

**MÉTODOS HÍBRIDOS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM
JANELAS DE TEMPO E MÚLTIPLOS ENTREGADORES**

Aldair Alberto Álvarez Díaz

São Carlos
Fevereiro de 2016

**MÉTODOS HÍBRIDOS PARA O PROBLEMA DE ROTEAMENTO DE VEÍCULOS COM
JANELAS DE TEMPO E MÚLTIPLOS ENTREGADORES**

Aldair Alberto Álvarez Díaz

Orientador: Prof. Dr. Pedro Munari

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Engenharia de Produção, da Universidade Federal de São Carlos como parte dos requisitos para a obtenção do título de Mestre em Engenharia de Produção.

São Carlos
Fevereiro de 2016

Ficha catalográfica elaborada pelo DePT da Biblioteca Comunitária UFSCar
Processamento Técnico
com os dados fornecidos pelo(a) autor(a)

A473m Álvarez Díaz, Aldair Alberto
Métodos híbridos para o problema de roteamento de
veículos com janelas de tempo e múltiplos
entregadores / Aldair Alberto Álvarez Díaz. -- São
Carlos : UFSCar, 2016.
106 p.

Dissertação (Mestrado) -- Universidade Federal de
São Carlos, 2016.

1. Roteamento de veículos. 2. Múltiplos
entregadores. 3. Métodos híbridos. 4. Branch-price-
and-cut. 5. Metaheurísticas. I. Título.



UNIVERSIDADE FEDERAL DE SÃO CARLOS

Centro de Ciências Exatas e de Tecnologia
Programa de Pós-Graduação em Engenharia de Produção

Folha de Aprovação

Assinaturas dos membros da comissão examinadora que avaliou e aprovou a Defesa de Dissertação de Mestrado do candidato Aldair Alberto Álvarez Diaz, realizada em 29/02/2016:

Pedro Augusto Munari Junior

Prof. Dr. Pedro Augusto Munari Junior
UFSCar

Eduardo Uchoa Barboza

Prof. Dr. Eduardo Uchoa Barboza
UFF

UFF

Reinaldo Morabito Neto

Prof. Dr. Reinaldo Morabito Neto
UFSCar

*Aos meus amados pais, Delcy e Julio,
aos meus adorados irmãos, Karen e Julio Jr.,
ao meu amor Karen,
e a minha avó Argenida (in memoriam).*

Vocês são meu maior impulso, a vocês dedico este sonho alcançado.

Agradecimentos

A Deus por mais uma oportunidade de crescimento pessoal e profissional, por cuidar de mim enquanto estive longe de casa e me dar as forças necessárias durante todo este tempo.

Aos meus pais Delcy e Julio e aos meus irmãos Karen e Julio Jr., pelo amor e apoio em todos os momentos. Devo tudo o que sou e tenho a eles. Apesar da distância estão sempre presentes, sendo meu maior presente e motivo para perseverar.

Agradeço especialmente ao meu orientador, Prof. Dr. Pedro Munari, por ser mais que orientador, pelos conselhos e conversas que me fizeram crescer e me tornar melhor tanto como pessoa quanto como pesquisador. Muito obrigado pela oportunidade de trabalhar ao seu lado, pela competência com que conduziu esta pesquisa e por estar sempre disponível para perguntas e por responder a todas elas.

À minha namorada Karen, pelo carinho, compreensão e apoio incondicionais. Obrigado por ser meu par perfeito, por me ajudar a melhorar cada dia e sempre ver o melhor de mim. Agradeço também a sua família pela admiração e apoio.

Aos professores da pós-graduação, em especial ao Prof. Dr. Reinaldo Morabito, ao Prof. Dr. Moacir Godinho e ao Prof. Dr. Roberto Tavares, pelos ensinamentos durante este período no Brasil. Agradeço também aos funcionários do DEP-UFSCar que de alguma forma contribuíram para a elaboração deste trabalho. Em especial ao Robson, Natália e Larissa, por sempre estarem dispostos a ajudar.

Aos meus amigos Yeyo e Luchy, pela amizade de toda a vida, por estarem ao meu lado, torcendo por mim mesmo nos momentos de ausência.

A todos meus amigos e colegas do Grupo de Pesquisa Operacional (GPO) da UFSCar que fizeram minha vida mais fácil e feliz durante o mestrado. Agradecimento especial para Alyne, Amélia, Cesar, Gabi, Juan, Karim, Pedro e Tamara.

Ao Prof. Jota, quem me motivou nos inícios desta carreira acadêmica na graduação.

À CAPES pelo apoio financeiro, sem o qual esta pesquisa não teria sido realizada.

*“To persevere is important for everybody.
Don’t give up, don’t give in. There’s always
an answer to everything.”*

Louis Zamperini

Resumo

Nesta dissertação aborda-se o problema de roteamento de veículos com janelas de tempo e múltiplos entregadores, uma variante do problema de roteamento de veículos recentemente proposta na literatura que, além das decisões de programação e roteamento, envolve a determinação do tamanho da tripulação de cada veículo. Esse problema surge na distribuição de bens em centros urbanos congestionados em que, devido aos tempos de serviço relativamente longos, pode ser difícil atender a todos os clientes durante o horário normal de trabalho. Diante dessa dificuldade, uma alternativa consiste em incluir a designação de entregadores extras, o que gera custos adicionais aos custos tradicionais de deslocamento e utilização de veículos. Neste problema, o objetivo é definir rotas para atender grupos de clientes minimizando o número de veículos usados, o número total de entregadores designados e a distância total percorrida. Para tratar o problema, são desenvolvidos diferentes métodos de solução. Inicialmente, são apresentadas duas abordagens metaheurísticas baseadas em Busca Local Iterada e Busca em Vizinhança Grande. Posteriormente, são propostos métodos híbridos de solução a partir da combinação dessas metaheurísticas com um método *branch-price-and-cut*. Experimentos computacionais usando instâncias encontradas na literatura confirmam a eficiência dos métodos de solução desenvolvidos para o problema.

Palavras-chave: Roteamento de veículos, múltiplos entregadores, métodos híbridos, *branch-price-and-cut*, metaheurísticas.

Abstract

In this dissertation we address the vehicle routing problem with time windows and multiple deliverymen, a variant of the vehicle routing problem that involves the additional decision of the crew size definition of the vehicles, besides scheduling and routing decisions. This problem arises in the distribution of goods in highly congested urban areas, where due to the relatively long service times, it may be difficult to serve all clients during regular working hours. Given this difficulty, an alternative consists in including the deliverymen assignment decision, which leads to extra costs in addition to travel and vehicle costs. The objective is to define routes to serve customer clusters minimizing the number of vehicles used, the number of allocated deliverymen and the traveled distance. In this study, we develop different solution methods to solve this problem. Initially, we present two metaheuristic approaches, which are based on Iterated Local Search and Large Neighborhood Search. Then we propose hybrid methods, combining these metaheuristics with a branch-price-and-cut method. Computational experiments using instances from the literature confirm the efficiency of the solution methods developed for the problem.

Key words: Vehicle routing, multiple deliverymen, hybrid methods, branch-price-and-cut, metaheuristics.

Lista de Figuras

1.1	Exemplo de uma rota típica no PRVME.	22
2.1	Estrutura básica de uma classe de método híbrido.	30
3.1	Movimento <i>Shift(2,0)</i>	43
3.2	Movimento <i>Swap(1,1)</i>	43
3.3	Movimento <i>Or-opt-1</i>	43
3.4	Movimento <i>2-opt</i>	43
4.1	Estrutura do método híbrido exato.	59
5.1	Gráfico de médias e intervalos de confiança de 95% de <i>Tukey</i> para os parâmetros da ILS.	67
5.2	Gráfico de médias e intervalos de confiança de 95% de <i>Tukey</i> para os parâmetros da LNS.	67
5.3	<i>Branch-price-and-cut</i> com heurística primal vs método híbrido exato nas instâncias tipo 1.	77
5.4	<i>Branch-price-and-cut</i> com heurística primal vs método híbrido exato nas instâncias tipo 2.	77
5.5	Método híbrido exato vs método híbrido com gerador de colunas híbrido nas instâncias tipo 1.	81
5.6	Método híbrido exato vs método híbrido com gerador de colunas híbrido nas instâncias tipo 2.	81
A.1	Solução para a instância C107.	94
A.2	Soluções para a instância R101.	95
A.3	Soluções para a instância RC102.	96
A.4	Solução para a instância C205.	97
A.5	Soluções para a instância R203.	98
A.6	Soluções para a instância RC205.	99

Lista de Tabelas

2.1	Características de métodos híbridos propostos na literatura para PRVs.	38
5.1	Fatores e níveis considerados no planejamento experimental.	66
5.2	Resultados das abordagens metaheurísticas para as instâncias tipo 1 (600 segundos).	68
5.3	Resultados das abordagens metaheurísticas para as instâncias tipo 2 (600 segundos).	69
5.4	Melhores resultados (agrupados) das abordagens metaheurísticas vs métodos da literatura.	70
5.5	Diferenças relativas entre os melhores resultados das abordagens metaheurísticas.	70
5.6	Resultados médios (agrupados) das abordagens metaheurísticas vs métodos da literatura.	71
5.7	Melhores resultados obtidos para o conjunto R1.	72
5.8	Método BPC_{PR} vs BPC_{MH} nas instâncias do tipo 1.	75
5.9	Método BPC_{PR} vs BPC_{MH} nas instâncias do tipo 2.	76
5.10	Método BPC_{MH} vs BPC_{GCH} nas instâncias do tipo 1.	79
5.11	Método BPC_{MH} vs BPC_{GCH} nas instâncias do tipo 2.	80
5.12	Soluções encontradas por todos os métodos híbridos nas instâncias do tipo 1.	82
5.13	Soluções encontradas por todos os métodos híbridos nas instâncias do tipo 2.	83
5.14	Fonte das soluções finais nos métodos híbridos.	84
5.15	Melhores resultados (agrupados) dos métodos de solução (600 segundos).	86
5.16	Proporções dos tempos que compõem as rotas nos diferentes tipos de instâncias.	87
5.17	Custo das soluções variando o número de entregadores permitidos a bordo.	88

Sumário

1	Introdução	21
2	Revisão da literatura	25
2.1	O problema de roteamento de veículos com janelas de tempo e múltiplos entregadores	25
2.2	Métodos de solução para o PRVJTME	28
2.3	Métodos híbridos para problemas de roteamento de veículos	29
3	Abordagens metaheurísticas	39
3.1	Heurística construtiva	39
3.2	Abordagem metaheurística baseada em ILS	40
3.2.1	Busca local	41
3.2.2	Perturbação	42
3.2.3	Heurística de redução de rotas	44
3.2.4	Heurística de redução de entregadores	44
3.3	Abordagem metaheurística baseada em LNS	45
3.3.1	Operadores de remoção	45
3.3.2	Operadores de inserção	47
3.4	Considerações finais	48
4	Métodos híbridos	51
4.1	Formulação por particionamento de conjuntos	51
4.2	Método <i>branch-price-and-cut</i>	52
4.2.1	Método de geração de colunas	53
4.2.2	Resolvendo o subproblema	54
4.2.3	Desigualdades válidas	55
4.2.4	Estratégia de ramificação	56
4.3	Heurística primal para o BPC	57
4.4	Método híbrido exato	58

4.5	Método híbrido heurístico	59
4.5.1	Uma metaheurística ILS para o ESPPRC	60
4.6	Considerações finais	62
5	Experimentos computacionais	63
5.1	Instâncias utilizadas	64
5.2	Implementação computacional	65
5.3	Ajuste dos parâmetros das metaheurísticas	66
5.4	Avaliação das abordagens metaheurísticas	67
5.5	Avaliação dos métodos híbridos	73
5.6	Fonte das soluções nos métodos híbridos	84
5.7	Comparação entre os diferentes tipos de métodos desenvolvidos	85
5.8	Análises adicionais	86
5.9	Considerações finais	88
6	Conclusões	89
	Apêndices	91
	Apêndice A Gráficos das soluções	93

Capítulo 1

Introdução

Processos de transporte estão envolvidos de múltiplas formas nos sistemas de produção, especialmente naqueles que tratam atividades de distribuição. Tais processos têm impacto na competitividade e nos níveis de serviço das indústrias. Por exemplo, processos de transporte podem representar entre 10% e 20% dos custos finais dos bens produzidos por uma empresa (TOTH; VIGO, 2002). Além disso, estima-se que os custos de distribuição podem representar até três quartos dos custos de logística de uma organização (BRÄYSY; GENDREAU, 2005), tornando necessário investir esforços para o aprimoramento desses processos. Dentre as atividades de distribuição, surge o problema de roteamento de veículos (PRV), o qual é enfrentado diariamente por muitas empresas que lidam com o transporte de produtos e/ou pessoas. Na prática, este problema desempenha um papel importante dentro dos sistemas de distribuição e, portanto, sua resolução é parte fundamental da gestão eficiente das operações das organizações.

Desde sua introdução por Dantzig e Ramser (1959), o PRV tem sido um dos problemas mais investigados dentro da literatura de Pesquisa Operacional (PO), principalmente por dois motivos. Em primeiro lugar, tem-se sua importância prática no apoio à tomada de decisões, já que surge naturalmente como um problema principal em contextos logísticos. Em segundo lugar, a dificuldade em obter soluções ótimas para instâncias de grande porte para estes problemas estimula o desenvolvimento de novas estratégias de solução. Além disso, dada a complexidade deste problema e a necessidade de uma análise integrada, é essencial que a tomada de decisões neste contexto tenha o apoio de abordagens quantitativas, em particular aquelas oferecidas pela PO.

Devido às diversas características que aparecem nos sistemas de produção do mundo real, o PRV estendeu-se além da simples entrega de bens, gerando um amplo número de variantes do problema. Essas variantes do PRV geralmente são formuladas com base em variações das seguintes características:

- frota de veículos utilizada;
- exigências dos clientes (janelas de tempo, número de entregas permitidas);

- rede de transporte e duração das rotas;
- natureza do serviço (entrega ou coleta, entregas opcionais);
- decisões adicionais a serem tomadas (tamanho da tripulação, entre outras).

Recentemente, alguns estudos foram apresentados na literatura explorando uma nova variante do PRV que envolve decisões de alocação do tamanho da tripulação nos veículos (PUREZA; MORABITO; REIMANN, 2012; FERREIRA; PUREZA, 2012). Esta variante, chamada de Problema de Roteamento de Veículos com Múltiplos Entregadores (PRVME), tem aplicações em muitos contextos industriais, principalmente em situações nas quais é necessária a entrega regular de produtos em centros urbanos bastante congestionados. Exemplos típicos são fábricas de refrigerantes, cervejas e laticínios que precisam reabastecer regularmente (diariamente ou a cada poucos dias), pequenos e médios estabelecimentos como lojas de conveniência, restaurantes, mercearias, entre outros. Estes estabelecimentos se encontram tipicamente em regiões com alta concentração comercial, nas quais se torna difícil até mesmo estacionar os veículos de entrega. Assim, os veículos estacionam em pontos estratégicos de uma região que possua um grupo de clientes, e as entregas são feitas a pé até esses clientes. Ao fazer as entregas dessa forma, o tempo de serviço no grupo de clientes pode ser relativamente longo, ocupando grande parte do tempo total da rota. Dessa forma, pode-se inviabilizar o atendimento de outros grupos de clientes durante o horário de trabalho permitido quando as entregas são realizadas por apenas um entregador. Em contextos assim, o uso de entregadores adicionais se torna importante, pois reduz o tempo de serviço em cada grupo de clientes e agiliza a entrega dos produtos. Um exemplo de uma rota típica nesse contexto pode ser observado na Figura 1.1, onde as entregas são feitas em duas fases. Inicialmente, o veículo chega até o local de estacionamento do grupo de clientes e, posteriormente, os entregadores levam os produtos até os clientes do grupo.

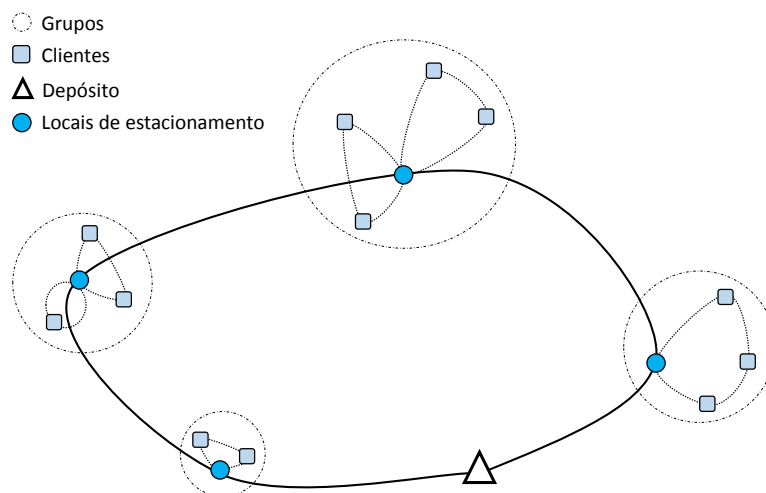


Figura 1.1: Exemplo de uma rota típica no PRVME.

Nesta dissertação aborda-se o problema de roteamento de veículos com janelas de tempo e múltiplos entregadores (PRVJTME) por meio de diferentes tipos de métodos de solução. Inicialmente foram desenvolvidas duas abordagens metaheurísticas baseadas em Busca Local Iterada e em Busca em Vizinhança Grande. Estas abordagens foram desenvolvidas com o intuito de obter rapidamente soluções com a melhor qualidade possível. Em seguida, desenvolveram-se métodos híbridos de solução combinando um método *branch-price-and-cut* com as abordagens metaheurísticas previamente desenvolvidas, abordando diferentes esquemas de cooperação entre esses componentes. Estes últimos métodos são o foco principal deste trabalho.

Uma das contribuições primárias deste trabalho é o desenvolvimento de métodos híbridos de solução para tratar o problema, já que até o momento a literatura concentrou-se no desenvolvimento de métodos metaheurísticos para resolver o PRVJTME. Ainda, conforme observado para outras variantes do PRV, acredita-se que métodos híbridos que explorem a estrutura do problema possam resultar em estratégias eficientes na prática, as quais possibilitem a obtenção de soluções de boa qualidade relativamente rápido quando comparadas com as heurísticas e metaheurísticas. Esse último aspecto corresponde a uma das principais justificativas deste estudo.

O objetivo deste trabalho é aprofundar o estudo do PRVJTME, focando no desenvolvimento de métodos híbridos de solução para o problema, os quais combinem técnicas e princípios usados em métodos exatos e heurísticos. Para atingir esse objetivo geral, devem-se alcançar os seguintes objetivos específicos: (i) estudar o problema de roteamento de veículos com janelas de tempo e múltiplos entregadores; (ii) desenvolver abordagens metaheurísticas para resolver o problema; e (iii) combinar as abordagens metaheurísticas desenvolvidas com um método *branch-price-and-cut*, com o objetivo de desenvolver métodos híbridos para resolver o problema.

Este projeto está estruturado dentro do contexto de Pesquisa Operacional, com uso ativo de técnicas de programação e modelagem matemática. Segundo Morabito e Pureza (2012), Bertrand e Fransoo (2002), esta pesquisa pode ser caracterizada como axiomática normativa dado que visa o desenvolvimento de estratégias e métodos, a fim de melhorar os resultados disponíveis na literatura para um problema previamente proposto e comparar seu desempenho com outras estratégias que tratam um mesmo problema.

