposições. Execução Paralela. Redução de Tempo.

ABSTRACT

This article studies the use of decomposition techniques to solve the Vehicle Routing Problem with Pickup, Delivery and Time Windows (VRPPDTW). The question raised by this work is: "Despite the time savings of the decomposition methods in solving the VRPPDTW, the loss of quality stands out?". We present seven adaptations of the decompositions elaborated by Santini et al. (2023). The experiments were carried out using both the instances and the state-of-the-art solver proposed by Sartori e Buriol (2020). With the results ready, we made comparisons with those obtained by Sartori e Buriol (2020), as well as bringing out the characteristics of the decompositions. Our analysis shows that the adaptations **Barycenter Grouping 2** and **Barycenter Grouping 4** have the best performance of all those presented.

Keywords: VRP. VRPPDTW. Decompositions. Parallel Execution. Time Reduction.

SUMÁRIO

PRIMEIRA PARTE – INTRODUÇÃO GERAL

1 INTRODUÇÃO

Os problemas de roteamento de veículos (PRVs) têm sido objeto de pesquisas intensas e de rápido crescimento há mais de sessenta anos. Isso se deve à sua importância econômica e ao seu interesse teórico. O uso de rotas de veículos eficientes proporciona uma vantagem competitiva direta para as empresas de transporte, que geralmente operam com margens de lucratividade limitadas. Além disso, o fato de esses problemas compartilharem uma estrutura simples, mas rica, generalizando o problema do caixeiro viajante, ajudou a elevar a família PRV a um dos principais bancos de testes para estudos de otimização combinatória e heurística (VIDAL; LAPORTE; MATL, 2020).

Apesar dos anos de estudo em resoluções exatas para o PRV, a maior parte dos trabalhos acadêmicos abordam os métodos de resolução heurísticos. Esse fato decorre, em sua maioria, dos métodos exatos não serem robustos o suficiente para resolverem instâncias de maior porte em tempo hábil, as quais são mais comuns atualmente. Além disso, outra vantagem das heurísticas é a sua flexibilidade, possibilitando a adição de variantes que rotineiramente surgem na prática. No domínio heurístico, as técnicas de decomposição provaram seu valor na solução de instâncias de grande porte, e também são usadas com sucesso para aumentar o desempenho das heurísticas em instâncias médias. As estratégias de decomposição são amplamente usadas na prática porque geralmente levam a planos de roteamento mais estruturados e intuitivos, por exemplo, atribuindo grupos de clientes geograficamente próximos à mesma rota (SANTINI et al., 2023).

Este trabalho pretende estender as contribuições de Santini et al. (2023), que apresentou decomposições as quais produzem resultados competitivos com os algoritmos sem decomposição da literatura para Capacitated Vehicle Routing Problem (CVRP), porém, utilizaremos outra variante do VRP, o Problema de Roteamento de Veículos com Coleta, Entrega e Janelas de Tempo (PRVCEJT). O principal objetivo é responder à seguinte questão: “Apesar da economia de tempo dos métodos de decomposição na resolução do PRVCEJT, a perda de qualidade se sobressai?”. Determinado nosso objetivo, elaboramos adaptações das decomposição baseadas em rotas, que respeitam as restrições da variante do VRP escolhida. Executamos nossos experimentos utilizando tanto o solver quanto as instâncias de Sartori e Buriol (2020). Essa escolha é alicerçada no fato de que este é um solver estado da arte que soluciona o PRVCEJT e estas instâncias são até cinco vezes maiores que as utilizadas por Santini et al. (2023). Por fim, en-

contramos espaço para trabalhos futuros como, por exemplo, o desenvolvimento de mais uma decomposição, além da experimentação de outras adaptações e instâncias.

O Apêndice A, elaborado no formato de artigo de acordo com as normas da Sociedade Brasileira de Computação (SBC), traz à tona a pesquisa discutida nesta seção. Pretende-se também utilizá-lo para uma futura publicação na área de Pesquisa Operacional.

REFERÊNCIAS

SANTINI, A. et al. Decomposition strategies for vehicle routing heuristics. **INFORMS journal on computing**, INFORMS, Linthicum, v. 35, n. 3, p. 543–559, 2023. ISSN 1091-9856.

SARTORI, C. S.; BURIOL, L. S. A study on the pickup and delivery problem with time windows: Matheuristics and new instances. **Computers & Operations Research**, v. 124, p. 105065, 2020. ISSN 0305-0548. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0305054820301829>.

VIDAL, T.; LAPORTE, G.; MATL, P. A concise guide to existing and emerging vehicle routing problem variants. **European journal of operational research**, Elsevier B.V, v. 286, n. 2, p. 401–416, 2020. ISSN 0377-2217.

SEGUNDA PARTE – ARTIGO

ARTIGO 1 – Artigo segundo às normas da Sociedade Brasileira de Computação

Aplicação de Decomposições Sobre o Problema de Roteamento de Veículos com Coleta, Entrega e Janelas de Tempo

Ismael Martins Silva[1], **Mayron César de Oliveira Moreira**[1], **Dilson Lucas Pereira**[1]

[1] Departamento de Ciência da Computação
Instituto de Ciências Exatas e Tecnológicas
Universidade Federal de Lavras – Lavras, MG – Brasil

ismael.silva@estudante.ufla.br, mayron.moreira@ufla.br

dilson.pereira@ufla.br

Abstract. *This article studies the use of decomposition techniques to solve the Vehicle Routing Problem with Pickup, Delivery and Time Windows (VRPPDTW). The question raised by this work is: "Despite the time savings of the decomposition methods in solving the VRPPDTW, the loss of quality stands out?". We present seven adaptations of the decompositions elaborated by [Santini et al. 2023]. The experiments were carried out using both the instances and the state-of-the-art solver proposed by [Sartori and Buriol 2020]. With the results ready, we made comparisons with those obtained by [Sartori and Buriol 2020], as well as bringing out the characteristics of the decompositions. Our analysis shows that the adaptations* ***Barycenter Grouping 2*** *and* ***Barycenter Grouping 4*** *have the best performance of all those presented.*

Resumo. *Este artigo estuda a utilização de técnicas de decomposição sobre a resolução do Problema de Roteamento de Veículos com Coleta, Entrega e Janelas de Tempo (PRVCEJT). A questão levantada por este trabalho é: "Apesar da economia de tempo dos métodos de decomposição na resolução do PRVCEJT, a perda de qualidade se sobressai?". Nós apresentamos sete adaptações das decomposições elaboradas por [Santini et al. 2023]. Os experimentos foram realizados utilizando tanto as instâncias, quanto o solver estado da arte proposto por [Sartori and Buriol 2020]. Com os resultados prontos, realizamos comparações com os obtidos por [Sartori and Buriol 2020], além de trazer à tona as características das decomposições. Nossas análises mostram que as adaptações* ***Agrupamento de Baricentro 2*** *e* ***Agrupamento de Baricentro 4****, possuem o melhor desempenho dentre as que apresentamos.*

1. Introdução

Os problemas de roteamento de veículos (PRVs) têm sido objeto de pesquisas intensas e de rápido crescimento há mais de sessenta anos. Isso se deve à sua importância econômica e ao seu interesse teórico. O uso de rotas de veículos eficientes proporciona uma vantagem competitiva direta para as empresas de transporte, que geralmente operam com margens de lucratividade limitadas. Além disso, o fato de esses problemas compartilharem uma estrutura simples, mas rica, generalizando o problema do caixeiro viajante, ajudou

a elevar a família PRV a um dos principais bancos de testes para estudos de otimização combinatória e heurística [Vidal et al. 2020].

O PRV clássico tem como objetivo construir um conjunto de rotas de veículos, as quais visitam um conjunto de clientes, com intuito de minimizar os custos de coleta e entrega [Laporte 2007]. A partir dele o cenário de pesquisa evoluiu drasticamente nas últimas duas décadas, surgindo características como: *performance ratios*; *cumulative objectives*; *workload balance*; *person-oriented*; *navigation complexity*; *probability of failure*; *safety risks*. Até o início dos anos 2000, a maioria dos estudos metodológicos se concentrava em um subconjunto limitado de problemas operacionais com atributos como janelas de tempo, vários depósitos, vários períodos e frotas heterogêneas. Desde então, o número de variantes de problemas cresceu rapidamente, refletindo a diversidade de aplicações. Os algoritmos de roteamento de veículos não são usados apenas para produzir rotas diárias, mas também servem como ferramentas de avaliação para outras decisões estratégicas e táticas, como localização de instalações, dimensionamento de frota, produção e gerenciamento de estoque [Vidal et al. 2020].

Apesar dos anos de estudo em resoluções exatas para o PRV, a maior parte dos trabalhos acadêmicos abordam os métodos de resolução heurísticos. Esse fato decorre, em sua maioria, dos métodos exatos não serem robustos o suficiente para resolverem instâncias de maior porte em tempo hábil, as quais são mais comuns atualmente. Além disso, outra vantagem das heurísticas é a sua flexibilidade, possibilitando a adição de variantes que rotineiramente surgem na prática. No domínio heurístico, as técnicas de decomposição provaram seu valor na solução de instâncias de grande porte, e também são usadas com sucesso para aumentar o desempenho das heurísticas em instâncias médias. As estratégias de decomposição são amplamente usadas na prática porque geralmente levam a planos de roteamento mais estruturados e intuitivos, por exemplo, atribuindo grupos de clientes geograficamente próximos à mesma rota [Santini et al. 2023].

Resumidamente, uma decomposição consiste em quebrar o problema original em vários subproblemas menores, os quais são resolvidos com maior facilidade e posteriormente, cada sub-solução é agrupada, criando assim uma solução para o problema original. O artigo de [Santini et al. 2023] investiga experimentalmente o impacto de diferentes técnicas de decomposição em meta-heurísticas de última geração para o problema de roteamento de veículos capacitado (PRVC). O objetivo do PRVC é determinar até p rotas, cada rota começando no depósito, visitando uma sequência de clientes e retornando ao depósito, de forma que a demanda total de clientes em cada rota não exceda a capacidade do veículo, que cada cliente seja visitado uma vez e que o custo total de viagem seja minimizado. Os autores optaram pelo PRVC por ele ser a variante clássica da família PRV.

[Santini et al. 2023] caracteriza as técnicas de decomposição em dois grupos, homogêneas e heterogêneas. Seus experimentos são concentrados em decomposições homogêneas, por criarem subproblemas da mesma natureza que o problema original e permitem usar o mesmo solucionador para o problema principal e os subproblemas. O trabalho deles conta com as instâncias de grande escala de [Uchoa et al. 2017], que possuem diversas características (por exemplo, distribuição de clientes, comprimento da rota, posicionamento do depósito, padrões de demanda) e ainda representam um desafio para as meta-heurísticas modernas. Este trabalho adotará estas mesmas características, porém,

utilizando uma variante do PRV mais complexa e com instâncias de maior porte.

De acordo com [Sartori and Buriol 2020], uma variante do VRP é o problema de coleta e entrega, no qual os veículos transportam mercadorias de um local de coleta para um local de entrega correspondente. Outras restrições incluem a precedência da coleta sobre a entrega e a capacidade máxima dos veículos, que nunca deve ser excedida durante o transporte. Quando os locais precisam ser atendidos dentro de uma janela de tempo, o problema é conhecido como problema de roteamento de veículos com coleta, entrega e janelas de tempo (PRVCEJT). Nesse caso, os veículos não podem operar antes do início ou depois do fim da janela de tempo, que é definida para cada local do cliente.

Formalizando o problema, define-se o PRVCEJT por uma frota de veículos M, um conjunto de clientes C e um grafo direcionado $G = (V, A)$. O conjunto de vértices do grafo pode ser representado por $V = P \cup D \cup \{0\}$, onde P representa o conjunto de pontos de coleta, D o conjunto de pontos de entrega e 0 representa o depósito. O conjunto de arcos denotado por A representa as conexões entre os vértices, seja depósito ou clientes. A cada arco (i, j), onde $i \neq j$, associamos um custo c_{ij} e um tempo t_{ij}, que pode incluir o tempo de atendimento no cliente i. Cada veículo m tem uma capacidade q_m e cada cliente i uma demanda d_i. Cada cliente i tem uma janela de tempo $[a_i, b_i]$. Um veículo deve chegar ao cliente antes do tempo b_i, pode chegar antes de a_i mas o cliente não será atendido antes. O depósito também possui uma janela de tempo $[a_0, b_0]$. Os veículos não podem sair do depósito antes de a_0 e devem estar de volta antes ou no horário b_0.

Uma solução para o PRVCEJT é um conjunto $s = \{r_1, r_2, ..., r_{|s|}\}$ de rotas que atendem a todas as solicitações. Cada rota $r \in s$ é uma sequência $r = (\pi_1, \pi_2, ..., \pi_{|r|})$, em que $\pi_v \in V, v = 1, ..., |r|$ indica os locais a serem visitados, de modo que todas as janelas de tempo sejam atendidas, a capacidade máxima do veículo nunca seja excedida, cada coleta seja visitada antes do local de entrega correspondente e ambos pertençam à mesma rota. Os locais $\pi_1 = \pi_{|r|} = 0$ são o depósito. O objetivo principal é minimizar o número de rotas em $|s|$ e secundário é minimizar o custo total das rotas $C(s) = \sum_{r \in s} C(r)$, sendo $C(r) = \sum_{v=2}^{|r|} C_{\pi_{(v-1)}\pi_v}$. Vale ressaltar aqui, que cada rota é um veículo utilizado para solucionar o problema, ou seja, o número de rotas também é o número de veículos.

Introduzidos os métodos de decomposição e o PRVCEJT, este trabalho pretende estender as contribuições de [Santini et al. 2023], que apresentou decomposições as quais produzem resultados competitivos com os algoritmos sem decomposição da literatura para CVRP, porém, utilizaremos outra variante do VRP, o PRVCEJT. O principal objetivo é responder à seguinte questão: "Apesar da economia de tempo dos métodos de decomposição na resolução do PRVCEJT, a perda de qualidade se sobressai?". Determinado nosso objetivo, elaboramos adaptações das decomposição baseados em rotas, que respeitam as restrições da variante do VRP escolhida. Executamos nossos experimentos utilizando tanto o solver quanto as instâncias de [Sartori and Buriol 2020]. Essa escolha é alicerçada no fato de que este é um solver estado da arte que soluciona o PRVCEJT e estas instâncias são até cinco vezes maiores que as utilizadas por [Santini et al. 2023]. Das onze decomposições adaptadas em conjunto com Arthur Henrique Sousa Cruz, Caio de Oliveira Lopes e Mayron César de Oliveira Moreira, esse trabalho apresentará sete delas, às quais foram divididas com base na similaridade de suas características (para mais detalhes consulte a Seção 4).

