

UNIVERSIDADE FERAL DE JUIZ DE FORA
FACULDADE DE ENGENHARIA & INSTITUTO DE CIÊNCIAS EXATAS
BACHARELADO EM ENGENHARIA COMPUTACIONAL

Heurísticas para o Problema de Roteamento de Veículos com Drones e Janelas de Tempo

Eugenio Belizário Ribeiro Faria

JUIZ DE FORA - MG
FEVEREIRO, 2022

Heurísticas para o Problema de Roteamento de Veículos com Drones e Janelas de Tempo

EUGENIO BELIZÁRIO RIBEIRO FARIA

Universidade Feral de Juiz de Fora
Faculdade de Engenharia & Instituto de Ciências Exatas
Departamentos de Mecânica Aplicada e Computacional & Ciência da Computação
Bacharelado em Engenharia Computacional

Orientador: Stênio Sã Rosário Furtado Soares
Coorientador: Luciana Brugiolo Gonçalves

JUIZ DE FORA - MG
FEVEREIRO, 2022

HEURÍSTICAS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM DRONES E JANELAS DE TEMPO

Eugenio Belizário Ribeiro Faria

MONOGRAFIA SUBMETIDA AO CORPO DOCENTE DA FACULDADE DE ENGENHARIA & INSTITUTO DE CIÊNCIAS EXATAS DA UNIVERSIDADE FERAL DE JUIZ DE FORA, COMO PARTE INTEGRANTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE BACHAREL EM ENGENHARIA COMPUTACIONAL.

Aprovada por:

Stênio Sã Rosário Furtado Soares
Doutor - Universidade Federal de Juiz de Fora

Luciana Brugiolo Gonçalves
Doutor - Universidade Federal de Juiz de Fora

Victor Ströele de Andrade Menezes
Doutor - Universidade Federal de Juiz de Fora

Lorenza Leão Oliveira Moreno
Doutor - Universidade Federal de Juiz de Fora

JUIZ DE FORA - MG
17 DE FEVEREIRO, 2022

Resumo

O problema de roteamento de veículos com drones e janelas de tempo é uma variação do problema clássico de roteamento de veículos onde o conjunto de clientes apresenta elementos que podem ser atendidos por caminhões ou drones e outros que, obrigatoriamente, devem ser atendidos por caminhão. Além disso, cada cliente tem uma demanda conhecida e um intervalo de tempo em que é permitida a entrega. No problema estudado, drones podem ser lançados de caminhões desde que estes estejam parados no depósito ou em algum cliente. Após a realização da entrega, cada drone lançado deve retornar ao mesmo caminhão de onde partiu. O objetivo é minimizar o deslocamento total necessário ao atendimento da demanda de cada cliente respeitando as restrições de janelas de tempo, capacidade de carga de drones e caminhões, além da autonomia e o alcance dos drones. Foi proposta uma abordagem GRASP (*Greedy Randomized Adaptive Search Procedure*) que utiliza diferentes heurísticas de construção de forma gulosa randomizada e, na fase de busca local, um algoritmo RVND (*Randomized Variable Neighborhood Descent*). Como estratégia para melhorar a qualidade da abordagem proposta, utiliza-se um algoritmo baseado em um modelo de Programação Linear Inteira que combina rotas de diferentes soluções.

Palavras-chave: Roteamento de Veículos, Drones, Metaheurísticas, MIP

Abstract

The vehicle routing problem with drones and time windows is a variation of the classic vehicle routing problem where the set of customers has elements that can be serviced by trucks or drones and others that must be serviced by truck. In addition, each customer has a known demand and a time interval in which delivery is allowed. In the problem studied, drones can be launched from trucks as long as they are stopped at the warehouse or at a customer. After the delivery is completed, each drone launched must return to the same truck it started from. The objective is to minimize the total displacement necessary to meet the demand of each client, respecting the restrictions of time windows, carrying capacity of drones and trucks, in addition to the autonomy and range of drones. A GRASP (Greedy Randomized Adaptive Search Procedure) approach was proposed, which uses different construction heuristics in a random greedy way and, in the local search phase, an RVND (Randomized Variable Neighborhood Descent) algorithm. As a strategy to improve the quality of the proposed approach, an algorithm based on an Integer Linear Programming model that combines routes from different solutions is used.

Keywords: Vehicle Routing Problem, Drones, Metaheuristics, MIP

Agradecimentos

A Deus, por ter permitido que eu tivesse saúde e determinação para não desanimar durante a realização deste trabalho e durante toda a graduação

Aos meus pais e familiares, em especial a minha mãe Juvania e minha tia Ivone.

A todos os professores e funcionários que me ajudaram nesta caminhada, em especial aos professores Stênio, Lorenza e Luciana, pela paciência e orientação nos últimos anos.

A Universidade Federal de Juiz de Fora, por todas as oportunidades fornecidas e por proporcionar um ensino público de qualidade.

A todos os amigos feitos durante esta caminhada, principalmente aqueles que entraram comigo e fizeram parte da turma X de algoritmos. Em especial ao amigo que mais esteve ao meu lado durante toda a graduação, André Felipe. Sem todas as nossas brincadeiras e piadas internas nos momentos mais tensos em trabalhos e vésperas de provas eu com certeza não teria chegado até aqui.

“Isto é mais difícil do que pensei que seria.”

Monica Geller (Friends)

Sumário

Lista de Figuras	6
Lista de Tabelas	7
Lista de Abreviações	8
1 Introdução	9
2 Revisão Bibliográfica	14
3 Definição do problema	19
4 Abordagens propostas	22
4.1 Representação da Solução	22
4.2 Algoritmos Construtivo	23
4.2.1 Guloso que insere no meio da rota	24
4.2.2 Guloso que insere no fim da rota	25
4.3 <i>Random Variable Neighborhood Descent</i> - RVND	27
4.4 GRASP Reativo	28
4.5 MIP	32
5 Experimentos Computacionais	35
5.1 Instâncias	35
5.2 Ajuste de Parâmetros	36
5.3 Ambiente de Teste e Configuração do Experimento	38
5.4 Análise de Resultados	38
5.4.1 Comparação entre resultados apresentados na literatura e os obtidos pela implementação proposta	39
5.4.2 Comparação do principal algoritmo da literatura com parâmetros originais (L3) com as meta-heurísticas propostas	41
5.4.3 Comparação do principal algoritmo da literatura com parâmetros melhorados com as abordagens propostas	42
5.4.4 Resultados de todas as implementações da literatura	43
5.4.5 Contribuição do algoritmo MIP	44
5.4.6 Comparação entre todos os algoritmos	45
6 Conclusão e trabalhos futuros	47
Referências Bibliográficas	49
A Apêndices	52
A.1 Resultados Detalhados do MIP	52
A.2 Custo Médio por Algoritmo	55
A.3 Melhor resultado obtido por algoritmo	57
A.4 Tempo Médio por Algoritmo	59

Lista de Figuras

1.1	Valor de soluções que poderiam ser providas por drones em diferentes setores em 2015 (em bilhões de dólares).	12
1.2	Modelo de drone utilizado pela DHL para realização de entregas	13
4.1	Exemplo de solução para o VPR-DTW	23
4.2	Movimentos utilizados no RVND	29
4.3	Visão geral do GRASP reativo implementado	32
4.4	Visão geral do MIP implementado	34

Lista de Tabelas

5.1	Quantidade de instâncias por classe e por número de clientes	35
5.2	Parâmetros ajustados pelo Irace para o MIP	37
5.3	Comparação entre os resultados da literatura com a implementação da literatura	40
5.4	Comparação de tempos, em segundos, entre os resultados da literatura e a implementação da mesma	40
5.5	Comparação do principal algoritmo da literatura com parâmetros originais (L3) com as meta-heurísticas propostas	41
5.6	Resultados comparativos do algoritmo da literatura com parâmetros melhoradas e das abordagens propostas	42
5.7	Resultado resumido da implementação da literatura	43
5.8	Resultado resumido do MIP	44
5.9	Resumo dos resultados quanto ao custo médio por algoritmo	45
5.10	Resumo dos resultados quanto ao melhor custo por algoritmo	46
5.11	Resumo dos tempo médio por algoritmo	46
A.1	Resultado detalhado do MIP	52
A.2	Custo Médio por Algoritmo	55
A.3	Melhor resultado por Algoritmo	57
A.4	Tempo Médio por Algoritmo	59

Lista de Abreviações

DCC	Departamento de Ciência da Computação
UFJF	Universidade Federal de Juiz de Fora
VRP	<i>Vehicle Routing Problem</i> (Problema de Roteamento de Veículos)
VRP-D	<i>Vehicle Routing Problem With Drones</i> (Problema de Roteamento de Veículos com Drones)
VRP-DTW	<i>Vehicle Routing Problem With Drones and Time Windows</i> (Problema de Roteamento de Veículos com Drones e Janelas de Tempo)
UAV	<i>Unmanned Aerial Vehicle</i> (Veículo Aéreo não tripulado)
MILP	<i>Mixed Integer Linear Programming</i> (Programação Linear Inteira)
TSP	<i>Travelling Salesman Problem</i> (Problema do Caixeiro Viajante)
RVND	<i>Random Variable Neighborhood Descent</i> (Busca com Vizinhança Variável Randômica)

1 Introdução

Com o grande crescimento de compras online, um dos aspectos mais importantes para a escolha de qual *e-commerce* utilizar é a velocidade da entrega dos pedidos (BERTÃO, 2021). Assim, realizar entregas de forma rápida é um dos maiores desafios de grandes varejistas. Por isto, diversos pesquisadores estão trabalhando para encontrar formas mais rápidas e mais eficientes para realizar o processo de entrega de mercadorias (RANIERI et al., 2018).

O problema de roteamento de veículos - VRP (do inglês *Vehicle Routing Problem*) é um problema clássico de otimização combinatória amplamente estudado na literatura devido as suas diversas aplicações, sendo uma das principais a redução de custos e de tempo de entrega de mercadorias. O seu objetivo é realizar o atendimento de consumidores por uma frota de veículos, que partem e retornam de um ou mais pontos denominados depósitos. Uma de suas variações mais promissoras visando realizar o processo de entrega de mercadorias de forma mais rápida e eficiente é o problema do roteamento de veículos com drones - VRP-D (do inglês *Vehicle Routing Problem With Drones*).

No VRP-D são utilizados caminhões e drones como veículos. Enquanto caminhões têm grande capacidade de carga e extensa autonomia, mas que, além de lentos e emitirem grande quantidade de gases poluentes, em muitos casos reais devem respeitar restrições de acesso a algumas vias em determinados horários, veículos aéreos não tripulados UAV (do inglês *Unmanned Aerial Vehicle*), popularmente conhecidos como drones, têm alta velocidade de deslocamento e poucos impactos ambientais, mas com capacidade de carga pequena e baixa autonomia. No VRP-D, a frota de drones é distribuída entre os diferentes veículos da frota de caminhões e cada drone só pode ser lançado do depósito ou de um caminhão parado em um cliente, tendo que retornar ao mesmo caminhão não necessariamente no mesmo ponto de partida. Dessa forma, deve-se manter as duas classes de veículos sincronizadas entre si, o que adiciona uma maior complexidade ao problema.

Já o problema de roteamento de veículos com drones e janelas de tempo - VRP-DTW (do inglês *Vehicle Routing Problem With Drones and Time Windows*), tratado neste

trabalho, mantém todas as características do VRP-D e acrescenta janelas de tempo para o atendimento de cada cliente. Restrições de janela de tempo são comuns em aplicações práticas de problemas roteamento de veículos e tornam o problema mais desafiador, já que não basta apenas definir uma sequência de clientes para cada veículo da frota, mas considerar também o tempo de deslocamento e o tempo de atendimento de cada cliente ao longo da rota.

Além da questão da velocidade de entrega, utilizar drones em complemento a caminhões no processo de entrega de mercadorias também traz uma redução nas emissões de gases causadores do efeito estufa, dado que drones utilizam energia elétrica como força motriz, tipicamente menos poluente que os motores a combustão utilizados em caminhões. Estudos mostram que o setor de transporte era responsável por 23.96% das emissões de CO_2 no mundo em 2015, índice ainda mais alarmante no Brasil, onde este setor representava 43.77% das emissões (SOLAYMANI, 2019), demonstrando ainda mais a necessidade de buscar formas alternativas de se realizar a logística das entregas de mercadorias.

Além disso, a utilização de drones em conjunto com caminhões no processo de entrega de mercadorias também pode trazer redução de custos, dado que, segundo Wohl- sen (2014), o custo do combustível para percorrer um quilômetro em um caminhão é de um pouco mais de cinquenta centavos de Euro, enquanto o custo para alimentar um drone pela mesma distância é de apenas dois centavos.

Vale ressaltar que através de avanços recentes na tecnologia empregada em dro- nes, estes dispositivos deixaram de ser objetos apenas de ficção e vêm se tornando uma realidade. Hoje, drones já são utilizados para aplicações nas áreas de:

- infraestrutura, como mostra Varghese et al. (2017), onde são utilizadas imagens obtidas por meio de drones para a verificação da integridade da infraestrutura de energia, incluindo linhas condutoras, postes e isoladores;
- entretenimento, como é exposto em Barrett (2018), onde foram utilizados 1.218 drones voando de forma sincronizada para formar desenhos 3D no céu da abertura das olimpíadas de inverno de 2018;

- segurança pública, seja atuando no reconhecimento de áreas de atuação, na análise de risco, realizando monitoramento em tempo real, visualização remota de áreas perigosas, dentre outras;
- agricultura, sendo utilizado na análise de plantação, demarcação de plantio, pulverização, dentre várias outras aplicações (GONÇALVES; CAVICHIOLI, 2021);
- mineração, na utilização de drones para obtenção de dados tridimensionais do terreno e na inspeção de superfícies de longa extensão e de áreas confinadas, utilizando câmeras térmicas por exemplo;
- transporte, sendo utilizado para socorro em desastres, transportando água, sangue e medicamentos para equipes de resgate (BRAVO; LEIRAS; OLIVEIRA, 2019), para transporte de órgãos, reduzindo o tempo de traslado dos mesmos (HAMPSON, 2019), dentre diversas outras aplicações.

Todas estas aplicações demonstram que a indústria de drones já é uma realidade. Entretanto, ainda existe muito espaço para crescimento na utilização de drones em diversas aplicações. A Figura 1.1, que possui dados retirados de Mazur et al. (2016), exhibe o valor monetário em 2015 de negócios e mão de obra em cada setor que podem ser substituídos por soluções alimentadas por drones, com o valor somado de todos os setores ultrapassando a marca de 127 bilhões de dólares.

A utilização de drones para realização de entregas de mercadorias também já é uma realidade. Existem iniciativas de diversas empresas neste sentido, como a DHL, que integra drones em sua rede de logística para realizar entregas para um local remoto na Alemanha durante o inverno (SCHNEIDER, 2017) e realiza serviços de entregas de forma regular utilizando drones na China desde 2019 (DHL. . . , 2019). A Amazon promete realizar entrega de pacotes aos clientes com segurança em 30 minutos ou menos usando drones (WELCH, 2015). Já a Alphabet realiza entregas de pequenas mercadorias para 100 casas no subúrbio de algumas cidades na Austrália desde 2019 (SUMMERS, 2019), dentre várias outras. A Figura 1.2 mostra o modelo de drone utilizado pela DHL para realização de entregas.