O restante desta dissertação está organizado como segue. No Capítulo 2, é descrito o PRVJTME e são revisados os trabalhos encontrados na literatura abordando o problema, seguido de uma revisão de trabalhos de aplicação de métodos híbridos a problemas de roteamento de veículos. No Capítulo 3, são descritas as abordagens metaheurísticas propostas para resolver o problema. Depois, no Capítulo 4 são apresentados os métodos híbridos de solução desenvolvidos a partir da combinação de um método *branch-price-and-cut* com abordagens metaheurísticas. No Capítulo 5, são apresentados os resultados

obtidos nos testes computacionais realizados com um conjunto de instâncias disponíveis na literatura. Comparações entre os métodos propostos são realizadas, assim como também é feita uma comparação de seus desempenhos com métodos da literatura. Finalmente, no Capítulo 6, apresentamos as principais conclusões e apontamos perspectivas para pesquisas futuras.

Capítulo 2

Revisão da literatura

Neste capítulo é descrito o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Entregadores (PRVJTME). Em seguida, são revisados trabalhos encontrados na literatura abordando o PRVJTME. Por fim, é apresentada a revisão de literatura sobre métodos híbridos para PRVs, destacando métodos que focam na combinação de metaheurísticas com princípios do método de geração de colunas, já que esse método é a base dos métodos híbridos desenvolvidos nesta dissertação.

2.1 O problema de roteamento de veículos com janelas de tempo e múltiplos entregadores

Em termos práticos, o PRVJTME envolve dois estágios: inicialmente, os clientes devem ser agrupados em torno de locais de estacionamento para, finalmente, estabelecer rotas para visitar esses grupos e realizar as respectivas entregas a cada cliente. Cada entrega deve ser realizada dentro de uma faixa de horário predeterminada para cada cliente, chamada de janela de tempo, enquanto o custo total de entrega é minimizado (número de veículos, alocação de entregadores e distância percorrida). Dada a complexidade do problema completo, as atividades de agrupamento e de roteamento são tratadas de forma separada (SENARCLENS DE GRANCY; REIMANN, 2014). Assim, no que tange ao PRVJTME, os grupos são assumidos como dados de entrada, cada um dos quais tem uma demanda acumulada e um tempo de serviço que inclui o transporte dos produtos desde o local de estacionamento do veículo até os clientes do grupo. Janelas de tempo passam a ser impostas pelo grupo de clientes. Na variante estudada nesta dissertação, assume-se que o tempo de serviço é linearmente dependente do número de entregadores designados à rota, mas vale ressaltar que esse pressuposto pode ser estendido, descrevendo os tempos de serviço em termos do número de entregadores através de funções não necessariamente lineares.

A primeira formulação matemática apresentada para o PRVJTME foi proposta por Pureza, Morabito

e Reimann (2012). Essa formulação é uma extensão da clássica formulação de fluxo de veículos para o PRVJT (DESAULNIERS; MADSEN; ROPKE, 2014), conforme apresentado a seguir.

Considere uma frota de veículos que está disponível em um único depósito e deve ser usada para visitar n grupos de clientes, de modo a atender as demandas d_i de cada um deles, $i = 1, \dots, n$. O objetivo é determinar rotas que tenham custo total mínimo e satisfaçam os requisitos descritos na sequência. Cada grupo deve ser visitado exatamente uma única vez e a visita deve satisfazer a janela de tempo $[w_i^a, w_i^b]$, isto é, o veículo não pode chegar ao grupo i após o instante w_i^b e deve esperar até w_i^a para iniciar o serviço, caso chegue antes deste instante. O tempo de serviço necessário para atender o grupo i com l entregadores é conhecido *a priori* e denotado por s_{il} . O tempo e o custo de viagem entre dois grupos i e j é dado por t_{ij} e c_{ij} , respectivamente. Assume-se que os veículos sejam idênticos e que haja um número suficiente de veículos para atender a demanda de todos os grupos. Os veículos devem retornar ao depósito após realizarem as rotas designadas e cada um deles tem uma capacidade máxima Q , o que limita o número de grupos que ele pode visitar, e um número máximo L de entregadores permitidos a bordo. O número total de entregadores designados deve respeitar o número de entregadores disponíveis, definido por E .

Sejam 0 e $n + 1$ os índices usados para representar o depósito. Esta notação é usada para diferenciar quando um veículo deixa o depósito (índice 0) de quando o veículo retorna a ele (índice $n + 1$). Assim, os grupos e o depósito ficam representados de forma uniforme por uma rede composta por $n + 2$ nós, numerados de 0 a $n + 1$. Demandas e tempos de serviço nulos são definidos para os nós correspondentes ao depósito, ou seja, $d_0 = d_{n+1} = 0$ e $s_{0l} = s_{n+1,l} = 0, \forall l$. Além disso, existem janelas de tempo associadas ao depósito, $[w_0^a, w_0^b] = [w_{n+1}^a, w_{n+1}^b]$, onde w_0^a define o instante no qual os veículos podem partir do depósito e w_{n+1}^b o instante máximo de retorno a ele. A variável de decisão binária x_{ijl} assume valor 1 se e somente se existe uma rota com l entregadores que visita o nó i e segue imediatamente para o nó j . As variáveis de decisão w_{il} e y_i determinam o instante de tempo em que o serviço se iniciará no nó i por uma rota com l entregadores e a demanda total já atendida por uma rota ao visitar o nó i , respectivamente. Usando os parâmetros e variáveis anteriormente definidos, o PRVJTME pode ser representado de acordo com a seguinte formulação (PUREZA; MORABITO; REIMANN, 2012):

$$\min \quad p_1 \sum_{j=1}^n \sum_{l=1}^L x_{0jl} + p_2 \sum_{j=1}^n \sum_{l=1}^L l x_{0jl} + p_3 \sum_{i=0}^{n+1} \sum_{j=0}^{n+1} \sum_{l=1}^L c_{ij} x_{ijl} \quad (2.1a)$$

$$\text{s.a} \quad \sum_{\substack{j=1 \\ j \neq i}}^{n+1} \sum_{l=1}^L x_{ijl} = 1, \quad i = 1, \dots, n, \quad (2.1b)$$

$$\sum_{\substack{i=0 \\ i \neq h}}^n x_{ihl} - \sum_{\substack{j=1 \\ j \neq h}}^{n+1} x_{hjl} = 0, \quad \begin{array}{l} h = 1, \dots, n, \\ l = 1, \dots, L, \end{array} \quad (2.1c)$$

$$y_i + d_j \sum_{l=1}^L x_{ijl} - Q(1 - \sum_{l=1}^L x_{ijl}) \leq y_j, \quad i, j = 0, \dots, n+1, \quad (2.1d)$$

$$w_{il} + (t_{ij} + s_{il})x_{ijl} - M_{ij}(1 - x_{ijl}) \leq w_{jl}, \quad \begin{array}{l} i, j = 0, \dots, n+1, \\ l = 1, \dots, L, \end{array} \quad (2.1e)$$

$$\sum_{j=1}^n \sum_{l=1}^L l x_{0jl} \leq E, \quad (2.1f)$$

$$d_i \leq y_i \leq Q, \quad i = 0, \dots, n+1, \quad (2.1g)$$

$$w_i^a \leq w_{il} \leq w_i^b, \quad \begin{array}{l} i = 0, \dots, n+1, \\ l = 1, \dots, L, \end{array} \quad (2.1h)$$

$$x_{ijl} \in \{0, 1\}, \quad \begin{array}{l} i, j = 0, \dots, n+1, \\ l = 1, \dots, L. \end{array} \quad (2.1i)$$

A função objetivo (2.1a) minimiza o custo total da solução, composto pela soma ponderada do número de veículos usados, o número de entregadores designados às rotas e a distância total percorrida. Os valores p_1 , p_2 e p_3 são os custos por veículo usado, entregador designado e unidade de distância percorrida, respectivamente. Esses valores definem a prioridade de cada um dos três componentes da função objetivo.

As restrições (2.1b) garantem que um único veículo em um único modo saia de cada grupo i . As restrições (2.1c) garantem a continuidade da rota, *i.e.*, se um veículo em modo l entra no grupo h , esse mesmo veículo em modo l deve sair do grupo h . Dizemos que um veículo viaja em modo l quando o tamanho de sua tripulação é l .

O conjunto de restrições (2.1d) define a relação entre as variáveis de fluxo x_{ijl} e as variáveis de carga y_i . Estas restrições também impedem a formação de subrotas que não incluam o depósito. De forma semelhante, as restrições (2.1e) definem a relação entre as variáveis de fluxo x_{ijl} e as variáveis de tempo de início do serviço w_{il} . O parâmetro $M_{ij} = w_i^b - w_j^a$ é usado para tornar inativa a restrição correspondente sempre que a variável x_{ijl} assumir valor zero.

As restrições (2.1f) limitam o número total de entregadores que podem ser designados, de acordo com o limite E . As equações (2.1g) impõem o limite de capacidade dos veículos, enquanto as restrições (2.1h) definem que as janelas de tempo dos grupos e do depósito sejam respeitadas. Finalmente, as restrições (2.1i) definem o domínio da variável x_{ijl} .

Por ser uma extensão da formulação de fluxo de veículos clássica, é de se esperar que a relaxação linear do modelo (2.1) também resulte em um limitante ruim para o valor ótimo do problema. Isto ocorre de fato, como indicam os resultados computacionais apresentados por Pureza, Morabito e Reimann

(2012) onde, o método *branch-and-cut* disponível no software de otimização CPLEX 11.0 resolveu apenas 1 das 56 instâncias contendo 100 grupos de clientes dentro de um limite de 10 horas em um computador Intel Core2 i7 2.67 GHz com 12 GB RAM. Para os demais problemas o *gap* de otimalidade ficou acima de 80% em média, mostrando assim a necessidade de desenvolver métodos de solução para o problema que resultem em estratégias eficientes e com um compromisso adequado em termos de qualidade de solução e esforço computacional.

Finalmente, note que o PRVJTME pode ser comparado com o PRV com frota heterogênea (PRVFH), onde as configurações dos veículos no PRVJTME equivaleriam aos diferentes tipos de veículos no PRVFH. Ressaltando-se que as configurações são restritas de forma global no PRVJTME, já que se limita o número total de entregadores designados, enquanto que no PRVFH cada configuração é restrita de forma individual (número de veículos disponíveis de cada tipo). No entanto, observe que as configurações no primeiro definem os tempos de serviço nos grupos de clientes enquanto que no último os tipos de veículos podem definir a capacidade do veículo, tempos de viagem, entre outros.

2.2 Métodos de solução para o PRVJTME

O primeiro trabalho a propor métodos de solução para o problema é o estudo de Pureza, Morabito e Reimann (2012), no qual são apresentadas duas abordagens metaheurísticas para a resolução do PRVJTME. A primeira é uma Busca Tabu (TS, do inglês *Tabu Search*) adaptativa, caracterizada por um mecanismo de intensificação e diversificação que muda os parâmetros da TS baseando-se na análise do padrão da trajetória das soluções (PUREZA; FRANÇA, 2001; FRANÇA; SOSA; PUREZA, 1999). A segunda abordagem é uma Otimização por Colônia de Formigas (ACO, do inglês *Ant Colony Optimization*), a qual constrói soluções através de um mecanismo probabilístico de inserção e, subsequentemente, utiliza mecanismos de busca local para melhorar as soluções (REIMANN; DOERNER; HARTL, 2012). Os resultados mostram que os algoritmos tiveram desempenhos semelhantes, pois nenhum deles dominou o outro em todas as instâncias utilizadas, em termos de qualidade de solução e tempo de execução.

Senarclens De Grancy e Reimann (2014) propuseram duas abordagens metaheurísticas, baseadas em ACO e GRASP (FEO; RESENDE, 1995). Visando ter algoritmos comparáveis, os autores utilizaram a mesma heurística de construção e de busca local em ambas as metaheurísticas. A heurística de construção foi baseada na heurística de inserção I1 de Solomon (1987), enquanto que para melhoria, uma busca local através de três operadores diferentes foi aplicada. Os resultados mostram que a ACO teve melhor desempenho para tempos de execução longos enquanto que a GRASP teve melhor desempenho para tempos de execução curtos.

Duas heurísticas foram desenvolvidas por Ferreira e Pureza (2012) para uma variante do problema que não considera janelas de tempo e descarta a exigência de que todos os grupos de clientes sejam

atendidos. A primeira heurística é uma adaptação do algoritmo de economias de Clarke e Wright (1964) enquanto a segunda heurística é uma TS que utiliza o mesmo mecanismo de adaptação da TS proposta por Pureza, Morabito e Reimann (2012). Os resultados mostram que a TS superou o desempenho da heurística de economias, em termos do número de grupos de clientes atendidos e do custo das soluções obtidas.

Em todos os trabalhos anteriormente mencionados, as instâncias utilizadas foram geradas baseadas nos problemas de Solomon (1987) para o PRVJT, modificando os tempos de serviço para se ajustarem ao problema com múltiplos entregadores.

Apesar da existência de trabalhos anteriores, há ainda uma considerável falta de pesquisa, por se tratar de um problema recentemente proposto na literatura. Por exemplo, na revisão de literatura realizada não se encontrou nenhum trabalho explorando a aplicação de métodos exatos ou híbridos para resolver o problema.

2.3 Métodos híbridos para problemas de roteamento de veículos

Diferentes métodos de solução foram aplicados nas últimas décadas para resolver variantes do PRV, passando pelos métodos exatos, métodos heurísticos e metaheurísticos, até combinações ou híbridos destes dois últimos tipos de métodos, os quais recentemente receberam a denominação de *matheuristics* (MANIEZZO; STÜTZLE; VOSS, 2010).

Dentre os métodos exatos são destacados métodos como: *branch-and-cut*, *branch-and-price*, programação dinâmica, etc. Estes métodos são capazes de encontrar a solução ótima do problema e provar sua otimalidade. Atualmente, métodos baseados em geração de colunas são as abordagens mais eficientes no que tange a métodos exatos de solução para PRVs (MUNARI; GONDZIO, 2013; BALDACCI; MINGOZZI; ROBERTI, 2012; ROPKE; CORDEAU, 2009; DESAULNIERS; LESSARD; HADJAR, 2008; FUKASAWA *et al.*, 2006).

Por outro lado, métodos de solução heurísticos e metaheurísticos são usados para gerar soluções factíveis, tipicamente de forma rápida. Contudo, estes métodos podem não ser capazes de encontrar a solução ótima do problema e nem fornecer informação de quão longe da solução ótima está a solução encontrada. Assim, não é possível saber se vale a pena investir mais tempo tentando encontrar uma melhor solução, sendo esta uma das maiores desvantagens desse tipo de método.

Nos últimos anos, muitas pesquisas voltaram sua atenção para abordagens cooperativas de solução que tentam tirar vantagem de cada uma das partes que as compõem. No começo, a cooperação era realizada principalmente entre diferentes métodos metaheurísticos como Algoritmos Genéticos (GA, do inglês *Genetic Algorithm*), Recozimento Simulado (SA, do inglês *Simulated Annealing*), Busca Tabu, Busca Local Iterada (ILS, do inglês *Iterated Local Search*), entre outros. Entretanto, atualmente mais

pesquisas desenvolvem hibridizações entre métodos de solução exatos e heurísticos. Nestes híbridos, conceitos de diferentes tipos de métodos são combinados visando explorar as vantagens particulares de seus componentes individuais e, de fato, métodos híbridos bem projetados frequentemente têm melhor desempenho que seus componentes isolados (JOURDAN; BASSEUR; TALBI, 2009). Em particular, métodos híbridos dados pela combinação de metaheurísticas com conceitos do método de geração de colunas têm recebido grande atenção na literatura, com resultados promissores na resolução de problemas de roteamento de veículos (RAIDL, 2015; ALVELOS; DE SOUSA; SANTOS, 2013; BALL, 2011; RAIDL; PUCHINGER, 2008)

Dentre os métodos híbridos para PRVs, uma das práticas mais comuns consiste na utilização de heurísticas e metaheurísticas na criação de soluções iniciais que se aproximem o máximo possível de soluções ótimas e armazenar suas rotas num conjunto chamado de “conglomerado (*pool*) de rotas”. Depois, a partir da combinação das rotas desse conjunto, tenta-se encontrar melhores soluções. Geralmente essa combinação é realizada resolvendo formulações por Particionamento de Conjuntos (PC) ou por Cobertura de Conjuntos (CC) através de softwares de otimização comerciais como o CPLEX (ILOG Inc., 2008), Gurobi (OPTIMIZATION, 2015), entre outros. Uma estrutura geral da interação desse tipo de método híbrido é apresentada na Figura 2.1, mas cabe ressaltar que a relação entre as partes que compõem este tipo de método pode envolver muitos mais componentes e informações.

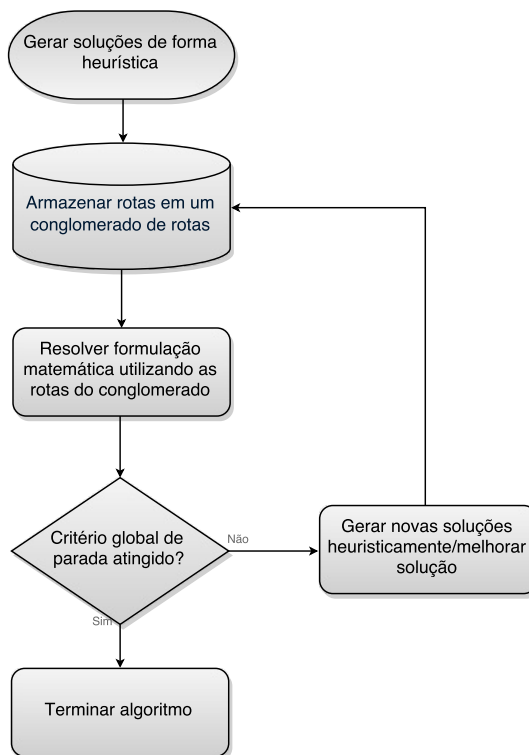


Figura 2.1: Estrutura básica de uma classe de método híbrido.

De acordo com a classificação de métodos híbridos para PRVs de Archetti e Speranza (2014), o

esquema de método híbrido recém descrito pode ser classificado dentro da categoria de métodos de heurísticas de melhoria, os quais usam modelos de programação matemática para melhorar soluções encontradas por uma heurística. Os autores completam a classificação propondo mais duas categorias, a saber, abordagens de decomposição e abordagens baseadas em geração de colunas. Na primeira categoria são incluídos os métodos que dividem os problemas em subproblemas menores e mais simples, os quais são resolvidos através de métodos de solução específicos ou por meio de softwares de otimização comerciais. A segunda categoria inclui métodos que utilizam princípios do método de geração de colunas.

Uma vez que muitos métodos híbridos apresentados na literatura são desenvolvidos utilizando formulações baseadas em PC para a combinação das rotas, a seguir se descreve uma formulação desse tipo para um problema de roteamento de veículos. O objetivo neste tipo de formulação é determinar o subconjunto de rotas de mínimo custo que atenda as restrições do problema. Seja P o conjunto de todas as rotas factíveis para o problema, λ_p é uma variável binária que assume o valor 1 se e somente se a rota p é selecionada para a solução, a_{ip} é um parâmetro binário que toma o valor 1 se e somente se o cliente i é atendido pela rota p e c_p corresponde ao custo associado à rota p . O problema pode ser formulado como:

$$\min \sum_{p \in P} c_p \lambda_p \quad (2.2a)$$

$$\text{s.a } \sum_{p \in P} a_{ip} \lambda_p = 1, \quad i = 1, \dots, n, \quad (2.2b)$$

$$\lambda_p \in \{0, 1\}, \quad \forall p \in P. \quad (2.2c)$$

Nessa formulação, a função objetivo (2.2a) minimiza o custo da solução, que é dado pela soma dos custos do conjunto de rotas que a compõem. As restrições (2.2b) garantem que cada cliente seja atendido por exatamente uma única rota, enquanto as restrições (2.2c) definem o domínio da variável λ_p .