O restante deste artigo está estruturado da seguinte forma. Na Seção 2 apresenta-se o funcionamento do solver utilizado, devido o mesmo ser fundamental para utilização das decomposições. Na Seção 3 é feito um *overview* do artigo de [Santini et al. 2023], o qual este trabalho se baseia. Na Seção 4, apresentam-se as principais características dos métodos de decomposição adaptados, justificando as escolhas feitas para este trabalho quanto às decomposições de instâncias. Na Seção 5, detalham-se as instâncias utilizadas no trabalho, bem como uma explicação dos testes e apresentação dos resultados obtidos, com uma comparação em relação aos resultados de [Sartori and Buriol 2020]. Por fim, na Seção 6, conclui-se o artigo, destacando também as futuras direções de pesquisa para este trabalho, e a Seção 7 fica reservada aos agradecimentos.

2. Solver para o PRVCEJT

[Sartori and Buriol 2020] estudam o PRVCEJT, estendem o algoritmo meta-heurístico proposto por [Curtois et al. 2017] e propõem um novo conjunto de instâncias para o problema. O método estendido usa os conhecidos componentes heurísticos Adaptive Guided Ejection Search (AGES) de [Curtois et al. 2017], Large Neighborhood Search (LNS) de [Ropke and Pisinger 2006] e um modelo de programação matemática (MP) do PRVCEJT como um problema de Set Partitioning (SP). O método itera através desses três componentes seguindo a abordagem do Iterated Local Search ([Lourenço et al. 2010]), recombinando rotas anteriores e novas em busca de soluções melhores.

A estrutura geral da meta-heurística é descrita no Algoritmo 1. Há vários parâmetros além da instância de entrada (Seção 4, [Sartori and Buriol 2020]). Na linha 1, uma solução inicial gulosa s é construída. Na linha 2, o conjunto de rotas, denotado como $\mathcal{P}$, é inicializado com rotas de s. O conjunto é usado no componente SP na linha 7, conforme detalhado na Seção 4.3. As variáveis $iter$ e $count$ mantêm o número total de iterações e o número de iterações sem melhoria, respectivamente. A melhor solução encontrada é indicada por s^*.

Algoritmo 1: Meta-heurística

Entrada:
Parâmetros para o AGES M_A, Z_A;
Parâmetros para a LNS M_L, L, w, $b_{min}, b_{max}, k_{min}, k_{max}$;
Parâmetros da Perturbação Z_M e μ;

1: $s \leftarrow$ solução_inicial()
2: $\mathcal{P} \leftarrow$ inicializar_conjunto(s)
3: $iter, count \leftarrow 0$
4: **repita**
5: $\quad s \leftarrow AGES(s, M_A, Z_A, \mu)$
6: $\quad s \leftarrow LNS(s, M_L, L, w, b_{min}, b_{max}, k_{min}, k_{max})$
7: $\quad s^M \leftarrow SP(s, \mathcal{P})$
8: $\quad s \leftarrow$ aceita_solução$(s, s^M, iter, count)$
9: $\quad s \leftarrow$ perturbação(s, Z_M, μ)
10: **até** condição de parada

Nas linhas 4-10, os componentes são iterados para buscar soluções aprimoradas. Primeiro, o AGES (Subseção 2.1) é aplicado para encontrar uma solução com um número reduzido de veículos. Em seguida, o LNS (Subseção 2.2) tenta reduzir o custo total da

solução atual, bem como o número de rotas. A fase SP (Subseção 2.3) adiciona as rotas de s à $\mathcal{P}$ e tenta recombinar as rotas geradas. Por fim, novas soluções são aceitas e perturbadas. Isso continua até que a condição de parada seja atingida, sendo o tempo máximo de execução, a utilizada por [Sartori and Buriol 2020].

2.1. Adaptive guided ejection search

O AGES de [Curtois et al. 2017] é o componente responsável pela minimização de veículos. Para buscar soluções com menos rotas, ele remove uma rota inteira r da solução atual e, em seguida, re-insere os pedidos de r nas rotas restantes.

2.2. Large neighborhood search

O componente LNS usado na meta-heurística é baseado no ALNS de [Ropke and Pisinger 2006] e no LNS de [Curtois et al. 2017]. Essa fase é composta de duas etapas seguidas por uma fase de aceitação da solução. Primeiro, uma série de pedidos são removidos da solução atual s, criando um conjunto de pedidos não roteados U e uma solução parcial s'. Em seguida, os pedidos em U são inseridos novamente em s', criando uma solução potencialmente diferente. Por fim, a nova solução é aceita usando a estratégia *Late Acceptance Hill Climbing* de [Burke and Bykov 2017] com uma lista de tamanho L. Esse processo é repetido por um número M_L de iterações sem melhorias.

2.3. Set Partitioning

O PRVCEJT pode ser formulado como um problema de SP na forma de MILP e resolvido de forma otimizada. [Dumas et al. 1991] apresenta uma formulação semelhante aplicada a um método de geração de colunas. [Sartori and Buriol 2020] usa um modelo semelhante, mas aplicando abordagens de geração de colunas com intuito de construir um conjunto de rotas $\mathcal{P}$ para atuar como conjunto $\mathcal{R}$. As rotas no conjunto são um subproduto da pesquisa heurística e, a cada iteração do Algoritmo 1, as rotas anteriores são recombinadas com novas rotas em busca de uma solução melhor.

2.4. Interações entre componentes

As contribuições de cada componente devem ser entendidas dentro de seu contexto no Algoritmo 1. O AGES reduz o número de veículos em uma solução, enquanto o LNS se concentra na minimização de custos e o SP intensifica essa mesma função. Por fim, o mecanismo de perturbação ([Sartori and Buriol 2020], Seção 4.5) é usado para evitar o retorno ao mesmo mínimo local, de modo que novas áreas do espaço de pesquisa sejam potencialmente exploradas.

Os resultados para os *benchmarks* padrão de [Li and Lim 2003] demonstraram que a meta-heurística de [Sartori and Buriol 2020] obteve soluções de boa qualidade em um tempo de execução razoável. Além disso, as modificações propostas proporcionaram melhor desempenho para o algoritmo de acordo com a análise de componentes. Entretanto, não foi possível alcançar valores de solução que correspondessem a todas as *best-known solutions* (BKS). No entanto, conforme verificado, nenhum método atual na literatura é capaz de atingir a qualidade BKS - e alguns valores BKS foram obtidos pela meta-heurística proposta ([Sartori and Buriol 2020]).

3. Técnicas de decomposição

O artigo de [Santini et al. 2023] tem como objetivo preencher a lacuna referente a caracterização e investigação sistemática do desempenho de diferentes técnicas de decomposição para VRPs. Eles discutem algumas características fundamentais das técnicas de decomposição heurística para VRPs e relacionam essas características a trabalhos recentes sobre esse tópico. Também investigam experimentalmente o impacto de diferentes técnicas de decomposição em meta-heurísticas de última geração para o Capacitated VRP (CVRP) a fim de derivar diretrizes metodológicas para aplicações futuras. Em seus experimentos, concentram-se em decomposições homogêneas, que criam subproblemas da mesma natureza que o problema original e permitem usar o mesmo solucionador para o problema principal e os subproblemas.

3.1. Características das técnicas de decomposição

No trabalho deles é definido que as estratégias de decomposição heurística derivam do princípio de dividir para conquistar. Elas definem um ou vários subproblemas de forma que sua solução contribua para a solução de um problema original complexo, seja produzindo uma nova solução, melhorando uma solução existente ou identificando regiões promissoras ou não promissoras do espaço de pesquisa. As técnicas de decomposição estão intimamente ligadas às estratégias de projeção ([Geoffrion 1970a; Geoffrion 1970b]), que envolvem a fixação de algumas das variáveis de decisão para concentrar a pesquisa em um subespaço menor. Com essa visão em mente, as técnicas de decomposição heurística podem ser caracterizadas em relação a:

1. a natureza dos subproblemas formados pela decomposição;
2. as informações usadas para definir os subproblemas;
3. os métodos de solução para os subproblemas;
4. as maneiras como as soluções dos subproblemas são usadas para resolver o problema principal.

3.1.1. Natureza dos subproblemas

Decomposição homogênea ou heterogênea. Uma decomposição é homogênea se levar a subproblemas que mantêm a mesma definição e estrutura do problema original. Essa decomposição permite o uso recursivo de uma única abordagem de solução. Por outro lado, uma decomposição heterogênea pode exigir técnicas personalizadas para a solução dos subproblemas.

Subproblemas independentes ou dependentes. Muitas decomposições formam vários subproblemas ao mesmo tempo. Os subproblemas são independentes quando o valor objetivo e a viabilidade de um subproblema não dependem dos outros. De modo geral, as decomposições independentes geralmente são desejáveis porque permitem o paralelismo de alto nível para reduzir o tempo de execução e permitem combinar as soluções dos subproblemas de forma direta.

3.1.2. Informações usadas para definir os subproblemas

As abordagens de decomposição geralmente são projetadas para obedecer a três princípios fundamentais:

1. As fixações de variáveis induzidas pela decomposição devem ser suportadas em soluções de alta qualidade;
2. As variáveis de decisão livres do subproblema não devem ser muito poucas ou muito numerosas, de modo que os subproblemas não sejam triviais, mas mais fáceis do que o problema original;
3. As variáveis de decisão livres devem estar relacionadas, de modo que sua solução contribua para a pesquisa.

Para atingir esses objetivos, [Santini et al. 2023] desenvolveu as estratégias a seguir.

Apoio em uma solução de elite. É comum confiar nas características de uma solução de alta qualidade (ou seja, de elite) para definir a decomposição. Por exemplo, a formação de subproblemas a partir dos clientes de um subconjunto de rotas selecionadas de uma solução de elite garante que as rotas que não estão selecionadas no momento apareçam juntas em pelo menos uma boa solução. Isso intensifica efetivamente o esforço de pesquisa em torno dessa solução.

Tamanho dos subproblemas. O tamanho dos subproblemas é um parâmetro essencial para controlar sua dificuldade e seu potencial de melhoria. Boas escolhas do tamanho do problema dependem do desempenho (tempo de execução e qualidade da solução) do algoritmo adotado para os subproblemas. O objetivo é escolher um tamanho de problema tal que o solucionador seja capaz de resolver os subproblemas de forma quase ideal em um tempo limitado.

Informações de relacionamento. Os subproblemas não devem ser triviais. Além de uma escolha adequada do tamanho do problema, a criação de problemas não triviais geralmente requer a seleção de clientes e variáveis de decisão que estejam relacionados. A relação entre as variáveis de decisão ou os clientes pode ser medida de diferentes maneiras:

- Relacionamento espacial;
- Relacionamento temporal, no caso de problemas com restrição de janela de tempo;
- Relacionamento histórico;
- Relacionamento conforme observado em padrões de soluções recentes.

As métricas de relação espacial e temporal derivam diretamente das características das instâncias e medem a proximidade no espaço ou no tempo entre duas visitas de clientes. Em contrapartida, a relação histórica e de padrão examina a frequência com que determinadas decisões são tomadas em conjunto em boas soluções no histórico de pesquisa.

3.1.3. Técnicas de solução para os subproblemas

As técnicas de solução para os subproblemas podem variar muito em termos de tempo de execução e qualidade da solução. Elas variam de heurísticas simples de reconstrução

gulosa (como é o caso da maioria dos programas de ruin-and-recreate) a meta-heurísticas mais sofisticadas aplicadas recursivamente nos subproblemas e de técnicas de enumeração especializadas a algoritmos completos de branch-cut-and-price ([Santini et al. 2023]).

3.1.4. Utilização de soluções de subproblema

Ao definir um subproblema apoiado em uma solução de elite, é possível explorar diretamente o resultado para substituir ou melhorar a solução de elite. Essa abordagem foi adotada na maioria das estratégias de decomposição heurística porque permite que se beneficie rapidamente do esforço despendido nos subproblemas. No entanto, há outras situações em que os atributos do problema tornam mais difícil explorar os resultados da decomposição ([Santini et al. 2023]).

[Santini et al. 2023] recomenda optar por decomposições **independentes** e **homogêneas** apoiadas em **soluções de elite**, pois isso facilita muito o uso do conhecimento obtido nas fases de decomposição. Seguindo essa abordagem, os estudos experimentais deles considera duas classes principais de decomposição: uma abordagem de particionamento que define subproblemas com base nas rotas de uma solução de elite (chamada de decomposição baseada em rotas) e uma abordagem de redução que fixa iterativamente as sequências de clientes (chamada de decomposição baseada em caminhos). Para cada uma dessas abordagens de decomposição, eles consideram diferentes noções de relação e diferentes definições de subproblema.

3.2. Decomposição baseada em rotas

Os métodos de decomposição de [Santini et al. 2023] baseados em rotas criam um conjunto de subproblemas independentes $\mathcal{S} = \{S_1, ..., S_k\}$, resolvem cada subproblema usando uma chamada recursiva ao solucionador e reconstroem uma solução para o problema original mesclando as soluções para os subproblemas. Cada subproblema é definido por um par $S_j = (V_j, p_j)$, em que V_j é um subconjunto de clientes e p_j é o número de veículos atribuídos ao subproblema j. Lembrando-se que as coordenadas geográficas (x_v, y_v) de cada local $v \in V$ são conhecidas. Além disso, é assumido que $(x_0, y_0) = (0, 0)$.

Eles descrevem uma solução CVRP como um conjunto de rotas $\mathcal{R} = \{R_1, ..., R_p\}$, em que cada rota é dada como uma sequência de clientes $R_i = (0, v_{i1}, ..., v_{ir_i}, 0)$. O número de clientes visitados pelo veículo i é r_i. Por conveniência, também é definido que o conjunto de clientes visitados pelo veículo i como $W_i = v_{i1}, ..., v_{ir_i}$. Se um veículo i não for usado, então $W_i = \emptyset$ e $r_i = 0$.

Em um método de decomposição baseado em rotas, o conjunto de vértices V_j de cada subproblema S_j é construído a partir de um subconjunto das rotas em uma determinada solução $\mathcal{R}$. Esse subconjunto é indexado por $\mathcal{I}_j \subseteq \{1, ..., p\}$, ou seja, $V_j = \bigcup_{i \in \mathcal{I}_j} W_i$. Define-se o número de veículos como $p_j = |\mathcal{I}_j|$ para garantir a viabilidade do subproblema S_j (desde que a solução $\mathcal{R}$ dada seja viável). Os métodos de decomposição baseados em rotas estudados diferem na forma como o conjunto de índices $\mathcal{I}_j$ é determinado.