Entretanto, este tipo de aplicação tem suas limitações, como a dependência de um

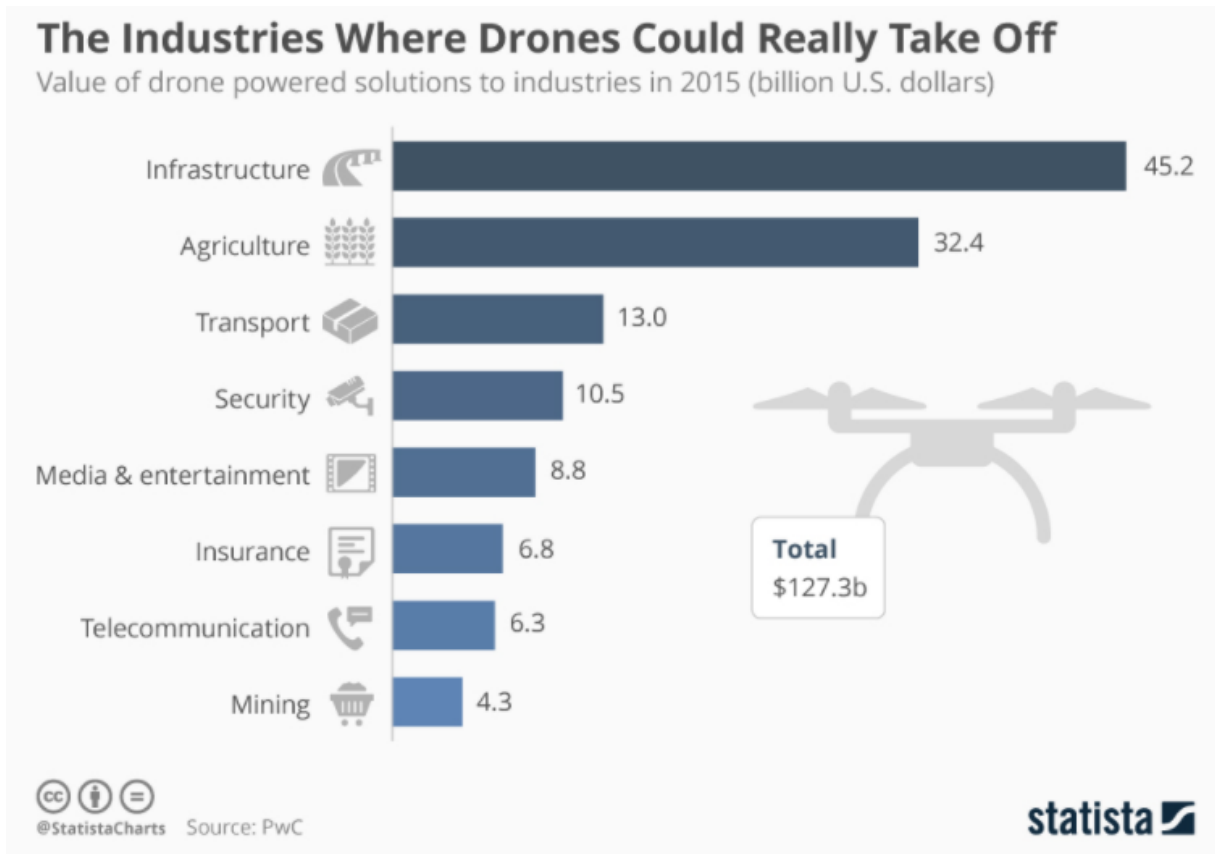


Figura 1.1: Valor de soluções que poderiam ser providas por drones em diferentes setores em 2015 (em bilhões de dólares). Fonte: McCarthy (2016)

depósito como base para voos de drones. Esta limitação é resolvida utilizando caminhões como uma base móvel dos drones e como veículo realizador de entregas, tendo assim uma frota multimodal e aproveitando o melhor que cada tipo de veículo.

Neste trabalho, foram utilizados algoritmos gulosos que conseguem atender um cliente através de uma rota de drone ou de caminhão para construir soluções base, um Busca de Vizinhança Variável Randômica (do inglês *Random Variable Neighborhood Search* - RVND) para melhoria das mesmas, e a metaheurística GRASP (do inglês *Greedy Randomized Adaptive Search Procedure*) que combina o RVND e os gulosos, e um modelo de programação linear inteira (do inglês *Mixed Integer Linear Programming* - MILP) para combinação de rotas de diferentes soluções.

O restante do trabalho está organizado da seguinte maneira: o Capítulo 2 apresenta uma revisão bibliográfica acerca da utilização de drones em problemas de logística; no Capítulo 3 é apresentada uma definição formal do VRP-DTW; os métodos computacionais utilizados para a resolução do problema são apresentados no Capítulo 4; a descrição



Figura 1.2: Drone utilizado pela DHL para realização de entregas. Fonte: DHL... (2019)

dos experimentos computacionais e a análise dos resultados obtidos é descrito no Capítulo 5 e, por fim, as conclusões e propostas de trabalhos futuros são apresentadas no Capítulo 6.

2 Revisão Bibliográfica

Neste capítulo, é apresentada uma revisão da literatura acerca da utilização de drones em problemas de logística, apresentando algumas das diversas variações deste tipo de aplicação.

O primeiro trabalho a combinar drones e caminhões no problema de roteamento de veículos foi Murray e Chu (2015). Nele são apresentadas duas variações do problema do caixeiro viajante - TSP (do inglês *traveling salesman problem*) chamadas *flying side-kick traveling salesman problem* (FSTPH) e *parallel drone scheduling traveling salesman problem* (PDSTSP).

O FSTPH consiste no problema de atender um conjunto de clientes com um único caminhão e um único drone, que possui diversas restrições, como autonomia de voo restrita, não conseguir transportar embalagens pesadas e poder visitar apenas um cliente por vez. O objetivo é minimizar o tempo necessário para atender todos os clientes e retornar com ambos os veículos para o depósito. Vale ressaltar que todas as restrições do TSP clássico são respeitadas. Murray e Chu (2015) apresentaram um modelo de programação linear inteira e uma heurística que inicialmente encontra uma solução para o TSP clássico e em seguida remove alguns clientes da rota e tenta atendê-los utilizando o drone.

Já no PDSTSP, o objetivo é minimizar o makespan, ou seja, o momento em que o caminhão volta ao depósito. Nesta variação são utilizados vários drones, mas eles desenvolvem rotas totalmente independentes do caminhão, não havendo cooperação entre eles. Os drones iniciam e finalizam suas rotas no depósito e atendendo um cliente por viagem. Para resolvê-lo, os autores apresentaram um modelo de programação linear inteira e uma heurística baseada na construção de uma solução inicial onde os drones atendem todos os clientes elegíveis (seja por distância a percorrer ou por peso da encomenda) e o caminhão atende o restante, aplicando em seguida uma busca local para melhoria desta solução base.

Em Agatz, Bouman e Schmidt (2018), os autores resolveram o TSP-D (do inglês

traveling salesman problem with drones), de forma bastante semelhante ao FSTSP apresentado por Murray e Chu (2015). As principais diferenças são que o drone pode retornar ao caminhão no mesmo cliente que foi lançado e que as duas classes de veículos passam a possuir velocidades diferentes. No trabalho foram propostas duas heurísticas baseadas em busca local e programação dinâmica para resolver o problema.

Wang, Poikonen e Golden (2017) introduziram o problema do roteamento de veículos com drones (VRP-D), uma generalização do TSP-D onde a frota é composta por vários caminhões equipados com um ou mais drones que podem ser despachados e recolhidos pelos caminhões no depósito ou em qualquer cliente. Os autores apresentam no trabalho análises de diversos cenários de pior caso onde são discutidos limites para economia de tempo ao utilizar drones em complemento aos caminhões.

Schermer, Moeini e Wendt (2018) propõem duas heurísticas para o VRP-D proposto por Wang, Poikonen e Golden (2017) utilizando como função objetivo a minimização do makespan. Ambas são compostas por duas fases: inicialização e aprimoramento. A fase de inicialização utiliza uma estratégia *route-first cluster-second*, onde primeiro se constrói uma única rota de TSP e posteriormente esta rota é decomposta em várias outras. Já a fase de aprimoramento é composta por vários movimentos de busca local. A diferença entre as heurísticas propostas é que a primeira ignora a utilização de drones na fase de inicialização e os insere apenas na fase de melhorias enquanto a segunda já considera os drones ao construir a solução.

O VPR-D proposto por Wang, Poikonen e Golden (2017) foi estendido de diversas maneiras, seja adicionando novas restrições ao problema ou até mesmo modificando a função objetivo para que o custo de transporte ou os custos ambientais sejam levados em consideração.

Wang e Sheu (2019) introduzem entregas a vários clientes com apenas uma viagem de drone, buscando reduzir o custo da logística. Os autores apresentam uma análise da variação dos custos com a duração máxima de voo de drones, chegando à conclusão de que, utilizando drones com maior autonomia, os custos são reduzidos em até 10%.

Já Chiang et al. (2019) focam nos impactos ambientais, avaliando as emissões de CO_2 durante o processo de entrega de mercadorias, levando em conta o peso do próprio

veículo e das mercadorias a serem entregues, além da distância percorrida, destacando que o uso de drones leva à soluções de baixo custo e com emissões menores do que o processo tradicional de entregas.

Liu et al. (2021) adicionam ao problema restrições relacionadas ao consumo de bateria dos drones conforme o peso do pacote, a eficiência do motor e a velocidade de voo. Eles concluem que o uso de drones pode refletir em maior economia quando existem mais pacotes leves, drones com capacidade de carga maior e com bateria de maior capacidade energética.

Di Puglia Pugliese e Guerriero (2017) adicionaram restrições de janelas de tempo e utilizaram como função objetivo minimizar o custo monetário das entregas, variação também tratada neste trabalho. Os autores apresentaram um modelo de programação linear inteira para o VRP-DTW que descreve e resolve o problema, e um estudo comparativo entre o roteamento de veículos com janelas de tempo e o roteamento de veículos com drones e janelas de tempo, chegando à conclusão que drones permitem uma redução no tempo de conclusão das entregas, mas caso tenham o mesmo custo por quilômetro que caminhões, eles não são vantajosos.

Por fim, em Macrina et al. (2020) é realizada uma profunda e criteriosa análise acerca da literatura sobre utilização de drones em problemas de logística e suas diversas variações, sendo uma excelente fonte de consulta.

Em Pugliese, Macrina e Guerriero (2021), foram usados para resolução do VRP-DTW um novo modelo de programação linear inteira, além de uma meta-heurística de duas fases embutida em uma estrutura *multi-start*. Na primeira fase, é obtida uma solução viável para o problema do roteamento de veículos com janelas de tempo para um subconjunto de clientes. A solução inicia com nenhuma rota, e a cada iteração, podem ser inseridos clientes em qualquer rota pré-existente ou em uma nova rota vazia criada, utilizando uma política de inserção mais barata.

Ainda sobre o trabalho de Pugliese, Macrina e Guerriero (2021), na primeira fase do algoritmo construtivo apresentado, a inserção ocorre entre dois clientes consecutivos, ou seja, as inserções podem ocorrer no “meio” da rota, sempre mantendo a viabilidade da mesma, garantindo que as janelas de tempo dos clientes e a capacidade máxima de carga

do caminhão sejam respeitadas. Em caso de empate no custo de inserção entre algumas posições, é escolhido randomicamente a posição de inserção dentre as com menor custo. Pelo algoritmo da literatura, a escolha do cliente a ser inserido na rota é feita através do identificador do cliente, em ordem crescente.

Já na segunda fase do algoritmo da literatura, são considerados os clientes restantes para obter uma solução viável para o problema do roteamento de veículos com drones e janelas de tempo. Para isto, partindo da solução inicial provida pela primeira fase, a cada iteração é adicionado um cliente em qualquer rota pré-existente ou em uma nova rota, seja esta inserção em uma rota de caminhão ou de drone. Assim como na primeira fase, é utilizada uma política de inserção mais barata e é mantida a viabilidade da rota, ou seja, as janelas de tempo dos clientes e a capacidade máxima do caminhão devem ser respeitadas e, se a rota a ser criada for de drone, a distância percorrida deve ser limitada e o drone possui um tempo máximo de espera pela abertura da janela de tempo do cliente a ser atendido e um tempo máximo de espera pela chegada do caminhão no nó de retorno. Em caso de empate de posições e/ou tipo de inserção, é feito um sorteio dentre as escolhas com menor custo. A escolha do cliente a ser inserido na rota a cada iteração é feita de forma totalmente randômica.

A estratégia MSTPH (do inglês *Multi-Start Two-Phase Heuristic*) funciona chamando o algoritmo de duas fases várias vezes variando o conjunto U utilizado pela primeira fase. Na primeira iteração, o conjunto de vértices U é composto pelos clientes que devem ser obrigatoriamente atendidos por caminhões. A cada iteração, adiciona-se um cliente a este conjunto. Para diversificar ainda mais o espaço de busca, a cada iteração do MSTPH, também tenta-se remover clientes deste conjunto, desde que este cliente removido não seja o que acabou de ser adicionado e que ele não seja um cliente que deve ser atendido obrigatoriamente por um caminhão e, caso seja encontrada uma solução com custo inferior após esta modificação, o conjunto U é atualizado. O algoritmo é finalizado quando não existem vértices a serem adicionados à U ou após um número de iterações sem melhora.

Além das estratégias construtivas descritas, Pugliese, Macrina e Guerriero (2021) propuseram dois algoritmos de melhoria de solução, sendo eles o FLS e o LS, que são integrados no MSTPH buscando soluções de maior qualidade. O procedimento FLS re-

move clientes atendidos por caminhão da rota e os tenta inserir como clientes de drone. Em particular, clientes que são atendidos por drones iniciando ou finalizando sua rota neles não são considerados para remoção. O procedimento é executado ao fim de cada término da segunda fase e a ordem de tentativa de remoção de clientes é a ordem em que estes aparecem na rota. Já o LS trabalha em cima da remoção de todas as rotas de drone da solução base. Dessa forma, a ideia principal do LS é, a partir de rotas contendo apenas entregas de caminhões, mover alguns clientes de sua posição atual na rota para alguma posição diferente na mesma rota ou até mesmo para alguma rota diferente sem se preocupar com os drones decolando ou pousando em cada uma das posições, chamando a segunda fase logo em seguida para reinserir os clientes removidos no primeiro momento. É utilizada uma política primeiro aprimorante, ou seja, sempre que se encontra uma rota viável com custo menor que a anterior, a rota é atualizada e o procedimento é reiniciado.

Assim, resumidamente pode-se dizer que em Pugliese, Macrina e Guerriero (2021) os autores propuseram duas abordagens, nomeadas de MSTPHFLS e MSTPH2. Enquanto MSTPHFLS basicamente executa a busca local FLS ao final de cada execução da segunda fase na estratégia multistart, MSTPH2, além de executar FLS ao final de cada execução da segunda fase, também tenta melhorar a solução através de chamadas do procedimento LS, ao final da execução do FLS.

A revisão da literatura correlata ao problema foco deste trabalho revela que existem muitas variantes de problemas de roteamento que envolvem a utilização de drones e que existem apenas dois trabalhos tratando da variante tratada neste trabalho, em específico (VRP-DTW), sendo que apenas um deles propõe o uso de metaheurísticas para o mesmo.

O presente trabalho utiliza como base de comparação dos resultados das abordagens propostas os resultados apresentados em Pugliese, Macrina e Guerriero (2021).

3 Definição do problema

O problema do roteamento de veículos com drones e janelas de tempo é uma variação do problema clássico do roteamento de veículos com janela de tempo, amplamente estudado na literatura. Nesta variação, a frota de veículos é formada por drones e caminhões, que devem operar de forma sincronizada. O objetivo é realizar com o menor curso a visita a cada cliente uma única vez para entrega de sua demanda, respeitando as janelas de tempo dos mesmos, onde cada caminhão pode possuir vários drones associados a ele.

Sejam os grafos completos $G_c(V, A_c)$ e $G_d(V, A_d)$, com $V = \{v_0, v_1, \dots, v_n\}$, onde v_0 representa o depósito, e $v_i, \forall i \geq 1$, representa os clientes a serem atendidos e, A_c e A_d representando, respectivamente, as arestas ponderadas pela distância a ser percorrida em uma entrega de caminhão e de drone. Dessa forma, denota-se por Ac_{ij} a distância a ser percorrida para um caminhão saindo do cliente v_i chegar ao cliente v_j , enquanto denota-se por Ad_{ij} a distância a ser percorrida para um drone saindo do cliente v_i chegar ao cliente v_j .

A entrega para cada cliente v_i deve ser realizada dentro de uma janela de tempo pré-definida $[a_i, b_i]$. É utilizada a definição clássica de janelas de tempo, ou seja, o tempo t_i em que a entrega é realizada ao cliente v_i deve ser obrigatoriamente realizada entre a_i e b_i , ou seja, $a_i \leq t_i \leq b_i$. Caso o veículo responsável pela entrega chegue ao cliente v_i antes do instante de tempo a_i , ele deve aguardar para realizar a entrega. Caminhões podem realizar esta operação de espera sem nenhum problema, enquanto drones podem esperar por no máximo T instantes de tempo.

Seja $K = \{k_0, k_1, \dots, k_m\}$ um conjunto contendo m caminhões idênticos, todos equipados com drones. Cada um desses caminhões possui uma capacidade de carga W_c e se desloca com velocidade constante s_c . Dessa forma, o tempo de viagem de um caminhão entre o nó v_i e v_j é dada por Ac_{ij}/s_c . Além disso, cada unidade de distância percorrida por um caminhão tem um custo de c_t .

Cada drone decola e pousa no seu próprio caminhão e pode ser utilizado na entrega de encomendas enquanto o caminhão realiza sua rota e deve iniciar e finalizar

suas rotas apenas enquanto o caminhão está parado em um cliente (atendido pelo próprio caminhão) ou no depósito, enquanto o caminhão ainda não iniciou ou já finalizou sua rota.

O processo de entrega de drone pode ser descrito como decolar, atender um cliente e retornar. Dessa forma, drones possuem capacidade unitária, sendo obrigados a voltar ao caminhão para recarregar a bateria e buscar a nova encomenda a cada entrega. Assim, uma rota de drone pode ser caracterizado pela tupla (x, y, z) , com $x, y, z \in V$ e x sendo o nó da decolagem do caminhão, y o cliente atendido pelo drone e z o nó de retorno ao caminhão.

Cada drone tem uma capacidade de carga W_d , podendo fazer entregas apenas de mercadorias mais leves que este valor, e voa a uma velocidade constante s_d . Dessa forma, o tempo de viagem utilizando drones entre o nó v_i e v_j é dado por Ad_{ij}/s_d . Como drones possuem bateria limitada, é imposta uma distância máxima E que cada drone pode percorrer para realizar cada entrega, assumindo que o gasto energético é proporcional à distância percorrida. Assim, a entrega de drone (x, y, z) só é viável se $Ad_{xy} + Ad_{yz} \leq E$. Essa afirmação só pode ser feita porque é considerado que a bateria do drone está totalmente carregada antes de iniciar a realização de uma entrega, afirmação totalmente plausível, dado que basta trocar a bateria do drone ao fim de cada entrega por outra com carga completa. Pelo mesmo motivo, o tempo de recarga dos drones é ignorado, não sendo necessário aguardar a recarga das baterias ao fim de cada entrega. Também têm-se um custo c_d para cada unidade de distância percorrida por um drone.