Para obter uma formulação por CC basta substituir a igualdade na restrição (2.2b) por uma desigualdade do tipo maior ou igual (\geq). Além de tudo, se o número de veículos for limitado (K veículos), pode-se acrescentar a seguinte restrição:

$$\sum_{p \in P} \lambda_p \leq K. \quad (2.2d)$$

Um dos primeiros trabalhos encontrados na literatura explorando a aplicação de métodos híbridos para resolver PRVs é a pesquisa de Rochat e Taillard (1995). Os autores propõem um método híbrido que utiliza uma formulação por PC como técnica de pós-otimização na aplicação de uma metaheurística de TS para resolver o PRV capacitado e o PRVJT. No trabalho, a formulação é resolvida através do software

de otimização CPLEX, usando colunas geradas a partir de um conjunto de rotas vindas das execuções prévias da TS. Os resultados mostram que o uso do modelo permite obter pequenas melhoras em relação às soluções obtidas pela TS.

Similarmente, Russell e Chiang (2006) propõem um *framework* híbrido baseado em Busca Dispersa (*Scattered Search*) para resolver o PRVJT. Na aplicação do *framework*, uma primeira fase baseada em TS é utilizada para transformar soluções iniciais (geradas heurísticamente) em versões melhoradas, das quais as melhores são escolhidas para compor um conjunto chamado de “conjunto de referência”. A segunda fase do método consiste na aplicação, sobre o conjunto, de vários operadores de combinação, entre eles uma formulação por CC que combina rotas do conjunto de referência em uma única solução superior. Posteriormente, é aplicada uma TS para tentar melhorar as soluções geradas pelas técnicas de combinação. Yildirim e Çatay (2014) também apresentam um método híbrido de duas fases baseado na combinação da metaheurística ACO e a resolução de uma formulação por PC para resolver o PRVJT. No método, um conjunto de soluções factíveis são geradas heurísticamente através da ACO enquanto suas rotas são armazenadas em um conglomerado de colunas, as quais são utilizadas em uma formulação por PC que é resolvida usando-se o software de otimização CPLEX 12.2. Uma vez gerada essa solução, alguns parâmetros da ACO são atualizados (feromônios) e a ACO é executada novamente a fim de tentar melhorar a solução. Todos estes passos são executados dentro de um processo iterativo até completar o número predefinido de iterações do método. Ambas abordagens mostraram resultados competitivos ao comparar seus resultados com as melhores soluções conhecidas na literatura para as instâncias utilizadas.

Alvarenga, Mateus e De Tomi (2007) e Muter, Birbil e Sahin (2010) também abordam o PRVJT usando métodos híbridos. Alvarenga, Mateus e De Tomi (2007) propõem um método híbrido de duas fases onde, na primeira fase, um conjunto de soluções são geradas através de um GA especializado, armazenando as rotas dos ótimos locais em um conglomerado de colunas com a intenção de serem utilizadas na segunda fase do método, que consiste na resolução de uma formulação por PC para obter a melhor combinação de rotas em uma única solução, tudo dentro de um processo iterativo até completar o tempo limite de execução. Os resultados mostram que o método híbrido desenvolvido foi capaz de gerar boas soluções quando comparado com resultados publicados na literatura, pois encontrou soluções melhores que as conhecidas até então para algumas das instâncias testadas.

Muter, Birbil e Sahin (2010) apresentam um *framework* híbrido, chamado de *MetaOpt*, o qual integra algoritmos metaheurísticos e exatos para resolver problemas de otimização combinatória que possam ser modelados através de formulações por CC. O *framework* guia uma metaheurística usando informação extraída do algoritmo exato para tentar explorar o espaço de soluções eficientemente. O *framework* foi testado resolvendo o PRVJT usando TS como metaheurística enquanto o software de

otimização CPLEX 11.0 foi utilizado no componente exato. No *framework*, uma formulação por CC é resolvida utilizando um conglomerado de colunas, o qual é preenchido com as rotas das soluções geradas através da metaheurística. Nos resultados, os autores encontraram soluções ótimas conhecidas da literatura e outros resultados que são comparáveis com métodos especializados para o PRVJT.

Subramanian *et al.* (2012) propõem um método híbrido para resolver PRVs com frota heterogênea, composto por uma ILS que utiliza o método de Descida em Vizinhança Variável (VND, do inglês *Variable Neighborhood Descent*) na fase de busca local e uma formulação por PC. No método, a ILS é executada armazenando em um conglomerado de colunas todas as rotas associadas aos mínimos locais encontrados pela metaheurística. Posteriormente, o modelo é resolvido sobre o conglomerado de colunas utilizando o software de otimização CPLEX 12.3. O software interage com a ILS durante sua execução, chamando a ILS cada vez que encontra uma solução incumbente, a fim de reforçar a diversificação do método. Uma variação desse método é utilizada por Subramanian, Uchoa e Ochi (2013) para resolver outras variantes do PRV (capacitado, assimétrico, com rotas abertas, com coleta e entrega e com múltiplos depósitos). No mesmo trabalho é utilizado o conglomerado de colunas geradas por meio da ILS junto com a resolução de uma formulação por PC, acrescentando um esquema dinâmico de controle de tamanho do modelo por PC para tornar viável a resolução de problemas de grande porte. Em ambos os trabalhos, o método foi testado em instâncias da literatura mostrando-se robusto ao ser competitivo para todas as variantes nas quais foi aplicado.

Villegas *et al.* (2013) propõem um método híbrido que combina uma formulação por PC com uma metaheurística híbrida entre GRASP e ILS. O método é usado para resolver duas variantes do Problema de Roteamento de Caminhões e Trailers (*Trucks and Trailers Routing Problem*). Na primeira fase do método, um conglomerado de colunas é gerado, armazenando as rotas das soluções geradas com a GRASP, a qual utiliza ILS na fase de melhoria. Na segunda fase, o modelo por PC é resolvido sobre o conglomerado de colunas usando o software de otimização Gurobi 5.0.1. Os resultados mostram que o método foi capaz de melhorar resultados de métodos previamente aplicados na literatura.

Kramer *et al.* (2015) abordam o Problema de Roteamento de Veículos com foco na redução da emissão de poluentes (PRP, do inglês *Pollution-Routing Problem*) através de um algoritmo híbrido, chamado de ILS-SP-SOA, que combina uma ILS com a resolução de um modelo por PC. O método gera e refina soluções através da ILS e de operadores de perturbação, complementado por um procedimento de otimização de velocidade (característica específica do PRP). Finalmente, as rotas associadas aos mínimos locais gerados são armazenadas em um conglomerado e usadas para resolver o modelo no software de otimização CPLEX 12.4 para gerar possíveis melhores soluções compostas pela combinação de rotas do conglomerado. Experimentos computacionais foram executados em problemas da literatura, como também foram propostos novos conjuntos de problemas de teste para a variante do problema. O método

também foi utilizado para resolver outros PRVs considerando aspectos ambientais, a saber, PRV considerando consumo de combustível (*Fuel Consumption VRP*) e PRV com minimização de energia (*Energy Minimizing VRP*), mostrando-se capaz de gerar soluções de alta qualidade e superar outros métodos da literatura.

Cacchiani, Hemmelmayr e Tricoire (2014) apresentam um método híbrido para o PRV periódico, combinando a resolução de uma formulação matemática com uma ILS. O método é baseado na resolução da relaxação linear de uma formulação do tipo de CC que utiliza colunas heurísticamente geradas pela ILS. Após isso, é utilizado um esquema iterativo de fixação e liberação de variáveis para integralizar e factibilizar a solução. As variáveis são fixadas quando são consideradas promissoras e liberadas depois de algum tempo a fim de diversificar a busca. O algoritmo é executado até atingir o número máximo de iterações ou o tempo limite de execução. O método desenvolvido foi testado em instâncias da literatura, mostrando-se competitivo quando comparado com os melhores desenvolvidos na literatura para o problema.

Os trabalhos apresentados até o momento utilizam, em essência, o conceito do conglomerado de colunas com o propósito de criar boas soluções por meio da combinação de rotas através da resolução de uma formulação matemática, como mostrado na Figura 2.1. Essa estratégia pode ser estendida através da incorporação do princípio do método de geração de colunas. Utilizando informação dual obtida da resolução da relaxação linear do problema mestre (formulação por PC ou CC) tenta-se gerar colunas com custo reduzido negativo para enriquecer o conglomerado de colunas, tentando se aproximar da solução ótima.

A estratégia que utiliza geração de colunas tem sido aplicada para resolver PRVs, obtendo bons resultados. Por exemplo, Parragh e Schmid (2013) propõem um método híbrido para resolver o *Dial-a-Ride Problem* - DARP (variante do problema de roteamento com pedidos de coleta e entrega). No método, inicialmente um conglomerado de colunas é preenchido com as rotas geradas a partir da aplicação de uma Busca em Vizinhança Grande (LNS, do inglês *Large Neighborhood Search*) que utiliza operadores de extração e inserção para gerar soluções iniciais. Em seguida, a relaxação linear de uma formulação por CC é resolvida sobre o conglomerado de colunas utilizando o software de otimização CPLEX 12.1. As variáveis duais são repassadas para uma Busca em Vizinhança Variável (VNS, do inglês *Variable Neighborhood Search*) que as utiliza para gerar colunas com custo reduzido negativo e enriquecer o conglomerado de colunas. Eventualmente é resolvida a formulação por CC com restrições de integralidade para obter uma solução inteira e posteriormente é executada a LNS como método de melhoria global. Esses passos são executados de forma iterativa até completar um número predefinido de iterações. O método é testado em instâncias da literatura, obtendo soluções de alta qualidade em relação às melhores soluções conhecidas na literatura.

De forma similar, Hauge *et al.* (2014) apresentam um método híbrido que combina TS com geração de colunas. No método, uma solução inicial e as colunas são geradas heurísticamente através de uma TS que utiliza a informação dual obtida da relaxação linear de uma formulação por PC para gerar colunas com custo reduzido negativo. A relaxação linear do modelo por PC é resolvida com as colunas vindas da TS. Esses passos são realizados dentro de um processo iterativo até atingir o limite de tempo ou não poder encontrar colunas com custo reduzido negativo. Finalmente o modelo por PC é resolvido conservando a restrição de integralidade para obter uma solução factível para o problema original. O método é aplicado para resolver um problema de coleta de resíduos industriais (aplicação do PRV com coleta e entrega simultâneas), sendo capaz de encontrar boas soluções para as instâncias testadas.

Danna e Le Pape (2005) desenvolvem um método híbrido através da combinação de um método *branch-and-price* com heurísticas de construção e de busca local em um esquema de cooperação para resolver o PRVJT. No método, o *branch-and-price* é usado para resolver uma formulação por PC do problema. Primeiramente, uma solução é construída por uma heurística e definida como a solução inicial do método. Em seguida, a execução do *branch-and-price* é iniciada e as heurísticas de melhoria são ativadas em intervalos regulares de tempo com a intenção de aprimorar a melhor solução atual do método. No *branch-and-price*, o problema mestre é resolvido usando o software de otimização CPLEX 8.1.0 e os subproblemas são resolvidos usando um algoritmo de programação dinâmica. Os resultados mostram que o método desenvolvido é capaz de gerar boas soluções para o problema, mantendo a capacidade do *branch-and-price* de gerar bons limitantes inferiores para o problema. Ademais, soluções ótimas para duas instâncias em aberto foram encontradas.

Nishi e Izuno (2014) propõem um método híbrido para o problema de roteirização e programação de navios o qual combina heurísticas com a técnica de geração de colunas. O problema é abordado usando a decomposição de *Dantzig-Wolfe* (DANTZIG; WOLFE, 1960) aplicada a uma formulação do problema. Inicialmente, uma solução é criada através de uma heurística construtiva e, após isso, a relaxação linear do problema mestre é resolvida usando as colunas correspondentes a essa solução inicial. O valor das variáveis duais é repassado para o subproblema que as utiliza para gerar novas colunas com custo reduzido negativo, onde o subproblema é formulado como um problema de programação inteira-mista (MIP) e consiste em encontrar um plano de visitas factível para cada navio. Esses passos são repetidos até não encontrar colunas com custo reduzido negativo, indicando que a solução ótima do problema relaxado foi encontrada, ou atingir o limite de tempo de execução. Ao final do método, se a solução obtida não é viável para o problema original, é aplicada uma heurística de fixação de variáveis para viabilizá-la. Tanto o problema mestre quanto o subproblema são resolvidos usando o software de otimização CPLEX 12.1. O método foi aplicado para resolver um caso real obtendo melhores soluções quando comparado com a operação da empresa.

Chen e Xu (2006) apresentam um método híbrido para resolver o PRV dinâmico. O método é baseado na técnica de geração de colunas e tem uma estrutura heurística de horizonte rolante, pois resolve uma série de problemas menores derivados da divisão do horizonte de planejamento em intervalos. Em cada estágio do método é resolvida uma formulação por PC com as rotas de um conglomerado de colunas. Inicialmente, o conglomerado é composto pelas rotas de uma solução inicial heurísticamente gerada. Depois, são usadas as mesmas colunas do conglomerado e os valores das variáveis duais obtidos da resolução da relaxação linear de uma formulação por PC para gerar colunas com custo reduzido negativo. Esse processo é repetido até completar o horizonte de planejamento da instância. No trabalho, é usado o software de otimização CPLEX 7.0 para resolver as formulações matemáticas e o método foi testado em instâncias baseadas nos problemas propostos por Solomon (1987) mostrando que o método é capaz de gerar boas soluções quando comparado com outros métodos também propostos pelos autores.

Conforme pode-se observar nos trabalhos revisados, a combinação da técnica de geração de colunas com heurísticas apresenta bons resultados. No entanto, vale ressaltar que existem outras formas de combinar métodos exatos e heurísticos. Por exemplo, existem técnicas baseadas em *Local Branching* (FISCHETTI; LODI, 2003), remoção e inserção (PISINGER; ROPKE, 2010), entre outros. A seguir são apresentadas algumas aplicações desses outros tipos de métodos híbridos.

Archetti, Speranza e Savelsbergh (2008) apresentam um método híbrido de duas fases para resolver o PRV com entregas fracionadas. O método combina uma TS com uma formulação de programação inteira-mista do problema. No método, uma TS identifica partes promissoras do espaço de soluções que serão exploradas por um modelo de programação inteira-mista. A TS é executada múltiplas vezes para identificar clientes que tendem a ter sua carga fracionada e também os arcos mais utilizados nas melhores soluções. Todas essas informações são consideradas como parte promissora do espaço de soluções, a qual é explorada através do MIP que as utiliza junto com as rotas geradas pela TS para gerar soluções completas. Para avaliar o desempenho do método foram utilizadas instâncias da literatura e se compararam seus resultados com uma metaheurística da literatura. Os resultados mostram que o método híbrido melhora o desempenho da metaheurística quando usada como método de solução independente.

Schmid *et al.* (2010) desenvolveram um método híbrido baseado na combinação de VNS e um algoritmo exato para o problema de distribuição de concreto pronto, formulando-o como um PRV com frota heterogênea. No método, a VNS é usada como algoritmo de geração de soluções iniciais e como algoritmo de melhoria global. Na parte exata é usada uma Busca em Vizinhança de Grande Porte (VLNS, do inglês *Very Large Neighborhood Search*) para auxiliar o software de otimização XPRESS na resolução de uma formulação matemática do problema, em um esquema similar ao método *Local Branching*. A VLNS fixa variáveis de decisão da solução incumbente e o modelo reduzido é resolvido pelo software

de otimização, em um processo iterativo. A aplicação do método em instâncias derivadas de um caso real de distribuição de concreto mostra que o método é capaz de encontrar soluções com *gap* médio de 12% para problemas de médio porte.

De Franceschi, Fischetti e Toth (2006) desenvolveram um método híbrido baseado em extração e realocação para o PRV com restrições de distância. O método inicia com uma solução gerada heurísticamente da qual são selecionados e extraídos nós. Esses nós são inseridos em um conglomerado, sobre o qual é aplicada uma heurística de recombinação para gerar subsequências que também serão adicionadas ao conglomerado. Durante essa fase, informação dual derivada da resolução da relaxação linear de um modelo matemático de realocação também pode ser usada para encontrar novas sequências úteis. Ao final, uma formulação matemática que tem estrutura de formulação por PC realoca as sequências nos espaços disponíveis. A formulação impõe que cada nó extraído pertença exatamente a uma das sequências que serão alocadas nos espaços disponíveis. O software de otimização CPLEX 8.0 foi usado para resolver a formulação. Os resultados mostram que o método foi efetivo ao ser capaz de melhorar resultados da literatura.

Similarmente, Salari, Toth e Tramontani (2010) propõem um método híbrido para o PRV com rotas abertas. No método são aplicados operadores aleatórios de extração de clientes das rotas de uma solução inicial (criada por meio de uma heurística rápida), para depois realocá-los através da resolução até a otimalidade de um modelo de programação inteira por CC. O anterior é feito dentro de uma estrutura heurística cíclica que perturba a solução no caso de não ter melhoria durante um número predefinido de iterações. Essa perturbação é executada visando mudar a região do espaço de soluções que está sendo explorada para reforçar a diversificação do método. O software de otimização usado para resolver a formulação foi o CPLEX 10.0. Resultados computacionais utilizando instâncias da literatura mostram que o método é capaz de encontrar boas soluções para o problema de forma rápida.

Finalmente, deve-se destacar a ampla aplicabilidade dos métodos híbridos para resolver muitos outros problemas envolvendo decisões de roteirização tais como: problema de roteamento e programação de técnicos (PILLAC; GUÉRET; MEDAGLIA, 2012), problema de roteamento de veículos seletivo (VALLE *et al.*, 2011), problema de roteamento de veículos com demandas estocásticas (MENDOZA; VILLEGAS, 2012) e o problema de balanceamento de carga de redes (SANTOS; DE SOUSA; ALVELOS, 2013).

A Tabela 2.1 sintetiza as pesquisas mencionadas nesta seção, mostrando os autores, o problema abordado e as partes principais do método de solução aplicado. Pode ser observado que o interesse na aplicação de métodos híbridos para problemas de roteamento de veículos cresceu consideravelmente na última década. Além do mais, na parte heurística dos diferentes métodos propostos, as metaheurísticas baseadas em buscas locais, como ILS, TS e LNS, foram as mais utilizadas.