[Santini et al. 2023] introduz o conceito de $baricentro$ de uma rota R_i não vazia,

e definido como o ponto no espaço euclidiano $\mathbb{R}^2$ com

$$b_i = (b_i^x, b_i^y) = \left(\frac{1}{\mid V_i \mid} \sum_{v \in V_i} x_v, \quad \frac{1}{\mid V_i \mid} \sum_{v \in V_i} y_v \right).$$

Ao usar os baricentros para criar subproblemas, são desconsideradas as rotas vazias. Os veículos correspondentes a essas rotas vazias, se houver, são distribuídos uniformemente entre os subproblemas criados. Para aumentar as chances de encontrar uma solução de alta qualidade para os subproblemas, é introduzido um parâmetro $m \in \mathbb{N}$ que denota um número-alvo de clientes a ser atribuído a cada subproblema.

3.3. Decomposição baseada em caminhos

Um método de decomposição baseado em caminhos cria um problema menor ao mesclar grupos de clientes. Primeiro [Santini et al. 2023] explica como mesclar os clientes que pertencem a um único caminho direcionado T. Esse caminho direcionado é definido por uma sequência de arcos consecutivos e denotado por uma sucessão ordenada dos clientes que ele visita: $T = (v_1, ..., v_{r_T})$. O objetivo é reduzir o tamanho do problema removendo todos os clientes visitados por T e substituindo-os por um único cliente, que representa todos os clientes em T visitados na ordem determinada $(v_1, ..., v_{r_T})$. O novo cliente v_T tem as seguintes características: sua demanda é a soma das demandas dos clientes que ele substitui, $q_{v_T} = \sum_{v \in T} q_v$; o custo de viagem de (para) um vértice $u \in V \setminus T$ para (de) v_T é, respectivamente, $c_{uv_T} = c_{uv_1}$ e $c_{v_T u} = c_{v_{r_T} u}$; e seu tempo de serviço é $\sum_{i=2}^{r_T} c_{v_{i-1} v_i}$.

Os métodos realizam a decomposição reduzindo todos os caminhos de um conjunto $\mathcal{T}$, resolvendo o subproblema reduzido e recuperando a solução correspondente para o problema original. O conjunto $\mathcal{T}$ contém caminhos extraídos de uma determinada solução de elite. Alguns desses caminhos podem se sobrepor ou ser consecutivos e, nesse caso, eles são mesclados. Também exclui-se os arcos de e para o depósito. Por exemplo, se os caminhos extraídos da solução forem $(0, v_1, v_2)$ e (v_2, v_3, v_4), o conjunto $\mathcal{T}$ conterá o único caminho (v_1, v_2, v_3, v_4). Um parâmetro p_L determina o comprimento dos caminhos extraídos da solução. No exemplo anterior, temos $p_L = 2$, ou seja, cada caminho contém três vértices. Observe que, quando $p_L = 1$, são extraídos arcos únicos.

O algoritmo resolve o subproblema reduzido chamando o solucionador recursivamente, ou seja, ele resolve o subproblema como um CVRP menor no qual a sequência de visitas e a direção dos clientes escolhidos são fixas. O subproblema também poderia ser descrito como um VRP agrupado capacitado ([Sevaux and Sörensen 2008; Battarra et al. 2014]), não fixando a direção e a sequência de visitas, ou como um problema de roteamento de arco capacitado ([Golden and Wong 1981]), fixando apenas a sequência de visitas. [Santini et al. 2023] decide evitar essas opções porque elas resultariam em uma decomposição heterogênea.

Para desenvolver um método de decomposição concreto, é necessário tomar duas decisões: (i) como selecionar os caminhos e (ii) quantos caminhos selecionar. Para a última decisão, sempre paramos de selecionar caminhos assim que o número de clientes no subproblema resultante atinge o número-alvo de clientes m (apresentado na Subseção 3.2). A primeira decisão, como selecionar caminhos, é determinada pelo algoritmo e cada decomposição ([Santini et al. 2023]).

4. Decomposições adaptadas

As decomposições geradas serão independentes e robustas, baseadas em soluções factíveis de boa qualidade, quanto à função objetivo. Isso possibilita ter subproblemas com mais possibilidades de respeitar as restrições do problema original, tais como precedência entre pontos de coleta e entrega, além das janelas de tempo. Essa decisão de metodologia é alicerçada em afirmações do artigo [Santini et al. 2023]. Nenhuma decomposição baseada em caminhos foi adaptada, pois, não encontramos um meio factível de implementá-lo sem violar a restrição do PRVCEJT: os respectivos pontos de coleta e entrega devem estar em uma mesma rota.

A seguir, a Tabela 1 apresenta as decomposições baseadas em rotas adaptadas do trabalho de [Santini et al. 2023], com suas respectivas técnicas de decomposição, heurísticas e variações. No presente trabalho, serão apresentadas apenas as decomposições DBS-r, DQ-r, DAB-1, DAB-2, DAB-3, DAB-4 e DAB-5. Elas foram selecionadas por suas características em comum, principalmente pela utilização da heurística *baricentro-r*, mas também pela técnica de decomposição *Agrupamento de Baricentro*. As demais decomposições ficam reservadas para outro trabalho.

Tabela 1. Decomposições adaptadas com suas respectivas características

Sigla	Decomposição	Métrica	Variação
DA	Aleatória	*baricentro-*β	Nenhuma
DBS	Baricentro Sweep	*baricentro-*β	Número de subproblemas fixo
DQ	Quadrante	*baricentro-*β	Divisão entre quadrantes
DAH	Agrupamento Hierárquico	*Agglomerative Clustering*	Número de subproblemas fixo
DBS-r	Baricentro Sweep	*baricentro-r*	Número de subproblemas fixo
DQ-r	Quadrante	*baricentro-r*	Divisão entre quadrantes
DAB-1	Agrupamento de Baricentro	*baricentro-r*	**K-Means**, com número de subproblemas fixo
DAB-2	Agrupamento de Baricentro	*baricentro-r*	**K-Means** define o número de subproblemas
DAB-3	Agrupamento de Baricentro	*baricentro-r*	**Hierárquico**, com número de subproblemas fixo
DAB-4	Agrupamento de Baricentro	*baricentro-r*	**Hierárquico**, **K-Means** define o número de subproblemas
DAB-5	Agrupamento de Baricentro	Matriz de custos	**Hierárquico**, com número de subproblemas fixo

4.1. Baricentro

Seja o conjunto de rotas $R_k = \{r_1, r_2, ..., r_{t-1}, r_t\}$ de um *input* S_k em que t é o número de veículos utilizados na solução. Considere Q_j como os pontos visitados pela rota j e P_j e D_j, como os pontos de coleta e entrega, respectivamente, da rota j. A rota Q_j é uma lista de pontos Q_j, tal que $Q_j = P_j \cup D_j$. Defini-se que o *baricentro-*β da rota

$j \in \{1, 2, ..., t-1, t\}$, é dado pela fórmula $\beta_j = (x, y)$, em que:

$$\beta_j = (x, y) = \begin{cases} x = \frac{1}{|Q_j|} \sum\limits_{u \in Q_j} x_u \\ \\ y = \frac{1}{|Q_j|} \sum\limits_{u \in Q_j} y_u \end{cases}$$

Com intuito de obter um *baricentro* que preserve a relação entre os pontos do PRVCEJT, defini-se que o *baricentro-r* da rota $j \in \{1, 2, ..., t-1, t\}$, considerando os seus respectivos pontos de coleta e entrega, é dado pela fórmula $b_j = (x, y, z, w)$, em que:

$$b_j = (x, y, z, w) = \begin{cases} x = \frac{1}{|P_j|} \sum\limits_{p \in P_j} x_p \\ y = \frac{1}{|P_j|} \sum\limits_{p \in P_j} y_p \\ z = \frac{1}{|D_j|} \sum\limits_{d \in D_j} z_d \\ w = \frac{1}{|D_j|} \sum\limits_{d \in D_j} w_d \end{cases}$$

A partir do *baricentro*, calculamos também o centro do problema, denotado como c'. O intuito dele é auxiliar as decomposições a não aglomerarem a maioria dos pontos em alguns subproblemas, enquanto os outros ficam com poucos ou sem pontos. Então, para o input S_k utilizando o *baricentro-*β, sendo $\beta = \{\beta_1, \beta_2, ..., \beta_{t-1}, \beta_t\}$ o conjunto com os *baricentros* de cada rota $r_j \in R_k$, calcula-se $c' = (x, y)$ como

$$c' = (x, y) = \begin{cases} x = \dfrac{Max(\beta^x) + Min(\beta^x)}{2} \\ \\ y = \dfrac{Max(\beta^y) + Min(\beta^y)}{2} \end{cases}$$

Para o input S_k utilizando o *baricentro-r*, sendo $b = \{b_1, b_2, ..., b_{t-1}, b_t\}$ o conjunto com os *baricentros* de cada rota $r_j \in R_k$, calcula-se $c' = (x, y, z, w)$ como

$$c' = (x, y, z, w) = \begin{cases} x = \dfrac{Max(b^x) + Min(b^x)}{2} \\ \\ y = \dfrac{Max(b^y) + Min(b^y)}{2} \\ \\ z = \dfrac{Max(b^z) + Min(b^z)}{2} \\ \\ w = \dfrac{Max(b^w) + Min(b^w)}{2} \end{cases}$$

4.2. Decomposição por Baricentro Sweep (DBS-r)

A decomposição por Baricentro Sweep de [Santini et al. 2023] primeiro classifica as rotas e, em seguida, usa a ordem resultante para criar os subproblemas. As rotas são ordenadas pelo aumento do ângulo polar $\vartheta_j \in [-\pi, \pi]$ de seu *baricentro*. Começando com $i = 1$, o subproblema $\mathcal{S}_i$ é criado adicionando (os índices de) rotas até que $\sum_{j \in \mathcal{S}_i} |r_j| \geq m$ (lembre-se de que $|r_j|$ denota o número de clientes na rota r_j). Em seguida, o subproblema $\mathcal{S}_i$ é "fechado", i é incrementado em um e o procedimento continua. Utilizando o *baricentro-r* e o centro c', com $\tau_j = \arctan|b_j^y / b_j^x|$, então o ângulo é

$$\vartheta_j = \begin{cases} \tau_j & \text{se } b_j^x \geq c'_x, b_j^y \geq c'_y \\ \pi - \tau_j & \text{se } b_j^x < c'_x, b_j^y \geq c'_y \\ -\tau_j & \text{se } b_j^x \geq c'_x, b_j^y < c'_y \\ -\pi + \tau_j & \text{se } b_j^x < c'_x, b_j^y < c'_y \end{cases}$$

A principal adaptação consiste em utilizar o *baricentro-r* e o centro c' para agrupar as rotas nos conjuntos Γ_1, Γ_2, Γ_3 e Γ_4, sendo ordenados internamente pelo ângulo ϑ_j. Os subproblemas são construídos utilizando o procedimento de [Santini et al. 2023] descrito acima, porém, as rotas são selecionadas de Γ_1, Γ_2, Γ_3 e Γ_4, nessa ordem. Os quatro conjuntos são definidos como

$$\Gamma_1 = \{r_j \in R_k; b_j^x \geq c'_x, b_j^y \geq c'_y\}, \qquad \Gamma_2 = \{r_j \in R_k; b_j^x \geq c'_x, b_j^y < c'_y\},$$
$$\Gamma_3 = \{r_j \in R_k; b_j^x < c'_x, b_j^y \geq c'_y\}, \qquad \Gamma_4 = \{r_j \in R_k; b_j^x < c'_x, b_j^y < c'_y\}.$$

4.3. Decomposição por Quadrante (DQ-r)

Esse método de [Santini et al. 2023] cria no máximo quatro subproblemas, um para cada quadrante, agrupando rotas com o baricentro no mesmo quadrante. O método não usa o parâmetro m, o que proporciona menos controle sobre o tamanho dos subproblemas criados. Além disso, para instâncias em que o depósito está no canto inferior esquerdo, o método fornece um subproblema idêntico ao problema original. Os conjuntos de índices para realizar a decomposição são os seguintes (apenas os não vazios são utilizados):

$$\mathcal{S}_1 = \{j \in R_k; b_j^x \geq 0, b_j^y \geq 0\}, \qquad \mathcal{S}_2 = \{j \in R_k; b_j^x < 0, b_j^y \geq 0\},$$
$$\mathcal{S}_3 = \{j \in R_k; b_j^x \geq 0, b_j^y < 0\}, \qquad \mathcal{S}_4 = \{j \in R_k; b_j^x < 0, b_j^y < 0\}.$$

A decomposição de quadrantes descrita acima está relacionada à usada por [Groer et al. 2011], com a diferença de que os autores dividem o plano euclidiano em duas metades em vez de quatro quadrantes. Em seguida, eles atribuem rotas a um dos dois meios planos se tiverem pelo menos um cliente no meio plano, o que leva a subproblemas sobrepostos que precisam ser resolvidos sequencialmente. Em vez disso, [Santini et al. 2023] exige que o baricentro esteja no quadrante e, portanto, seus subproblemas não se sobrepõem.