Para cada cliente v_i , há uma demanda associada, chamada de q_i . Para que o cliente v_i seja passível de entrega de drone, $q_i \leq W_d$. Vale ressaltar que todos os clientes são passíveis de entrega através de caminhões, mas apenas os que atendem a esta restrição são passível de entrega de drone. Entretanto, o somatório dos pesos de todas as entregas efetuadas por um caminhão e por todos os drones associados a esse caminhão deve ser menor que W_c , capacidade do caminhão.

A função objetivo do problema de roteamento de veículos com drones e janelas de tempo é apresentada na Equação 3.1, onde x_{ij}^k é uma variável binária que assume valor 1 se a aresta (v_i, v_j) está sendo usada na rota do caminhão k e o valor 0 caso contrário,

enquanto y_{abc}^k é outra variável binária que assume o valor 1 se as arestas (v_a, v_b) e (v_b, v_c) estão sendo utilizadas na rota de algum drone associado ao k -ésimo caminhão, e o valor 0 caso contrário.

$$\min Z = c_t \cdot \left(\sum_{i \in V} \sum_{j \in V} \sum_{k \in K} Ac_{ij} \cdot x_{ij}^k \right) + c_d \cdot \left(\sum_{a \in V} \sum_{b \in V} \sum_{c \in V} \sum_{k \in K} (Ad_{ab} + Ad_{bc}) \cdot y_{abc}^k \right) \quad (3.1)$$

Vale ressaltar a flexibilidade desta função objetivo. Como basicamente se tem uma constante multiplicando a distância percorrida por drone e outra constante multiplicando a distância percorrida por caminhão, pode-se, variando c_t e c_d , utilizá-la tanto para prevermos questões monetárias, dado que o custo tende a ser maior quanto maior é a distância percorrida por drones e caminhões, quanto questões ambientais, dado que as emissões de CO_2 são diretamente proporcionais à distância percorrida. Além disso, ela deixa explícita a diferença de “custo” entre utilizar drones e caminhões para realizar uma dada entrega.

Assim, para resolver VRP-DTW deseja-se definir um conjunto de rotas e entregas de drones de forma que todos os clientes $v_i \in V$ sejam atendidos exatamente uma vez dentro de sua janela de tempo com o custo mínimo de transporte.

4 Abordagens propostas

Neste capítulo são descritas a representação da solução utilizada, as abordagens propostas para resolver o VRP-DTW, assim como uma breve análise das possíveis limitações da implementação feita das abordagens da literatura. Em linhas gerais, são propostas duas maneiras de se resolver o VRP-DTW: um GRASP (do inglês *Greedy Randomized Adaptive Search Procedure*) reativo e um algoritmo MIP (do inglês *Mixed Integer Programming*) que combina rotas de diferentes soluções.

4.1 Representação da Solução

Uma solução para o VRP-DTW nada mais é que um conjunto de rotas que se iniciam e finalizam no depósito, onde todos os clientes são atendidos exatamente uma vez. Dessa forma, para descrever a representação de uma solução, basta descrever a representação de cada uma das rotas que compõe a mesma.

Neste trabalho, utilizou-se uma representação similar à utilizada em Pugliese, Macrina e Guerriero (2021), onde basicamente são usadas duas estruturas: uma que representa as rotas de caminhão e outra que representam as rotas de drone. Para cada rota de caminhão, basicamente é utilizado um vetor contendo os ids dos vértices visitados. É importante ressaltar que a ordem do ids no vetor representa a ordem de visita dos clientes. Em particular, toda rota de caminhão é iniciada e finalizada com o nó de id 0 (depósito). Já para rotas de drones, usa-se um vetor de tuplas de três elementos, sendo eles: o nó de partida, o cliente atendido pelo drone e o nó de retorno ao caminhão, nesta ordem. Vale ressaltar que a ordem das tuplas no vetor é indiferente nesta representação, enquanto a ordem dos vértices em cada tupla é relevante.

Por exemplo, para a solução apresentada na Figura 4.1, na qual os círculos representam os clientes e depósito, com o id indicado no interior, as entregas de caminhão são representadas por setas com linhas contínuas, e entregas de drone por setas com linhas tracejadas, possui a seguinte representação, sendo a rota 1 representada em azul na figura

e a rota 2 em vermelho:

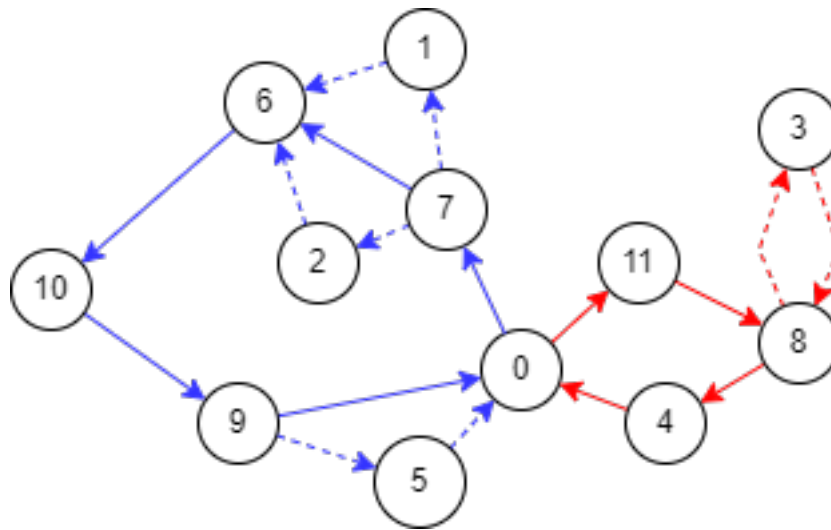


Figura 4.1: Exemplo de solução para o VPR-DTW

- Rota 1:
 - Caminhão: [0, 7, 6, 10, 9, 0]
 - Drone: [(7, 1, 6), (7, 2, 6), (9, 5, 0)]
- Rota 2:
 - Caminhão: [0, 11, 8, 4, 0]
 - Drone: [(8, 3, 8)]

4.2 Algoritmos Construtivo

Para construir soluções iniciais e que serão posteriormente refinadas por um algoritmo de busca local, foram desenvolvidos dois diferentes tipos de algoritmos gulosos, ambos randomizados, de forma que o primeiro realiza inserções no meio da rota, enquanto o segundo realiza inserções sempre no final das rotas. Como ambos os procedimentos não são realizados de forma totalmente gulosa, é possível obter uma variedade de soluções, já que não se escolhe o melhor candidato, e sim um dentre os melhores candidatos.

4.2.1 Guloso que insere no meio da rota

O algoritmo guloso que insere no meio da rota foi desenvolvido em dois níveis, onde a partir de uma solução vazia, a cada iteração é escolhido um cliente (primeiro nível de randomização) e uma posição na solução parcial para o inserir (segundo nível de randomização). O cliente, uma vez colocado na solução, nunca mais será removido, ou seja, não há revisão de decisões já tomadas. O procedimento se repete até que todos os clientes sejam atendidos.

O Algoritmo 1 mostra o pseudocódigo do método construtivo proposto. Ele recebe como parâmetro um valor α , que define o percentual dos elementos a serem considerados em cada iteração.

Optou-se por utilizar dois critérios de seleção de clientes a entrar na solução a cada iteração do guloso: o peso da entrega a ser realizada para o cliente v e o tempo de fechamento da sua janela de tempo. Dessa forma, para o primeiro critério, o algoritmo tende a inserir na solução os clientes com maior peso em um primeiro momento, ou seja, os clientes que obrigatoriamente devem ser atendidos por caminhões, favorecendo assim a formação de rotas básicas de caminhões antes das primeiras tentativas de entregas via drone. Já para o segundo critério, os vértices que serão inseridos primeiramente na solução são aqueles que tem sua janela de tempo fechando mais cedo. Assim, a função *selecionaCliente*(V, α) seleciona um dos $\alpha\%$ melhores clientes do conjunto de candidatos a partir de um dos critérios detalhados acima, recebido via parâmetro.

As funções *getPossiveisInsercoesCaminhao*(S, v) e *getPossiveisInsercoesDrones*(S, v) nas linhas 6 e 8 procuram por possíveis posições na solução atual na qual o cliente pode ser inserido via caminhão ou drone, respectivamente. Vale ressaltar que além das posições nas rotas existentes, é considerada a possibilidade de criação de uma nova rota.

Já a função *selecionaInsercao*(*caminhao*, *drone*, α) (linha 12) recebe por parâmetro todas as inserções possíveis na solução parcial, seja via drone ou via caminhão, e retorna uma das $\alpha\%$ inserções mais baratas. Assim, é dada prioridade a inserções com menor impacto no custo atual da solução.

Por fim, a função *inserirClienteNaSolucao*($s, v, movimento$) na linha 13 simplesmente insere o cliente v através do veículo e na posição determinadas pela função

selecionaInsercao(caminhao, drone, α).

Algoritmo 1: Algoritmo Construtivo Inseire Meio

Entrada: α , *instancia*
Saída: S

```

1 início
2    $V = \{v_1, v_2, \dots, v_n\}$ ;
3    $S = \emptyset$ ;
4   while  $V \neq \emptyset$  do
5      $v = selecionaCliente(V, \alpha)$ ;
6      $caminhao = getPossiveisInsercoesCaminhao(S, v)$ ;
7     se clientePodeSerAtendidoPorDrone( $v$ ) então
8       |  $drone = getPossiveisInsercoesDrones(S, v)$ ;
9     senão
10      |  $drone = \emptyset$ ;
11    fim se
12     $movimento = selecionaInsercao(caminhao, drone, \alpha)$ ;
13     $s = inserirClienteNaSolucao(s, v, movimento)$ ;
14     $V = V - v$ ;
15  end while
16  return  $S$ 
17 fim

```

4.2.2 Guloso que insere no fim da rota

O algoritmo guloso que insere no fim da rota, tem um funcionamento similar ao guloso apresentado na Seção 4.2.1, diferenciando tanto pelo fato de não inserir clientes no meio de uma rota, trabalhando sempre com a inserção no final da mesma, quanto pelo fato da seleção do cliente a ser inserido na solução a cada iteração não ser randomizada.

O Algoritmo 2 mostra o funcionamento desta estratégia gulosa. No laço compreendido entre as linhas 4 e 11, a seleção do cliente a ser inserido pelo guloso na solução se dá pelo tempo de fechamento da janela de tempo dos clientes. Dessa forma, a função *selecionaCliente* (linha 5) retorna o cliente que ainda não foi adicionado à solução com fechamento de janela mais próximo de ocorrer. Este cliente será adicionado ao final de alguma rota como entrega de caminhão.

A função *getPossiveisRotasParaInserir*(S, v) na linha 6 faz uma busca em todas as rotas existentes na solução atual para verificar em quais é possível realizar a inserção do cliente v . Já a função *getPossiveisInsercoes*($S, V, rotas$) (linha 7) retorna todos os movimentos possíveis a partir das rotas definidas no passo anterior. Assim, sendo o último

cliente inserido na rota k o cliente c_k , o que esta função faz é tentar, para todas as rotas retornadas no passo anterior, realizar movimentos via drone com decolagem ocorrendo em c_k , atendendo um cliente ainda não inserido na solução, e retornando ao caminhão em v , retornando todas as combinações possíveis. Em particular, é retornada à combinação sem nenhuma rota de drone, contendo apenas a rota de caminhão entre c_k e v . Assim, tem-se as seguintes possibilidades de movimento: apenas a inserção de um cliente ao fim de uma rota; a inserção de um cliente v ao fim da rota k e de um outro cliente como rota de drone decolando de c_k e retornando em v ; a inserção de um cliente v ao fim da rota k e de dois outros clientes como rota de drone decolando de c_k e retornando em v ; e assim sucessivamente.

Já a função *selecionaInsercao*(*possiveisInsercoes*, α) (linha 7) basicamente escolhe uma das possíveis inserções dentre as $\alpha\%$ melhores possíveis. Como temos inserções com diferentes números de clientes atendidos, utilizou-se o custo médio de atendimento como critério de escolha.

Por fim, a função *realizaInsercao*(S , *insercao*) da linha 9 efetua a inserção escolhida no passo anterior, retornando a solução atualizada e os vértices incluídos na solução através da inserção escolhida.

Algoritmo 2: Algoritmo Construtivo Insere Final

Entrada: α , *instancia*
Saída: S

```

1 início
2    $V = \{v_1, v_2, \dots, v_n\}$ ;
3    $S = \emptyset$ ;
4   while  $V \neq \emptyset$  do
5      $v = \text{selecionaCliente}(V)$ ;
6      $rotas = \text{getPossiveisRotasParaInserir}(S, v)$ 
7      $possiveisInsercoes = \text{getPossiveisInsercoes}(S, V, rotas)$ 
8      $insercao = \text{selecionaInsercao}(possiveisInsercoes, \alpha)$ ;
9      $S, verticesInseridos = \text{realizaInsercao}(S, insercao)$ ;
10     $V = V - verticesInseridos$ ;
11  end while
12  return  $S$ 
13 fim
```

4.3 *Random Variable Neighborhood Descent* - RVND

Uma heurística de busca local consiste em percorrer a vizinhança de uma solução na tentativa de encontrar uma solução melhor que a atual. A vizinhança de uma solução consiste em um conjunto finito de soluções obtidas a partir de alterações realizadas na sua estrutura, definidas por uma operação denotada movimento. Cada movimento define uma vizinhança e, uma vez verificadas todas as soluções desta vizinhança, a melhor solução encontrada é denotada ótimo local para aquela vizinhança.

Uma vez que um dado movimento pode levar a soluções que violem as restrições do problema, pode-se optar por submeter as soluções geradas a algum procedimento de viabilização, ou simplesmente descartar todas soluções inviáveis da vizinhança. Neste trabalho, considerou-se apenas soluções viáveis.

O uso de diferentes heurísticas de busca local para uma dada solução permite que diferentes regiões próximas a ela sejam intensificadas no processo de busca. Uma técnica bastante explorada na literatura para diferentes problemas de otimização quando se tem diferentes movimentos é o método de descida randômica com vizinhança variável - RVND, do inglês *Random Variable Neighborhood Descendent*, uma variação do VND (do inglês *Variable Neighborhood Descendent*) desenvolvido por Mladenović e Hansen (1997), no qual se executa uma sequência aleatória de buscas locais, modificando o movimento a ser realizado durante a busca.

Neste trabalho, utilizou-se uma variação do RVND, de forma que foram utilizados 10 movimentos, sendo os oito primeiros são selecionados normalmente conforme o RVND padrão, enquanto os dois últimos foram fixados. Os oito movimentos selecionados randomicamente são exemplificados na Figura 4.2. Nesta figura, setas com linhas contínuas representam rotas de caminhões, e setas com linhas tracejadas representam rotas de drones. O vértice em destaque em cada uma das figuras representa o nó em que o movimento foi realizado. Vale ressaltar que o depósito é representado pelo nó 0. A Figura 4.2a representa a solução base antes de qualquer movimento ser realizado.

As Figuras 4.2b, 4.2c, 4.2d, 4.2e exemplificam os movimentos intra-rota utilizados. O movimento *or-opt1-truck-drone* tenta remover um nó atendido por caminhão e passar a atendê-lo por meio de um drone. Já o movimento *or-opt1-truck-truck* modifica a posição

na rota onde o cliente está sendo atendido por caminhão. O movimento *or-opt1-drone-truck* tenta remover um cliente atendido por drone e passar a atendê-lo via caminhão. Por fim, o movimento *or-opt1-drone-drone* tenta trocar a posição de partida e/ou de chegada do drone no caminhão ao atender o cliente.

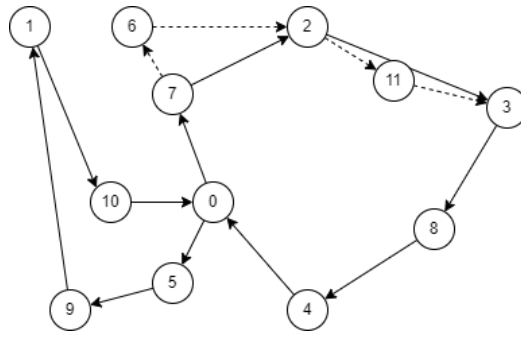
Já as Figuras 4.2f, 4.2g, 4.2h, 4.2i exemplificam os movimentos inter-rota utilizados. Dessa forma, o movimento *shift-truck-truck* tenta remover uma entrega de caminhão de uma rota e inseri-lo como entrega de caminhão em outra rota. O movimento *shift-truck-drone* remove uma entrega de caminhão de uma rota e o atende via drone em outra rota. Já o movimento *shift-drone-truck* remove um cliente atendido via drone em uma rota e o insere como entrega de caminhão em outra rota. Por fim, o movimento *shift-drone-drone* remove uma entrega via drone de uma rota e tenta realizar a entrega em outra rota também através de um drone.