Autor	Problema	Abordagem
(ROCHAT; TAILLARD, 1995)	PRVJT	TS+PC (Cplex)
(RUSSELL; CHIANG, 2006)	PRVJT	TS+CC (Cplex)
(DE FRANCESCHI; FISCHETTI; TOTH, 2006)	PRV com restrições de distância	Remoção heurística+PC (Cplex)
(CHEN; XU, 2006)	PRV dinâmico	Heurística de horizonte rolante+PC (Cplex)
(ALVARENGA; MATEUS; DE TOMI, 2007)	PRVJT	GA+PC (GLPK)
(ARCHETTI; SPERANZA; SAVELSBERGH, 2008)	PRV com entregas fracionadas	TS+MIP (Solver não informado)
(MUTER; BIRBIL; SAHIN, 2010)	PRVJT	TS+CC (Cplex)
(SCHMID <i>et al.</i> , 2010)	Problema de distribuição de concreto	VNS (fixando com VLNS)+MIP (Xpress)
(SALARI; TOTH; TRAMONTANI, 2010)	PRV com rotas abertas	Remoção heurística+CC (Cplex)
(VALLE <i>et al.</i> , 2011)	PRV seletivo	GRASP+CC (Cplex)
(SUBRAMANIAN <i>et al.</i> , 2012)	PRV com frota heterogênea	ILS+PC (Cplex)
(PILLAC; GUÉRET; MEDAGLIA, 2012)	PR e programação de técnicos	LNS+SC (Gurobi)
(MENDOZA; VILLEGAS, 2012)	PRV com demandas estocásticas	Heurísticas para TSP+PC (Gurobi)
(PARRAGH; SCHMID, 2013)	<i>Dial-a-ride problem</i>	LNS+CC (Cplex)
(VILLEGAS <i>et al.</i> , 2013)	PR de caminhões e trailers	GRASP+CC (Gurobi)
(SUBRAMANIAN; UCHOA; OCHI, 2013)	PRVs com frota homogênea	ILS+PC (Cplex)
(SANTOS; DE SOUSA; ALVELOS, 2013)	Balanceamento de carga de redes	GRASP+PC (Cplex)
(YILDIRIM; ÇATAY, 2014)	PRVJT	ACO+PC (Cplex)
(CACCHIANI; HEMMELMAYR; TRICOIRE, 2014)	PRV periódico	ILS+CC (Cplex)
(NISHI; IZUNO, 2014)	PR e programação de navios	Heurística gulosa+PC (Cplex)
(HAUGE <i>et al.</i> , 2014)	PRV com coleta e entrega simultâneas	TS+PC (Cplex)
(KRAMER <i>et al.</i> , 2015)	<i>Pollution-Routing problem</i>	ILS+PC (Cplex)

Tabela 2.1: Características de métodos híbridos propostos na literatura para PRVs.

Capítulo 3

Abordagens metaheurísticas

Neste capítulo apresentamos as abordagens metaheurísticas desenvolvidas para resolver o PRVJTME. Elas são baseadas em diferentes buscas locais e utilizam a mesma heurística de construção. A primeira abordagem desenvolvida é baseada na metaheurística Busca Local Iterada (ILS) e é apresentada na Seção 3.2. A segunda é baseada na metaheurística Busca em Vizinhança Grande (LNS) e sua descrição e componentes são mostrados na Seção 3.3. A heurística de construção usada em ambas as abordagens é descrita na Seção 3.1.

As metaheurísticas ILS e LNS foram aplicadas de forma bem-sucedida para resolver outras variantes do PRV, por exemplo: PRV com frota heterogênea (SUBRAMANIAN *et al.*, 2012; PENNA; SUBRAMANIAN; OCHI, 2013); PRV com janelas de tempo (PISINGER; ROPKE, 2007); PRV dinâmico (HONG, 2012); PRV com múltiplas rotas (AZI; GENDREAU; POTVIN, 2014); PRV com entregas fracionadas (SILVA; SUBRAMANIAN; OCHI, 2015); PRV com coletas e entregas e janelas de tempo (ROPKE; PISINGER, 2006); PRV com *backhauls* (CUERVO *et al.*, 2014). Dessa forma, acredita-se que essas metaheurísticas possam ter grande potencial para obter soluções de boa qualidade em tempos de execução aceitáveis também para o PRVJTME.

3.1 Heurística construtiva

Para obter uma solução inicial para o PRVJTME foi desenvolvida uma heurística construtiva baseada na usada por Senarclens De Grancy e Reimann (2014), a qual está fundamentada na heurística de inserção I1 de Solomon (1987). Em nossa aplicação da heurística, as rotas são construídas sequencialmente inicializando-as com o grupo mais distante em relação ao depósito e que ainda não foi atendido, definindo a tripulação inicial do veículo como o número máximo de entregadores permitidos a bordo (3 entregadores). Em seguida, novos grupos são inseridos na rota minimizando um critério ponderado de minimização de tempo adicional e de distância adicional percorrida ao atender o grupo. Uma vez que

não possam ser inseridos mais grupos na rota atual, uma nova rota é inicializada e o processo é repetido até completar uma solução que atenda a todos os grupos.

3.2 Abordagem metaheurística baseada em ILS

A primeira abordagem metaheurística desenvolvida utiliza em seu núcleo a metaheurística Busca Local Iterada (LOURENÇO; MARTIN; STÜTZLE, 2003). Essa metaheurística aplica uma busca local repetidamente sobre um conjunto de soluções obtidas da perturbação dos ótimos locais previamente visitados, com o propósito de gerar soluções de alta qualidade. Para implementar um algoritmo de busca local iterada são necessários 4 componentes básicos: (i) uma solução inicial; (ii) uma busca local para melhoria; (iii) um procedimento de perturbação que permita escapar dos ótimos locais; e (iv) um critério de aceitação que determine a partir de qual solução deve-se continuar a busca. Para mais informações e aplicações da ILS veja Lourenço, Martin e Stützle (2010).

Na abordagem proposta, a solução inicial gerada pela heurística construtiva, descrita na Seção 3.1, é melhorada por meio da aplicação dos operadores da ILS e de um conjunto de heurísticas de melhoria adicionais. A abordagem metaheurística baseada em ILS foi implementada de forma que pode ser usada focando tanto no número de rotas quanto na redução da distância total percorrida. A estrutura da abordagem é mostrada no Algoritmo 1.

A abordagem é executada iterativamente até completar o critério de parada, que pode ser o número de iterações ou o tempo de execução do algoritmo (linha 27). Inicialmente, é gerada uma solução S_0 usando a heurística construtiva (linha 2). No início de cada execução do laço principal, aplica-se a busca local RVND sobre S_0 (linha 5) de modo que a solução resultante é definida como a solução incumbente S^+ (a busca local RVND é descrita na Seção 3.2.1); em seguida, a abordagem entra em um ciclo de melhoria (linhas 7-23) perturbando a incumbente atual por meio do operador respectivo (linha 8) e melhorando a solução resultante através da aplicação da heurística de busca local RVND (linha 9) e da heurística de redução de entregadores (linha 10). Este ciclo de melhoria compreende duas fases. A primeira fase completa-se quando são atingidas as primeiras *MaxIterILS* iterações consecutivas sem melhoria (linhas 13-15). A partir desse momento começa a segunda fase, que é aplicada também até atingir *MaxIterILS* iterações sem melhoria (linhas 16-18). Ao final do ciclo, a melhor solução global da abordagem é atualizada (linhas 24-26) e um novo ciclo é iniciado caso o critério de parada não tenha sido atingido.

As diferentes fases da abordagem são necessárias para se considerar os diferentes termos da função objetivo. Na primeira fase, a abordagem tem enfoque na redução de rotas da solução e, para isso, o procedimento de perturbação (linha 8) é efetuado utilizando-se a heurística de redução de rotas da Seção 3.2.3. A segunda fase da abordagem focaliza na redução da distância total percorrida e por isso

Algoritmo 1: Abordagem metaheurística baseada em ILS.

Entrada: Instância, Parâmetros;
Saída: Melhor solução S^* ;

```
1 início
2    $S_0 \leftarrow$  Gerar solução inicial;
3    $S^* \leftarrow S_0$ ;
4   repita
5      $S^+ \leftarrow$  RVND( $S_0$ );
6      $iterILS \leftarrow 0$ ;
7     enquanto  $iterILS < 2 * MaxIterILS$  faça
8        $S' \leftarrow$  Perturbar( $S^+$ );
9        $S' \leftarrow$  RVND( $S'$ );
10       $S' \leftarrow$  Redução entregadores( $S'$ );
11      se  $f(S') < f(S^+)$  então
12         $S^+ \leftarrow S'$ ;
13        se  $iterILS < MaxIterILS$  então
14           $iterILS \leftarrow 0$ ;
15        fim
16      senão
17         $iterILS \leftarrow MaxIterILS$ ;
18      fim
19    fim
20    senão
21       $iterILS \leftarrow iterILS + 1$ ;
22    fim
23  fim
24  se  $f(S^+) < f(S^*)$  então
25     $S^* \leftarrow S^+$ ;
26  fim
27 até atingir o critério de parada;
28 fim
```

a perturbação é realizada utilizando o procedimento de perturbação da Seção 3.2.2. Cabe ressaltar que, na primeira fase, nem sempre a heurística de redução de rotas consegue modificar a solução. Nesse caso, aplica-se o procedimento de perturbação da segunda fase. A seguir, são descritos cada um dos componentes da abordagem metaheurística baseada em ILS.

3.2.1 Busca local

A busca local desempenha o papel da intensificação dentro da ILS. No algoritmo, a busca local é realizada por meio de uma heurística de Busca de Vizinhança Variável (MLADENOVIC; HANSEN, 1997) com Ordenação Aleatória, sendo conhecida por RVND (do inglês *Randomized Variable Neighborhood Descent*). A heurística aplica repetidamente um conjunto de buscas locais menores, chamadas de estruturas de vizinhança, para tentar melhorar progressivamente a solução. O funcionamento da heurística RVND é descrito a seguir. Dado um conjunto de estruturas de vizinhança $V = \{v^1, \dots, v^m\}$, a heurística

escolhe aleatoriamente uma delas para melhorar a solução atual. Se a estrutura de vizinhança não melhora a solução, ela é eliminada do conjunto V e a busca continua com as outras estruturas. Por outro lado, toda vez que uma estrutura melhora a solução atual, o conjunto é restabelecido a sua forma inicial. O algoritmo termina quando o conjunto V se torna vazio. Só movimentos factíveis são admitidos e a estratégia de primeira melhoria é utilizada nas estruturas de vizinhança. Além disso, sempre que uma estrutura de vizinhança consegue melhorar a solução, é executado o procedimento de redução de rotas descrito na Seção 3.2.3, buscando-se um aprimoramento adicional. O conjunto de estruturas de vizinhança é composto pelos movimentos apresentados a seguir:

- Estruturas de vizinhança *inter-rotas*:
 - **Shift($k,0$)**: k grupos adjacentes são transferidos de uma rota r_1 para a rota r_2 , $k = \{1, 2, 3\}$ (veja Figura 3.1).
 - **Swap($1,1$)**: Troca o grupo c_1 da rota r_1 com o grupo c_2 da rota r_2 (veja Figura 3.2).
 - **Swap($2,1$)**: Troca dois grupos adjacentes c_1 e c_2 da rota r_1 com o grupo c_3 da rota r_2 .
 - **Swap($2,2$)**: Troca dois grupos adjacentes c_1 e c_2 da rota r_1 com dois grupos adjacentes c_3 e c_4 da rota r_2 .
- Estruturas de vizinhança *intra-rotas*:
 - **Or-opt-1**: Tenta melhorar a rota por meio do reposicionamento de um grupo para outra posição na rota (veja Figura 3.3).
 - **2-opt**: Dados dois pares de grupos adjacentes não consecutivos (i, i^+) , (j, j^+) , seus arcos são trocados pelos arcos (i, j) , (i^+, j^+) e a direção entre (i^+, j) é invertida (veja Figura 3.4).

3.2.2 Perturbação: heurística de remoção e inserção

A perturbação é a responsável pela diversificação dentro da ILS, permitindo que a metaheurística consiga escapar de ótimos locais. Em nossa implementação, um dos procedimentos de perturbação consiste em uma heurística de remoção e inserção, baseado no procedimento proposto por Melechovsky (2012) e seu funcionamento é descrito a seguir. Para uma rota aleatória, são escolhidos aleatoriamente até nP grupos, os quais são removidos da rota. Posteriormente, testa-se a inserção desses grupos previamente extraídos nas rotas restantes, realizando a primeira inserção factível encontrada e inserindo-os na mesma ordem que foram removidos. Se no final existem grupos sem inserção factível, uma nova rota atendendo só a esses grupos é criada.

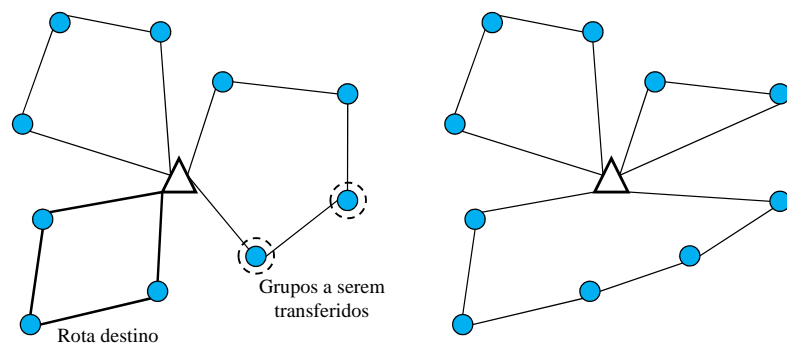


Figura 3.1: Movimento *Shift(2,0)*.

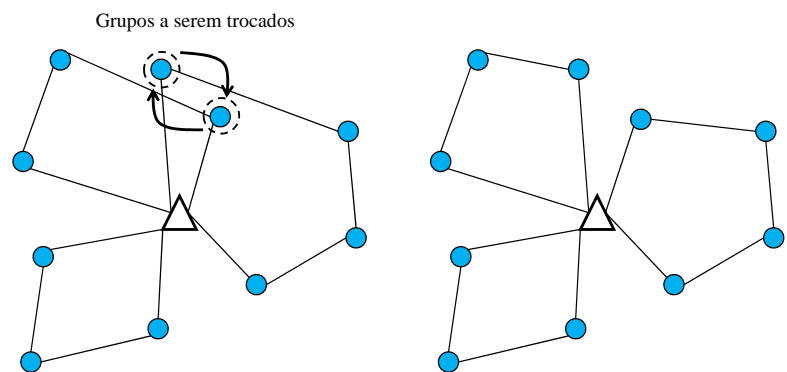


Figura 3.2: Movimento *Swap(1,1)*.

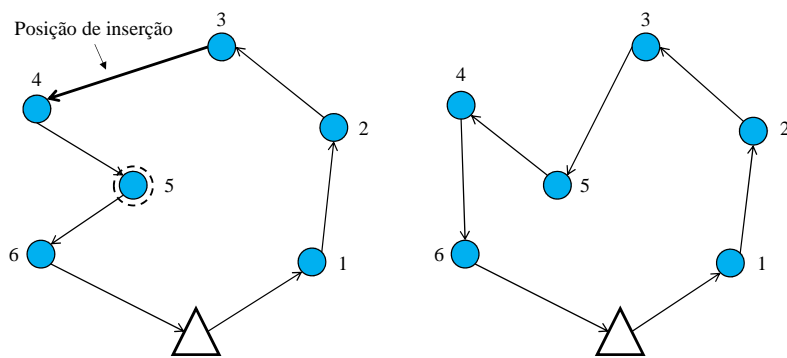


Figura 3.3: Movimento *Or-opt-1*.

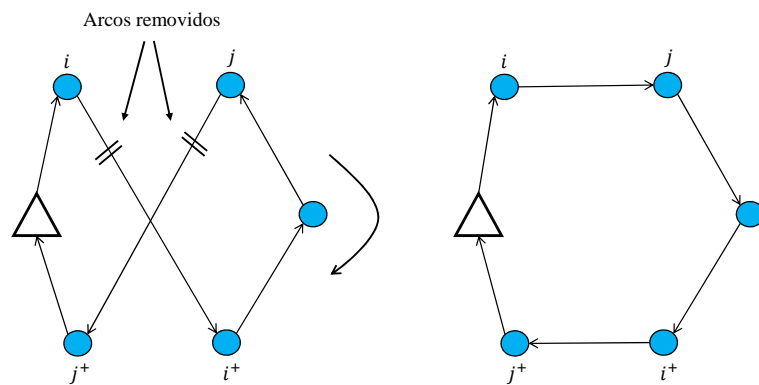


Figura 3.4: Movimento *2-opt*.

3.2.3 Heurística de redução de rotas

Levando em consideração a prioridade dos objetivos do problema e que diminuindo a quantidade de rotas utilizadas pode-se reduzir também o número total de entregadores designados, foi implementada uma extensão da heurística utilizada por Senarclens De Grancy e Reimann (2014) para redução de rotas. Tomando como entrada uma solução completa, a heurística toma uma rota por vez, remove todos os grupos dela e tenta reinseri-los em sua melhor posição factível nas rotas restantes. Se todos os grupos foram realocados com sucesso, atualiza-se a solução com número de rotas reduzido. No caso de não poder realocar algum grupo, as tripulações das rotas destino são temporariamente aumentadas em uma unidade (se a tripulação for menor que a máxima possível), a fim de aumentar suas folgas e facilitar a inserção factível desses grupos. Se aumentando a tripulação das rotas torna-se possível inserir os grupos, então as tripulações temporárias das rotas que receberam grupos são mantidas. A heurística continua até alocar todos os grupos de clientes da rota inicialmente eliminada ou até todas as rotas terem experimentado o tamanho máximo de tripulação e ainda ter grupos não atendidos. Vale lembrar que esta heurística além de ser usada diretamente para diminuir o número de rotas de uma solução, também é utilizada como procedimento de perturbação durante a primeira fase da abordagem ILS, quando ela é usada focalizando na redução do número de rotas da solução.

3.2.4 Heurística de redução de entregadores

Considerando que as rotas de uma solução podem ter mais entregadores que o necessário, seja por construção ou devido aos movimentos de melhoria, uma heurística foi projetada com o intuito de tentar reduzir o número total de entregadores de uma solução dada. A heurística funciona conforme a seguinte descrição. Seja $S = \{r_1, r_2, \dots, r_n\}$ uma solução factível composta pelas rotas r_1, r_2, \dots, r_n . Define-se $crew_i$ como o tamanho da tripulação da rota r_i , $\forall r_i \in S$. Para cada rota da solução, a heurística aplica o seguinte procedimento. Inicialmente, o tamanho da tripulação é reduzido em uma unidade (se $crew_i > 1$). Se a rota continua sendo factível com a nova tripulação, a solução S é atualizada e continua-se a aplicação do procedimento de redução com a rota atual. Caso uma redução da tripulação torne a rota infactível, a heurística busca o primeiro grupo da rota que se tornou infactível e o remove dela. A extração do primeiro grupo infactível realiza-se com a intenção de aumentar a folga da rota a partir do ponto de remoção e facilitar o atendimento dos grupos posteriores a ele. Seguidamente, a factibilidade da rota é analisada e, caso ainda seja infactível, o processo de remoção de grupos é repetido até que a rota se torne factível. Finalmente, os grupos que foram removidos da rota são inseridos na solução por meio de uma heurística de inserção gulosa, que os aloca na melhor posição possível em todas as rotas e, caso algum grupo de clientes não possua inserção factível, cria novas rotas para atender a esses grupos. A solução resultante deste processo é denotada por S' e, caso ela possua o mesmo número de rotas da

solução inicial, então S' substitui a solução S . Este procedimento de redução é aplicado para cada uma das rotas $r_i \in S$.

3.3 Abordagem metaheurística baseada em LNS

A segunda abordagem proposta é baseada na metaheurística LNS, introduzida por Shaw (1998). Essa metaheurística foi projetada visando superar as dificuldades das buscas locais tradicionais, as quais só fazem pequenas modificações nas soluções e conseqüentemente podem não ser capazes de se deslocar entre áreas promissoras do espaço de soluções. Na metaheurística LNS, a solução incumbente é iterativamente melhorada por meio de operações alternadas de destruição e reparação, através da aplicação de um conjunto de algoritmos simples que são usados para modificar a solução. Para informação mais detalhada sobre LNS, sugere-se o trabalho de Pisinger e Ropke (2010).