Seguindo a proposta de [Santini et al. 2023], nós utilizamos o *baricentro-r* para distribuir as rotas em um plano cartesiano adaptado. Para a adaptação do plano são utilizados os menores e maiores valores das coordenadas dos baricentros, $b_j = (x, y, z, w)$,

além de calcular o centro c' do *input* S_k. Assim como [Santini et al. 2023], descartamos os conjuntos vazios. Tendo em vista que o *baricentro-r* possui quatro coordenadas, para simular os quadrantes de um plano cartesiano, nós utilizamos as dezesseis combinações de cada coordenada para construir os subproblemas:

$$\mathcal{S}_1 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y \geq c'_y, \\ b_j^z \geq c'_z, b_j^w \geq c'_w \end{matrix} \right\}, \quad \mathcal{S}_2 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y \geq c'_y, \\ b_j^z < c'_z, b_j^w \geq c'_w \end{matrix} \right\},$$
$$\mathcal{S}_3 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y \geq c'_y, \\ b_j^z \geq c'_z, b_j^w < c'_w \end{matrix} \right\}, \quad \mathcal{S}_4 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y < c'_y, \\ b_j^z \geq c'_z, b_j^w \geq c'_w \end{matrix} \right\},$$
$$\mathcal{S}_5 = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y \geq c'_y, \\ b_j^z \geq c'_z, b_j^w \geq c'_w \end{matrix} \right\}, \quad \mathcal{S}_6 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y \geq c'_y, \\ b_j^z < c'_z, b_j^w < c'_w \end{matrix} \right\},$$
$$\mathcal{S}_7 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y < c'_y, \\ b_j^z < c'_z, b_j^w \geq c'_w \end{matrix} \right\}, \quad \mathcal{S}_8 = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y \geq c'_y, \\ b_j^z < c'_z, b_j^w \geq c'_w \end{matrix} \right\},$$
$$\mathcal{S}_9 = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y < c'_y, \\ b_j^z \geq c'_z, b_j^w < c'_w \end{matrix} \right\}, \quad \mathcal{S}_{10} = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y \geq c'_y, \\ b_j^z \geq c'_z, b_j^w < c'_w \end{matrix} \right\},$$
$$\mathcal{S}_{11} = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y < c'_y, \\ b_j^z \geq c'_z, b_j^w \geq c'_w \end{matrix} \right\}, \quad \mathcal{S}_{12} = \left\{ j \in R_k;\ \begin{matrix} b_j^x \geq c'_x, b_j^y < c'_y, \\ b_j^z < c'_z, b_j^w < c'_w \end{matrix} \right\},$$
$$\mathcal{S}_{13} = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y \geq c'_y, \\ b_j^z < c'_z, b_j^w < c'_w \end{matrix} \right\}, \quad \mathcal{S}_{14} = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y < c'_y, \\ b_j^z < c'_z, b_j^w \geq c'_w \end{matrix} \right\},$$
$$\mathcal{S}_{15} = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y < c'_y, \\ b_j^z \geq c'_z, b_j^w < c'_w \end{matrix} \right\}, \quad \mathcal{S}_{16} = \left\{ j \in R_k;\ \begin{matrix} b_j^x < c'_x, b_j^y < c'_y, \\ b_j^z < c'_z, b_j^w < c'_w \end{matrix} \right\}.$$

4.4. Decomposição por Agrupamento de Baricentro (DAB-1 à 5)

Esse método cria $k = \lceil n/m \rceil$ subproblemas, dividindo as rotas em k grupos e criando um subproblema para cada grupo. Para agrupar rotas com baricentros próximos, [Santini et al. 2023] usa o popular algoritmo *k-means* ([MacQueen et al. 1967]) usando os baricentros como pontos. Ele também utiliza o método *k-means++* ([Arthur and Vassilvitskii 2007]) para gerar o grupo inicial.

Nossas adaptações podem ser resumidas quanto aos algoritmos de agrupamento utilizados, o número de subproblemas, além é claro, da métrica utilizada como parâmetro dos algoritmos de agrupamento. Com relação aos algoritmos, são eles: *Mini Batch K-Means*, disponível em `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.MiniBatchKMeans.html`; *Hierarchical Clustering*, disponível em `https://scikit-learn.org/stable/modules/clustering.html#hierarchical-clustering`. O número de subproblemas pode ser um valor $k \in \mathbb{N}$ passado como parâmetro para a decomposição ou, ser calculado a partir dos pontos do *input* S_k utilizando o algoritmo *Mini Batch K-Means*.

Quanto às métricas, a principal é o *baricentro-r* por si só. A segunda é uma *Matriz de Custos* entre pontos de coleta, calculada a partir do *baricentro-r*. Primeiro, montamos uma matriz auxiliar M_b a partir do *baricentro-r*. Para $i \in \{1, 2, ..., t-1, t\}$ e

$j \in \{1, 2, ..., t-1, t\}$, $M_b(i, j)$ é dado por

$$M_b = \begin{cases} 0 & \text{para } i = j \text{ e } i + t = j + t \\ \sqrt{(b_j^x - b_i^x)^2 + (b_j^y - b_i^y)^2} & \text{para } (i, j) \text{ e } (j, i) \\ \sqrt{(b_j^z - b_i^x)^2 + (b_j^w - b_i^y)^2} & \text{para } (i, j + t) \text{ e } (j + t, i) \\ \sqrt{(b_j^x - b_i^z)^2 + (b_j^y - b_i^w)^2} & \text{para } (i + t, j) \text{ e } (j, i + t) \\ \sqrt{(b_j^z - b_i^z)^2 + (b_j^w - b_i^w)^2} & \text{para } (i + t, j + t) \text{ e } (j + t, i + t) \\ \sqrt{(b_i^z - b_i^x)^2 + (b_i^w - b_i^y)^2} & \text{para } (i, i + t) \text{ e } (i + t, i) \end{cases}$$

Seja U a concatenação dos respectivos pontos de coleta e entrega, define-se $u_i = (p_i, d_i)$, com $u \in U$, $p \in P$ e $d \in D$. Como estamos utilizando a matriz auxiliar, os pontos (p_i, d_i) de u_i correspondem com os índices $(i, i+t)$ de M_b. Sendo assim, o custo $C(u_i, u_j)$ entre os pontos de coleta u_i e u_j, para $i \in \{1, 2, ..., t-1, t\}$ e $j \in \{1, 2, ..., t-1, t\}$ é dado por

$$C(u_i, u_j) = \begin{cases} 0 & \text{se } i = j \\ C(p_i, d_i, p_j, d_j) & \text{se } i \neq j \end{cases}$$

$$C(p_i, d_i, p_j, d_j) = C(i, i + t, j, j + t)$$

$$C(i, i + t, j, j + t) = min \begin{pmatrix} M_b(i, j), & M_b(i, j + t), \\ M_b(i + t, j), & M_b(i + t, j + t), \\ M_b(j, i), & M_b(j, i + t), \\ M_b(j + t, i), & M_b(j + t, i + t) \end{pmatrix}$$

Definidos os algoritmos de agrupamento, número de pontos e as métricas, são apresentadas as cinco configurações das decomposições por Agrupamento de Baricentro:

1. **DAB-1:** Utiliza a métrica *baricentro-r*, com número de subproblemas k definido a priori e o algoritmo *Mini Batch K-Means*;
2. **DAB-2:** Difere da **DAB-1** apenas no número de pontos, os quais são calculados utilizando o *Mini Batch K-Means*;
3. **DAB-3:** Utiliza a métrica *baricentro-r*, com número de subproblemas k definido a priori e o algoritmo *Hierarchical Clustering*;
4. **DAB-4:** Difere da **DAB-3** apenas no número de pontos, os quais são calculados utilizando o *Mini Batch K-Means*;
5. **DAB-5:** Utiliza a métrica *Matriz de Custos*, com número de subproblemas k definido a priori e o algoritmo *Hierarchical Clustering*.

5. Testes computacionais

Os testes foram executados em um computador com uma CPU Intel Core i7-8700 3.40 GHz x 12, 16GB de RAM e Ubuntu 20.04.4. Todos os métodos de decomposição foram implementados em Python 3.10 e o solver utilizado foi o algoritmo heurístico estado da arte do PRVCEJT, proposto por [Sartori and Buriol 2020], que possui implementação na linguagem C++, disponibilizada pelo próprio autor [Sartori 2022]. O critério de parada do mesmo é o tempo máximo de execução, o qual se estende aos métodos de decomposição.

Na Subseção 5.1 são descritas as instâncias utilizadas para os testes, advindas do trabalho de [Sartori and Buriol 2020]. A Subseção 5.2 apresenta uma comparação entre os valores das soluções obtidas aplicando os métodos de decomposição em relação aos resultados obtidos por [Sartori and Buriol 2020] em sua pesquisa, levando em conta custo, número de veículos e tempo de execução utilizado na resolução do problema, além de informações importantes das decomposições. E por fim, a Subseção 5.3 trará uma análise dos resultados obtidos.

5.1. Instâncias e tempo de execução

As instâncias utilizadas foram extraídas de [Sartori and Buriol 2020], as quais podem ser divididas em quatro tipos, sendo eles:

- "bar": Instâncias geradas na cidade de Barcelona.
- "ber": Instâncias geradas na cidade de Berlim.
- "nyc": Instâncias geradas na cidade de Nova York.
- "poa": Instâncias geradas na cidade de Porto Alegre.

Cada um desses tipos possui instâncias com diferentes números de pontos, variando entre os valores de 100, 200, 400, 600, 800, 1000, 1500, 2000, 2500, 3000, 4000 ou 5000 pontos. Foram selecionadas aleatoriamente quatro instâncias de cada tipo e tamanho e, em cada uma delas, foram aplicados os sete métodos de decomposição de pontos apresentados neste trabalho. Em suma, para cada decomposição e cada tamanho de instância, foram executados 16 testes. Com relação ao tempo máximo de execução, ele também foi extraído de [Sartori and Buriol 2020], os quais estão associados exclusivamente ao número de pontos das instâncias:

- 300 segundos: Instâncias com 100 pontos.
- 900 segundos: Instâncias com 200 ou 400 pontos.
- 1800 segundos: Instâncias com 600 pontos.
- 3600 segundos: As demais instâncias.

5.2. Testes

Os testes foram realizados utilizando como critério de parada os tempos de execução apresentados, assim como em [Sartori and Buriol 2020]. Porém, com a diferença a seguir: seria descontado do tempo máximo de execução de cada instância o tempo utilizado para a decomposição da mesma, de forma a já considerar a decomposição como parte do processo de resolução das instâncias, em termos de tempo. Além disso, o tempo restante foi dividido igualmente entre os subproblemas e os mesmos também foram executados em paralelo para melhor aproveitar os recursos computacionais disponíveis, que por consequência também reduz o tempo de execução, de forma a acelerar os testes para que fosse possível a obtenção de resultados em um tempo hábil.

Apesar de usarmos uma semente para o controle de eventos aleatórios no algoritmo e para propiciar maior reprodutibilidade dos testes, o algoritmo de [Sartori and Buriol 2020] ainda possui fatores estocásticos. Para contornar essa questão e realizar uma análise mais concreta, optou-se por executar cada um dos testes selecionados 5 vezes, assim como feito por [Sartori and Buriol 2020]. Dessa forma, tanto o valor em custo quanto o número de veículos da solução de uma instância seriam considerados como a média dentre as cinco execuções da mesma. Além dos arquivos de

saída gerados pelas decomposições (subproblemas) e das saídas geradas pelo algoritmo de [Sartori and Buriol 2020], o algoritmo que executa as decomposições gera, para todas as instâncias, uma imagem com os pontos exibidos em um plano cartesiano, de forma a facilitar a visualização do resultado das decomposições.

As Figuras 1, 2, 3, 4, 5, 6, 7 e 8 abaixo mostram algumas dessas imagens. As Figuras 1 e 2 tratam-se de duas instâncias distintas, porém, com a mesma decomposição. Já as Figuras 2, 3, 4, 5, 6, 7 e 8 referem-se à mesma instância, porém, sendo submetida a decomposições distintas.

Figura 1. Decomposição por Baricentro Sweep, instância "ber-n200-4"

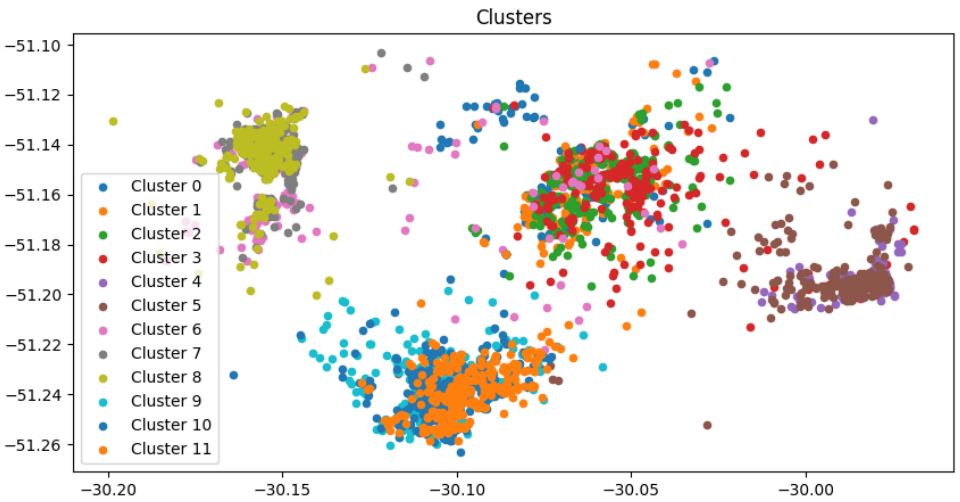

Figura 2. Decomposição por Baricentro Sweep, instância "poa-n3000-3"

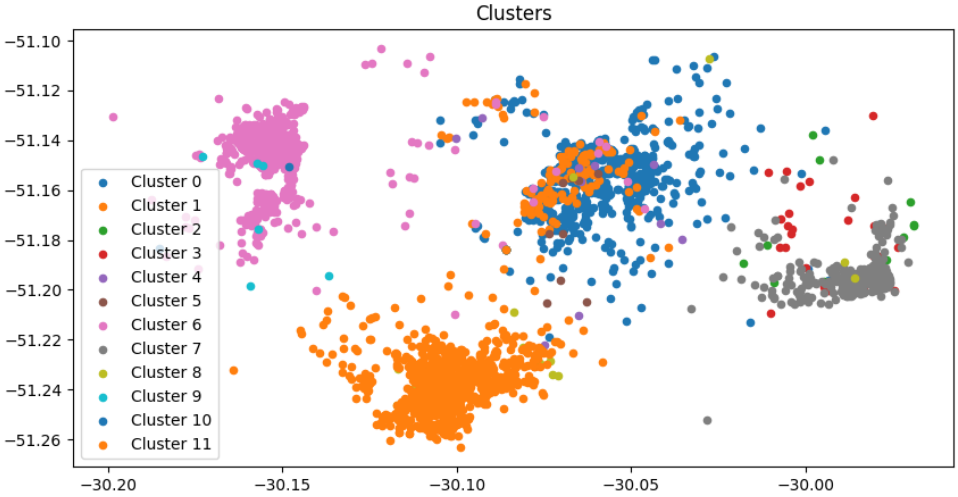

Figura 3. Decomposição por Quadrante, instância "poa-n3000-3"