Os dois movimentos restantes fixados no final do RVND são o `clientReinsert(2)` e o `clientReinsert(3)`, que trabalham com a remoção e reinserção de k nós da solução atual, de forma que, para o primeiro movimento, utilizou-se $k = 2$ e, caso não se encontre nenhuma solução melhor que a inicial, tenta-se utilizar $k = 3$. Estes dois movimentos finais do RVND não são exaustivos, ou seja, não são testadas todas as possibilidades de movimento, sendo o número de tentativas de modificações tentadas um parâmetro a ser passado ao algoritmo.

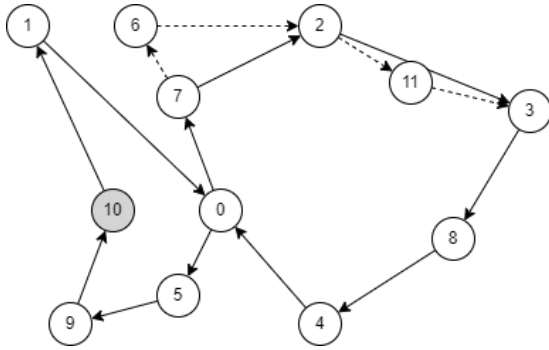
Vale ressaltar que caso não se encontre nenhum vizinho melhor, as funções de movimentos retornam a solução passada por parâmetro. Além disso, pode-se optar por utilizar a estratégia primeiro aprimorante ou a melhor aprimorante. O funcionamento do RVND proposto está descrito no Algoritmo 3, onde a função *getMovimentoRandomico* basicamente seleciona aleatoriamente, com igualdade de probabilidades, um dos movimentos contidos no vetor movimentos.

4.4 GRASP Reativo

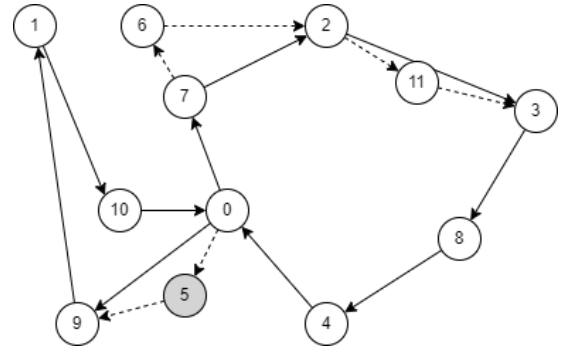
Um algoritmo GRASP é uma metaheurística que combina as etapas de construção e de busca local dentro de um processo iterativo. A versão que é utilizada neste trabalho foi proposta inicialmente por (PRAIS; RIBEIRO, 2000), na qual o grau de aleatoriedade



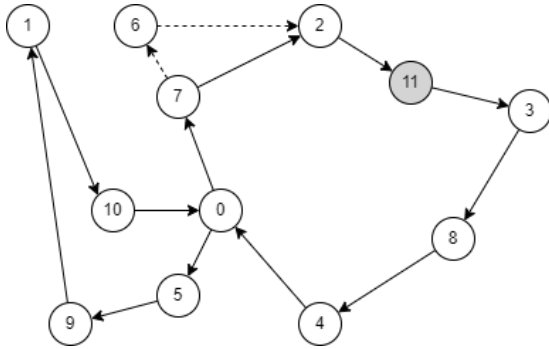
(a) Rota Original



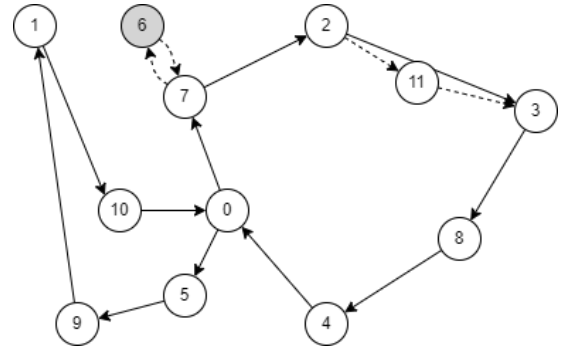
(b) Movimento Or-opt1-truck-truck



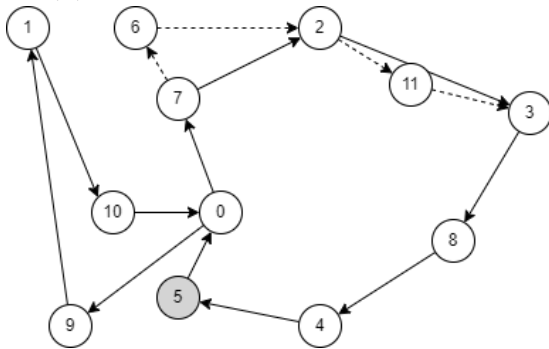
(c) Movimento Or-opt1-truck-drone



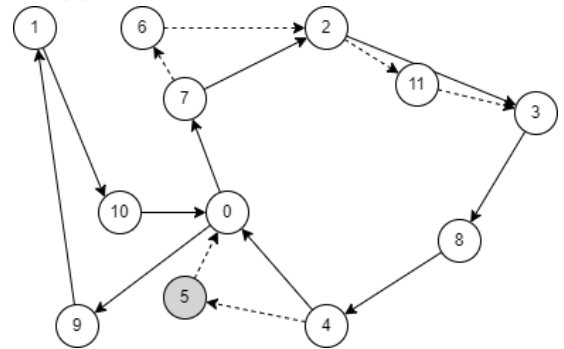
(d) Movimento Or-opt1-Drone-truck



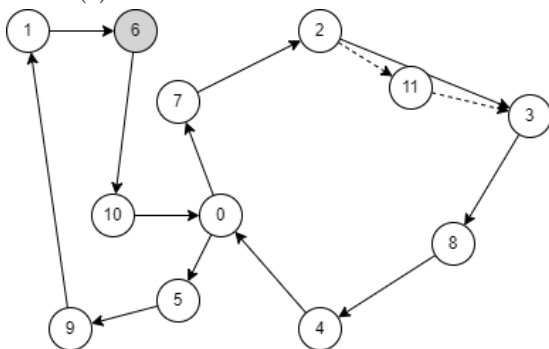
(e) Movimento Or-opt1-drone-drone



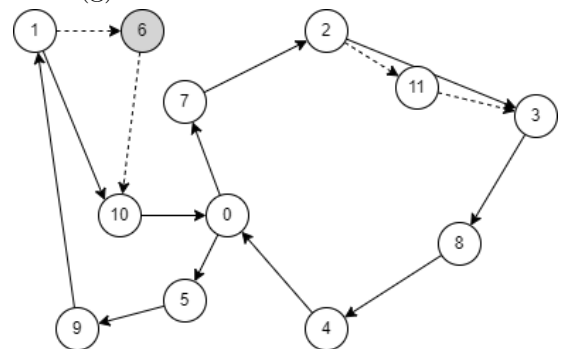
(f) Movimento Shift-truck-truck



(g) Movimento Shift-truck-drone



(h) Movimento Shift-drone-truck



(i) Movimento Shift-drone-drone

Figura 4.2: Movimentos utilizados no RVND

Algoritmo 3: RVND

Entrada: S_0
Saída: S^*

```

1 início
2    $S = S_0$ ;
3    $movimentos = [movimento1(), \dots, movimento8()];$ 
4   while  $movimentos \neq \emptyset$  do
5      $movimento = getMovimentoRandomico(movimentos);$ 
6      $S^* = movimento(S);$ 
7     se  $custo(S^*) < custo(S)$  então
8        $movimentos = [movimento1(), \dots, movimento8()];$ 
9        $S = S^*$ ;
10    senão
11       $movimentos = movimentos - movimento;$ 
12    fim se
13    se  $movimentos == \emptyset$  então
14       $S^* = clientReinsert(S, 2)$ 
15      se  $custo(S^*) == custo(S)$  então
16         $S^* = clientReinsert(S, 3)$ 
17        se  $custo(S^*) < custo(S)$  então
18           $movimentos = [movimento1(), \dots, movimento8()];$ 
19           $S = S^*$ ;
20        fim se
21      senão
22         $movimentos = [movimento1(), \dots, movimento8()];$ 
23         $S = S^*$ ;
24      fim se
25    fim se
26  end while
27  return  $S^*$ ;
28 fim

```

no algoritmo construtivo é ajustada durante a execução do algoritmo de acordo com os resultados obtidos ao longo da execução. Esta variação é denominada GRASP Reativo.

O funcionamento desta metaheurística pode ser visto no Algoritmo 4. Em resumo, a partir de uma lista de diferentes valores de α recebida por parâmetro, a cada iteração é selecionado o α a ser utilizado nesta iteração. Nas primeiras iterações, todas as opções de valores de α possuem a mesma probabilidade de seleção. A cada bloco de iterações, estas probabilidades são atualizadas conforme o histórico de qualidade das soluções obtidas pelo algoritmo para cada valor de α , de forma que aqueles valores de α que estão trazendo melhores resultados sejam priorizados em relação aos outros.

Uma vez escolhido o valor de α a ser utilizado, uma solução é construída utilizando a estratégia apresentada na Seção 4.2.1, utilizando o critério de peso da encomenda. A partir da solução obtida na fase de construção, utiliza-se o RVND apresentado na Seção 4.3 como etapa de busca local do algoritmo. Estas duas etapas são executadas até que um dado critério de parada seja atendido. Por fim, é retornada à solução com menor custo obtida durante todo o processo.

Algoritmo 4: GRASP Reativo

Entrada:

lista α , numeroIteracoes, numeroIteracoesAtualizarProbabilidade

Saída: Solução S , inteiro numeroIteracoes, float[] lista α

```

1 início
2   probabilidades = iniciaProbabilidades(lista $\alpha$ );
3   S* =  $\emptyset$ ;
4   for i = 0; i < numeroIteracoes; i ++ do
5      $\alpha = escolhe\alpha(probabilidades, lista\alpha);$ 
6     S = construtivo( $\alpha$ );
7     S = RVND(S);
8     se custo(S) < custo(S*) então
9       | S* = S;
10    fim se
11    se i mod numeroIteracoesAtualizarProbabilidade == 0 então
12      | probabilidades = atualizaProbabilidades(probabilidades, lista $\alpha$ );
13    fim se
14  end for
15  return S*
16 fim

```

Uma representação gráfica de seu funcionamento pode ser visto na Figura 4.3.

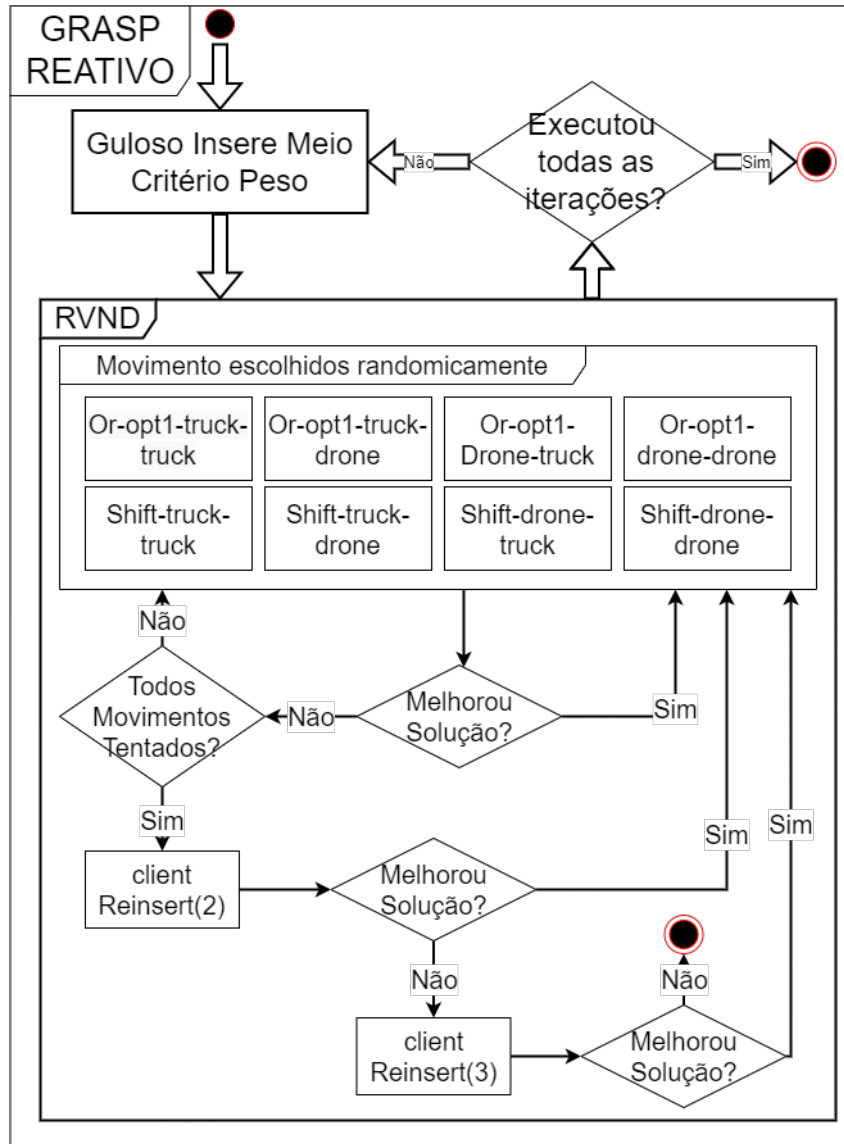


Figura 4.3: Visão geral do GRASP reativo implementado

4.5 MIP

A fim de se obter resultados ainda melhores, foi desenvolvido um modelo de programação linear inteira através do qual se combinam rotas de diferentes soluções, formando uma nova solução a partir destas. Dessa forma, basicamente, o que este modelo faz é minimizar o somatório dos custos das rotas escolhidas (Equação 4.1), forçando que em todas as rotas escolhidas os clientes sejam visitados apenas uma vez (Equação 4.2).

As Equações 4.1, 4.2 e 4.3 mostram o modelo desenvolvido. A variável r_i representa se a i -ésima rota está na solução (quando seu valor é 1) ou não (quando seu valor é 0). $Custo(i)$ representa o custo da rota i , enquanto A_i^v é uma entrada binária que

representa se a rota i atende o cliente v .

$$\text{Minimizar } z = \sum_{i \in R} \text{custo}(i) * r_i \quad (4.1)$$

$$\sum_{i \in R} r_i * A_i^v = 1, \forall v \in V; \quad (4.2)$$

$$r_i \in (0, 1), \forall i \in R; \quad (4.3)$$

Como conjunto de rotas para serem selecionadas pelo MIP utilizou-se todas as rotas de todas as soluções geradas pelos seguintes procedimentos:

- GRASP descrito na Seção 4.2.1, utilizando o peso da encomenda como critério de escolha do cliente a ser atendido no guloso e a estratégia primeiro aprimorante no RVND;
- GRASP descrito na Seção 4.2.1, utilizando o peso da encomenda como critério de escolha do cliente a ser atendido no guloso e a estratégia melhor aprimorante no RVND;
- GRASP descrito na Seção 4.2.1, utilizando o tempo de fechamento da janela de tempo como critério de escolha do cliente a ser atendido no guloso e a estratégia primeiro aprimorante no RVND;
- GRASP descrito na Seção 4.2.1, utilizando o tempo de fechamento da janela de tempo como critério de escolha do cliente a ser atendido no guloso e a estratégia melhor aprimorante no RVND;
- Construtivo descrito na Seção 4.2.2 seguido do RVND utilizando a estratégia primeiro aprimorante
- Construtivo descrito na Seção 4.2.2 seguido do RVND utilizando a estratégia melhor aprimorante
- Rotas unitárias, ou seja, rotas contendo apenas um cliente atendido, construídas artificialmente

Uma representação gráfica do funcionamento deste algoritmo pode ser visto na Figura 4.4.