Na aplicação da abordagem ao PRVJTME, a metaheurística LNS é utilizada como método de melhoria global acompanhado do conjunto de heurísticas de melhoria adicionais, que foram usadas na abordagem baseada em ILS. A estrutura geral da abordagem é mostrada no Algoritmo 2. A abordagem é executada iterativamente até completar o critério de parada, executando múltiplas vezes a metaheurística LNS. Inicialmente, é gerada uma solução inicial (linha 2) por meio da heurística construtiva descrita na Seção 3.1, que no começo de cada iteração da abordagem é armazenada como a melhor solução inicial (linha 5). Em seguida, a abordagem entra em um ciclo de melhoria (linhas 7-16), onde inicialmente são aplicados os operadores de remoção e inserção (linha 8). Em nossa implementação, primeiro aplica-se um dos operadores de remoção da Seção 3.3.1 e, logo depois, um dos dois operadores de inserção da Seção 3.3.2. Em ambos os casos, os operadores são escolhidos aleatoriamente. Depois, são aplicadas as heurísticas de redução de rotas, de redução de entregadores e a heurística de busca de vizinhança variável com ordenação aleatória (RVND) (linhas 9-11). Ao final do ciclo, é atualizada a melhor solução da execução atual da abordagem, aceitando só soluções melhoradas (linhas 12-14). Finalmente, toda vez que a metaheurística executa *MaxIterLNS* iterações, a melhor solução global é atualizada (linhas 17-19) e a abordagem metaheurística é reiniciada.

A seguir, serão descritos detalhadamente cada um dos componentes da abordagem metaheurística baseada em LNS.

3.3.1 Operadores de remoção

Para a abordagem proposta foram usados quatro operadores de remoção diferentes. Cada um deles toma como entrada uma solução completa e retorna uma solução da qual foram extraídos q grupos. Os operadores são descritos a seguir.

Algoritmo 2: Abordagem metaheurística baseada em LNS.

Entrada: Instância, Parâmetros;**Saída:** Melhor solução S^* ;

```
1 início
2    $S_0 \leftarrow$  Gerar solução inicial;
3    $S^* \leftarrow S_0$ ;
4   repita
5      $S^+ \leftarrow S_0$ ;
6      $iterLNS \leftarrow 0$ ;
7     enquanto  $iterLNS < MaxIterLNS$  faça
8        $S' \leftarrow$  Aplicar operador de remoção e inserção em  $S^+$ ;
9        $S' \leftarrow$  Redução rotas( $S'$ );
10       $S' \leftarrow$  Redução entregadores( $S'$ );
11       $S' \leftarrow$  RVND( $S'$ );
12      se  $f(S') < f(S^+)$  então
13        |  $S^+ \leftarrow S'$ ;
14      fim
15       $iterLNS \leftarrow iterLNS+1$ ;
16    fim
17    se  $f(S^+) < f(S^*)$  então
18      |  $S^* \leftarrow S^+$ ;
19    fim
20  até atingir o critério de parada;
21 fim
```

- **Remoção aleatória:** este operador escolhe q grupos de forma aleatória e os remove da solução. Como apontado por Pisinger e Ropke (2007), este procedimento claramente tem o efeito de diversificar a busca.
- **Remoção da pior posição:** a ideia deste procedimento é remover os grupos que são muito caros, ou que de alguma maneira aumentam o custo da solução. Seja i um grupo qualquer, i^- seu predecessor e i^+ seu sucessor na rota. Em nossa aplicação, o custo c_i do grupo i é computado como

$$c_i = d_{i^-,i} + d_{i,i^+} - d_{i^-,i^+} \quad (3.1)$$

onde d_{ij} é a distância entre dois grupos i e j . Depois, o operador de remoção escolhe repetidamente um grupo i com maior custo até remover q grupos da solução. A escolha dos grupos a serem removidos tem um componente aleatório controlado por um parâmetro p , como descrito a seguir. Seja L o número de grupos que ainda estão na solução. Toda vez que um novo grupo vai ser removido, é gerado um número aleatório y entre zero e um e calcula-se k tal que $k = \lceil y^p L \rceil$. Em seguida, o k -ésimo grupo com maior custo é removido e o valor de L é atualizado. Esse procedimento é repetido até remover q grupos da solução. Note que se p for grande, grupos de maior custo têm maior probabilidade de serem removidos, enquanto grupos de menor custo podem ser

escolhidos para menores valores de p . Essa estratégia é usada a fim de evitar situações nas quais os mesmos grupos sejam removidos com frequência muito mais elevada que os demais.

- **Remoção relacionada:** o propósito deste procedimento é remover grupos que de alguma maneira estão relacionados e conseqüentemente se espera que sejam fáceis de trocar em uma posterior fase de inserção. Na implementação, a medida de relação entre dois grupos i e j foi tomada como a distância d_{ij} entre eles, realizando a remoção da seguinte forma. O algoritmo inicialmente seleciona um grupo i de forma aleatória. Em seguida, escolhe um dos grupos previamente selecionados e seleciona um novo grupo que esteja mais relacionado com ele. O procedimento se repete até selecionar q grupos, os quais são removidos. Do mesmo modo que no operador de remoção da pior posição, o processo de seleção dos grupos relacionados tem um componente probabilístico controlado pelo parâmetro p .
- **Remoção orientada por tempo:** este procedimento é uma variação da remoção relacionada, na qual a medida de relação entre dois grupos é baseada no tempo de início do serviço. Assim, tenta-se remover grupos que são atendidos aproximadamente no mesmo tempo, pois espera-se que eles sejam fáceis de trocar. O procedimento funciona como segue. Um grupo r é escolhido aleatoriamente e os $B = 2q$ grupos mais próximos dele, em termos de distância, são marcados como potenciais grupos a serem escolhidos. A medida de proximidade em tempo entre os grupos r e i é medida como

$$\Delta_{ri} = |w_r - w_i| \quad (3.2)$$

onde w_r e w_i são os tempos de início do serviço nos grupos r e i , respectivamente. Dentre os B grupos marcados, são escolhidos os $q - 1$ grupos mais próximos de r de acordo com a medida Δ_{ri} . Ao final, r e os $q - 1$ grupos escolhidos são removidos da solução. Semelhante ao operador de remoção relacionada, o processo de seleção dos grupos mais relacionados tem um componente probabilístico controlado pelo parâmetro p .

3.3.2 Operadores de inserção

Uma vez que a solução foi destruída pelo operador de remoção, deve-se efetuar a reparação dela para torná-la factível novamente. Para isso, foram utilizados dois operadores de inserção. Cada um deles toma como entrada uma solução incompleta e retorna uma solução completa, inserindo os q grupos que foram extraídos na fase de remoção. Os dois operadores de inserção utilizados são descritos a seguir.

- **Inserção gulosa:** como indicado por seu nome, o procedimento de inserção gulosa insere os grupos na posição mais barata possível. Formalmente este procedimento pode ser definido da

seguinte forma. Seja Δf_{ir} o incremento na função objetivo ao inserir o grupo i na melhor posição possível na rota r . Se o grupo i não pode ser inserido na rota r , Δf_{ir} é definido como infinito. Seguindo o procedimento guloso, determina-se

$$(i', r') = \arg \min_{i,r} \Delta f_{ir} \quad (3.3)$$

e insere-se o grupo i' na melhor posição da rota r' . Esta operação é realizada até inserir todos os grupos removidos ou não existir inserção factível para algum deles. Nesse último caso, novas rotas são criadas (inicializadas com a máxima tripulação possível) para esses grupos até completar uma solução atendendo 100% dos grupos.

- **Inserção tipo *regret*:** este procedimento surge como resultado dos inconvenientes do operador de inserção guloso, pois ele tende a deixar para o final a alocação de grupos de difícil inserção, quando não se têm muitas possibilidades de inserção. A inserção tipo *regret* tenta contornar essa dificuldade através da incorporação de um componente de antecipação no processo de seleção do grupo a ser inserido. Seja Δf_i^q o aumento na função objetivo ao inserir o grupo i na q -ésima melhor inserção possível na solução. Seguindo o procedimento, em cada iteração é escolhido um grupo i' tal que

$$i' = \arg \max_i (\Delta f_i^2 - \Delta f_i^1) \quad (3.4)$$

e o grupo é inserido na posição correspondente à rota de menor custo de inserção. Em outras palavras, o procedimento maximiza a diferença do custo de inserção entre a segunda melhor e a melhor inserção do grupo i' , o que significa que grupos com menos posições factíveis de inserção tendem a ser inseridos primeiro. O procedimento continua até inserir todos os grupos ou não existir inserção factível restante para algum grupo. Como ocorre no procedimento de inserção guloso, se no final ainda restarem grupos sem inserção factível, são criadas novas rotas para alocar esses grupos.

3.4 Considerações finais

Neste capítulo, foram descritas duas abordagens metaheurísticas baseadas em Busca Local Iterada e Busca em Vizinhança Grande para resolver o PRVJTME. Essas abordagens foram escolhidas devido a terem sido aplicadas de forma bem-sucedida para outras variantes do PRV, tendo potencial para também resolver de forma adequada o PRVJTME. Devido à complexidade do problema e suas diferentes características, as abordagens metaheurísticas foram reforçadas por meio de um conjunto de heurísticas adicionais (redução de rotas e redução de entregadores), já que o simples uso dos operadores originais

de cada metaheurística (perturbação e busca local para a ILS e operações de remoção e inserção para a LNS) não permite obter boas soluções para o problema.

Os algoritmos foram calibrados por meio de um planejamento experimental a fim de determinar uma combinação adequada dos parâmetros de cada um deles. Experimentos computacionais utilizando instâncias da literatura foram realizados para avaliar o desempenho das abordagens metaheurísticas desenvolvidas. A descrição do procedimento de calibração e os resultados dos testes realizados encontram-se no Capítulo 5. Adicionalmente, foi testado o software *ParamILS* (HUTTER; HOOS; STÜTZLE, 2007) para tentar encontrar de forma automática a melhor configuração dos parâmetros das abordagens metaheurísticas. Os resultados dos testes realizados mostraram diferentes configurações alternativas para os parâmetros, indicando que não há uma alta sensibilidade dos algoritmos em relação aos parâmetros.

Por último, apesar de serem abordagens metaheurísticas diferentes, deve-se destacar a semelhança entre elas já que as operações de remoção e inserção na LNS podem ser vistas como uma perturbação mais agressiva na ILS.

Capítulo 4

Métodos híbridos

Neste capítulo são descritos os métodos híbridos propostos para resolver o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Entregadores. Estes híbridos são baseados na combinação de métodos metaheurísticos, como as abordagens metaheurísticas descritas no Capítulo 3, com um método *branch-price-and-cut* (BPC) para o PRVJTME, o qual é baseado no BPC desenvolvido por Munari e Morabito (2015). Nas seguintes seções é descrita uma formulação por particionamento de conjuntos para o PRVJTME e as diferentes estratégias propostas para derivar os métodos híbridos de solução para o PRVJTME, partindo do BPC previamente mencionado. Esses métodos foram desenvolvidos visando aproveitar da melhor forma possível as características dos métodos disponíveis.

Vale lembrar que até o momento a literatura concentrou-se no desenvolvimento de métodos metaheurísticos para resolver o PRVJTME. Assim, a principal contribuição deste trabalho é o desenvolvimento de métodos híbridos para resolver o problema.

4.1 Formulação por particionamento de conjuntos

Em nossa aplicação, o método híbrido é usado para resolver uma formulação por PC para o PRVJTME. Essa formulação é baseada no modelo por PC para o PRV clássico (veja o modelo (2.2)), modificado para considerar as diferentes configurações (número de entregadores) que podem ter os veículos. Essa formulação também resulta da aplicação da decomposição de *Dantzig-Wolfe* à formulação de fluxo de veículos (2.1), apresentada no Capítulo 2 (MUNARI; MORABITO, 2015).

A formulação é descrita na sequência: seja P^l o conjunto de todas as rotas factíveis usando l entregadores, $l = 1, \dots, L$; λ_p^l uma variável de decisão binária que toma o valor 1 se e somente se a rota $p \in P^l$ é selecionada; a_{pi}^l um parâmetro binário com valor 1 se e somente se o grupo de clientes i é atendido pela rota $p \in P^l$; e c_p^l corresponde ao custo associado à rota $p \in P^l$. Com essa notação, é

apresentada a formulação:

$$\min \sum_{l=1}^L \sum_{p \in P^l} c_p^l \lambda_p^l \quad (4.1a)$$

$$\text{s.a.} \sum_{l=1}^L \sum_{p \in P^l} a_{pi}^l \lambda_p^l = 1, \quad i = 1, \dots, n, \quad (4.1b)$$

$$\sum_{l=1}^L \sum_{p \in P^l} l \lambda_p^l \leq E, \quad (4.1c)$$

$$\lambda_p^l \in \{0, 1\}, \quad l = 1, \dots, L, p \in P^l. \quad (4.1d)$$

A função objetivo (4.1a) minimiza o custo total do conjunto de rotas que compõem a solução. As restrições (4.1b) garantem que cada grupo seja atendido por exatamente uma única rota em um único modo l , as restrições (4.1c) limitam o número total de entregadores que podem ser designados, enquanto as restrições (4.1d) definem o domínio da variável λ_p^l .

Dada a formulação (4.1a)-(4.1d), pode-se obter um limitante inferior para o valor ótimo resolvendo sua relaxação linear, que é obtida substituindo as restrições de integralidade (4.1d) pelas restrições

$$\lambda_p^l \geq 0, \quad l = 1, \dots, L, p \in P^l. \quad (4.1e)$$

4.2 Método *branch-price-and-cut*

Uma das principais vantagens da formulação (4.1a)-(4.1d) é que o valor dado por sua relaxação linear fornece um limitante que é mais apertado que o limitante dado pela relaxação linear da formulação de fluxo de veículos e, por conseguinte, proporciona uma melhor aproximação do valor ótimo do problema. Esses melhores limitantes podem ser usados dentro de um algoritmo *branch-and-bound* (ARENALES *et al.*, 2007) com a intenção de reduzir o número de nós que devem ser explorados e, assim, aprimorar o funcionamento do método. Entretanto, devido ao grande número de rotas que podem ter os conjuntos $P^l, l = 1, \dots, L$, não é possível enumerar todos esses conjuntos e, portanto, deve-se aplicar o método de geração de colunas para resolver a relaxação linear do modelo por PC em cada nó da árvore. A combinação do método de geração de colunas com o *branch-and-bound* resulta no método conhecido como *branch-and-price*. Adicionalmente, desigualdades válidas podem ser acrescentadas com o objetivo de melhorar os limitantes e, dessa forma, refinar o desempenho do *branch-and-price*, produzindo então um método *branch-price-and-cut* (DESROSIERS *et al.*, 2010).

4.2.1 Método de geração de colunas

Geração de colunas é um método usado para resolver problemas de programação linear (PL) com um número muito grande de variáveis (colunas). Geração de colunas baseia-se no fato de que não é necessário enumerar todas as possíveis variáveis para resolver o PL até a otimalidade. Portanto, o método trabalha com um problema auxiliar de menor tamanho que contém um subconjunto das variáveis do problema original. Em cada iteração do método o problema auxiliar é resolvido e, usando sua solução dual, são geradas mais colunas que serão adicionadas ao problema auxiliar. O método finaliza quando se garante que a solução ótima do problema auxiliar é também a solução ótima do problema original. Dessa forma, pode-se resolver o PL usando somente um subconjunto de suas variáveis, enquanto novas variáveis são geradas à medida que elas são requeridas.

O PL a ser resolvido é chamado de *Problema Mestre* (PM) o qual, em nossa aplicação, consiste da relaxação linear da formulação (4.1a)-(4.1d). Para resolver o PM, considera-se um subconjunto de variáveis $\bar{P}^l \subseteq P^l, l = 1, \dots, L$, que gera o seguinte problema auxiliar chamado de *Problema Mestre Restrito* (PMR):

$$\min \quad \sum_{l=1}^L \sum_{p \in \bar{P}^l} c_p^l \lambda_p^l \quad (4.2a)$$

$$\text{s.a} \quad \sum_{l=1}^L \sum_{p \in \bar{P}^l} a_{pi}^l \lambda_p^l = 1, \quad i = 1, \dots, n, \quad (4.2b)$$

$$\sum_{l=1}^L \sum_{p \in \bar{P}^l} l \lambda_p^l \leq E, \quad (4.2c)$$

$$\lambda_p^l \geq 0, \quad l = 1, \dots, L, p \in \bar{P}^l. \quad (4.2d)$$

O processo de geração de colunas é dado conforme a seguinte descrição. Sejam λ^* e (\bar{u}, \bar{v}) , as soluções primais e duais ótimas do PMR, estando $\bar{u} \in \mathbb{R}^n$ e $\bar{v} \in \mathbb{R}_-$ associadas com as restrições (4.2b) e (4.2c), respectivamente. Com esta solução dual, o custo reduzido das variáveis do PM é dado por $c_p^l - \sum_{i=1}^n a_{pi}^l \bar{u}_i - l \bar{v}$. Em cada iteração do método de geração de colunas, resolve-se o PMR, obtendo-se as soluções primal e dual correspondentes e, a seguir, deve-se verificar se ainda existe alguma coluna com custo reduzido a ser gerada. Contudo, dado que não é possível enumerar todas as colunas dos conjuntos $P^l, l = 1, \dots, L$, deve-se resolver um problema de otimização (chamado de subproblema) que permite gerar colunas assim que são necessárias. Esse subproblema é dado por:

$$z_{SP}^l(\bar{u}, \bar{v}) = \min_{p \in P^l} \left\{ c_p^l - \sum_{i=1}^n a_{pi}^l \bar{u}_i - l \bar{v} \right\}, \quad l = 1, \dots, L. \quad (4.3)$$

Se $z_{SP}^l(\bar{u}, \bar{v})$ for negativo, então a variável associada λ_p^l pode ser adicionada ao conjunto \bar{P}^l e o

novo PMR é resolvido. O PL é resolvido até a otimalidade quando não é possível gerar mais colunas com custo reduzido negativo, ou seja, quando $z_{SP}^l(\bar{u}, \bar{v}) \geq 0 \quad \forall l = 1, \dots, L$, indicando que as soluções atuais λ^* e (\bar{u}, \bar{v}) são as soluções primal e dual ótimas do PM.

No método padrão de geração de colunas, o PMR é resolvido até a otimalidade em cada iteração, tipicamente por um método tipo simplex, de modo que o subproblema recebe um ponto extremo do conjunto de soluções duais. No entanto, o uso de pontos extremos causa instabilidade no método, afetando negativamente o processo de geração de colunas. De fato, a alta oscilação entre pontos extremos de iterações consecutivas tipicamente resulta em um lento progresso do algoritmo de geração de colunas, especialmente nas últimas iterações e, conseqüentemente, técnicas de estabilização devem ser usadas a fim de se evitar tal fenômeno (LÜBBECKE; DESROSIERS, 2005; VANDERBECK, 2005).

4.2.2 Resolvendo o subproblema

Nesta seção descreve-se o método usado no BPC para resolver o subproblema no método de geração de colunas. O subproblema consiste em um Problema de Caminho Mínimo Elementar com Restrições de Recursos (ESPPRC, do inglês *Elementary Shortest Path Problem with Resource Constraints*). Nesse problema, dado um grafo $\mathcal{G}(\mathcal{V}, \mathcal{A})$ definido pelo conjunto de vértices \mathcal{V} mais o conjunto de arcos \mathcal{A} , busca-se o caminho de custo mínimo desde um nó origem até um nó destino. Em nosso caso, o conjunto de vértices \mathcal{V} está associado ao conjunto de grupos de clientes mais os nós correspondentes ao depósito (nó 0 representando a saída do depósito e nó $n + 1$ representando o retorno ao depósito). Além disso, o nó origem e o nó destino correspondem aos nós 0 e $n + 1$, respectivamente. Cada arco no grafo tem um custo associado e o custo de um caminho é a soma dos custos dos arcos usados nele. Adicionalmente, deve-se atender às restrições de recursos para garantir a factibilidade dos caminhos gerados com relação a janelas de tempo, capacidade dos veículos e elementaridade.