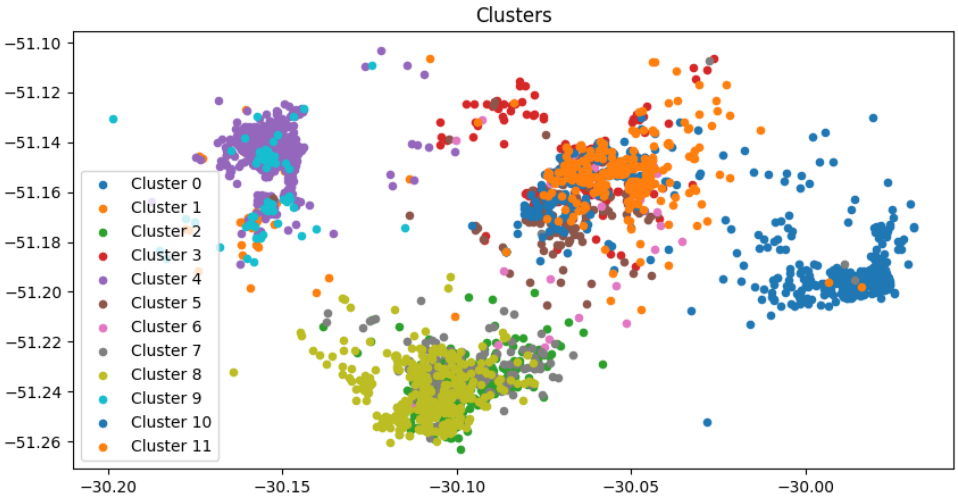

Figura 4. Decomposição por Agrupamento de Baricentro 1, instância "poa-n3000-3"

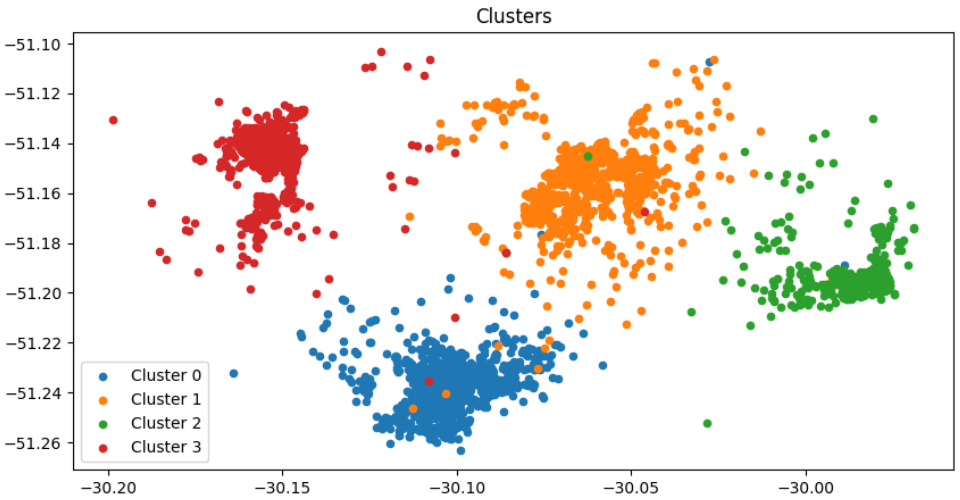

Figura 5. Decomposição por Agrupamento de Baricentro 2, instância "poa-n3000-3"

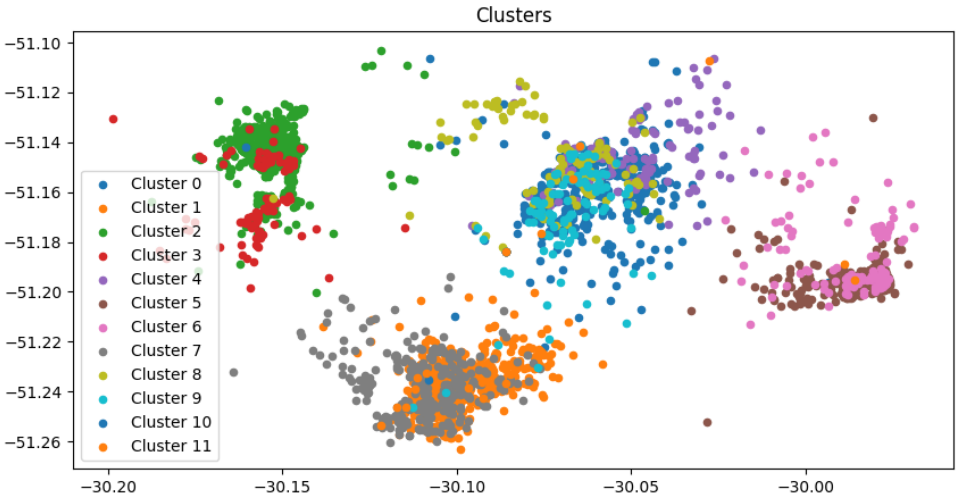

Figura 6. Decomposição por Agrupamento de Baricentro 3, instância "poa-n3000-3"

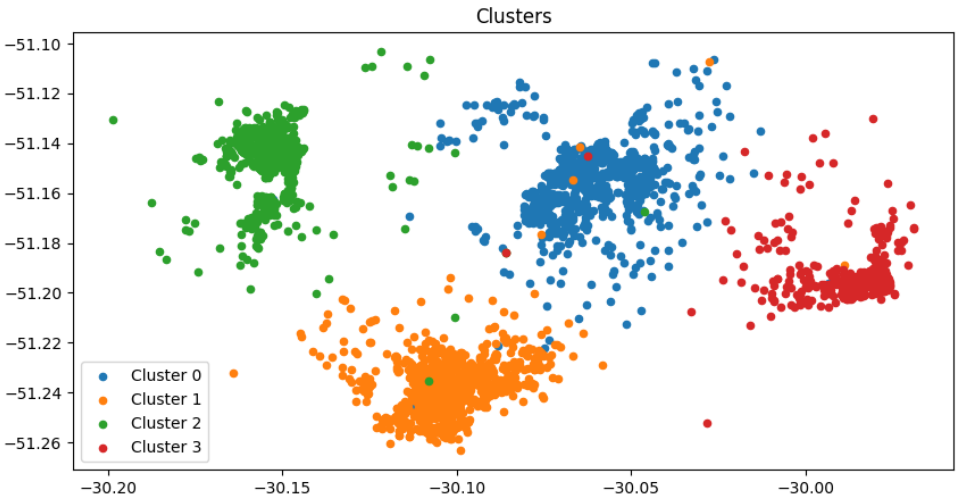

Figura 7. Decomposição por Agrupamento de Baricentro 4, instância "poa-n3000-3"

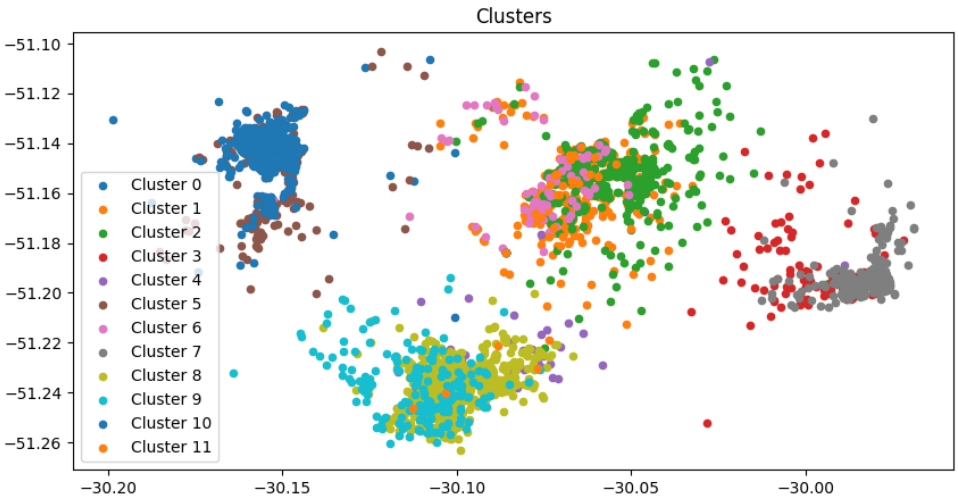

Figura 8. Decomposição por Agrupamento de Baricentro 5, instância "poa-n3000-3"

A seguir, as Tabelas 2, 3, 4, 5 e 6, referentes aos gráficos das Figuras 9, 10, 11, 12 e 13, respectivamente, apresentam os resultados obtidos neste trabalho comparados com os obtidos por [Sartori and Buriol 2020], assim como valores importantes das decomposições. As comparações levam em consideração o custo das soluções e o número de veículos. Como o tempo de execução de [Sartori and Buriol 2020] é o tempo máximo de execução, utilizamos as duas métricas de tempo a seguir: tempo total utilizado; e tempo de decomposição. Além desses valores, também trazemos o número de subproblemas ge-

rados pelas decomposições, devido à sua influência na qualidade da solução e no tempo de execução. Sobre as tabelas, temos os seguintes aspectos:

- **n** é o número de pontos das instâncias;
- $\mathbf{c}_{sb}$ é a média do custo das instâncias de [Sartori and Buriol 2020];
- $\mathbf{v}_{sb}$ é a média do número de veículos das instâncias de [Sartori and Buriol 2020];
- $\mathbf{t}_{max}$ é o tempo máximo de execução;
- k é o número de subproblemas definido a priori;
- **DBS-r**, **DQ-r**, **DAB-1**, **DAB-2**, **DAB-3**, **DAB-4** e **DAB-5** representam as decomposições **Decomposição por Baricentro Sweep**, **Decomposição por Quadrante**, **Decomposição por Agrupamento de Baricentro 1**, **Decomposição por Agrupamento de Baricentro 2**, **Decomposição por Agrupamento de Baricentro 3**, **Decomposição por Agrupamento de Baricentro 4** e **Decomposição por Agrupamento de Baricentro 5**, respectivamente;
- **%** indica que o valor das células da coluna é em porcentagem.

O valor de melhora/piora da Tabela 2 é calculado da seguinte forma. Seja $\mathbf{c}_d(n)$ a média dos custos da decomposição d, e $\mathbf{c}_{sb}(n)$ a média do custo das instâncias de [Sartori and Buriol 2020], ambos para as instâncias de tamanho n, calculamos:

$$\delta\mathbf{c}_d(n) = 1 - \frac{\mathbf{c}_d(n)}{\mathbf{c}_{sb}(n)}$$

$\delta\mathbf{c}_d(n)$ maior do que zero indica uma melhora no custo obtido pela decomposição d para as instâncias de tamanho n, em comparação com o valor obtido por [Sartori and Buriol 2020]. Já valores menores do que zero indicam uma piora no custo.

Tabela 2. Custo

n	$\mathbf{c}_{sb}$	DBS-r %	DQ-r %	DAB-1 %	DAB-2 %	DAB-3 %	DAB-4 %	DAB-5 %
100	1014,20	**3,88**	-5,37	**4,16**	-0,53	**6,31**	-0,92	**6,27**
200	1856,92	-13,76	-17,60	-11,33	-11,41	-11,63	-11,55	-12,85
400	3395,44	-19,93	-18,21	-15,66	-12,05	-16,82	-11,86	-17,63
600	4856,68	-14,41	-9,65	-10,61	-6,14	-12,95	-5,58	-13,16
800	6523,16	-0,75	**4,90**	**2,91**	**7,03**	**1,43**	**7,07**	**0,77**
1000	9006,36	-2,36	**1,08**	**0,91**	**5,18**	-0,39	**5,55**	-0,83
1500	12087,64	-0,39	**2,04**	**1,92**	**5,35**	**1,21**	**5,62**	**1,04**
2000	16833,24	-4,67	-2,05	-1,88	**1,33**	-2,71	**1,58**	-2,64
2500	20294,56	-6,00	-4,48	-3,98	-1,48	-4,42	-1,26	-4,53
3000	24728,40	-7,93	-7,35	-6,25	-3,82	-6,67	-3,90	-6,96
4000	34402,44	**12,87**	**13,22**	**13,72**	**15,38**	**13,84**	**15,35**	**13,67**
5000	42829,28	**14,49**	**13,42**	**13,56**	**15,89**	**14,72**	**16,27**	**14,75**

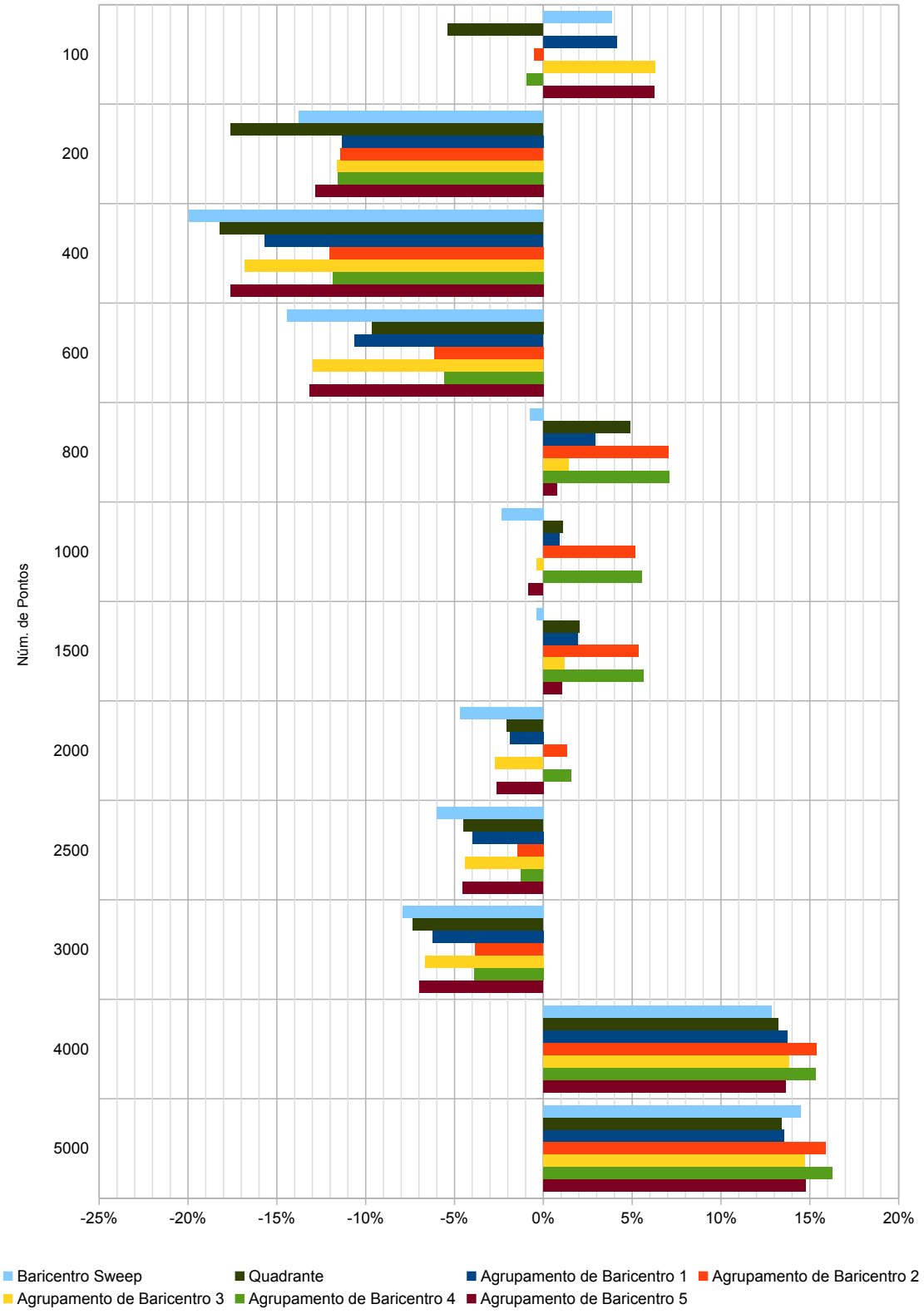

Figura 9. Gráfico do *Custo*

O valor de melhora/piora da Tabela 3 é calculado da seguinte forma. Seja $\mathbf{v}_d(n)$ a média dos números de veículos da decomposição d, e $\mathbf{v}_{sb}(n)$ a média do números de veículos das instâncias de [Sartori and Buriol 2020], ambos para as instâncias de tamanho n, calculamos:

$$\delta\mathbf{v}_d(n) = 1 - \frac{\mathbf{v}_d(n)}{\mathbf{v}_{sb}(n)}$$

$\delta\mathbf{v}_d(n)$ maior do que zero indica uma melhora no número de veículos obtido pela decomposição d para as instâncias de tamanho n, em comparação com o valor obtido por [Sartori and Buriol 2020]. Já valores menores do que zero indicam uma piora no número de veículos.