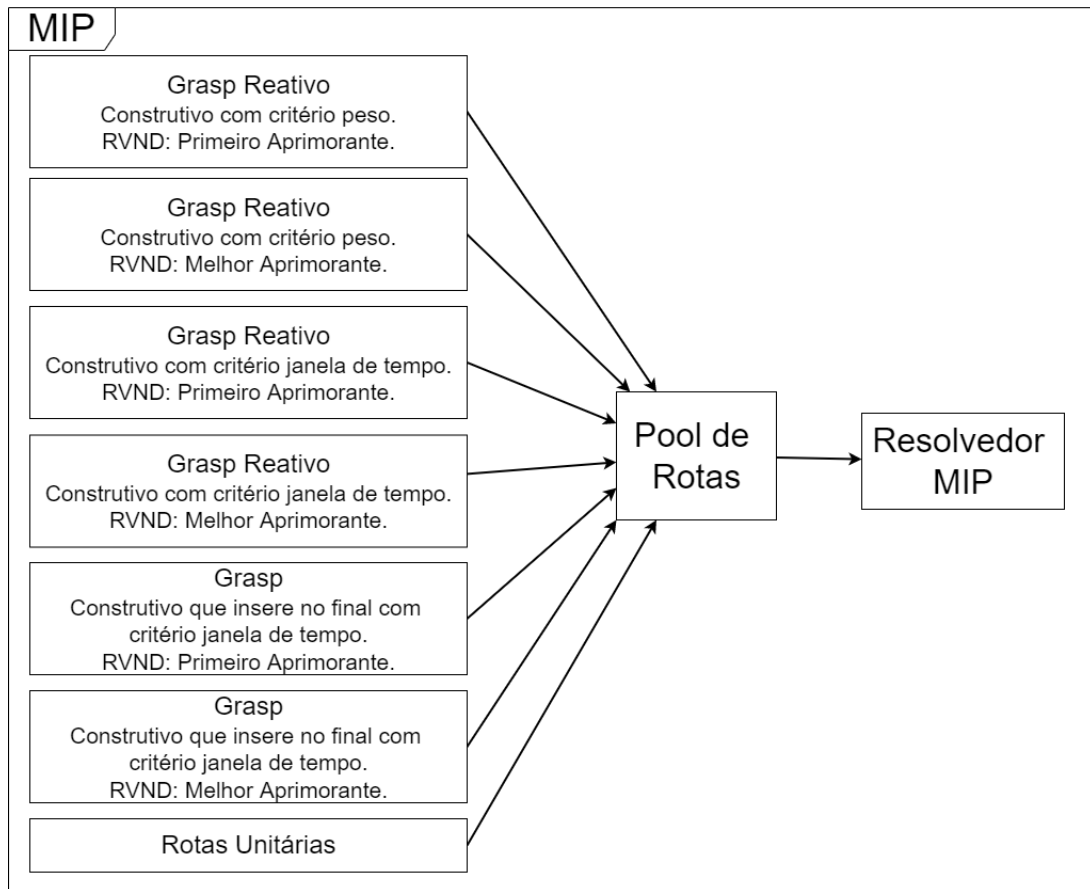


Figura 4.4: Visão geral do MIP implementado

5 Experimentos Computacionais

Neste capítulo, os experimentos computacionais realizados são detalhados e os resultados obtidos são analisados.

5.1 Instâncias

As instâncias usadas foram retiradas de Solomon (1987), sendo utilizadas as classes randômicas (R2), clusterizadas (C2) e randômicas clusterizadas (RC2). Foram feitas diversas adaptações nestas instâncias, conforme descrito em Pugliese, Macrina e Guerriero (2021). Assim, são mantidas as posições originais de todos os clientes e do depósito. Também são mantidas as demandas e as janelas de tempo. Na tabela 5.1 pode-se ver a quantidade de instâncias por classe e por número de clientes:

Tabela 5.1: Quantidade de instâncias por classe e por número de clientes

	25	50	100	TOTAL
R	11	11	11	33
C	8	8	8	24
RC	8	8	8	24
TOTAL	27	27	27	81

É assumido que 80% dos clientes podem ser atendidos via drone, sendo eles os 80% com menor demanda. O tempo de serviço para atendimento via drone é igual ao tempo de serviço da instância original, enquanto o tempo de serviço para atendimento via drone é definido como metade do tempo original. O tempo máximo que um drone pode ficar parado esperando a janela de tempo de um cliente abrir é determinado de acordo com o tipo da instância, sendo 10 unidades de tempo para instâncias randômicas e randômicas clusterizadas e 90 unidades de tempo para as clusterizadas. O custo de transporte por unidade de distância via caminhão é 25, enquanto o custo por unidade de distância via drone é 1.

Além disso, a capacidade de carga de cada caminhão é a mesma da instância original e a métrica para medir distâncias entre clientes utilizada para caminhões é a

métrica de Manhattan, simulando quarteirões, enquanto a métrica utilizada para drones é a euclidiana, simulando entregas em linha reta.

5.2 Ajuste de Parâmetros

Os ajustes de parâmetros para todos os algoritmos apresentados foram feitos a partir de testes empíricos e do framework *iterated race* (IRACE), desenvolvido por López-Ibáñez et al. (2016), pelo qual é realizada uma varredura entre os parâmetros candidatos para selecionar a melhor configuração. Abaixo é descrito com mais detalhes como foi feito o ajuste de parâmetros para cada um dos algoritmos utilizados.

Os parâmetros necessários para o RVND são definir se será utilizado melhor ou primeiro aprimorante, e o número de vezes que será tentado melhorar a solução através do `clientReinsert(2)` e do `clientReinsert(3)`. A definição da utilização de primeiro ou melhor aprimorante é feita de acordo com o algoritmo fonte da solução base, e será descrito caso a caso. Já o número de tentativas de melhorias através do `clientReinsert(2)` e `clientReinsert(3)` foi definido empiricamente, de forma que se tenha um ganho no custo da solução sem impactar no tempo gasto. Dessa forma, definiu-se utilizar, para o `clientReinsert(2)`, o número de vértices da instância dividido por 2.5 e, para o `clientReinsert(3)`, o número de vértices da instância dividido por 1.25. Assim, como as instâncias possuem 25, 50 ou 100 vértices, para o `clientReinsert(2)` são feitas 10, 20 e 40 tentativas, e para o `clientReinsert(3)` são feitas 20, 40 e 80 tentativas, respectivamente.

Para o GRASP, os parâmetros a serem ajustados são o conjunto de valores para α a ser utilizado, o número de iterações, o número de iterações para se atualizar a probabilidade de escolha de cada alfa e se a estratégia utilizada no RVND é primeiro ou melhor aprimorante. Como valores de alfa utilizados definiu-se empiricamente utilizar $\{0.1, 0.2$ e $0.3\}$. Foram dadas 1000 iterações para o GRASP, sendo que a cada 50 iterações são atualizadas as probabilidades de cada alfa ser escolhido, e optou-se por utilizar a estratégia primeiro aprimorante.

Para o MIP, os parâmetros a serem ajustados são alfas utilizados em cada GRASP, alfa utilizado no guloso que insere no final e o número de execuções de cada um dos algoritmos. Para os GRASP's utilizados, definiu-se utilizar $\{0.1, 0.2$ e $0.3\}$, os mesmos

utilizados na execução isolada do GRASP. Para o guloso que insere no final foi utilizado alfa 0.2, dado que o que se deseja nestas execuções é ter uma boa variedade de rotas, e utilizando alfa de 0.1 eram obtidas muitas soluções iguais. Por fim, para a definição do número de iterações de cada algoritmo, foi utilizado o Irace, onde cada um dos algoritmos tinha a possibilidade de ser executado por 0, 20, 50, 80, 100, 150, 200, 250, 300, 350, 400, 450, 500, 550, 600, 650, 700, 750, 800, 850 ou 900 iterações. Além disso, foi limitado o somatório de todas as execuções superiormente em 1200 e inferiormente em 700. Foi dado um máximo de 6000 experimentos, e utilizadas 27 instâncias para a realização dos ajustes, sendo oito clusterizadas, onze randomizadas e oito clusterizadas e randomizadas. A Tabela 5.2 exibe o número de iterações escolhido para cada versão do GRASP, onde a coluna GRASP descreve o tipo de GRASP utilizado (reativo ou não) e a versão do construtivo utilizada (inserindo no meio ou no fim da rota); a coluna Crit. Construtivo descreve o critério para seleção do vértice a entrar na rota pelo construtivo; a coluna RVND mostra se foi utilizada a versão primeiro ou melhor aprimorante para o RVND; e a coluna selecionado apresenta a quantidade de iterações para o GRASP selecionado pelo IRACE.

Grasp	Crit. Construtivo	RVND	Selecionado
GRASP reativo meio	Peso entrega	Primeiro Aprimorante	500
GRASP reativo meio	Peso entrega	Melhor Aprimorante	450
GRASP reativo meio	Janela de tempo	Primeiro Aprimorante	50
GRASP reativo meio	Janela de tempo	Melhor Aprimorante	80
GRASP fim	Janela de tempo	Primeiro Aprimorante	100
GRASP fim	Janela de tempo	Melhor Aprimorante	0

Tabela 5.2: Parâmetros ajustados pelo Irace para o MIP

Para os algoritmos da literatura, foram realizados dois experimentos computacionais distintos: um utilizando os parâmetros utilizados originalmente no artigo, e um com estes parâmetros aumentados. Isto só foi possível pela implementação ter sido feita em C++, o que levou a uma melhora de tempo considerável quando considerado com os resultados reportados no artigo. Assim, para o MSTPHFLS é necessário passar apenas um parâmetro, que é o número de tentativas de adição sem melhora de vértices no conjunto de clientes atendidos na primeira fase do algoritmo. Originalmente, o valor foi 2, e foi feito um experimento computacional extra utilizando 10. Já para o MSTPH2,

além do parâmetro representando o número de tentativas de adição sem melhora de vértices no conjunto de clientes atendidos na primeira fase, também é necessário um outro parâmetro, responsável por definir o número de execuções sem melhora da busca local LS. Originalmente, o valor utilizado para ambos os parâmetros foi 2, mas também foram coletados resultados utilizando 10 para ambos os parâmetros.

5.3 Ambiente de Teste e Configuração do Experimento

Todos os algoritmos descritos neste trabalho foram desenvolvidos na linguagem C++, com os modelos matemáticos sendo resolvidos através do resolvedor Gurobi Optimization, LLC (2021), na versão 9.1.2. Os testes foram executados em uma máquina com processador core i5 7200U possuindo um clock base de 2.50 GHz, 8 GB de memória RAM com 2400 MHz, sistema operacional Mint 20.2 com kernel Linux 5.4.0-96-generic. O compilador utilizado foi o g++ (Ubuntu 9.3.0-17ubuntu1 20.04) 9.3.0. Cada instância foi executada 30 vezes e foram aferidos o tempo de execução e o custo da solução encontrada.

5.4 Análise de Resultados

Nesta seção, são descritos os resultados obtidos, assim como feita uma análise comparativa dos mesmos. Primeiramente é feita uma comparação entre os resultados obtidos na implementação da literatura e os resultados expostos no artigo base de comparação, justificando assim a implementação da mesma. Depois, são comparados os resultados da implementação da literatura utilizando os parâmetros originais e parâmetros melhorados com os resultados dos algoritmos propostos, seguido de uma análise comparativa entre todos os algoritmos da literatura implementados; depois, é feita uma análise dos resultados obtidos pelo MIP e, por fim, é realizada uma análise comparativa dentre todos os algoritmos implementados.

São exibidas diversas tabelas com resumos de resultados, onde as linhas denotadas por Todas apresentam a média de todas as instâncias, as linhas denotadas por 25, 50 e 100 representam, respectivamente, a média das instâncias conforme o número de clientes (25, 50 ou 100), enquanto as linhas denotadas por C, R e RC apresentam as médias das

instâncias conforme o tipo (clusterizadas, randomizadas ou randomizadas e clusterizadas). As colunas nomeadas como L1 apresentam os resultados obtidos pela implementação do algoritmo MSTPHFLS desenvolvido neste trabalho, utilizando os parâmetros originais do artigo base (PUGLIESE; MACRINA; GUERRIERO, 2021), as colunas L2 trazem os resultados da implementação do algoritmo MSTPHFLS utilizando parâmetros ajustados empiricamente neste trabalho. Já as colunas denotadas por L3 mostram os resultados obtidos a partir da implementação do algoritmo MSPTPH2 utilizando os parâmetros originais, enquanto nas colunas L4 são apresentados os resultados obtidos com a implementação do algoritmo MSTPH2 com parâmetros definidos empiricamente neste trabalho e as colunas nomeadas como Literatura referem-se aos resultados apresentados no artigo base.

5.4.1 Comparação entre resultados apresentados na literatura e os obtidos pela implementação proposta

Com o objetivo de validar os resultados obtidos a partir da implementação do melhor algoritmo da literatura com os resultados apresentados no artigo de referência para estes mesmos algoritmos, realizou-se um experimento em que utilizou-se os mesmos parâmetros sugeridos pelos autores, versão denotada aqui como L3. Destaca-se, entretanto que embora as instâncias usadas neste trabalho tenham sido obtidas do mesmo *benchmark* usados na referência, o conjunto de dados em cada experimento poderia ser diferente. O objetivo aqui era verificar possíveis distorções quanto aos dados efetivamente usados em cada instância.

As Tabelas 5.3 e 5.4 apresentam a comparação dos resultados de custo e tempo, respectivamente, obtidos por L3 com os apresentados no artigo base, onde valores em negrito indicam os melhores resultados. Nelas, pode-se ver um resumo dos resultados por classe e tamanho de instâncias.

Nas tabelas 5.3 e 5.4, podemos perceber que o tempo e o custo das soluções da implementação da literatura (L3) com os resultados expostos em Pugliese, Macrina e Guerriero (2021) não são 100% compatíveis. Existem algumas justificativas para isto. Com relação ao tempo, a implementação do artigo foi feita na linguagem JAVA, sabida-

Tabela 5.3: Comparação entre os resultados da literatura com a implementação da literatura

	25		50		100	
	Artigo	L3	Artigo	L3	Artigo	L3
R	6190.42	6860,8	10055.77	9739,41	19427.71	13559,98
C	7921.65	4683,8	12204.18	8418,44	18873.51	13713,98
RC	9073.14	6904	15330.14	12833,29	23647.34	20660,61
Todas	7557.52	6221,8	12255.11	10262,35	20513.76	15666,82

Tabela 5.4: Comparação de tempos, em segundos, entre os resultados da literatura e a implementação da mesma

	25		50		100	
	Artigo	L3	Artigo	L3	Artigo	L3
R	56.67	0,036	1419.10	0,217	4031.59	2,294
C	23.08	0,022	1206.49	0,375	3101.14	5,762
RC	46.01	0,03	1169.04	0,341	3208.22	2,817
Todas	43.56	0,03	1282.01	0,301	3511.94	3,477

mente de menor desempenho que a linguagem C++ utilizado pela implementação feita neste trabalho. Já em termos de custo, o primeiro e principal ponto é que o artigo não descreve com clareza como são tratadas as instâncias. Os autores descrevem que, para gerar a instância para o problema tratado, tomam-se as instâncias disponibilizadas por Solomon (1987), selecionam-se aleatoriamente alguns nós contidos nesta instância, e, a partir disto, faz-se o processo descrito na Seção 5.1. Em momento algum é descrito quais instâncias são utilizadas para gerar as instâncias de tamanho 25, por exemplo, se toma-se simplesmente as instâncias de Solomon (1987) de tamanho 25 e seleciona-se todos os nós, ou se é feita uma seleção de 25 nós nas instâncias de tamanho 50, por exemplo. Este foi o motivo principal pelo qual fez-se necessária uma implementação dos algoritmos da literatura, ou seja, para que fosse possível, sem nenhuma dúvida, realizar uma comparação de execuções utilizando instâncias idênticas.

Entretanto, também existem alguns pontos na descrição dos algoritmos do artigo base que não ficam totalmente claros, o que leva à possibilidade de que as implementações destes algoritmos aqui apresentados não reflitam fielmente o que os autores desenvolveram.

O principal ponto de dúvida está relacionado à busca local LS no processo de troca de um nó de atendido por caminhão de uma rota para outra rota. Lá é descrito, em tradução livre, que no procedimento inter-rota entre duas rotas diferentes k e k' ,

é construída uma meta-rota $r^{kk'}$, onde k e k' são unidas e a estratégia entre rotas é executada, e após o movimento, a meta-rota é dividida. Entretanto, não é descrito como é feita esta união de rotas, o que pode ser ou uma tarefa complexa, devido as janelas de tempo de cada cliente, ou uma tarefa simples, caso se ignore alguma restrição do problema neste ponto. Dessa forma, dada a falta de clareza na descrição do algoritmo, foi implementado como procedimento entre rotas a remoção de um vértice da rota k e inserção do mesmo na rota k' , na posição em que esta inserção não infringe nenhuma restrição, sem que nenhuma meta-rota seja explicitamente criada no procedimento.

Diante da dificuldade de comparação entre os resultados das abordagens propostas neste trabalho com os resultados apresentados no artigo de referência, seja pela incerteza quanto à forma com que as instâncias usadas no experimento dos autores foram geradas, ou pela diferença de desempenho de códigos gerados em C++ e Java, a análise que segue é baseada na implementação do melhor algoritmo dos autores.

5.4.2 Comparação do principal algoritmo da literatura com parâmetros originais (L3) com as meta-heurísticas propostas

Esta seção tem por objetivo comparar os algoritmos propostos neste trabalho com a implementação do melhor algoritmo da literatura com parâmetros originais. Para isto, a Tabela 5.5 mostra os resultados comparativos de custo médio, melhor custo e tempo entre os algoritmos propostos e a implementação do MSTPH2 da literatura com parâmetros originais, coluna L3 da Tabela.