Geralmente esse tipo de problema é resolvido através de algoritmos de Programação Dinâmica (PD), como algoritmos de *label-setting* (IRNICH; DESAULNIERS, 2005; BEASLEY; CHRISTOFIDES, 1989; DESROCHERS, 1988). O princípio do algoritmo consiste em associar a cada caminho parcial, um rótulo (*label*). Um rótulo \mathcal{L}_i consiste em um conjunto de atributos que caracterizam o caminho parcial que termina no nó i . No algoritmo, as rotas são obtidas pela extensão iterativa desses rótulos, já que através da prolongação factível de um rótulo, novos nós são inseridos no caminho parcial. Adicionalmente, o ESPPRC também pode ser resolvido utilizando outros tipos de métodos, tais como heurísticas e metaheurísticas (XU *et al.*, 2003; SAVELSBERGH; SOL, 1998) ou combinação de delas com PD (SPLIET; DESAULNIERS, 2015; DESAULNIERS; LESSARD; HADJAR, 2008). Além do mais, recentemente Lozano, Duque e Medaglia (2015) propuseram um algoritmo de *propagação de pulso* para abordar o ESPPRC como subproblema ao resolver a relaxação linear de uma formulação por PC para o PRVJT. Propagação

de pulso refere-se à exploração recursiva de caminhos parciais que são estendidos até alcançar o nó destino ou serem descartados por uma regra de poda definida. Os resultados encontrados pelos autores foram promissores quando comparados com os de métodos considerados como o estado da arte para resolver o ESPPRC.

No presente BPC, o ESPPRC é resolvido por um algoritmo de PD. O algoritmo de *label-setting* utiliza a técnica de extensão bidirecional para realizar a prolongação dos caminhos (RIGHINI; SALANI, 2008; CHABRIER, 2006), o que significa que os rótulos são estendidos tanto progressivamente desde o nó origem para seus sucessores, quanto regressivamente desde o nó destino para seus predecessores. Sejam \mathcal{F}_i e \mathcal{B}_i rótulos que terminam no nó i e que foram estendidos progressivamente e regressivamente, respectivamente. Uma vez que as operações de extensão progressiva e regressiva são concluídas, cada rótulo \mathcal{F}_i é acoplado com cada rótulo \mathcal{B}_i para todo $i \in \mathcal{V}$, se o caminho resultante é factível e tem custo total negativo. O caminho com menor custo é a solução ótima do ESPPRC. Os demais caminhos correspondem a soluções subótimas que também podem ser usadas para gerar colunas para o PMR. Regras de dominância devem ser aplicadas visando minimizar o número de rótulos que podem ser gerados durante a execução do algoritmo. Essas regras são baseadas na eliminação de rótulos para os quais garante-se não levar a uma solução ótima. Além disso, o algoritmo usa a técnica de identificação de nós inatingíveis *a priori*, visando que mais rótulos possam ser dominados (FEILLET *et al.*, 2004).

Como mostrado por Dror (1994), o ESPPRC é um problema NP-difícil e, portanto, encontrar a solução ótima toda vez que o subproblema é resolvido pode ser computacionalmente custoso. Dessa forma, para acelerar o algoritmo de PD geralmente são usadas algumas simplificações, já que somente na última iteração do método de geração de colunas precisa-se encontrar a solução ótima do subproblema a fim de mostrar que não existem colunas com custo reduzido negativo a serem geradas (PECIN *et al.*, 2014; MUNARI; GONDZIO, 2013; BALDACCI; MINGOZZI; ROBERTI, 2012).

4.2.3 Desigualdades válidas

Semelhante ao método *branch-and-cut*, desigualdades válidas podem ser adicionadas para fortalecer os valores dos limitantes, reduzir o número de nós na árvore e, conseqüentemente, melhorar o desempenho geral do método BPC. A maior preocupação em relação ao uso de desigualdades válidas é que algumas delas podem requerer mudanças complexas no subproblema, o que compromete o desempenho dos algoritmos que são usados para resolvê-lo. Portanto, existe um *trade-off* entre a qualidade dos limitantes e o esforço adicional quando desigualdades válidas são incluídas no método.

Desigualdades válidas do tipo *subset-row* (SR), propostas por Jepsen *et al.* (2008), são comumente usadas em métodos BPC para variantes do PRV. As desigualdades válidas do tipo SR mais usadas são as baseadas em conjuntos de três nós, como segue. Dado um conjunto de nós (grupos de clientes) $\mathcal{S} =$

$\{i_1, i_2, i_3\} \subset \{1, 2, \dots, n\}$, a desigualdade SR correspondente deve assegurar que só uma rota/coluna visite pelo menos dois dos grupos em \mathcal{S} . Para o PRVJTME, Munari e Morabito (2015) incorporaram o número de entregadores da seguinte forma

$$\sum_{l=1}^L \sum_{p \in I_{\mathcal{S}}} \lambda_p^l \leq 1, \quad (4.4)$$

onde $I_{\mathcal{S}} \in \bar{P}$ é o conjunto de colunas no PMR que correspondem a rotas atendendo pelo menos dois grupos de clientes do conjunto \mathcal{S} . Geralmente, só as desigualdades mais violadas são adicionadas ao PMR. As variáveis duais associadas aos cortes inseridos no PRM devem ser consideradas no subproblema, o que pode impactar negativamente no método para resolvê-lo dada a estrutura das desigualdades do tipo SR. Várias ideias tem sido propostas na literatura com a intenção de reduzir o impacto negativo advindo pelo uso destas desigualdades (BALDACCI; MINGOZZI; ROBERTI, 2012; DESAULNIERS; LESSARD; HADJAR, 2008).

4.2.4 Estratégia de ramificação

Seguindo a proposta de Munari e Morabito (2015), utiliza-se uma estratégia de ramificação composta de três regras diferentes, conforme descrito a seguir. Elas são apresentadas a seguir na ordem de prioridade na qual são aplicadas.

Seja $\bar{\lambda}$ a solução ótima do PM no nó atual. A primeira regra de ramificação calcula o número de veículos usados na solução, conforme a Equação (4.5).

$$nV = \sum_{l=1}^L \sum_{p \in \bar{P}^l} \bar{\lambda}_p^l \quad (4.5)$$

Se nV for fracionário, dois novos nós filhos são criados impondo $\sum_{l=1}^L \sum_{p \in \bar{P}^l} \lambda_p^l \leq \lfloor nV \rfloor$ para o primeiro e $\sum_{l=1}^L \sum_{p \in \bar{P}^l} \lambda_p^l \geq \lceil nV \rceil$ para o segundo. Caso nV seja inteiro, a estratégia segue para a segunda regra de ramificação, a qual computa o número de entregadores usados na solução, de acordo com a Equação (4.6).

$$nE = \sum_{l=1}^L \sum_{p \in \bar{P}^l} l \bar{\lambda}_p^l \quad (4.6)$$

Se nE for fracionário, dois novos nós filhos são criados, estabelecendo $\sum_{l=1}^L \sum_{p \in \bar{P}^l} l \lambda_p^l \leq \lfloor nE \rfloor$ para um nó e $\sum_{l=1}^L \sum_{p \in \bar{P}^l} l \lambda_p^l \geq \lceil nE \rceil$ para o outro. Semelhante à regra anterior, caso nE seja inteiro, a próxima regra de ramificação passa a ser aplicada.

Finalmente, a terceira regra ramifica nos fluxos dos arcos, conforme descrito a seguir. Seja \bar{x}_p^l o vetor binário de incidência tal que, $\bar{x}_{pij}^l = 1$ se e somente se a rota $p \in \bar{P}^l$ visita o nó i e viaja diretamente

para o nó j . A regra calcula

$$\bar{x}_{ij}^l = \sum_{p \in \bar{P}^l} \bar{x}_{pij}^l \bar{\lambda}_p^l, \quad \forall l = 1, \dots, L; i, j = 0, 1, \dots, n+1, \quad (4.7)$$

e selecciona um componente fracionário \bar{x}_{ij}^l , criando dois novos nós filhos, impondo $\sum_{l=1}^L x_{ij}^l = 0$ para um deles e $\sum_{l=1}^L x_{ij}^l = 1$ para o outro.

Note que as duas primeiras regras de ramificação (associadas a nV e nE) só modificam o problema mestre dos nós filhos, enquanto a última regra da estratégia modifica tanto os subproblemas quanto o problema mestre dos nós filhos. No caso de $\sum_{l=1}^L x_{ij}^l = 0$, deve-se impor em todos os subproblemas a remoção do arco (i, j) , ao passo que no PMR precisa-se excluir todas as colunas associadas a rotas que visitam o nó i e vão diretamente para o nó j , para $l = 1, \dots, L$. Em contrapartida, no nó filho com $\sum_{l=1}^L x_{ij}^l = 1$ são excluídos todos os arcos (i, j) para os subproblemas definidos por $P^{l'}$ tal que $l' \neq l$, ao tempo que no subproblema definido por P^l são eliminados todos os arcos partindo do nó i , com exceção daqueles que vão diretamente para j . Já no PMR, são removidas todas as colunas associadas a rotas que visitam o nó i e vão diretamente para o nó j , com exceção daquelas que o fazem em modo l .

4.3 Heurística primal para o BPC

A primeira hibridização realizada consiste no desenvolvimento de uma heurística primal para incorporar ao método BPC. O termo *heurística primal* geralmente refere-se a procedimentos baseados nas ferramentas de métodos exatos para obter soluções factíveis para o problema, truncando um procedimento exato ou construindo soluções a partir da relaxação da qual o método depende (JONCOUR *et al.*, 2010). Na literatura existem tentativas de desenvolvimento de heurísticas primais de propósito geral, baseadas na exploração das estruturas dos problemas. Algumas das heurísticas encontradas na literatura são do tipo gulosas, residuais, de arredondamento, de busca local e de mergulho (*diving*) (SADYKOV *et al.*, 2015; LÜBBECKE; PUCHERT, 2012; JONCOUR *et al.*, 2010).

É bem sabido que a disponibilidade de bons limitantes primais no início do método *branch-and-bound* pode reduzir consideravelmente o tamanho da árvore. Observando-se isso, e com o propósito de obter rapidamente soluções inteiras para o problema, utiliza-se uma heurística primal no BPC. Na heurística primal desenvolvida para o BPC, tenta-se aproveitar a estrutura da formulação por PC do problema mestre. Dado que as colunas no PMR correspondem a rotas factíveis do problema, pode-se buscar uma solução completa e factível para o problema a partir da combinação de um subconjunto dessas rotas. A heurística consiste em impor restrições de integralidade nas variáveis λ_p^l do PMR e resolver o respectivo problema de programação inteira utilizando um software de otimização de propósito geral ao finalizar o método de geração de colunas em cada nó da árvore. O fortalecimento do BPC por meio

dessa heurística representa a primeira estratégia de hibridização realizada.

4.4 Método híbrido exato: melhorando os limitantes primais do BPC por meio de metaheurísticas

A partir da primeira estratégia de hibridização proposta, é explorada a utilização das metaheurísticas apresentadas no Capítulo 3 e do método *branch-price-and-cut* em um esquema cooperativo de solução. Nesta estratégia, as metaheurísticas e o BPC interagem com o propósito de encontrar melhores soluções inteiras para o problema. O uso das metaheurísticas dentro deste método híbrido é focado em três componentes do BPC: (i) fornecer colunas para inicializar o PMR no nó raiz da árvore; (ii) sugerir uma solução incumbente inicial para o método; e (iii) tentar melhorar a solução toda vez que o BPC encontre uma nova solução inteira.

O funcionamento e os componentes deste método são esquematizados na Figura 4.1. Inicialmente, as metaheurísticas ILS e LNS são executadas de forma sequencial e todas as rotas correspondentes a soluções intermediárias são armazenadas em um conglomerado de rotas. Colunas são criadas a partir dessas rotas, adicionando-as ao PMR até gerar o número predefinido de colunas iniciais. Ainda, durante a execução das metaheurísticas a melhor solução encontrada é armazenada e no final desta primeira fase é definida como a incumbente inicial da árvore. Esta primeira fase do método é representada por meio do retângulo direito da figura.

Completando a geração das colunas iniciais, a heurística primal desenvolvida para o BPC (círculo inferior na figura) é executada, ou seja, o PMR inicial é resolvido impondo restrições de integralidade nas variáveis λ_p^l , usando um software de otimização de propósito geral com a intenção de tentar encontrar uma melhor solução dada pela combinação das rotas iniciais geradas pelas metaheurísticas. Se uma melhor solução for encontrada, então atualiza-se a incumbente e se aplicam as metaheurísticas sobre ela, atualizando-a caso seja melhorada. No final desta fase, começa a execução do método BPC, representada através do retângulo esquerdo da figura.

Continuando com a execução do método, sempre que o BPC consegue encontrar uma nova solução inteira, seja porque a resolução do PMR resultou em uma solução inteira (círculo superior na figura), ou porque a heurística primal encontrou uma solução inteira dada pela combinação das colunas do PMR, as abordagens metaheurísticas são executadas em série para tentar melhorá-la. Uma única execução das abordagens metaheurísticas é aplicada e se uma delas consegue melhorar a solução, então uma nova execução das abordagens é realizada tentando melhorar ainda mais a solução. Esse esquema adaptativo não é computacionalmente dispendioso e permite empregar as abordagens metaheurísticas mais frequentemente se elas forem bem-sucedidas para melhorar a solução. O método é executado até

atingir um dos critérios de parada. O primeiro deles consiste na prova da otimalidade da solução por parte do método, enquanto o segundo consiste no tempo limite de execução predefinido do algoritmo.

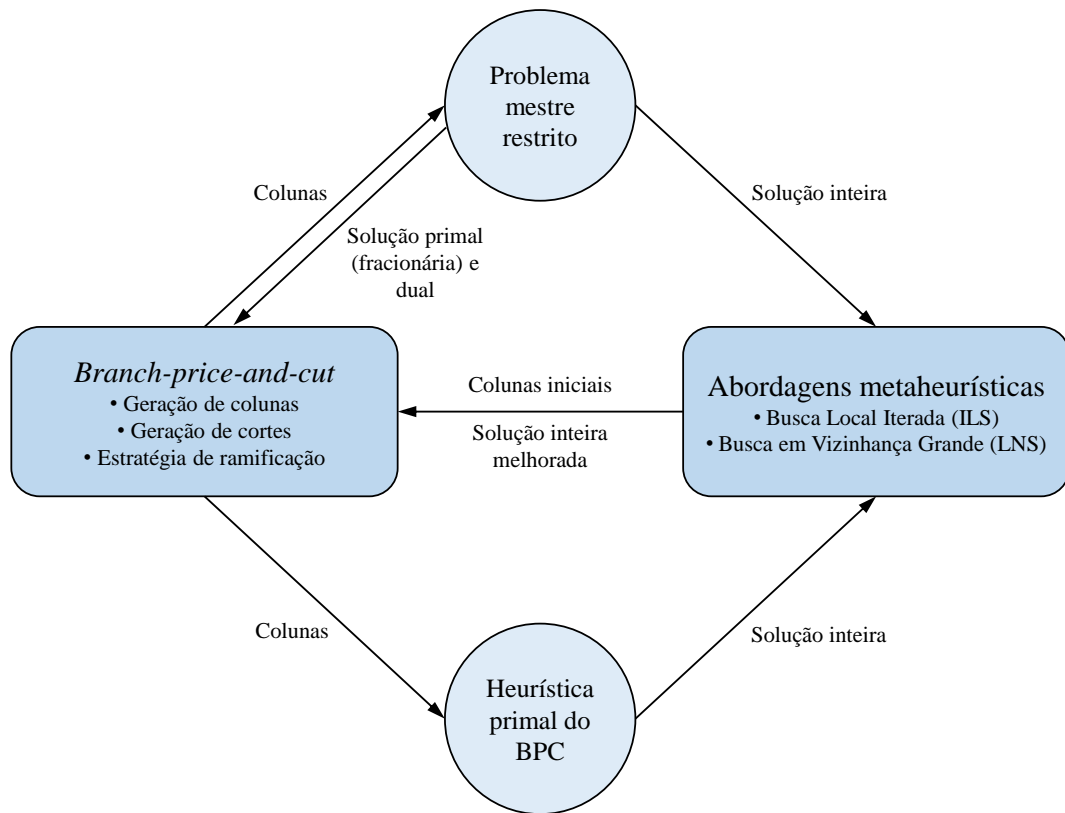


Figura 4.1: Estrutura do método híbrido exato.

Uma das ideias que fundamenta este método híbrido é o esquema de cooperação realizado entre as partes do método. Dado que as metaheurísticas foram desenvolvidas especialmente para obter boas soluções para o problema, elas ajudam o BPC a encontrar melhores soluções inteiras de forma mais precoce. Em contrapartida, o BPC ajuda as metaheurísticas fornecendo diferentes soluções iniciais para elas.

4.5 Método híbrido heurístico: um gerador de colunas metaheurístico

Partindo do fato de que o maior esforço em algoritmos do tipo *branch-and-price* reside na resolução dos subproblemas (MUNARI; GONDZIO, 2013; BARNHART *et al.*, 1998), foi proposta uma metaheurística de Busca Local Iterada (ILS) para resolver o ESPPRC. Esta metaheurística foi desenvolvida com o intuito de gerar rapidamente colunas com custo reduzido negativo e, dessa forma, tentar acelerar a resolução do PM em cada nó do árvore. A partir da incorporação da ILS na geração de colunas, foram desenvolvidas duas variantes do método híbrido heurístico. Na primeira variante, tenta-se resolver os subproblemas utilizando unicamente a metaheurística ILS desenvolvida, enquanto que na segunda variante

do método, uma abordagem de duas fases é aplicada. Sempre que a metaheurística conseguir gerar colunas com custo reduzido negativo, elas são adicionadas ao PMR correspondente e, caso ela não o consiga, o algoritmo de programação dinâmica é chamado para provar a otimalidade ou para gerar mais colunas com custo reduzido negativo. Nas seguintes subseções descrevemos a ILS desenvolvida e seus componentes.

4.5.1 Uma metaheurística ILS para o ESPPRC

Como explicado no Capítulo 3, a metaheurística ILS baseia-se na aplicação de uma busca local de forma repetida sobre a perturbação dos ótimos locais previamente visitados com a intenção de melhorar progressivamente a solução e explorar eficientemente o espaço de busca de forma simultânea. Neste caso, a metaheurística tenta encontrar rotas com o menor custo reduzido possível, partindo de um conjunto inicial de rotas. Vale a pena lembrar que devem-se considerar as variáveis duais associadas aos cortes no PRM no momento de calcular o custo reduzido das colunas geradas.

A estrutura da metaheurística ILS desenvolvida para resolver o ESPPRC é mostrada no Algoritmo 3. A metaheurística é executada de forma iterativa até completar um dos três critérios de parada (linha 5): explorar todas as colunas iniciais (\mathcal{R}), gerar o número máximo de colunas ($maxCol$) ou atingir o tempo limite de execução ($maxTempo$). Para cada rota inicial, a metaheurística aplica inicialmente a busca local (linha 7), que em nosso caso consiste em uma Busca em Vizinhança Variável com Ordenação Aleatória (RVND); em seguida, o algoritmo entra em um ciclo de melhoria (linhas 8-21) perturbando a solução através do operador respectivo (linha 9) e melhorando por meio da aplicação da heurística de busca local (linha 10). O ciclo se repete até completar $MaxIterILS$ iterações consecutivas sem melhoria, atualizando a melhor solução em cada iteração (linhas 11-12) e adicionando-a ao conjunto \mathcal{R}^* , caso seu custo reduzido seja negativo (linhas 13-15). Note que a perturbação aplica-se sempre à melhor solução atual (linha 9) e que nosso critério de aceitação (linha 11) admite a rota somente se seu custo reduzido foi melhorado. Os componentes usados na ILS são descritos a seguir.