Tabela 3. Número de Veículos

n	$\mathbf{v}_{sb}$	**DBS-r** **%**	**DQ-r** **%**	**DAB-1** **%**	**DAB-2** **%**	**DAB-3** **%**	**DAB-4** **%**	**DAB-5** **%**
100	6,56	**1,87**	-9,57	**7,96**	**0,53**	**6,44**	-3,09	**6,63**
200	13,40	-11,01	-18,00	-7,46	-8,77	-9,61	-10,17	-8,40
400	23,56	-26,54	-25,74	-23,62	-19,00	-24,04	-18,85	-23,73
600	33,60	-13,50	-9,93	-11,16	-4,69	-13,65	-4,43	-14,10
800	45,92	-13,16	-8,50	-11,28	-4,67	-12,45	-5,46	-12,48
1000	55,92	-7,09	-4,35	-4,30	**1,49**	-5,93	**1,09**	-4,90
1500	84,52	**1,00**	**2,85**	**2,17**	**6,13**	**1,06**	**6,01**	**1,43**
2000	116,56	**4,85**	**5,92**	**5,85**	**9,35**	**4,34**	**8,95**	**5,24**
2500	127,80	-9,18	-8,56	-8,38	-5,05	-9,27	-5,39	-9,03
3000	170,80	-9,98	-10,42	-9,95	-6,80	-10,71	-7,40	-9,96
4000	237,08	**3,95**	**4,01**	**4,01**	**5,74**	**3,58**	**5,25**	**4,26**
5000	269,88	**12,43**	**11,67**	**12,33**	**14,24**	**12,10**	**14,11**	**12,29**

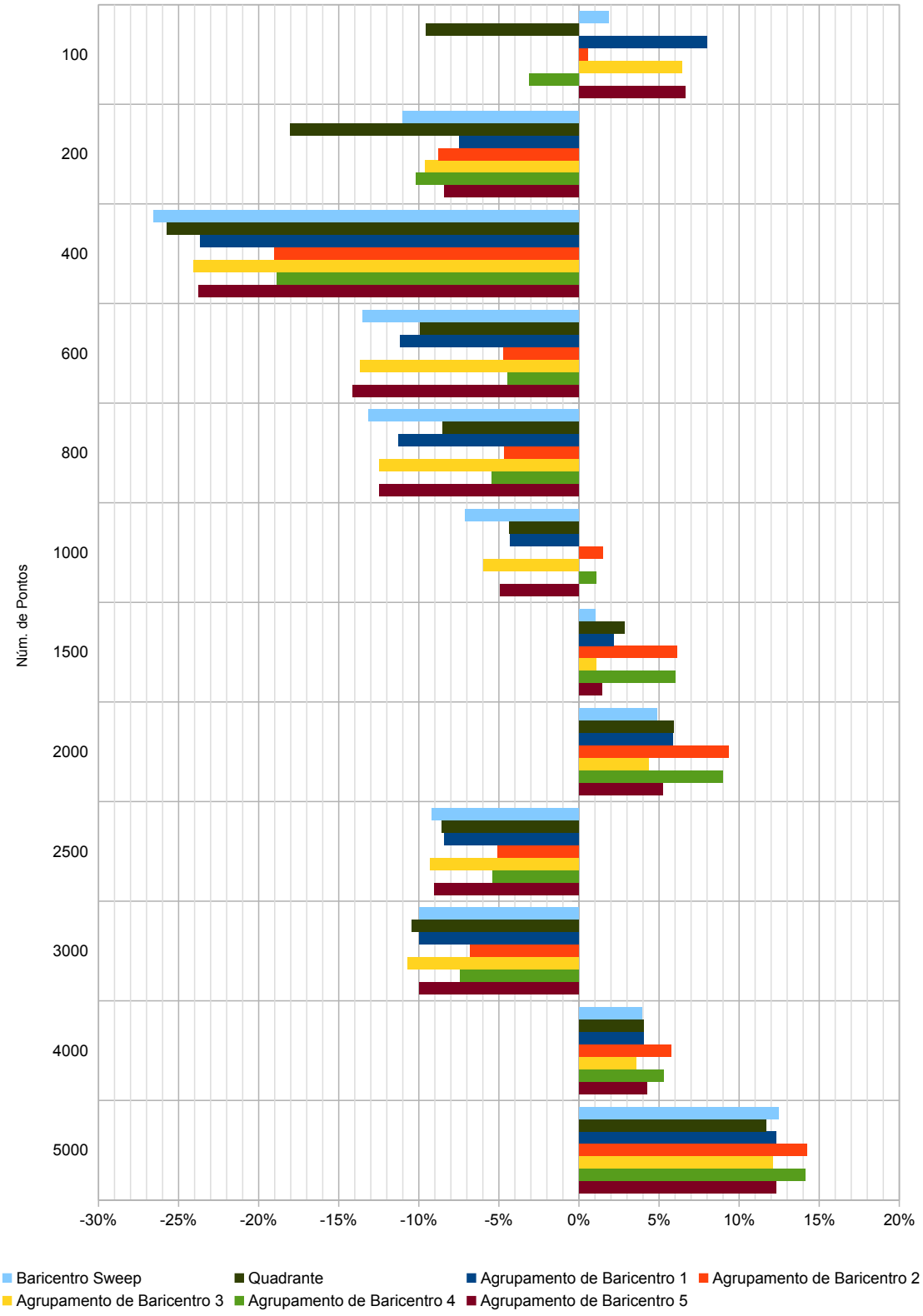

Figura 10. Gráfico do *Número de Veículos*

O valor de tempo de execução da Tabela 4 é calculado da seguinte forma. Seja $\mathbf{t}_{exec,d}(n)$ a média dos tempos de execução da decomposição d, e $\mathbf{t}_{max}(n)$ o tempo máximo de execução, ambos para as instâncias de tamanho n, calculamos:

$$\mathbf{t}_{uti,d}(n) = \frac{\mathbf{t}_{exec,d}(n)}{\mathbf{t}_{max}(n)}$$

$\mathbf{t}_{uti,d}(n)$ representa o percentual do tempo máximo de execução utilizado pela decomposição d para as instâncias de tamanho n, para fase de decomposição mais a fase de solução.

Tabela 4. Tempo de Execução

n	$\mathbf{t}_{max}$	**DBS-r** **%**	**DQ-r** **%**	**DAB-1** **%**	**DAB-2** **%**	**DAB-3** **%**	**DAB-4** **%**	**DAB-5** **%**
100	300	50,11	25,08	55,77	37,74	50,13	29,52	50,13
200	900	27,13	17,95	31,49	29,89	25,15	25,97	25,11
400	900	14,45	14,77	16,90	27,72	12,74	24,56	12,72
600	1800	10,24	16,40	11,26	23,62	8,41	22,71	8,41
800	3600	9,33	20,39	11,09	26,31	8,49	24,95	8,48
1000	3600	8,95	13,62	10,47	24,41	8,44	23,79	8,44
1500	3600	8,90	14,89	10,16	23,69	8,55	23,20	8,55
2000	3600	8,95	14,09	10,42	23,19	8,67	21,67	8,62
2500	3600	9,05	14,54	9,95	20,69	8,72	19,74	8,74
3000	3600	9,02	12,76	9,84	21,73	8,82	20,91	8,83
4000	3600	9,05	14,44	10,04	24,66	8,99	23,83	9,02
5000	3600	9,25	14,54	9,86	20,53	9,27	20,13	9,31

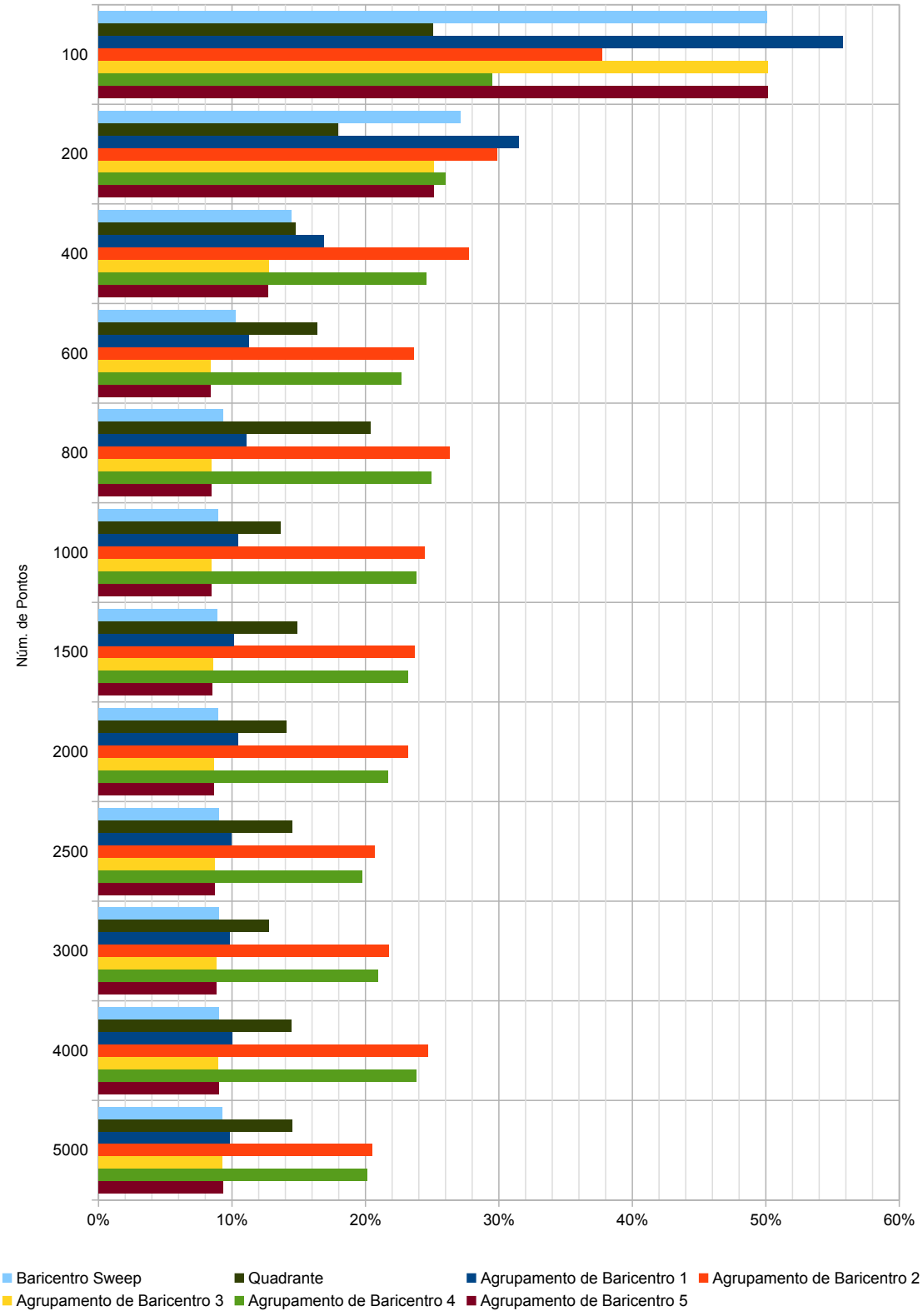

Figura 11. Gráfico do *Tempo de Execução*

O valor de tempo de decomposição da Tabela 5 é calculado da seguinte forma. Seja $\mathbf{t}_{dec,d}(n)$ a média dos tempos de decomposição da decomposição d, e $\mathbf{t}_{max}(n)$ o tempo máximo de execução, ambos para as instâncias de tamanho n, calculamos:

$$\mathbf{t'}_{dec,d}(n) = \frac{\mathbf{t}_{dec,d}(n)}{\mathbf{t}_{max}(n)}$$

$\mathbf{t'}_{dec,d}(n)$ representa o percentual do tempo máximo de execução utilizado pela decomposição d para as instâncias de tamanho n, na fase de decomposição.