Tabela 5.5: Comparação do principal algoritmo da literatura com parâmetros originais (L3) com as meta-heurísticas propostas

	Custo Médio			Melhor Custo			Tempo Médio (s)		
	L3	Mip	grasp	L3	Mip	grasp	L3	Mip	grasp
25	6899,62	6343,03	6595,84	6221,37	6098,11	6249,41	0,03	1,19	0,87
50	11408,80	10721,65	11500,16	10266,59	10220,37	10643,04	0,30	5,79	4,04
100	17535,81	17379,51	17934,70	15713,89	16046,33	16222,70	3,48	59,44	28,48
C	9804,14	9747,66	10463,51	8937,71	9004,63	9497,25	2,05	16,58	12,38
R	11132,55	10930,37	11231,04	10055,61	10314,52	10391,18	0,85	17,23	11,24
RC	15213,37	13972,80	14628,34	13462,92	13223,33	13469,42	1,06	34,45	9,74
Todas	11948,08	11481,40	12010,23	10733,95	10788,27	11038,38	1,27	22,14	11,13

Analisando a Tabela 5.5, percebe-se que o algoritmo da literatura, quando executado com parâmetros originais, tem resultados médios piores que o MIP proposto, porém

sendo executado com tempos cerca de 17.5 vezes menores. Quando olha-se para o melhor custo obtido em cada algoritmo, o MIP obtém melhores resultados em instancias de tamanho 25 e 50 e de classe Randomizada e Clusterizada (RC), mas perde nas de tamanho 100 e classe Clusterizadas (C) e Randomizadas (R). Já o GRASP possui resultados equivalentes à literatura, mas possui tempos aproximadamente 9 vezes maiores.

Dessa forma, fica claro que, uma vez que foi feita uma reimplementação do algoritmo da literatura em uma linguagem mais performática, é possível dar mais tempo de processamento para ele.

5.4.3 Comparação do principal algoritmo da literatura com parâmetros melhorados com as abordagens propostas

Conforme justificado na Subseção 5.4.2, foi dado mais tempo de execução para MSTPH2. Nesta seção, é feita uma comparação dos algoritmos propostos com a implementação do MSTPH2 da literatura com parâmetros definidos empiricamente neste trabalho (L4). Para isto, a Tabela 5.6, exibe a comparação entre os resultados entre estas abordagens.

Reforçando, o algoritmo MSTPH2 recebe dois parâmetros, sendo eles o número de tentativas de adição sem melhora de vértices no conjunto de clientes atendidos na primeira fase do algoritmo e o número de execuções sem melhora da busca local LS. Originalmente, em L3 e no artigo base de comparação, ambos são setados com o valor 2. Após testes empíricos, em L4, foram utilizados ambos com o valor 10, o que proveu uma melhora significativa nos custos das soluções sem que o tempo explodisse.

Tabela 5.6: Resultados comparativos do algoritmo da literatura com parâmetros melhoradas e das abordagens propostas

	Custo Médio			Melhor Custo			Tempo Médio (s)		
	L4	Mip	grasp	L4	Mip	grasp	L4	Mip	grasp
25	6511,28	6343,03	6595,84	6143,93	6098,11	6249,41	0,21	1,19	0,87
50	10689,73	10721,65	11500,16	10015,37	10220,37	10643,04	2,45	5,79	4,04
100	16453,40	17379,51	17934,70	15441,04	16046,33	16222,70	29,63	59,44	28,48
C	9331,19	9747,66	10463,51	8962,17	9004,63	9497,25	16,57	16,58	12,38
R	10494,57	10930,37	11231,04	9848,42	10314,52	10391,18	7,82	17,23	11,24
RC	14099,99	13972,80	14628,34	13046,63	13223,33	13469,42	9,01	34,45	9,74
Todas	11218,14	11481,40	12010,23	10533,44	10788,27	11038,38	10,76	22,14	11,13

Na tabela 5.6 fica claro o ganho do algoritmo da literatura quando se dá mais tempo de processamento. Se com parâmetros originais ele sempre perdia no custo médio para a abordagem que utiliza o MIP, com mais iterações ele passou a ter os melhores

resultados na maioria das vezes. Na Subsessão 5.4.6 será feita uma comparação mais detalhada entre a literatura implementada e os resultados dos algoritmos aqui propostos.

5.4.4 Resultados de todas as implementações da literatura

Esta Subseção tem por objetivo exibir os resultados de todas as implementações da literatura feitas, fazendo uma breve análise sobre os mesmos. Dessa forma, um resumo dos resultados obtidos pode ser visto na Tabela 5.7, onde a coluna Melhor representa o melhor resultado obtido dentre as quatro implementações testadas. As colunas referenciadas por Custo Médio apresentam a distância percentual média entre a média de todos os resultados obtidos por cada algoritmo em 30 execuções e a coluna melhor, enquanto as colunas referenciadas por Melhor Custo apresentam a distância percentual média entre melhor custo obtido por cada algoritmo nas 30 execuções e a coluna melhor. Por fim, as colunas referenciadas por Tempo Médio (s) apresentam o tempo médio, em segundos, da execução de cada um dos algoritmos.

Reforçando, as colunas nomeadas como L1 apresentam os resultados obtidos pela implementação do algoritmo MSTPHFLS desenvolvido neste trabalho, utilizando os parâmetros originais do artigo base (PUGLIESE; MACRINA; GUERRIERO, 2021), as colunas L2 trazem os resultados da implementação do algoritmo MSTPHFLS utilizando parâmetros ajustados empiricamente neste trabalho. Já as colunas denotadas por L3 mostram os resultados obtidos a partir da implementação do algoritmo MSPTPH2 utilizando os parâmetros originais, enquanto nas colunas L4 são apresentados os resultados obtidos com a implementação do algoritmo MSTPH2 com parâmetros definidos empiricamente neste trabalho.

Tabela 5.7: Resultado resumido da implementação da literatura

	Melhor	Custo Médio (%)				Melhor Custo (%)				Tempo Médio (s)			
		L1	L2	L3	L4	L1	L2	L3	L4	L1	L2	L3	L4
25	6137,593	37,60	25,70	12,16	5,97	15,82	12,81	1,37	0,12	0,001	0,002	0,030	0,208
50	9998,333	40,05	29,63	13,49	6,66	20,24	16,29	2,64	0,20	0,001	0,004	0,301	2,452
100	15343,370	39,73	28,61	14,02	7,04	21,78	17,83	2,11	0,71	0,004	0,013	3,477	29,634
C	8853,125	39,45	27,59	9,34	4,47	20,97	18,74	0,92	0,84	0,002	0,007	2,053	16,572
R	9831,333	31,34	22,14	13,75	7,14	13,65	10,25	2,39	0,21	0,002	0,006	0,849	7,818
RC	13043,000	49,50	36,40	16,39	7,84	25,32	19,98	2,68	0,03	0,002	0,006	1,063	9,010
Todas	10493,099	39,12	27,98	13,23	6,55	19,28	15,65	2,04	0,34	0,002	0,006	1,269	10,765

Como era de se esperar, os resultados utilizando parâmetros utilizando parâme-

tros ajustados empiricamente neste trabalho são superiores aos utilizando parâmetros originais, apesar de para isso necessitarem de mais tempo de execução. Além disso, pode-se observar que os melhores resultados foram obtidos pelo MSTPH2 (L3 e L4), o que também era esperado, dado que o algoritmo basicamente executa uma busca local a mais que o MSTPHFLS.

5.4.5 Contribuição do algoritmo MIP

Esta Subseção tem por objetivo demonstrar a efetividade do MIP. Para isto, um resumo dos resultados obtidos pelo mesmo pode ser visto na tabela 5.8. A coluna Melhor Solução Antes MIP apresenta o custo da melhor solução recebida pelo MIP para realizar a combinação de rotas. A coluna melhoria após MIP apresenta representa a melhoria percentual média em relação a melhor solução original recebida pelo MIP. A coluna Execução do MIP apresenta o tempo gasto unicamente para realizar a combinação de rotas das soluções recebidas, enquanto a coluna total apresenta o tempo total da execução do MIP, incluindo o tempo de construção das soluções originais.

Tabela 5.8: Resultado resumido do MIP

	Custo Médio		Tempo Médio (s)	
	Melhor Solução Antes MIP	Melhoria após MIP (%)	Execução do MIP	Total
25	6576,3247	-3,259%	0,1262	1,1923
50	11461,9778	-5,669%	0,5908	5,7864
100	17857,2173	-2,701%	20,3708	59,4358
C	10403,3833	-4,799%	0,9326	16,5774
R	11227,0374	-3,198%	1,3142	17,2302
RC	14541,9000	-3,886%	20,9842	34,4473
Todas	11965,1733	-3,876%	7,0293	22,1382

Fica nítido que, para instâncias do tipo Randomizadas e Clusterizadas (RC) e instâncias contendo 100 clientes, o MIP encontra mais dificuldade para combinar as rotas, levando tempo bastante superior para realizar a operação que para instâncias de outros tipos. Além disso, o MIP conseguiu obter melhoras mais notáveis em instâncias de tamanho 50 e em instâncias do tipo clusterizadas. Além disso, fica claro que a utilização do MIP é justificada, dada que na média, ele melhora em 3.876% a melhor solução recebida por ele.

5.4.6 Comparação entre todos os algoritmos

A Tabela 5.9 apresenta os custos médios das soluções encontradas por cada algoritmo agrupados por número de clientes e por tipo de instância. O detalhamento deste tipo de resultado pode ser visto no Anexo A.2. A coluna ML representa o melhor resultado médio das implementações da literatura, e o restante das colunas são diferenças percentuais em relação a este valor.

Tabela 5.9: Resumo dos resultados quanto ao custo médio por algoritmo

	ML	L1	L2	L3	L4	MIP	GRASP
25	6511,3	29,71%	18,51%	5,81%	0,00%	-2,34%	1,29%
50	10689,7	31,13%	21,38%	6,34%	0,00%	0,66%	7,13%
100	16453,4	30,33%	20,00%	6,50%	0,00%	6,93%	10,48%
C	9331,2	33,13%	21,88%	4,60%	0,00%	3,47%	9,39%
R	10494,6	22,44%	13,89%	6,15%	0,00%	2,82%	6,19%
RC	14100	38,57%	26,39%	7,92%	0,00%	-1,45%	3,37%
Todas	11218,1	30,39%	19,96%	6,21%	0,00%	1,75%	6,30%

Percebe-se que o MIP consegue obter melhores resultados médios em instâncias de tamanho 25 e também em instâncias de tipo randomizadas e clusterizadas, mesmo quando comparada ao algoritmo da literatura implementado com parâmetros melhorados. Além disso, pode-se verificar que, na média, o GRASP tem resultados comparáveis a implementação da literatura que utiliza os mesmos parâmetros indicados no artigo (L3).

Por fim, quando se analisa os resultados detalhados no Anexo A.2, pode-se perceber que o GRASP tem uma média de resultados melhor que o melhor da literatura em 15 instâncias, o que representa 18.5% do total de instâncias, enquanto o MIP tem médias melhores em 36 instâncias (44% do total de instâncias).

A Tabela 5.10 apresenta os melhores custos dentre as 30 execuções para cada instância das soluções encontradas por cada algoritmo agrupados por número de clientes e por tipo de instância, além de exibir os resultados médios para todas as 81 instâncias na linha Todas. O detalhamento deste tipo de resultado pode ser visto no anexo A.3. A coluna ML representa o melhor resultado das implementações da literatura, e o restante das colunas são diferenças percentuais para ele.

Percebe-se que o MIP consegue obter os melhores resultados em instâncias com

Tabela 5.10: Resumo dos resultados quanto ao melhor custo por algoritmo

	ML	L1	L2	L3	L4	MIP	GRASP
25	6137,6	15,82%	12,81%	1,37%	0,12%	-0,58%	1,76%
50	9998,3	20,24%	16,29%	2,64%	0,20%	2,17%	6,05%
100	15343,4	21,78%	17,83%	2,11%	0,71%	5,30%	6,49%
C	8853,1	20,97%	18,74%	0,92%	0,84%	1,01%	5,62%
R	9831,3	13,65%	10,25%	2,39%	0,21%	4,05%	5,23%
RC	13043	25,32%	19,98%	2,68%	0,03%	1,17%	3,27%
Todas	10493,1	19,28%	15,65%	2,04%	0,34%	2,30%	4,77%

25 clientes, mas na média, fica a 2.30% dos melhores resultados obtidos pela literatura. Já o GRASP acaba perdendo por 4.77% quando comparado aos melhores resultados da literatura. Analisando os resultados instância a instância, conforme apresentado no Anexo A.3, o GRASP obteve o melhor resultado em 19 instâncias (23.5% de todas as instâncias), enquanto o MIP encontra melhores resultados em 31 instâncias (38.27% das instâncias).

Quanto ao tempo médio, em segundos, a tabela 5.11 mostra o comportamento dos algoritmos quanto o número de clientes da instância e quanto ao seu tipo. O detalhamento instância a instância deste resultado pode ser visto no anexo A.4.

Tabela 5.11: Resumo dos tempo médio por algoritmo

	L1	L2	L3	L4	MIP	GRASP
25	0,001	0,002	0,030	0,208	1,192	0,873
50	0,001	0,004	0,301	2,452	5,786	4,043
100	0,004	0,013	3,477	29,634	59,436	28,484
C	0,002	0,007	2,053	16,572	16,577	12,377
R	0,002	0,006	0,849	7,818	17,23	11,244
RC	0,002	0,006	1,063	9,010	34,447	9,738
Todas	0,002	0,006	1,269	10,765	22,138	11,133

Pode-se perceber que o algoritmo que usa MIP, conforme já citado anteriormente, tem um grande impacto em seu tempo tempo quando utilizado para resolver instâncias do tipo RC e de tamanho 100, o que acaba elevando bastante sua média de tempo, que acaba sendo o dobro da média de tempo do L4. Já o GRASP proposto possui tempos compatíveis com L4, sendo inclusive mais rápido em instâncias do tipo clusterizadas.

6 Conclusão e trabalhos futuros

Este trabalho propôs dois métodos computacionais para resolver o problema de roteamento de veículos com drones e janelas de tempo: um GRASP reativo e um algoritmo baseado no modelo de programação linear inteira que combina rotas provenientes de diferentes versões da heurística GRASP. Além disso, foi feita a implementação dos algoritmos propostos por Pugliese, Macrina e Guerriero (2021), utilizados como base de comparação com os algoritmos propostos.

Para o GRASP reativo, na fase construtiva foi utilizado um algoritmo guloso randomizado que insere clientes no meio das rotas existentes ou em uma nova rota, utilizando como critério de escolha do cliente a ser inserido na rota a cada iteração o peso de sua encomenda, priorizando solicitações maiores e conseqüentemente formando rotas de caminhão em um primeiro momento. Já para a fase de busca local foi utilizado um algoritmo RVND modificado, onde foram utilizados os movimentos Or-opt1-truck-truck, Or-opt1-truck-drone, Or-opt1-Drone-truck, Or-opt1-drone-drone, Shift-truck-truck, Shift-truck-drone, Shift-drone-truck e Shift-drone-drone escolhidos aleatoriamente na primeira fase do algoritmo, e, caso nenhum destes movimentos melhore a solução, são chamados os movimentos clientReinsert(2) e clientReinsert(3), nesta ordem. Foi utilizada uma estratégia melhor aprimorante para todos os movimentos utilizados na abordagem GRASP proposta.

Já o algoritmo baseado em modelo de programação linear inteira trabalha com um conjunto de soluções de entrada, desmembrando-as e formando novas soluções. Resumidamente, o algoritmo combina rotas de diferentes soluções, garantindo que todos os clientes sejam atendidos exatamente uma vez. Como conjunto de soluções de entrada, utilizou-se o algoritmo GRASP descrito no parágrafo anterior em quatro variantes: utilizando como critério de seleção de cliente a ser atendido a cada iteração na fase gulosa o peso da encomenda e o tempo de fechamento da janela de tempo, e para o RVND utilizando as estratégias melhor e primeiro aprimorante. Além disso, também foi utilizado um GRASP simples, que utiliza como construtivo um algoritmo guloso que insere no final

das rotas, utilizando como critério de cliente a ser atendido a cada iteração o tempo de fechamento de sua janela de tempo, em conjunto com o RVND primeiro aprimorante.

Além disso, foram feitas implementações de dois algoritmos propostos por Pugliese, Macrina e Guerriero (2021), o MSTPH2 e o MSTPHFLS, de forma a tornar possível comparações entre a literatura e os algoritmos propostos neste trabalho.

Todos os algoritmos foram submetidos a um conjunto de 81 instâncias da literatura, sendo 27 de cada tamanho (25, 50 ou 100 clientes), 24 do tipo clusterizada, 33 do tipo randomizada e 24 do tipo randomizada e clusterizada.

O GRASP implementado obteve resultados superiores aos da implementação da literatura em aproximadamente 23% de todas as instâncias, ficando em média a 4,77% dos melhores resultados da literatura implementada, enquanto o MIP encontrou resultados melhores em aproximadamente 38% das instâncias, ficando em média a 2,30% dos melhores resultados da implementação da literatura.

Como trabalhos futuros, sugere-se explorar outras meta-heurísticas para este problema, como um ILS (do inglês *Iterated Local Search*), utilizando como perturbação para a solução a remoção de alguns clientes da solução, e utilizar movimentos mais variados no RVND, realizando um estudo mais aprofundado sobre as instâncias onde se obteve os piores resultados, de forma a aplicar movimentos de busca local que melhoram os mesmos.