4.5.1.1 Busca Local

A busca local aplicada na ILS é uma Busca de Vizinhança Variável com Ordenação Aleatória (veja Seção 3.2.1). Nesta aplicação utilizaram-se três estruturas de vizinhança, a saber: *inserção*, *extração* e *troca*.

O operador de *inserção* tenta acrescentar iterativamente nós à rota atual até completar o número máximo de nós inseridos ou o número máximo de nós cuja inserção pode ser testada. A execução das inserções é aplicada de forma que testa-se primeiro a inserção de nós associados às variáveis duais de maior valor, utilizando o mesmo procedimento aplicado nos operadores de remoção da Seção 3.3.1, e verificando se a inserção do nó gera uma redução do custo reduzido da rota. Por outro lado, o operador

Algoritmo 3: Estrutura da Busca Local Iterada para o ESPPRC.

Entrada: Conjunto de rotas $\mathcal{R} = \{r_1, \dots, r_n\}$, parâmetros ($maxCol$, $maxTempo$, $MaxIterILS$);
Saída: Conjunto de rotas com custo reduzido negativo \mathcal{R}^* ;

```
1 início
2    $i \leftarrow 1$  (rotas exploradas);
3    $\mathcal{R}^* \leftarrow \emptyset$  (rotas geradas);
4   tempoDecorrido  $\leftarrow 0$ ;
5   enquanto ( $i \leq n$ ) & ( $|\mathcal{R}^*| < maxCol$ ) & ( $tempoDecorrido < maxTempo$ ) faça
6     iterILS  $\leftarrow 0$ ;
7      $r^* \leftarrow RVND(r_i)$ ;
8     enquanto iterILS < MaxIterILS faça
9        $r' \leftarrow Perturbar(r^*)$ ;
10       $r' \leftarrow RVND(r')$ ;
11      se  $f(r') < f(r^*)$  então
12         $r^* \leftarrow r'$ ;
13        se  $f(r^*) < 0$  então
14          Adicionar  $r^*$  a  $\mathcal{R}^*$ ;
15        fim
16      iterILS  $\leftarrow 0$ ;
17    fim
18    senão
19      iterILS  $\leftarrow iterILS + 1$ ;
20    fim
21  fim
22  Atualizar tempoDecorrido;
23   $i \leftarrow i + 1$ ;
24 fim
25 fim
```

de *extração* tenta retirar de forma iterativa nós da rota atual se a respectiva remoção resulta em uma redução do custo reduzido da rota. Finalmente, o procedimento de *troca* testa o intercâmbio de cada par de nós adjacentes da rota atual sempre que a permuta resulte em uma redução do custo reduzido da rota.

Note que no momento de aplicar as estruturas de vizinhança deve-se conferir a factibilidade das operações em termos de adjacência dos nós (ramificação nos arcos do grafo) e das restrições do problema (capacidade e janelas de tempo).

4.5.1.2 Perturbação

A perturbação é o mecanismo responsável pela diversificação dentro da metaheurística ILS. Em nossa implementação, o procedimento de perturbação consiste na remoção de um número aleatório de nós da rota atual. Este simples procedimento possibilita que a metaheurística consiga escapar dos ótimos locais e desloca a busca para outras áreas do espaço de soluções.

4.6 Considerações finais

Neste capítulo, descreveram-se os métodos híbridos propostos para resolver o PRVJTME. Diferentes estratégias de hibridização foram detalhadas, todas geradas a partir da combinação de um método BPC com diversos métodos metaheurísticos. Uma heurística primal foi desenvolvida com a intenção de melhorar a capacidade do BPC para encontrar soluções inteiras. A partir desse método desenvolveu-se um método híbrido exato, combinando as metaheurísticas apresentadas no Capítulo 3 com o BPC. Este método visa melhorar ainda mais a habilidade do BPC para encontrar soluções inteiras para o problema. Posteriormente, desenvolveu-se um gerador de colunas metaheurístico onde estuda-se o uso de uma metaheurística para resolver e apoiar na resolução do subproblema. Usando instâncias da literatura, avaliaram-se os métodos híbridos desenvolvidos e os resultados encontram-se no Capítulo 5.

Adicionalmente, considerando que resolver diretamente a formulação (2.1), apresentada no Capítulo 2, não é viável quando é utilizado um software de otimização como CPLEX, desenvolveu-se uma heurística de programação matemática tipo *relax-and-fix* (DILLENBERGER *et al.*, 1994). Foram realizados experimentos utilizando diferentes estratégias de fixação de variáveis, no entanto, dado que a estrutura dessa formulação não é propícia para que esse tipo de método possa ter o desempenho apropriado, os resultados não foram promissores. Assim, só se faz menção dela como uma estratégia de solução que foi testada, mas não será descrita nesta dissertação.

Capítulo 5

Experimentos computacionais

Neste capítulo analisamos e comparamos os resultados dos métodos apresentados nos Capítulos 3 e 4. Inicialmente, na Seção 5.1 são apresentadas as instâncias usadas para testar os métodos propostos. Na Seção 5.2 são descritos alguns detalhes de implementação computacional dos métodos de solução desenvolvidos. A seguir, na Seção 5.3, é apresentada uma análise estatística realizada com a intenção de calibrar as abordagens metaheurísticas propostas. Em seguida, as abordagens metaheurísticas foram testadas como métodos de solução independentes. Seus desempenhos são comparados entre si, bem como com métodos da literatura. Os resultados são mostrados na Seção 5.4. Posteriormente, foram avaliados os métodos híbridos desenvolvidos, de modo a determinar os ganhos obtidos pela inclusão de seus componentes. Os resultados são apresentados na Seção 5.5. Uma análise da fonte das soluções nos métodos híbridos é mostrada na Seção 5.6. Na Seção 5.7 é feita uma comparação entre todos os métodos de solução desenvolvidos. Finalmente, na Seção 5.8 são feitas algumas análises adicionais.

Nas tabelas que apresentam resultados neste capítulo, os rótulos *Inst*, *Custo*, *Veic*, *Dist*, *Entr* e *Tempo* indicam a instância resolvida, o custo total da solução, o número de veículos usados, a distância total percorrida, o número de entregadores designados e o tempo de execução do método (em segundos), respectivamente. Os melhores resultados para cada instância/conjunto, considerando o custo total da solução, são destacados em negrito. No caso de empate entre métodos, considerando precisão de duas casas decimais, destaca-se aquela com menor distância total percorrida. Adicionalmente, em algumas das tabelas deste capítulo os tempos reportados podem ser menores ao tempo limite correspondente, o qual indica que foi provada a otimalidade da solução antes de atingir o tempo limite de execução correspondente (estas soluções são marcadas com *). Essa situação só pode acontecer para os métodos híbridos que conservam o componente exato do BPC. Por outro lado, quando os tempos reportados são maiores que o tempo máximo correspondente é porque ao atingir o limite de tempo nos métodos híbridos, sempre é executada a heurística primal proposta para o BPC.

5.1 Instâncias utilizadas

Nos experimentos, foram usadas as instâncias propostas por Pureza, Morabito e Reimann (2012), as quais são baseadas nas conhecidas instâncias de Solomon (1987) para o PRVJT. Cada uma delas possui 100 grupos de clientes. No total são 56 instâncias agrupadas em 6 conjuntos, a saber: R1, R2, C1, C2, RC1 e RC2. As instâncias do tipo R apresentam grupos de clientes distribuídos aleatoriamente no espaço geográfico bidimensional, enquanto as instâncias do tipo C apresentam grupos distribuídos de forma aglomerada. As instâncias do tipo RC apresentam uma mistura dessas características (aleatória e aglomerada). Por outro lado, as instâncias do tipo 1 se caracterizam por horizontes de planejamento curtos, veículos com capacidade relativamente pequena e janelas de tempo estreitas. Já as instâncias do tipo 2 se caracterizam por horizontes de planejamento longos, veículos de grande capacidade e janelas de tempo amplas. Note que as características das instâncias do tipo 1 permitem que poucos grupos sejam atendidos por cada rota, enquanto que as características das instâncias do tipo 2 permitem que muitos grupos possam ser atendidos pela mesma rota. Devido suas características, as instâncias tipo 2 resultam mais difíceis de serem resolvidas por métodos tipo BPC, como observado na literatura (MUNARI; MORABITO, 2015; MUNARI; GONDZIO, 2013; DESAULNIERS; LESSARD; HADJAR, 2008).

Nas instâncias originais de Solomon não há diferenciação dos tempos de serviço de acordo com o número de entregadores. Assim, os tempos de serviço precisam ser modificados de forma a representar o tempo de entrega da demanda acumulada dos clientes de cada grupo, considerando que as entregas devem ser feitas a pé pelos entregadores da rota. A modificação nos tempos de serviço, conforme a proposta de Pureza, Morabito e Reimann (2012), é dada por:

$$s_{il} = \min\{rs * q_i, T - \max\{w_i^a, d_{0i}\} - d_{i0}\}/l, \quad i = 1, \dots, n, l = 1, 2, 3. \quad (5.1)$$

onde q_i é a demanda do grupo i , rs é a taxa de serviço (nos experimentos adotou-se $rs = 2$), T é o tempo máximo de retorno ao depósito (encerramento da janela de tempo do depósito) e $d_{0i} = d_{i0}$ é a distância entre o depósito e o grupo i . A lógica para essa definição dos tempos de serviço consiste no fato de que a demanda de cada grupo representa a demanda acumulada dos clientes que o integram e, portanto, o tempo de serviço deve estar relacionado com ela. Ao mesmo tempo, deve-se manter a factibilidade da instância, o qual é garantido com o segundo termo da equação.

Os custos na função objetivo foram definidos com os mesmos valores usados por Pureza, Morabito e Reimann (2012): $p_1 = 1,0$, $p_2 = 0,1$ e $p_3 = 0,0001$; os quais priorizam a minimização do número de veículos seguido da minimização do número de entregadores e, finalmente, a minimização da distância total percorrida.

5.2 Implementação computacional

Todos os algoritmos foram implementados em linguagem C++ e os experimentos foram realizados em um computador com processador Intel Core i7 3.40 GHz com 16 GB de memória RAM, usando-se um único núcleo do processador. O software de otimização usado para a implementação da heurística primal, apresentada na Seção 4.3, foi o CPLEX 12.4 por meio da biblioteca *Concert* (CPLEX, 2009). O critério de parada para todas as execuções dos métodos foi o tempo, mas os métodos híbridos exatos podem parar por otimalidade. Nesta pesquisa, a árvore BPC foi administrada usando o *framework* do *branch-and-price* de pontos interiores de Munari e Gondzio (2013) e Munari e Morabito (2015).

Adicionalmente, nesta seção são comentados alguns detalhes de implementação usados para melhorar o desempenho dos algoritmos. Para o BPC, um PMR deve ser fornecido para iniciar o processo de geração de colunas, o qual deve ter solução factível a fim de garantir que uma solução dual será repassada para o subproblema. O BPC é inicializado com um conjunto de colunas correspondentes a rotas realizando visitas individuais a cada grupo de clientes em cada modo, as quais são mantidas em todos os nós da árvore. Além do mais, são usadas colunas iniciais geradas por meio de um algoritmo baseado na heurística de economias de Clarke e Wright (1964) e buscas locais simples. Estas últimas colunas não são usadas nos métodos híbridos que utilizam as abordagens metaheurísticas.

No *framework* do *branch-and-price* adotado é usado o método primal-dual de geração de colunas (GONDZIO; GONZÁLEZ-BREVIS; MUNARI, 2013), em que os PMRs são resolvidos usando um algoritmo de pontos interiores, obtendo-se soluções primais e duais sub-ótimas bem centralizadas. Devido a essa centralidade, as soluções duais oscilam menos de uma iteração para a outra, contribuindo para melhorar o desempenho do método de geração de colunas, em termos do número de iterações e tempo de execução (MUNARI; GONDZIO, 2013; ROUSSEAU; GENDREAU; FEILLET, 2007). Para melhorar a estratégia de ramificação, o BPC usa a técnica *strong branching* para selecionar dentre os candidatos aquele cuja ramificação é mais promissora. Ainda, o processo de ramificação é realizado usando a abordagem *early branching*, conforme descrito por Munari e Gondzio (2013).

Nos métodos híbridos, cada execução da heurística primal foi limitada a 30 segundos, enquanto que a execução das metaheurísticas dentro deles é limitada a 5 segundos toda vez que elas são chamadas. O número inicial de colunas geradas na inicializar o BPC durante a primeira fase dos métodos híbridos foi definido como 6000 e 3000 colunas para as instâncias do tipo 1 e 2, respectivamente. Estes valores foram definidos através de testes computacionais preliminares.

Como conjunto de soluções iniciais para a ILS usada no método de geração de colunas dos métodos híbridos se utilizaram as rotas com menor custo reduzido associadas às variáveis do PMR atual. Elas são boas candidatas para obter colunas com custo reduzido negativo já que o seu é próximo de zero.

5.3 Ajuste dos parâmetros das metaheurísticas

As abordagens metaheurísticas foram calibradas de acordo com a técnica de planejamento de experimentos (MONTGOMERY, 2008), de forma similar ao apresentado em (NADERI; RUIZ; ZANDIEH, 2010). Para isso, foi realizado um experimento fatorial completo de modo a testar os parâmetros em diferentes níveis, mostrados na Tabela 5.1. Os níveis testados para cada fator foram determinados realizando testes preliminares.

Algoritmo	Parâmetro/fator	Níveis
ILS	$MaxIter_{ILS}$	100; 150; 200
	nP	1; 3; 5
LNS	$MaxIter_{LNS}$	500; 1000; 1500
	q	até 0,2n; 0,3n; 0,4n

Tabela 5.1: Fatores e níveis considerados no planejamento experimental.

As combinações resultaram em 9 configurações diferentes para cada algoritmo. Um conjunto de 20 instâncias foi utilizado para calibrar os algoritmos, as quais foram escolhidas aleatoriamente dentre as 56 instâncias disponíveis, realizando 5 réplicas para cada combinação configuração/instância, resultando em um total de $9 \times 20 \times 5 = 900$ observações. Um tempo limite de 5 minutos foi imposto para cada execução dos algoritmos, os quais foram realizados em um computador Dell Precision T7600 CPU E5-6280 2.70 GHz com 192 GB RAM, usando-se um único núcleo do processador. O desempenho das abordagens em cada instância foi medido como o *gap* do custo da solução encontrada (Alg_{sol}) em relação ao custo da melhor solução encontrada por todas as configurações (Min_{sol}) conforme definido na Equação (5.2).

$$gap = \frac{Alg_{sol} - Min_{sol}}{Min_{sol}} \quad (5.2)$$

Os resultados dos experimentos foram analisados por meio de uma análise de variância (ANOVA) usando o software estatístico *Statgraphics Centurion XV*, verificando os pressupostos de homocedasticidade e independência dos erros por meio das técnicas correspondentes, não encontrando-se nenhuma evidência para questionar sua validade. Em relação à normalidade, encontrou-se que os erros não se ajustam a uma distribuição normal. No entanto, é bem sabido que a ANOVA é robusta a pequenas violações do pressuposto de normalidade (LIX; KESELMAN; KESELMAN, 1996; GLASS; PECKHAM; SANDERS, 1972; BOX; ANDERSEN, 1955) e daí concluímos que a validade do experimento não fica comprometida. Os gráficos de médias e intervalos de *Tukey* com 95% de confiança são mostrados na Figuras 5.1 e 5.2, cabendo ressaltar que a sobreposição entre intervalos indica que não existem diferenças estatisticamente significativas entre as médias.

Conforme pode ser observado nas Figuras 5.1 e 5.2, os resultados mostraram que não existem diferenças estatisticamente significativas no desempenho de nenhum dos métodos, nos diferentes níveis

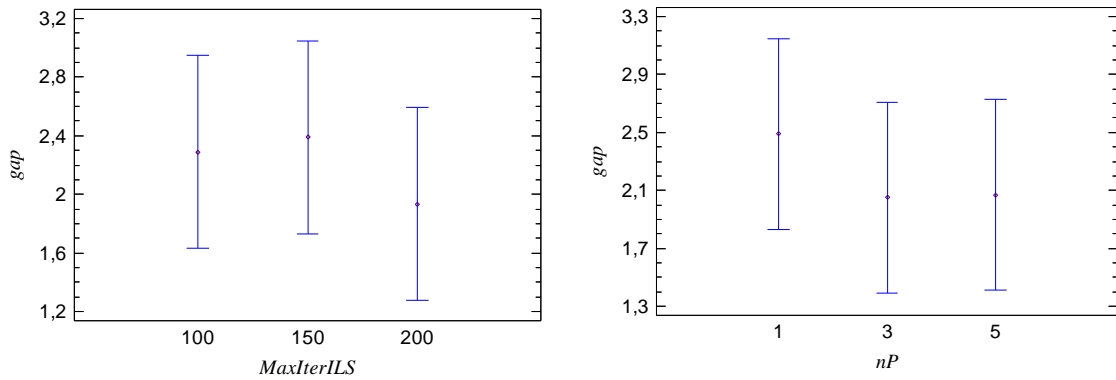


Figura 5.1: Gráfico de médias e intervalos de confiança de 95% de *Tukey* para os parâmetros da ILS.

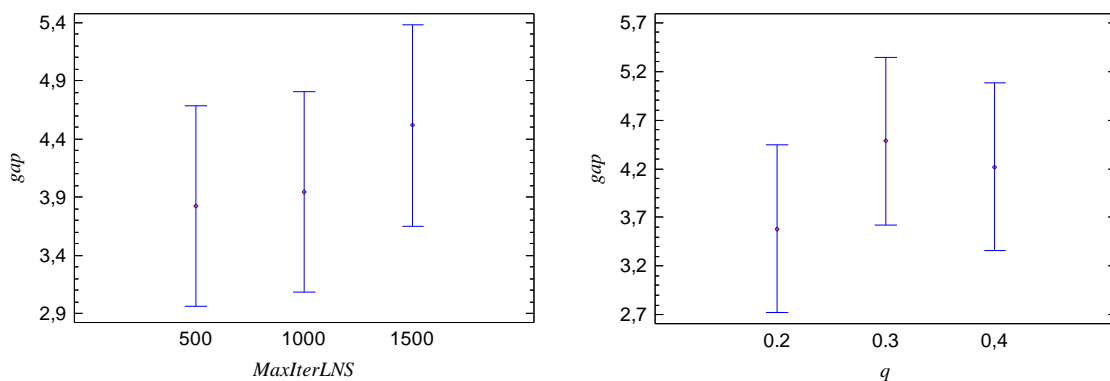


Figura 5.2: Gráfico de médias e intervalos de confiança de 95% de *Tukey* para os parâmetros da LNS.

de seus parâmetros. Este resultado pode ser explicado pela grande dependência que as abordagens metaheurísticas têm das heurísticas adicionais (redução de rotas e de entregadores), devido às características específicas e a complexidade do problema. Note que as heurísticas adicionais não possuem parâmetros e, conseqüentemente, fazem com que a sensibilidade das abordagens em relação aos parâmetros seja baixa.

Apesar de não existir diferenças estatisticamente significativas, os valores dos parâmetros foram definidos com base nos melhores resultados dos testes, ficando fixados como segue: $MaxIterILS = 200$, $nP = 3$, $MaxIterLNS = 1000$ e $q = rand(0,1; 0,2)$. Por outro lado, o valor do componente de aleatorização nos operadores de remoção foi $p = 3$, conforme a proposta de Ropke e Pisinger (2006). Este valor é suficientemente grande para permitir a remoção de grupos de maior custo toda vez que os operadores são usados e, ainda, permite evitar as situações nas quais os mesmos grupos são removidos uma e outra vez.