Tabela 5. Tempo de Decomposição

n	$\mathbf{t}_{max}$	**DBS-r** **%**	**DQ-r** **%**	**DAB-1** **%**	**DAB-2** **%**	**DAB-3** **%**	**DAB-4** **%**	**DAB-5** **%**
100	300	0,08	0,09	0,11	0,13	0,08	0,13	0,08
200	900	0,03	0,04	0,05	0,05	0,04	0,05	0,04
400	900	0,06	0,06	0,06	0,07	0,06	0,07	0,06
600	1800	0,04	0,05	0,05	0,06	0,04	0,06	0,04
800	3600	0,03	0,04	0,03	0,04	0,03	0,04	0,03
1000	3600	0,04	0,05	0,05	0,06	0,04	0,06	0,04
1500	3600	0,08	0,10	0,09	0,11	0,09	0,11	0,09
2000	3600	0,14	0,16	0,16	0,18	0,15	0,18	0,15
2500	3600	0,21	0,26	0,24	0,27	0,22	0,26	0,23
3000	3600	0,30	0,37	0,34	0,38	0,33	0,38	0,34
4000	3600	0,52	0,64	0,59	0,68	0,56	0,67	0,59
5000	3600	0,81	1,04	0,91	1,18	0,87	1,17	0,89

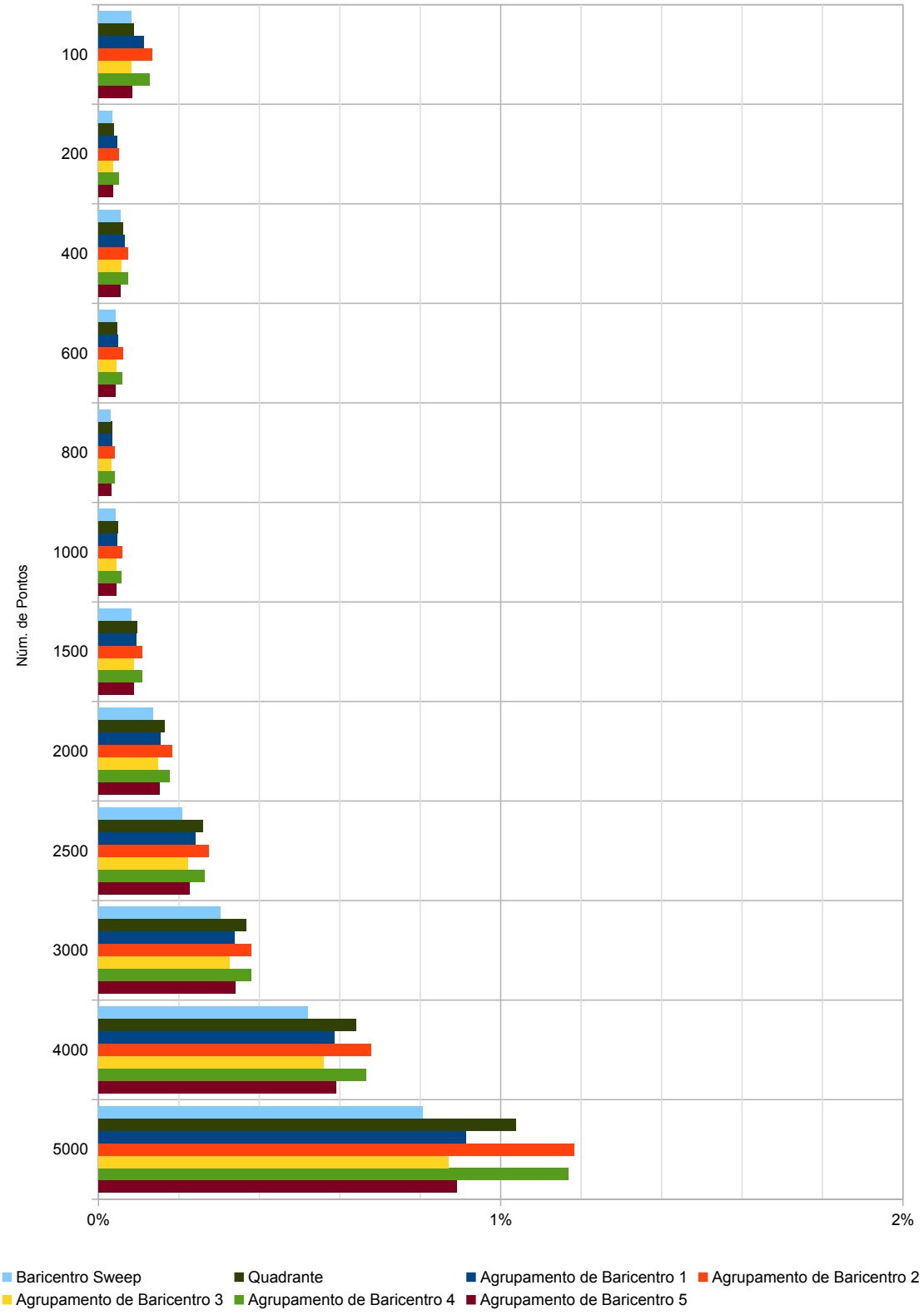

Figura 12. Gráfico do *Tempo de Decomposição*

O número de subproblemas da Tabela 6 é a média de cada decomposição *d*, para as instâncias de tamanho n.

Tabela 6. Número de Subproblemas

n	k	**DBS-r**	**DQ-r**	**DAB-1**	**DAB-2**	**DAB-3**	**DAB-4**	**DAB-5**
100	2	2,00	4,94	1,89	2,94	2,00	3,59	2,00
200	4	3,75	7,81	3,54	3,64	4,00	4,06	4,00
400	8	7,19	9,38	6,66	3,95	8,00	4,39	8,00
600	12	10,00	8,88	9,45	4,53	12,00	4,75	12,00
800	12	11,00	8,06	9,83	4,26	12,00	4,64	12,00
1000	12	11,38	10,31	10,10	4,44	12,00	4,58	12,00
1500	12	11,50	10,50	10,40	4,65	12,00	4,83	12,00
2000	12	11,56	11,69	10,47	4,80	12,00	5,11	12,00
2500	12	11,56	11,25	10,76	5,26	12,00	5,58	12,00
3000	12	11,81	12,44	10,91	5,15	12,00	5,40	12,00
4000	12	11,88	11,06	10,95	5,01	12,00	5,25	12,00
5000	12	11,94	11,63	11,36	5,59	12,00	5,71	12,00

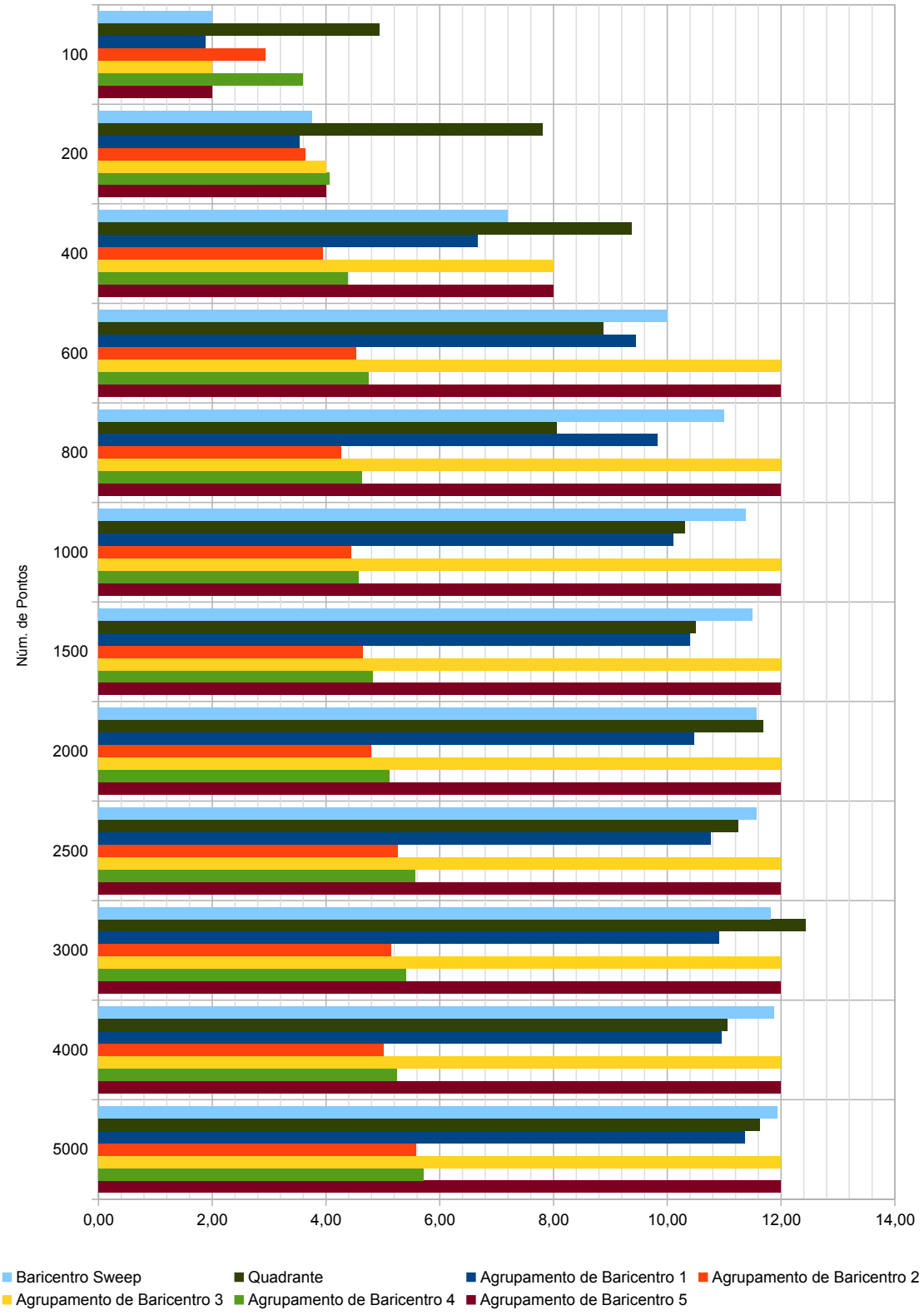

Figura 13. Gráfico do *Número de Subproblemas*

A seguir, as Figuras 14, 15, 16, 17 e 18 resumem os dados das Tabelas 2, 3, 4, 5 e 6, respectivamente, por meio da representação gráfica de *boxplot*. Neste tipo de gráfico, círculos são *outliers* (valores destoantes), o traço na parte superior da haste vertical representa o ponto de máximo (desconsiderando os *outliers*), o traço na parte inferior da haste representa o ponto de mínimo (desconsiderando os *outliers*), a base da caixa retangular é o primeiro quartil, a linha dentro da caixa é a segundo quartil (mediana) e o topo da caixa é o terceiro quartil. O primeiro quartil marca o ponto no qual 25% dos valores estão abaixo e os outros 75% estão acima, o segundo quartil marca o ponto no qual 50% dos valores estão abaixo e os outros 50% estão acima, e o terceiro quartil marca o ponto no qual 75% dos valores estão abaixo e os outros 25% estão acima.