Além disso, sugere-se utilizar fontes mais variadas de soluções para o MIP, dando para o mesmo um conjunto de soluções mais variado para exploração. Ademais, pode-se realizar testes com os algoritmos propostos em variações do problema de roteamento de veículos com drones e janelas de tempo, como a variante que fixa o número de drones associado a cada caminhão. Por fim, como trabalho futuro, pode-se observar que existem inúmeras aplicações que consistem em variações do problema tratado, seja quanto à função objetivo, como considerar emissões de gases poluentes, por exemplo, ou mesmo quanto às restrições, levando em conta frota heterogênea, velocidade da via variando ao longo do tempo, entre outras.

Referências Bibliográficas

- AGATZ, N.; BOUMAN, P.; SCHMIDT, M. Optimization approaches for the traveling salesman problem with drone. *Transportation Science*, v. 52, n. 4, p. 965–981, 2018. Disponível em: <<https://doi.org/10.1287/trsc.2017.0791>>.
- BARRETT, B. Inside the olympics opening ceremony world-record drone show. *Wired*, 2018. Disponível em: <<https://www.wired.com/story/olympics-opening-ceremony-drone-show/>>. Acesso em: 28 nov. 2021.
- BERTÃO, N. Velocidade de entrega de produtos está no topo da lista de expectativa do consumidor. *Valor Investe*, 2021. Disponível em: <<https://valorinveste.globo.com/objetivo/gastar-bem/noticia/2021/09/28/velocidade-de-entrega-de-produtos-esta-no-topo-da-lista-de-expectativa-do-consumidor.ghtml>>. Acesso em: 28 nov. 2021.
- BRAVO, R. Z. B.; LEIRAS, A.; OLIVEIRA, F. L. C. The use of uavs in humanitarian relief: An application of pomdp-based methodology for finding victims. *Production and Operations Management*, v. 28, n. 2, p. 421–440, 2019. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1111/poms.12930>>.
- CHIANG, W.-C. et al. Impact of drone delivery on sustainability and cost: Realizing the uav potential through vehicle routing optimization. *Applied Energy*, v. 242, p. 1164–1175, 2019. ISSN 0306-2619. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0306261919305252>>.
- DHL Express launches its first regular fully-automated and Intelligent Urban Drone Delivery Service. 2019. Disponível em: <<https://www.dhl.com/tw-en/home/press/press-archive/2019/dhl-express-launches-its-first-regular-fully-automated-and-intelligent-urban-drone-delivery-service.html>>. Acesso em: 28 nov. 2021.
- Di Puglia Pugliese, L.; GUERRIERO, F. Last-mile deliveries by using drones and classical vehicles. In: SFORZA, A.; STERLE, C. (Ed.). *Optimization and Decision Science: Methodologies and Applications*. Cham: Springer International Publishing, 2017. p. 557–565. ISBN 978-3-319-67308-0.
- GONÇALVES, V. P.; CAVICHIOLI, F. A. Estudo das funcionalidades dos drones na agricultura. *Revista Interface Tecnológica*, v. 18, n. 1, p. 321–331, jul. 2021. Disponível em: <<https://revista.fatectq.edu.br/index.php/interfacetecnologica/article/view/1126>>.
- Gurobi Optimization, LLC. *Gurobi Optimizer Reference Manual*. 2021. Disponível em: <<https://www.gurobi.com>>.
- HAMPSON, M. Drone delivers human kidney: The organ was flown several kilometers by a drone without incurring damage - [news]. *IEEE Spectrum*, v. 56, n. 1, p. 7–9, Jan 2019. ISSN 1939-9340.
- LIU, Y. et al. Two-echelon routing problem for parcel delivery by cooperated truck and drone. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, v. 51, n. 12, p. 7450–7465, Dec 2021. ISSN 2168-2232.

- LÓPEZ-IBÁÑEZ, M. et al. The irace package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, v. 3, p. 43–58, 2016. ISSN 2214-7160. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2214716015300270>>.
- MACRINA, G. et al. Drone-aided routing: A literature review. *Transportation Research Part C: Emerging Technologies*, v. 120, p. 102762, 2020. ISSN 0968-090X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0968090X20306744>>.
- MAZUR, M. et al. Clarity from above: Pwc global report on the commercial applications of drone technology. *Warsaw: Drone Powered Solutions, PriceWater house Coopers*, 2016. Disponível em: <<https://www.pwc.pl/pl/pdf/clarity-from-above-pwc.pdf>>.
- MCCARTHY, N. *The industries where drones could really take off [infographic]*. Forbes Magazine, 2016. Disponível em: <<https://www.forbes.com/sites/niallmccarthy/2016/09/02/the-industries-where-drones-could-really-take-off-infographic/?sh=690a4c884e33>>. Acesso em: 27 jan. 2022.
- MLADENOVIĆ, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, n. 11, p. 1097–1100, 1997. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0305054897000312>>.
- MURRAY, C. C.; CHU, A. G. The flying sidekick traveling salesman problem: Optimization of drone-assisted parcel delivery. *Transportation Research Part C: Emerging Technologies*, v. 54, p. 86–109, 2015. ISSN 0968-090X. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0968090X15000844>>.
- PRAIS, M.; RIBEIRO, C. Parameter variation in grasp procedures. *Investigación Operativa*, v. 9, p. 1–20, 01 2000.
- PUGLIESE, L. D. P.; MACRINA, G.; GUERRIERO, F. Trucks and drones cooperation in the last-mile delivery process. *Networks*, v. 78, n. 4, p. 371–399, 2021. Disponível em: <<https://onlinelibrary.wiley.com/doi/abs/10.1002/net.22015>>.
- RANIERI, L. et al. A review of last mile logistics innovations in an externalities cost reduction vision. *Sustainability*, v. 10, n. 3, 2018. ISSN 2071-1050. Disponível em: <<https://www.mdpi.com/2071-1050/10/3/782>>.
- SCHERMER, D.; MOEINI, M.; WENDT, O. Algorithms for solving the vehicle routing problem with drones. In: NGUYEN, N. T. et al. (Ed.). *Intelligent Information and Database Systems*. Cham: Springer International Publishing, 2018. p. 352–361. ISBN 978-3-319-75417-8.
- SCHNEIDER, D. Air traffic control for delivery drones [top tech 2017]. *IEEE Spectrum*, v. 54, n. 1, p. 32–33, January 2017. ISSN 1939-9340.
- SOLAYMANI, S. Co2 emissions patterns in 7 top carbon emitter economies: The case of transport sector. *Energy*, v. 168, p. 989–1001, 2019. ISSN 0360-5442. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0360544218323569>>.
- SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265, 1987. Disponível em: <<https://doi.org/10.1287/opre.35.2.254>>.

SUMMERS, N. *Alphabet's wing launches Drone Delivery Service in Australia*. 2019. Disponível em: <<https://www.engadget.com/2019-04-09-alphabet-wing-drone-delivery-service-australia.html>>. Acesso em: 28 nov. 2021.

VARGHESE, A. et al. Power infrastructure monitoring and damage detection using drone captured images. In: *2017 International Joint Conference on Neural Networks (IJCNN)*. [S.l.: s.n.], 2017. p. 1681–1687. ISSN 2161-4407.

WANG, X.; POIKONEN, S.; GOLDEN, B. The vehicle routing problem with drones: several worst-case results. *Optimization Letters*, v. 11, n. 4, p. 679–697, Apr 2017. ISSN 1862-4480. Disponível em: <<https://doi.org/10.1007/s11590-016-1035-3>>.

WANG, Z.; SHEU, J.-B. Vehicle routing problem with drones. *Transportation Research Part B: Methodological*, v. 122, p. 350–364, 2019. ISSN 0191-2615. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0191261518307884>>.

WELCH, A. A cost-benefit analysis of amazon prime air. University of Tennessee at Chattanooga, 2015. Disponível em: <<https://scholar.utc.edu/honors-theses/47>>.

WOHLSEN, M. The next big thing you missed: Amazon's delivery drones could work—they just need trucks. *Wired: Business*, 2014. Disponível em: <<https://www.wired.com/2014/06/the-next-big-thing-you-missed-delivery-drones-launched-from-trucks-are-the-future-of-shipping/>>. Acesso em: 28 nov. 2021.

A Apêndices

A.1 Resultados Detalhados do MIP

Tabela A.1: Resultado detalhado do MIP

Instância	Custo Médio		Tempo Médio (s)	
	Antes Combinar Rotas	Melhoria após Mip (%)	Execução do MIP	Total
C201_025	5581,9000	0,000%	0,0878	0,8544
C201_050	10014,9333	-4,718%	0,2885	3,6072
C201_100	19253,8667	-7,372%	5,3101	25,6449
C202_025	5630,5667	-0,088%	0,1021	1,0152
C202_050	9323,2667	-4,417%	0,3050	4,5655
C202_100	18505,7333	-9,386%	4,2143	36,8408
C203_025	5279,2667	-1,613%	0,1138	1,2104
C203_050	9102,4333	-6,103%	0,3273	6,0991
C203_100	17320,5000	-8,032%	2,1817	51,3357
C204_025	4151,4667	-0,261%	0,1061	1,3148
C204_050	8383,5667	-4,502%	0,3930	9,3545
C204_100	18290,0000	-15,281%	2,1521	87,0529
C205_025	5135,6667	-6,542%	0,0871	0,9360
C205_050	9355,7000	-4,935%	0,2480	3,9528
C205_100	15637,2667	-5,473%	1,2421	28,9494
C206_025	4452,4333	-1,018%	0,0805	0,9699
C206_050	8594,9000	-4,597%	0,2199	4,7130
C206_100	15982,2000	-7,676%	0,9079	35,1558
C207_025	4553,7000	-0,666%	0,0882	1,1925
C207_050	8689,3333	-5,017%	0,2366	5,8301
C207_100	16685,1333	-5,886%	1,9166	39,7636
C208_025	4448,2333	-0,414%	0,0813	1,0625
C208_050	8369,1333	-2,397%	0,2468	5,2751
C208_100	16940,0000	-8,787%	1,4454	41,1616
R201_025	10016,8333	-2,700%	0,1597	1,1167
R201_050	14843,5667	-5,694%	3,8264	7,7803
R201_100	17346,2667	-0,150%	4,5316	28,7349
R202_025	8430,4000	-8,514%	0,1415	1,3045
R202_050	13512,6667	-10,625%	1,0079	5,9320
R202_100	16487,2333	-0,001%	3,8392	33,3281
R203_025	8056,7333	-8,763%	0,1640	1,4400
R203_050	12571,1667	-9,633%	1,5607	7,5935
R203_100	16576,5333	-0,005%	6,6317	45,5120
R204_025	7132,3333	-8,240%	0,2619	1,6355

Instância	Custo Médio		Tempo Médio (s)	
	Antes Combinar Rotas	Melhoria após Mip (%)	Execução do MIP	Total
R204_050	9453,5333	-5,886%	0,4025	8,0320
R204_100	14058,0667	-0,265%	2,1599	67,4643
R205_025	6733,2000	-1,540%	0,1247	1,1996
R205_050	10769,2000	0,000%	0,4127	4,8631
R205_100	15852,3667	0,000%	1,3376	32,8554
R206_025	6649,5000	-6,751%	0,1459	1,3045
R206_050	10514,1000	-2,726%	0,7968	6,2226
R206_100	15370,7667	0,000%	2,1840	39,2136
R207_025	6872,3000	-9,210%	0,2195	1,4427
R207_050	9994,9000	-1,493%	0,8458	7,1512
R207_100	15033,3667	-0,180%	2,5914	48,1068
R208_025	6575,5333	-12,937%	0,2399	1,5778
R208_050	9195,5000	-4,430%	0,4297	7,7672
R208_100	14049,9667	-0,350%	3,2768	74,0063
R209_025	6110,0333	-0,160%	0,0882	1,0251
R209_050	10815,0000	-0,029%	0,5685	5,3580
R209_100	15553,8333	-0,061%	1,0779	34,0879
R210_025	8061,1333	-3,837%	0,1534	1,2661
R210_050	10869,2000	-0,642%	0,6052	5,6883
R210_100	15560,9667	0,000%	2,1628	35,9247
R211_025	5293,4333	-0,703%	0,1059	1,1546
R211_050	8395,5000	-0,012%	0,2353	5,5338
R211_100	13737,1000	0,000%	1,0802	42,9738
RC201_025	8668,9333	-1,793%	0,1057	1,0310
RC201_050	18621,1333	-16,493%	0,5981	4,8663
RC201_100	27214,1000	-0,173%	268,9678	293,9409
RC202_025	7722,6333	-3,047%	0,1198	1,2335
RC202_050	16045,5667	-12,337%	0,4102	5,1394
RC202_100	22910,6667	-0,399%	26,1491	56,5909
RC203_025	7848,0000	-2,257%	0,1478	1,2541
RC203_050	14966,7667	-12,282%	0,3739	5,7335
RC203_100	20026,0333	-0,934%	4,4575	42,4806
RC204_025	6342,3333	-0,488%	0,1113	1,3145
RC204_050	11890,8333	-5,171%	0,3426	6,3057
RC204_100	18227,2333	-1,616%	2,7755	60,3128
RC205_025	8328,9667	-3,495%	0,0975	1,0378
RC205_050	15477,7000	-12,798%	0,2913	4,7179
RC205_100	25594,5000	-0,904%	188,0433	218,1379
RC206_025	7015,3333	-2,115%	0,1036	1,1026
RC206_050	14485,0000	-7,631%	0,4537	4,8357
RC206_100	22009,8333	0,000%	4,9828	33,2205
RC207_025	6656,1667	-0,828%	0,0951	1,1254
RC207_050	13865,6667	-6,276%	0,3067	4,7567
RC207_100	19368,0333	-0,004%	1,9788	32,3707

Instância	Custo Médio		Tempo Médio (s)	
	Antes Combinar Rotas	Melhoria após Mip (%)	Execução do MIP	Total
RC208_025	5813,7333	0,000%	0,0745	1,0698
RC208_050	11353,1333	-2,223%	0,2196	4,5580
RC208_100	18553,3000	0,000%	2,4138	39,6000

A.2 Custo Médio por Algoritmo

Tabela A.2: Custo Médio por Algoritmo

Instância	ML	Custo Médio (%)					
		L1	L2	L3	L4	MIP	GRASP
C201_025	5773,73	20,087	8,935	1,191	0,000	-3,323	-1,142
C201_050	9370,53	35,783	21,152	4,193	0,000	1,834	7,218
C201_100	13866,87	52,459	39,356	2,821	0,000	28,612	36,348
C202_025	5465,97	15,311	9,405	1,498	0,000	2,920	3,229
C202_050	8975,73	27,467	16,198	1,109	0,000	-0,716	3,522
C202_100	13781,57	38,628	27,532	0,448	0,000	21,675	30,498
C203_025	4731,60	29,682	22,079	7,296	0,000	9,775	11,837
C203_050	8816,57	30,377	20,682	4,282	0,000	-3,059	4,965
C203_100	16040,90	64,545	48,832	8,685	0,000	-0,696	17,498
C204_025	4044,23	11,021	2,924	9,934	0,000	2,384	4,244
C204_050	8122,93	22,693	18,887	7,384	0,000	-1,437	3,484
C204_100	17557,77	53,973	39,902	8,007	0,000	-11,748	5,434
C205_025	4872,57	31,906	24,442	7,661	0,000	-1,495	6,863
C205_050	8226,90	12,665	5,412	1,268	0,000	8,108	12,853
C205_100	14048,27	63,847	54,976	10,071	0,000	5,219	11,304
C206_025	4555,30	19,391	11,882	0,952	0,000	-3,253	-1,968
C206_050	8054,63	7,180	3,118	1,065	0,000	1,803	5,897
C206_100	13619,07	50,217	31,077	7,365	0,000	8,343	17,392
C207_025	4484,90	22,423	9,982	2,412	0,000	0,858	2,337
C207_050	8301,67	32,571	23,004	5,072	0,000	-0,581	4,300
C207_100	14213,53	46,333	27,470	7,073	0,000	10,479	17,662
C208_025	4430,93	18,401	9,434	2,012	0,000	-0,026	0,324
C208_050	8081,53	36,638	27,232	2,580	0,000	1,077	3,034
C208_100	14510,87	51,563	21,248	6,027	0,000	6,482	18,265
R201_025	9647,23	23,960	19,317	4,469	0,000	1,028	4,925
R201_050	13617,10	24,177	16,022	5,704	0,000	2,800	8,509
R201_100	17273,87	28,404	23,000	6,891	0,000	0,269	1,010
R202_025	7619,70	42,493	34,724	6,823	0,000	1,219	10,665
R202_050	11940,57	27,059	20,763	5,317	0,000	1,142	14,303
R202_100	15930,83	19,452	8,424	5,155	0,000	3,491	3,337
R203_025	7883,73	33,646	29,433	8,615	0,000	-6,761	2,060
R203_050	10395,03	41,201	33,607	9,604	0,000	9,285	19,632
R203_100	15546,37	19,152	8,895	3,642	0,000	6,621	7,060
R204_025	7279,17	35,017	29,199	7,424	0,000	-10,091	-2,499
R204_050	9296,07	24,898	17,496	6,397	0,000	-4,291	2,256
R204_100	11789,80	16,807	6,490	7,397	0,000	18,923	18,490
R205_025	6392,77	37,769	23,337	5,822	0,000	3,704	5,097
R205_050	10665,03	12,792	8,095	3,376	0,000	0,977	1,350
R205_100	14263,63	5,226	1,810	4,163	0,000	11,138	11,907
R206_025	6569,13	26,152	14,524	5,362	0,000	-5,610	1,564