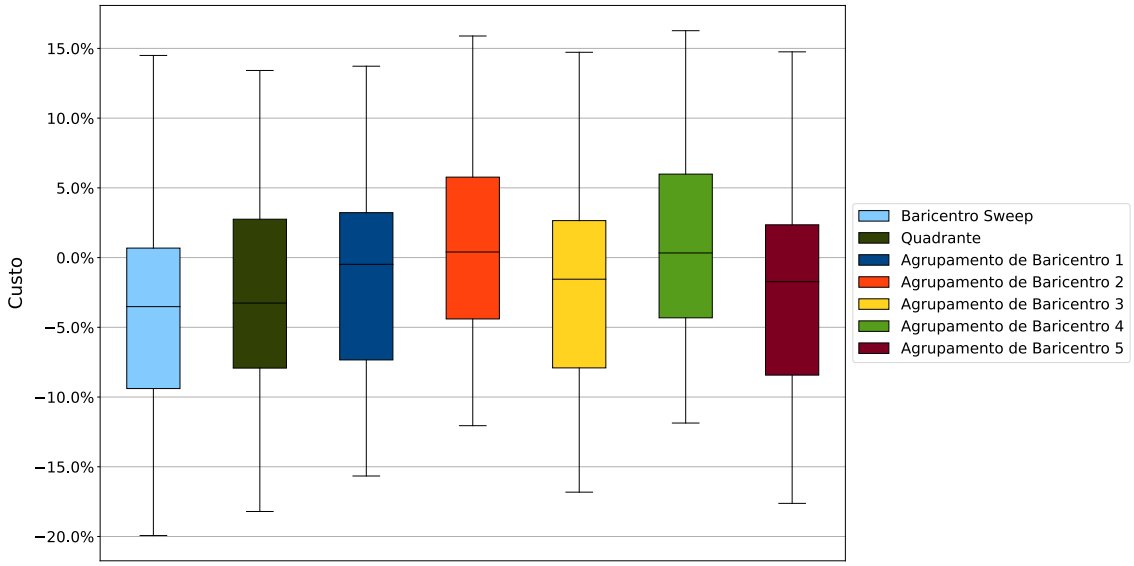

Figura 14. Gráfico resumo do *Custo*

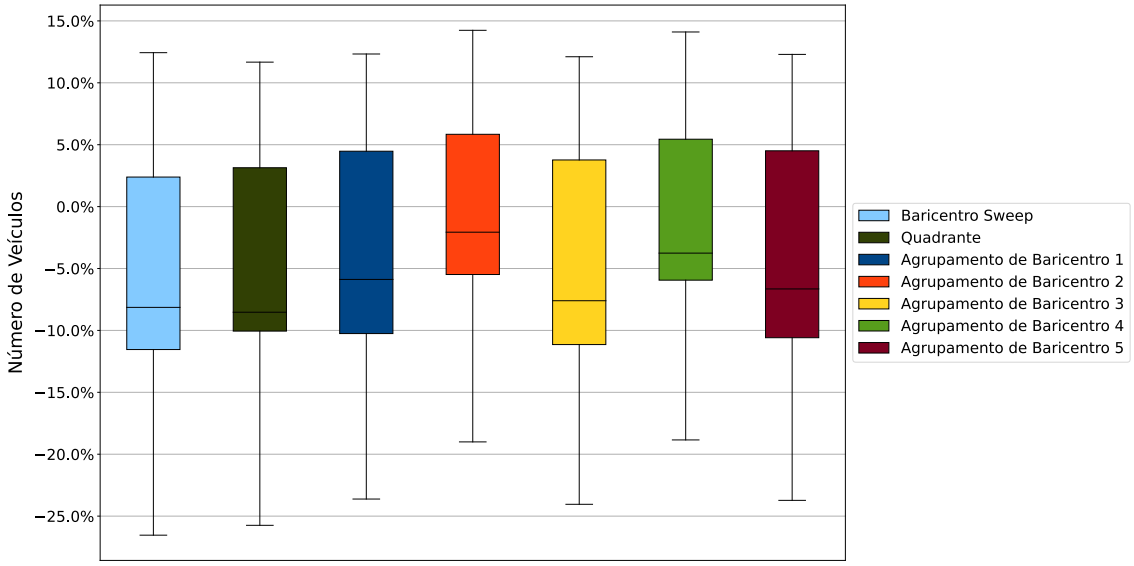

Figura 15. Gráfico resumo do *Número de Veículos*

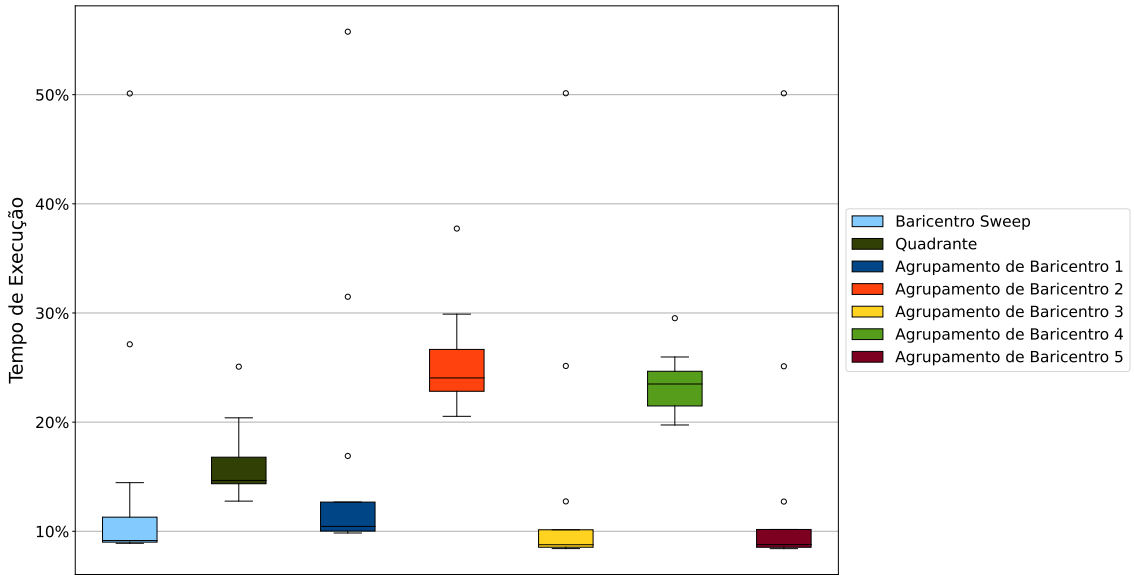

Figura 16. Gráfico resumo do *Tempo de Execução*

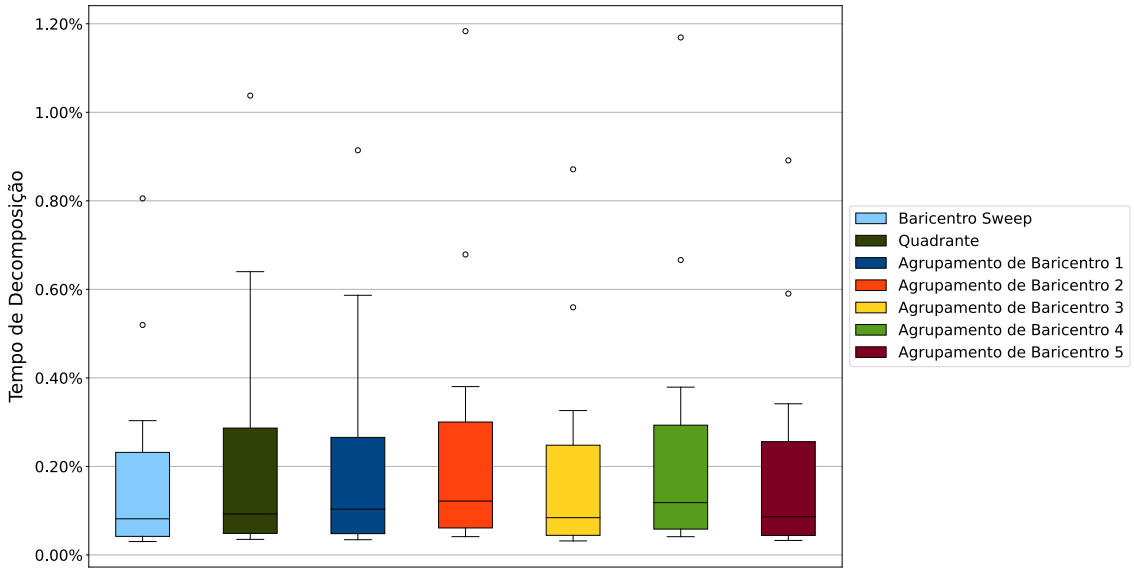

Figura 17. Gráfico resumo do *Tempo de Decomposição*

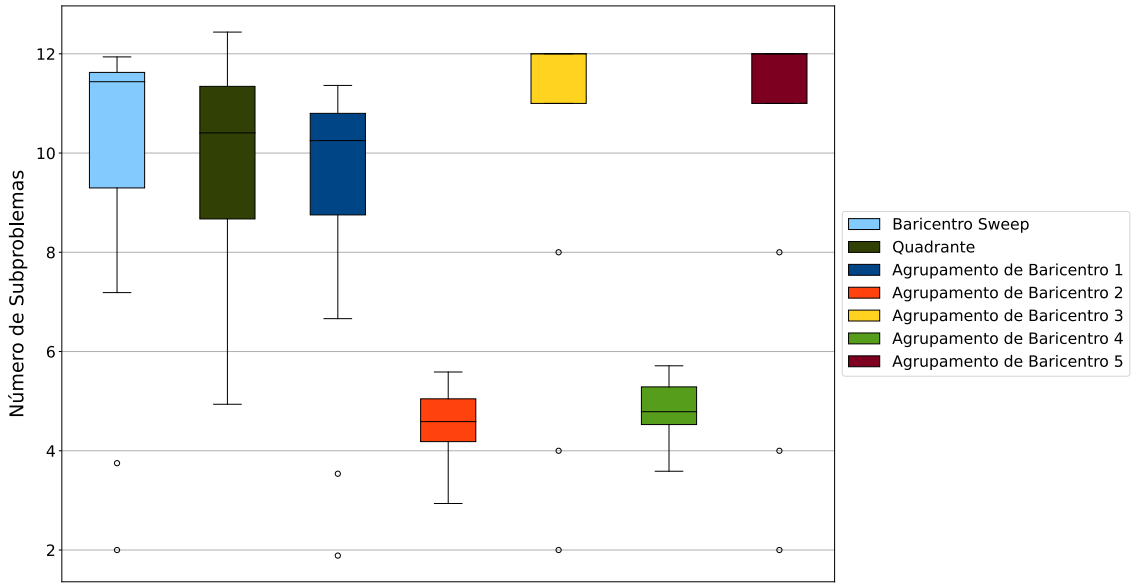

Figura 18. Gráfico resumo do *Número de Subproblemas*

5.3. Análise dos Testes

Com relação ao custo, observando os dados das tabelas e gráficos, podemos dizer que as decomposições apresentaram resultados promissores, principalmente levando em consideração o tempo utilizado para solucionar o problema. A piora aproxima-se de 20%, lembrando que esse valor está concentrado nas instâncias menores, como as de 400 pontos, já para instâncias maiores, como as de 3000 pontos, a piora não atingiu 8%. A melhora nas instâncias com até 2000 pontos chega na casa dos 7%, porém, para as instâncias com 4000 ou mais pontos, a melhora supera os 12%.

Com relação ao número de veículos, observamos algo similar ao que ocorreu com o custo. A piora varia de 3% à 26%, sendo que os piores valores são observados principalmente nas instâncias menores, como as de 400 pontos. A melhora varia de 0,5% à 14%, entretanto, os melhores valores estão concentrados nas maiores instâncias, como as de 5000 pontos. Com relação a utilização do tempo máximo de execução, os maiores valores chegam à 50% parando na casa dos 55%, sendo que esses estão concentrados nas menores instâncias, as de 100 pontos. Para as demais instâncias, o tempo utilizado não excede os 29%, chegando na casa dos 9% até mesmo para as maiores instâncias, como as de 5000 pontos.

Colocando as decomposições lado-a-lado, temos que as decomposições **Agrupamento de Baricentro 2** e **Agrupamento de Baricentro 4** são as que apresentam maior melhora e menor piora, tanto para o custo, quanto para o número veículos. Porém, são as que utilizam mais do tempo máximo de execução, devido o número de subproblemas criados pelas mesmas ser bem menor, se comparado com as outras decomposições, principalmente nas instâncias maiores. Isso nos mostra que o tempo extra utilizado na fase de decomposição para calcular o número de subproblemas k, utilizando o *Mini Batch K-Means*, é benéfico para a decomposição, pois quanto menor for o número de subproblemas maior será a parcela do tempo restante dividido igualmente para cada um, dando

assim mais tempo para o solver otimizar a solução.

Resgatando a questão lançada no início deste artigo: "Apesar da economia de tempo dos métodos de decomposição na resolução do PRVCEJT, a perda de qualidade se sobressai?". Observando os resultados de um aspecto geral, vemos que para a parcela de instâncias testadas, dentre as elaboradas por [Sartori and Buriol 2020], as decomposições apresentaram perda de qualidade relevante, por mais que não seja em todos os casos. Sendo assim, surge espaço para futuras investigações com objetivo de correlacionar as características das decomposições com a qualidade da solução, visto que em nossa análise detectamos que mudar a obtenção do número de subproblemas é suficiente para aprimorar os resultados obtidos.

6. Conclusões e trabalhos futuros

Este trabalho apresenta decomposições baseadas em rotas para o problema Problema de Roteamento de Veículos com Coleta, Entrega e Janelas de Tempo (PRVCEJT), estendendo a o trabalho realizado por [Santini et al. 2023]. Apesar de não obtermos os melhores resultados, as decomposições adaptadas por nós apresentam soluções promissoras, principalmente para instâncias grandes. Dentre os próximos passos dessa pesquisa, que ficam a cargo de trabalhos futuros, destacamos a importância da utilização das instâncias de [Sartori and Buriol 2020] por completo, melhorando assim a qualidade dos testes.

Outros passos seriam: adicionar um novo parâmetro as decomposições, de tal modo que o tempo de execução de um subproblema seja proporcional ao número de pontos do mesmo; coletar a relação entre o número de pontos de um subproblema e à qualidade de sua solução; otimizar a fase de decomposição (por exemplo, retirando a etapa de plotagem da mesma ou desenvolver em *C++*), dando assim mais tempo para os subproblemas; desenvolver uma variação da **Decomposição por Agrupamento de Baricentro 5** que calcule o número de subproblemas utilizando o *Mini Batch K-Means*, tendo em vista os benefícios observados nesta técnica. Implementado-os, será possível realizar a investigação destacada na Subseção 5.3, além de obtermos uma nova decomposição.

Por fim, mas não menos importante, contribuirá significativamente para os trabalhos futuros a análise de todas as decomposições não apresentadas neste artigo, descritas na Seção 4. Seguindo esses passos, o maior e mais detalhado volume de dados a serem coletados implicará na necessidade de mais tempo para a execução dos testes e análise dos mesmos, porém, a qualidade dos resultados obtidos permitirá, por exemplo, a distinção dos cenários mais adequados para a aplicação de cada decomposição.

7. Agradecimentos

Essa pesquisa foi desenvolvida com amparo financeiro da Fundação de Desenvolvimento Científico e Cultural (FUNDECC), acordo 032/2020.

Referências

[Arthur and Vassilvitskii 2007] Arthur, D. and Vassilvitskii, S. (2007). k-means: the advantages of careful seeding. In *Proceedings of the eighteenth annual ACM-SIAM symposium on discrete algorithms*, pages 1027–1035. Society for Industrial and Applied Mathematics.

[Battarra et al. 2014] Battarra, M., Erdoğan, G., and Vigo, D. (2014). Exact algorithms for the clustered vehicle routing problem. *Operations research*, 62(1):58–71.

[Burke and Bykov 2017] Burke, E. K. and Bykov, Y. (2017). The late acceptance hill-climbing heuristic. *European Journal of Operational Research*, 258(1):70–78.

[Curtois et al. 2017] Curtois, T., Landa-Silva, D., Qu, Y., and Laesanklang, W. (2017). Large neighbourhood search with adaptive guided ejection search for the pickup and delivery problem with time windows. *Euro Journal of Transportation and Logistics*, 7.

[Dumas et al. 1991] Dumas, Y., Desrosiers, J., and Soumis, F. (1991). The pickup and delivery problem with time windows. *European Journal of Operational Research*, 54(1):7–22.

[Geoffrion 1970a] Geoffrion, A. M. (1970a). Elements of large-scale mathematical programming part i: Concepts. *Management science*, 16(11):652–675.

[Geoffrion 1970b] Geoffrion, A. M. (1970b). Elements of large scale mathematical programming part ii: Synthesis of algorithms and bibliography. *Management science*, 16(11):676–691.

[Golden and Wong 1981] Golden, B. L. and Wong, R. T. (1981). Capacitated arc routing problems. *Networks*, 11(3):305–315.

[Groer et al. 2011] Groer, C., Golden, B., and Wasil, E. (2011). A parallel algorithm for the vehicle routing problem. *INFORMS journal on computing*, 23(2):315–330.

[Laporte 2007] Laporte, G. (2007). What you should know about the vehicle routing problem. *Naval Research Logistics (NRL)*, 54(8):811–819.

[Li and Lim 2003] Li, H. and Lim, A. (2003). A metaheuristic for the pickup and delivery problem with time windows. *International journal on artificial intelligence tools*, 12(2):173–186.

[Lourenço et al. 2010] Lourenço, H. R., Martin, O. C., and Stützle, T. (2010). *Iterated Local Search: Framework and Applications*, pages 363–397. Springer US, Boston, MA.

[MacQueen et al. 1967] MacQueen, J. et al. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1, pages 281–297. Oakland, CA, USA.

[Ropke and Pisinger 2006] Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, 40(4):455–472.

[Santini et al. 2023] Santini, A., Schneider, M., Vidal, T., and Vigo, D. (2023). Decomposition strategies for vehicle routing heuristics. *INFORMS journal on computing*, 35(3):543–559.

[Sartori 2022] Sartori, C. S. (2022). Solver - a matheuristic approach to the pickup and delivery problem with time windows. `https://github.com/cssartori/math-pdptw`. Acessado: 24-08-2022.

[Sartori and Buriol 2020] Sartori, C. S. and Buriol, L. S. (2020). A study on the pickup and delivery problem with time windows: Matheuristics and new instances. *Computers & Operations Research*, 124:105065.

[Sevaux and Sörensen 2008] Sevaux, M. and Sörensen, K. (2008). Hamiltonian paths in large clustered routing problems. In *workshop on Metaheuristics for Logistics and Vehicle Routing, EU/ME'08*, pages 411–417, Troyes, France.

[Uchoa et al. 2017] Uchoa, E., Pecin, D., Pessoa, A., Poggi, M., Vidal, T., and Subramanian, A. (2017). New benchmark instances for the capacitated vehicle routing problem. *European Journal of Operational Research*, 257(3):845–858.

[Vidal et al. 2020] Vidal, T., Laporte, G., and Matl, P. (2020). A concise guide to existing and emerging vehicle routing problem variants. *European journal of operational research*, 286(2):401–416.