Instância	ML	Custo Médio (%)					
		L1	L2	L3	L4	MIP	GRASP
R206_050	9945,77	17,907	7,632	6,537	0,000	2,833	4,880
R206_100	15001,30	13,835	4,016	3,730	0,000	2,463	1,405
R207_025	6961,90	30,143	15,183	4,426	0,000	-10,378	-2,819
R207_050	9509,07	29,287	17,522	7,738	0,000	3,540	5,138
R207_100	13847,63	24,515	12,241	7,376	0,000	8,367	9,151
R208_025	6977,20	21,180	14,815	7,305	0,000	-17,949	-6,147
R208_050	8462,87	18,595	10,831	7,792	0,000	3,843	8,349
R208_100	10870,63	15,054	8,048	9,934	0,000	28,794	29,589
R209_025	6130,07	13,857	5,943	1,958	0,000	-0,487	-0,415
R209_050	10172,23	14,849	8,303	5,189	0,000	6,288	6,539
R209_100	14443,00	13,259	7,056	4,237	0,000	7,626	8,528
R210_025	7694,60	34,412	21,016	5,746	0,000	0,744	4,339
R210_050	9963,60	18,591	9,314	8,476	0,000	8,388	8,131
R210_100	14231,93	23,785	7,728	7,312	0,000	9,338	9,316
R211_025	6090,17	15,006	6,747	5,860	0,000	-13,694	-11,829
R211_050	8254,93	8,685	3,063	7,238	0,000	1,690	2,039
R211_100	11653,73	9,404	3,860	5,874	0,000	17,877	18,232
RC201_025	8356,97	54,570	34,538	7,779	0,000	1,873	4,560
RC201_050	15597,90	64,794	49,908	11,120	0,000	-0,308	18,248
RC201_100	24567,43	47,150	40,280	6,812	0,000	10,581	12,178
RC202_025	7153,90	72,129	47,347	9,414	0,000	4,661	10,652
RC202_050	14256,53	75,791	58,712	8,026	0,000	-1,336	17,072
RC202_100	21552,80	33,643	24,609	11,682	0,000	5,876	7,472
RC203_025	7697,60	45,396	33,341	9,188	0,000	-0,347	2,933
RC203_050	13449,60	68,279	54,774	10,551	0,000	-2,387	12,712
RC203_100	20143,60	24,709	17,349	10,796	0,000	-1,512	-0,539
RC204_025	6572,17	40,565	17,256	8,809	0,000	-3,968	-3,675
RC204_050	11957,40	12,130	6,829	3,806	0,000	-5,699	1,956
RC204_100	18533,47	17,299	12,381	6,345	0,000	-3,241	-2,898
RC205_025	8286,53	17,519	11,817	2,998	0,000	-3,001	1,071
RC205_050	14058,67	67,102	45,744	11,055	0,000	-3,996	11,843
RC205_100	24002,03	24,362	18,497	5,530	0,000	5,671	6,707
RC206_025	7311,37	47,429	29,109	10,020	0,000	-6,079	-5,429
RC206_050	14062,93	48,115	36,262	9,250	0,000	-4,859	1,065
RC206_100	23431,63	16,440	10,801	3,774	0,000	-6,068	-5,747
RC207_025	6992,17	27,995	10,825	5,243	0,000	-5,594	-5,122
RC207_050	13854,73	39,745	26,933	8,836	0,000	-6,202	1,993
RC207_100	21057,70	29,165	23,346	9,562	0,000	-8,028	-7,698
RC208_025	5828,97	14,695	2,199	6,594	0,000	-0,261	-0,755
RC208_050	11212,23	21,170	9,801	8,146	0,000	-0,994	1,196
RC208_100	18461,43	15,561	10,773	4,696	0,000	0,498	1,169
Média	11218,14	30,389	19,964	6,214	0,000	1,749	6,302
Desvio Padrão	4900,29	16,820	13,921	2,794	0,000	8,034	8,504

A.3 Melhor resultado obtido por algoritmo

Tabela A.3: Melhor resultado por Algoritmo

Instância	ML	Melhor Custo (%)					
		L1	L2	L3	L4	MIP	GRASP
C201_025	5741	7,403	6,967	0,000	0,000	-2,787	-2,787
C201_050	9133	7,577	7,577	1,029	0,000	0,887	4,424
C201_100	13707	23,966	23,141	0,467	0,000	11,979	11,118
C202_025	5142	13,963	13,633	0,000	1,381	7,682	7,915
C202_050	8687	7,908	7,908	2,705	0,000	-2,671	2,463
C202_100	13614	22,146	17,989	0,610	0,000	5,384	12,046
C203_025	4638	20,979	19,340	1,315	0,000	1,790	11,923
C203_050	8486	19,927	19,927	3,040	0,000	-5,491	5,303
C203_100	15294	51,589	40,473	1,347	0,000	-7,663	12,037
C204_025	3966	0,933	0,681	0,126	0,000	0,656	0,530
C204_050	7504	24,787	24,174	8,089	0,000	-1,666	5,237
C204_100	13784	61,615	56,355	0,000	18,326	5,238	20,647
C205_025	4803	24,526	23,444	0,271	0,000	-4,102	3,664
C205_050	8200	3,329	0,756	0,024	0,000	2,768	3,646
C205_100	13361	54,367	53,177	0,786	0,000	3,405	4,334
C206_025	4389	9,000	6,699	0,000	0,456	-0,023	0,046
C206_050	8005	1,874	1,349	0,012	0,000	-2,461	0,874
C206_100	13032	37,185	24,969	0,023	0,000	5,955	10,704
C207_025	4348	6,348	6,348	0,000	0,000	0,437	0,230
C207_050	8080	18,688	18,688	0,173	0,000	-5,433	-1,757
C207_100	13185	26,356	26,174	1,896	0,000	8,017	9,640
C208_025	4361	7,957	2,568	0,046	0,000	0,092	0,115
C208_050	8005	22,386	22,311	0,025	0,000	-2,136	-0,462
C208_100	13010	28,555	25,035	0,192	0,000	4,427	13,051
R201_025	9452	9,278	9,278	0,222	0,000	-3,417	4,771
R201_050	12794	10,607	10,607	4,791	0,000	4,682	11,122
R201_100	16269	25,865	21,212	3,891	0,000	1,666	-1,438
R202_025	7419	26,230	22,618	0,135	0,000	-1,496	1,847
R202_050	10595	28,193	20,595	9,750	0,000	9,929	19,811
R202_100	14842	13,927	5,269	2,479	0,000	1,428	2,048
R203_025	7172	25,195	24,916	0,474	0,000	0,181	6,637
R203_050	9364	32,315	32,315	6,717	0,000	13,424	19,233
R203_100	14477	7,957	6,721	0,000	0,767	5,650	8,137
R204_025	6676	34,362	18,859	3,655	0,000	-5,917	-2,397
R204_050	8214	21,804	21,804	0,000	4,541	2,873	7,524
R204_100	11139	3,708	3,016	2,918	0,000	16,079	11,141
R205_025	6070	16,491	14,498	0,049	0,000	0,148	0,478
R205_050	10370	2,285	2,054	0,183	0,000	-0,656	-0,511
R205_100	14066	2,837	2,645	1,635	0,000	5,951	4,003
R206_025	6062	12,290	11,333	0,000	0,000	-1,881	4,635

Instância	ML	Melhor Custo (%)					
		L1	L2	L3	L4	MIP	GRASP
R206_050	9222	8,913	8,903	2,266	0,000	2,017	6,745
R206_100	14078	6,073	2,401	1,477	0,000	-2,365	-0,980
R207_025	6080	12,944	11,793	4,243	0,000	-3,158	1,086
R207_050	8634	23,940	14,871	5,328	0,000	6,984	9,578
R207_100	12240	13,088	13,088	0,090	0,000	15,801	8,766
R208_025	6226	20,270	8,962	3,534	0,000	-14,375	-7,485
R208_050	8067	9,161	3,397	4,673	0,000	5,343	3,595
R208_100	10652	3,258	3,258	0,385	0,000	23,817	20,081
R209_025	6086	0,723	0,723	0,000	0,115	-0,033	-1,660
R209_050	9810	5,260	5,260	1,040	0,000	4,873	5,280
R209_100	14201	4,211	3,183	0,000	0,000	4,648	5,633
R210_025	7036	34,011	19,315	3,368	0,000	5,884	7,717
R210_050	8834	18,553	9,826	0,000	0,000	16,787	6,079
R210_100	13484	4,769	2,796	3,063	0,000	8,254	4,613
R211_025	5374	9,062	0,000	11,444	1,358	-5,489	-5,359
R211_050	7899	1,836	1,393	0,544	0,000	3,481	2,937
R211_100	11530	1,180	1,180	0,390	0,000	12,697	9,063
RC201_025	8169	4,125	4,125	0,061	0,000	0,906	0,820
RC201_050	14293	56,342	37,704	0,000	0,147	2,484	11,040
RC201_100	22324	49,162	36,597	2,450	0,000	11,727	14,953
RC202_025	6664	37,530	34,274	0,000	0,000	4,412	8,298
RC202_050	13226	59,769	44,723	0,703	0,000	3,811	11,992
RC202_100	19616	35,338	21,008	2,345	0,000	4,563	6,301
RC203_025	6682	45,346	44,897	4,579	0,000	8,007	8,126
RC203_050	12772	54,087	43,697	1,597	0,000	1,425	10,507
RC203_100	19112	13,630	12,777	3,987	0,000	-1,816	-3,872
RC204_025	6256	9,495	9,495	1,151	0,000	-0,096	-0,032
RC204_050	11251	11,181	6,862	0,000	0,587	-3,457	0,142
RC204_100	16687	18,248	16,798	5,286	0,000	2,721	2,900
RC205_025	8071	11,721	11,609	0,012	0,000	-1,338	0,074
RC205_050	13441	41,961	21,271	0,573	0,000	-1,652	3,876
RC205_100	22914	18,740	14,546	4,547	0,000	4,163	-0,960
RC206_025	6614	18,657	11,612	0,665	0,000	-1,134	-1,089
RC206_050	12434	15,433	15,433	4,069	0,000	3,482	8,453
RC206_100	20919	23,103	13,003	8,643	0,000	-2,204	-3,776
RC207_025	6615	7,725	7,725	1,572	0,000	-3,084	-3,084
RC207_050	12011	33,952	32,062	13,105	0,000	1,382	6,561
RC207_100	19191	27,039	24,704	5,331	0,000	-5,966	-4,862
RC208_025	5603	0,500	0,125	0,125	0,000	2,606	2,606
RC208_050	10624	4,452	4,452	0,857	0,000	-2,297	-0,377
RC208_100	17543	10,027	10,027	2,679	0,000	-0,462	-0,114
Média	10493,10	19,278	15,646	2,040	0,342	2,300	4,767
Desvio Padrão	4472,42	15,537	13,049	2,759	2,097	6,094	6,069

A.4 Tempo Médio por Algoritmo

Tabela A.4: Tempo Médio por Algoritmo

Instância	Tempo Médio (s)					
	L1	L2	L3	L4	MIP	GRASP
C201_025	<0.001	0,002	0,016	0,119	0,854	0,625
C201_050	0,001	0,005	0,369	2,682	3,607	2,519
C201_100	0,004	0,017	4,299	21,299	25,645	14,633
C202_025	<0.001	0,001	0,018	0,102	1,015	0,807
C202_050	0,001	0,004	0,417	3,247	4,566	3,680
C202_100	0,005	0,014	4,583	22,971	36,841	26,639
C203_025	<0.001	0,001	0,038	0,254	1,210	0,980
C203_050	0,002	0,005	0,412	3,664	6,099	5,080
C203_100	0,003	0,014	6,691	53,101	51,336	38,692
C204_025	<0.001	0,001	0,026	0,170	1,315	1,066
C204_050	0,001	0,004	0,304	3,224	9,355	7,873
C204_100	0,003	0,013	8,027	78,819	87,053	65,953
C205_025	<0.001	0,002	0,023	0,140	0,936	0,677
C205_050	0,001	0,004	0,289	2,753	3,953	2,690
C205_100	0,004	0,014	6,163	46,558	28,949	20,148
C206_025	0,001	0,001	0,016	0,116	0,970	0,723
C206_050	0,001	0,004	0,290	2,978	4,713	3,432
C206_100	0,003	0,013	5,478	48,788	35,156	27,032
C207_025	0,001	0,002	0,021	0,117	1,192	0,960
C207_050	0,001	0,004	0,389	2,727	5,830	4,559
C207_100	0,004	0,014	5,538	46,867	39,764	30,451
C208_025	<0.001	0,001	0,019	0,123	1,063	0,814
C208_050	0,001	0,005	0,533	3,895	5,275	4,122
C208_100	0,003	0,013	5,321	53,011	41,162	32,878
R201_025	0,001	0,001	0,036	0,221	1,117	0,730
R201_050	0,001	0,004	0,175	1,510	7,780	2,726
R201_100	0,004	0,012	1,374	11,520	28,735	15,551
R202_025	0,001	0,002	0,047	0,295	1,305	0,930
R202_050	0,001	0,004	0,247	2,064	5,932	3,685
R202_100	0,004	0,014	2,342	19,631	33,328	21,217
R203_025	0,001	0,002	0,040	0,286	1,440	1,057
R203_050	0,001	0,004	0,318	2,581	7,593	4,924
R203_100	0,004	0,013	4,052	29,477	45,512	29,036
R204_025	0,001	0,001	0,040	0,311	1,636	1,163
R204_050	0,001	0,003	0,247	2,207	8,032	6,092
R204_100	0,004	0,012	3,108	29,948	67,464	39,078
R205_025	<0.001	0,002	0,034	0,206	1,200	0,820
R205_050	0,001	0,003	0,157	1,167	4,863	2,916
R205_100	0,004	0,011	0,869	12,309	32,855	22,334
R206_025	0,001	0,002	0,037	0,282	1,305	0,895

Instância	Tempo Médio (s)					
	L1	L2	L3	L4	MIP	GRASP
R206_050	0,001	0,004	0,209	1,789	6,223	3,908
R206_100	0,004	0,012	2,367	20,590	39,214	27,540
R207_025	0,001	0,002	0,042	0,233	1,443	1,005
R207_050	0,001	0,004	0,276	2,208	7,151	4,973
R207_100	0,003	0,012	4,049	30,537	48,107	35,293
R208_025	<0.001	0,001	0,032	0,239	1,578	1,115
R208_050	0,001	0,004	0,214	2,059	7,767	5,582
R208_100	0,003	0,012	2,388	27,589	74,006	43,457
R209_025	<0.001	0,001	0,019	0,128	1,025	0,703
R209_050	0,001	0,003	0,234	1,783	5,358	3,516
R209_100	0,003	0,013	1,334	14,212	34,088	24,903
R210_025	<0.001	0,002	0,045	0,358	1,266	0,941
R210_050	0,001	0,004	0,224	2,200	5,688	3,850
R210_100	0,004	0,014	2,282	25,778	35,925	24,832
R211_025	0,001	0,001	0,025	0,170	1,155	0,760
R211_050	0,001	0,003	0,084	1,019	5,534	3,495
R211_100	0,004	0,012	1,071	13,074	42,974	32,036
RC201_025	0,001	0,002	0,030	0,246	1,031	0,726
RC201_050	0,001	0,005	0,282	2,253	4,866	3,093
RC201_100	0,003	0,011	2,995	17,648	293,941	16,710
RC202_025	<0.001	0,002	0,033	0,202	1,234	0,935
RC202_050	0,001	0,005	0,534	3,390	5,139	3,720
RC202_100	0,003	0,014	3,200	32,752	56,591	22,082
RC203_025	0,001	0,002	0,039	0,285	1,254	0,933
RC203_050	0,001	0,005	0,497	3,521	5,734	4,402
RC203_100	0,004	0,013	3,885	37,168	42,481	28,801
RC204_025	<0.001	0,002	0,028	0,256	1,315	1,033
RC204_050	0,001	0,003	0,196	1,814	6,306	4,839
RC204_100	0,004	0,011	3,155	28,488	60,313	36,873
RC205_025	<0.001	0,001	0,020	0,161	1,038	0,763
RC205_050	0,001	0,005	0,322	2,784	4,718	3,556
RC205_100	0,003	0,010	2,875	25,398	218,138	22,177
RC206_025	0,001	0,001	0,035	0,249	1,103	0,809
RC206_050	0,001	0,004	0,331	2,347	4,836	3,235
RC206_100	0,003	0,011	1,914	14,241	33,221	19,673
RC207_025	0,001	0,002	0,034	0,213	1,125	0,839
RC207_050	0,001	0,004	0,335	2,425	4,757	3,511
RC207_100	0,004	0,011	2,694	22,654	32,371	22,651
RC208_025	0,001	0,002	0,020	0,148	1,070	0,761
RC208_050	0,001	0,004	0,231	1,907	4,558	3,194
RC208_100	0,003	0,010	1,818	15,678	39,600	28,394