5.4 Avaliação das abordagens metaheurísticas

Com o propósito de validar as abordagens metaheurísticas, seus desempenhos foram comparados entre si, bem como com os desempenhos de outras abordagens metaheurísticas encontradas na literatura. As

abordagens foram testadas usando 600 segundos como tempo limite de execução e cada instância foi resolvida cinco vezes.

Nas Tabelas 5.2 e 5.3 são reportados os melhores resultados para cada instância, considerando as cinco execuções realizadas. Os resultados mostram que a ILS encontrou a melhor solução em 9 das 12 instâncias do conjunto R1, em 5 das 8 instâncias do conjunto RC1, em 9 das 11 instâncias do conjunto R2 e em 4 das 8 instâncias do conjunto RC2. Em relação aos conjuntos C1 e C2, ambas abordagens obtiveram as mesmas soluções para todas as instâncias. Contudo, apesar do domínio da ILS em relação ao número de melhores soluções encontradas, a ILS domina a LNS em termos de resultados médios agrupados somente em um dos seis conjuntos de instâncias, é dominada pela LNS em três conjuntos e empata para os outros dois - veja Tabela 5.4. Essa observação indica que quando a LNS consegue superar a ILS em uma instância, o faz com diferenças significativas de forma que lhe permite dominá-la em termos de resultados médios.

Inst	ILS (melhor de 5)					LNS (melhor de 5)				
	Custo	Veic	Dist	Entr	Tempo	Custo	Veic	Dist	Entr	Tempo
R101	23,68	19	1757,40	45	600	23,68	19	1755,04	45	600
R102	20,95	17	1530,92	38	601	20,95	17	1531,36	38	603
R103	15,84	13	1410,73	27	601	15,94	13	1369,12	28	603
R104	12,71	10	1062,58	26	601	12,71	10	1072,41	26	603
R105	17,64	14	1412,52	35	600	17,64	14	1412,52	35	600
R106	15,04	12	1352,09	29	600	15,03	12	1328,21	29	601
R107	12,81	10	1140,79	27	603	12,91	10	1139,96	28	600
R108	11,70	9	965,31	26	601	11,70	9	1011,37	26	601
R109	15,03	12	1291,62	29	600	14,43	11	1281,84	33	605
R110	13,92	11	1187,49	28	600	13,92	11	1219,12	28	603
R111	13,02	10	1161,96	29	602	13,11	10	1149,03	30	607
R112	11,80	9	987,14	27	600	11,80	9	989,70	27	603
RC101	20,38	16	1791,30	42	601	20,28	16	1842,68	41	602
RC102	18,07	14	1697,19	39	601	17,86	14	1644,59	37	603
RC103	15,24	12	1408,14	31	601	15,24	12	1409,48	31	601
RC104	14,02	11	1242,45	29	603	14,02	11	1247,64	29	608
RC105	18,77	15	1698,62	36	602	18,27	14	1727,28	41	602
RC106	16,75	13	1493,28	36	602	16,75	13	1517,93	36	605
RC107	15,23	12	1325,32	31	600	15,33	12	1312,43	32	601
RC108	14,22	11	1203,36	31	600	14,22	11	1236,24	31	604
C101	11,08	10	828,94	10	600	11,08	10	828,94	10	600
C102	11,08	10	828,94	10	600	11,08	10	828,94	10	601
C103	11,08	10	828,06	10	600	11,08	10	828,06	10	600
C104	11,08	10	819,81	10	601	11,08	10	819,81	10	600
C105	11,08	10	828,94	10	600	11,08	10	828,94	10	600
C106	11,08	10	828,94	10	600	11,08	10	828,94	10	601
C107	11,08	10	828,94	10	600	11,08	10	828,94	10	600
C108	11,08	10	828,94	10	600	11,08	10	828,94	10	601
C109	11,08	10	827,26	10	601	11,08	10	827,26	10	603

Tabela 5.2: Resultados das abordagens metaheurísticas para as instâncias tipo 1 (600 segundos).

Inst	ILS (melhor de 5)					LNS (melhor de 5)				
	Custo	Veic	Dist	Entr	Tempo	Custo	Veic	Dist	Entr	Tempo
R201	4,93	4	1344,99	8	601	4,93	4	1345,59	8	607
R202	4,82	4	1161,70	7	603	4,02	3	1247,40	9	604
R203	3,70	3	983,76	6	606	3,70	3	983,97	6	601
R204	2,69	2	883,10	6	600	2,69	2	859,71	6	611
R205	3,80	3	1039,14	7	605	3,81	3	1067,70	7	603
R206	3,70	3	959,96	6	604	3,70	3	968,03	6	600
R207	2,70	2	950,19	6	600	3,59	3	875,41	5	633
R208	2,57	2	740,83	5	609	2,57	2	746,23	5	634
R209	3,70	3	982,55	6	606	3,70	3	1003,84	6	606
R210	3,70	3	992,55	6	603	3,70	3	993,90	6	608
R211	3,59	3	886,11	5	608	3,59	3	886,16	5	604
RC201	5,15	4	1478,24	10	601	5,15	4	1478,24	10	602
RC202	4,93	4	1257,67	8	602	4,93	4	1258,72	8	600
RC203	3,91	3	1127,09	8	600	3,91	3	1115,92	8	610
RC204	3,69	3	887,12	6	619	3,69	3	892,48	6	617
RC205	5,04	4	1437,01	9	603	4,95	4	1535,65	8	603
RC206	4,03	3	1265,14	9	600	4,03	3	1261,36	9	604
RC207	3,91	3	1120,86	8	600	3,91	3	1116,27	8	613
RC208	3,79	3	919,75	7	601	3,79	3	921,32	7	627
C201	3,36	3	588,88	3	600	3,36	3	588,88	3	600
C202	3,36	3	588,88	3	601	3,36	3	588,88	3	604
C203	3,36	3	585,27	3	603	3,36	3	585,27	3	607
C204	3,36	3	584,49	3	607	3,36	3	584,49	3	600
C205	3,36	3	588,49	3	601	3,36	3	588,49	3	603
C206	3,36	3	588,49	3	600	3,36	3	588,49	3	603
C207	3,36	3	587,31	3	600	3,36	3	587,31	3	604
C208	3,36	3	588,29	3	600	3,36	3	588,29	3	601

Tabela 5.3: Resultados das abordagens metaheurísticas para as instâncias tipo 2 (600 segundos).

Seguidamente, os resultados das abordagens foram comparados com resultados disponíveis na literatura. Especificamente, foi feita uma comparação com os resultados agrupados das abordagens de Busca Tabu (TS-PMR) e Otimização por Colônia de Formigas (ACO-PMR) de Pureza, Morabito e Reimann (2012). Comparou-se diretamente com os resultados reportados pelos autores, os quais foram obtidos em um computador com processador Intel Core2 2.40 GHz com 2 GB de memória RAM, já que não pretende-se avaliar o desempenho dos métodos em termos de tempos de execução, mas sim em relação à qualidade das soluções obtidas.

A Tabela 5.4 mostra as médias dos melhores resultados (agrupados) gerados pelas diferentes abordagens metaheurísticas para cada um dos conjuntos de instâncias. Observa-se que, em termos de qualidade das soluções obtidas, as abordagens metaheurísticas propostas nesta pesquisa superaram em todos os conjuntos de instâncias as abordagens TS-PMR e ACO-PMR. As diferenças relativas entre esses melhores resultados encontrados pelas diferentes abordagens metaheurísticas são mostradas na Tabela 5.5. A segunda e a terceira coluna mostram a porcentagem de redução do custo que foi atingido pela ILS em

relação TS-PMR e ACO-PMR, respectivamente. Os valores da quarta e quinta coluna se interpretam de forma análoga, mas em relação à LNS. Observa-se que a ILS e a LNS conseguiram superar os resultados médios para 4 dos 6 conjuntos de instâncias de ambas abordagens (TS-PMR e ACO-PMR), indicando que as abordagens metaheurísticas propostas nesta pesquisa são superiores em termos dos resultados encontrados. Deve-se se mencionar que não foi possível comparar todos os resultados obtidos com as abordagens desenvolvidas por Senarclens De Grancy e Reimann (2014), já que nesse artigo só foram reportados os resultados para o conjunto R1, os quais são abordados detalhadamente na Tabela 5.7.

Método		Conjunto de instâncias					
		R1	RC1	C1	R2	RC2	C2
TS-PMR (melhor de 5)	Custo	15,70	16,64	11,08	3,75	4,45	3,36
	Veic	12,33	13,00	10,00	2,90	3,40	3,00
	Dist	1258,00	1527,90	830,70	1034,00	1230,40	597,20
	Entr	32,42	34,90	10,00	7,50	9,30	3,00
	Tempo	640,10	677,10	265,10	425,40	419,10	246,80
ACO-PMR (melhor de 5)	Custo	15,77	16,70	11,08	3,86	4,58	3,36
	Veic	12,50	13,00	10,00	3,10	3,60	3,00
	Dist	1261,50	1480,10	833,60	1064,20	1296,00	609,30
	Entr	31,40	35,50	10,00	6,50	8,50	3,00
	Tempo	575,80	508,60	375,20	600,60	462,00	243,30
ILS (melhor de 5)	Custo	15,34	16,59	11,08	3,63	4,31	3,36
	Veic	12,17	13,00	10,00	2,91	3,38	3,00
	Dist	1271,71	1482,46	827,64	993,17	1186,61	587,51
	Entr	30,50	34,38	10,00	6,18	8,13	3,00
	Tempo	600,75	601,25	600,22	604,09	603,25	601,50
LNS (melhor de 5)	Custo	15,32	16,50	11,08	3,64	4,29	3,36
	Veic	12,08	12,88	10,00	2,91	3,38	3,00
	Dist	1271,64	1492,29	827,64	998,00	1197,49	587,51
	Entr	31,08	34,75	10,00	6,27	8,00	3,00
	Tempo	602,42	603,25	600,67	610,09	609,50	602,75

Tabela 5.4: Melhores resultados (agrupados) das abordagens metaheurísticas vs métodos da literatura.

Conjunto	ILS		LNS	
	<i>gap_{TS}</i>	<i>gap_{ACO}</i>	<i>gap_{TS}</i>	<i>gap_{ACO}</i>
R1	2,27%	2,70%	2,43%	2,86%
RC1	0,33%	0,68%	0,85%	1,20%
C1	0,00%	0,00%	0,00%	0,00%
R2	3,29%	6,05%	3,04%	5,80%
RC2	3,23%	5,98%	3,49%	6,23%
C2	0,00%	0,00%	0,00%	0,00%

Tabela 5.5: Diferenças relativas entre os melhores resultados das abordagens metaheurísticas.

A Tabela 5.6 apresenta os resultados médios agrupados para cada conjunto de instâncias, considerando as cinco execuções de cada abordagem. Semelhante aos resultados mostrados na Tabela 5.4, constata-se que as abordagens metaheurísticas baseadas em ILS e LNS superaram em todos os conjuntos

os resultados médios das abordagens ACO-PMR e TS-PMR.

Método		Conjunto de instâncias					
		R1	RC1	C1	R2	RC2	C2
TS-PMR (média de 5)	Custo	16,20	17,12	11,08	3,89	4,50	3,37
	Veic	12,80	13,40	10,00	3,00	3,40	3,00
	Dist	1272,40	1511,40	847,70	1046,80	1251,30	653,50
	Entr	32,70	35,70	10,00	7,90	9,70	3,00
	Tempo	658,20	686,50	245,70	393,00	400,50	261,90
ACO-PMR (média de 5)	Custo	15,90	17,08	11,08	3,89	4,69	3,36
	Veic	12,60	13,40	10,00	3,10	3,70	3,00
	Dist	1263,20	1496,30	838,80	1070,30	1307,90	623,70
	Entr	31,70	35,30	10,00	6,80	8,60	3,00
	Tempo	508,80	472,80	389,10	527,50	455,60	263,80
ILS (média de 5)	Custo	15,45	16,68	11,08	3,76	4,35	3,36
	Veic	12,25	13,10	10,00	3,05	3,43	3,00
	Dist	1260,58	1478,90	827,64	988,71	1203,95	587,51
	Entr	30,73	34,35	10,00	6,07	8,03	3,00
	Tempo	601,03	600,98	600,44	606,27	604,20	603,29
LNS (média de 5)	Custo	15,51	16,60	11,08	3,73	4,31	3,36
	Veic	12,30	12,98	10,00	3,02	3,38	3,00
	Dist	1267,65	1488,90	827,64	997,41	1217,48	587,51
	Entr	30,87	34,80	10,00	6,15	8,10	3,00
	Tempo	602,45	602,65	601,89	613,05	608,95	606,42

Tabela 5.6: Resultados médios (agrupados) das abordagens metaheurísticas vs métodos da literatura.

Finalmente, na Tabela 5.7 são detalhados os resultados encontrados no conjunto R1. Estas instâncias foram detalhadamente abordadas por Pureza, Morabito e Reimann (2012) e Senarclens De Grancy e Reimann (2014) já que suas características fazem com que elas representem mais acertadamente a importância dos tempos de serviço. Os resultados mostram que as abordagens propostas são capazes de encontrar a melhor solução em 7 das 12 instâncias do conjunto. Pode-se observar que as abordagens propostas nesta pesquisa conseguiram superar ligeiramente o desempenho da TS-PMR, sendo que nossa ILS diminuiu em 1,30% e 5,92% o número médio de veículos e entregadores utilizados respectivamente, enquanto a LNS conseguiu reduzir 2,03% e 4,11% o número médio de veículos e entregadores utilizados, respectivamente. Comparando com ACO-SR, a ILS obteve resultados com o mesmo número de veículos e reduzindo 0,81% o número médio de entregadores usados, enquanto a LNS diminuiu em 0,75% o número de veículos utilizando 1,07% mais entregadores.

Vale ressaltar que para a instância R101, cuja solução ótima é conhecida, nossas abordagens encontraram soluções usando a quantidade ótima de veículos e entregadores, com um incremento de 2,17% e 2,03% na distância total percorrida, respectivamente.

Método		Instância											Média	Soma	
		R101	R102	R103	R104	R105	R106	R107	R108	R109	R110	R111			R112
TS-PMR (melhor de 5)	Custo	23,67	21,05	16,33	12,91	17,84	15,23	13,01	12,80	15,43	14,12	13,11	12,90	15,70	188,41
	Veic	19	17	13	10	14	12	10	10	12	11	10	10	12,33	148
	Dist	1740,00	1520,00	1285,00	1057,00	1446,00	1323,00	1112,00	967,00	1296,00	1217,00	1137,00	996,00	1258,00	15096,00
	Entr	45	39	32	28	37	31	29	27	33	30	30	28	32,42	389
	Temp	645	655	959	692	463	492	473	953	428	620	616	686	640,17	7682
ACO-SR (melhor de 5)	Custo	23,67	20,95	15,94	12,70	17,64	15,13	12,81	11,70	15,12	13,92	13,01	11,80	15,37	184,40
	Veic	19	17	13	10	14	12	10	9	12	11	10	9	12,17	146
	Dist	1725,46	1533,40	1371,63	1045,68	1412,52	1301,34	1108,92	967,18	1229,72	1154,95	1134,16	996,32	1248,44	14981,28
	Entr	45	38	28	26	35	30	27	26	30	28	29	27	30,75	369
	Temp	960	960	960	960	960	960	960	960	960	960	960	960	960,00	11520
ILS (melhor de 5)	Custo	23,68	20,95	15,84	12,71	17,64	15,04	12,81	11,70	15,03	13,92	13,02	11,80	15,34	184,13
	Veic	19	17	13	10	14	12	10	9	12	11	10	9	12,17	146
	Dist	1757,40	1530,92	1410,73	1062,58	1412,52	1352,09	1140,79	965,31	1291,62	1187,49	1161,96	987,14	1271,71	15260,57
	Entr	45	38	27	26	35	29	27	26	29	28	29	27	30,50	366
	Temp	600	601	601	601	600	600	603	601	600	600	602	600	600,75	7209
LNS (melhor de 5)	Custo	23,68	20,95	15,94	12,71	17,64	15,03	12,91	11,70	14,43	13,92	13,11	11,80	15,32	183,83
	Veic	19	17	13	10	14	12	10	9	11	11	10	9	12,08	145
	Dist	1755,04	1531,36	1369,12	1072,41	1412,52	1328,21	1139,96	1011,37	1281,84	1219,12	1149,03	989,70	1271,64	15259,70
	Entr	45	38	28	26	35	29	28	26	33	28	30	27	31,08	373
	Temp	600	603	603	603	600	601	600	601	605	603	607	603	602,42	7229

Tabela 5.7: Melhores resultados obtidos para o conjunto R1.

5.5 Avaliação dos métodos híbridos

Nesta seção é feita uma comparação do desempenho dos métodos híbridos, com a intenção de avaliar os ganhos obtidos pelos diversos esquemas de hibridização desenvolvidos. Lembrando que foram desenvolvidos quatro métodos híbridos, listados a seguir:

- Método *branch-price-and-cut* utilizando a heurística primal (BPC_{PR}).
- Método híbrido exato, que consiste no mesmo método anterior adicionando as metaheurísticas ILS e LNS para obter e melhorar soluções primais para o problema (BPC_{MH}).
- O método anterior utilizando o gerador de colunas híbrido, o qual incorpora a metaheurística ILS no método de geração de colunas junto com o algoritmo de programação dinâmica (BPC_{GCH}). Este método continua sendo exato.
- Método híbrido heurístico, que utiliza só a metaheurística ILS para resolver a geração de colunas (BPC_{GC-ILS}).

Inicialmente, é feita uma avaliação dos métodos híbridos que somente incluem componentes para obter melhores soluções primais, em relação ao BPC usado como método de solução autônomo. Nesse sentido, testes preliminares permitiram identificar a necessidade de incorporar a heurística primal ao BPC, pois sem ela o método apresenta um desempenho muito inferior, não encontrando soluções factíveis para a maioria das instâncias dentro do tempo limite de execução. Portanto, comparou-se o desempenho do método BPC_{MH} em relação ao BPC_{PR} .

As Tabelas 5.8 e 5.9 mostram os resultados obtidos por ambos os métodos dentro de uma hora de execução e uma comparação entre seus resultados. Nas tabelas, as colunas 2 a 7 mostram os resultados do método BPC_{PR} , as colunas 8 a 13 mostram os resultados do método BPC_{MH} e as colunas 14 e 15 representam a proporção entre o custo da solução e o tempo de execução dos dois métodos.

Para as instâncias tipo 1 os resultados mostram que, em termos do custo das soluções, os métodos tiveram desempenhos semelhantes. Destaca-se que para o conjunto C1, o uso de metaheurísticas dentro do BPC acelerou consideravelmente o desempenho do método, já que em média o BPC_{PR} gastou 2,59 vezes o tempo que o BPC_{MH} usou para encontrar as soluções e, mesmo assim, nem sempre o primeiro conseguiu encontrar a melhor solução para as instâncias desse conjunto. Note que nas instâncias dos conjuntos RC1 e C1 que são resolvidas até a otimalidade por ambos métodos, o BPC_{MH} sempre consegue provar mais rapidamente sua otimalidade. Contudo, para as instâncias R101 e R102, o método BPC_{MH} foi menos rápido devido ao tempo gasto na fase inicial deste método (execução das metaheurísticas e a heurística primal) enquanto que o método BPC_{PR} consegue resolvê-las rapidamente. Em relação ao número de nós explorados por cada método, observa-se uma redução significativa

no número médio de nós explorados ao incluir as metaheurísticas. Esse fato é resultado do tempo gasto pelas metaheurísticas dentro do método BPC_{MH} , o que faz com que esse método consiga explorar uma menor quantidade de nós dentro do limite de tempo disponível.

No que se refere aos conjuntos de instâncias tipo 2, podem ser observados ganhos significativos em termos do custo das soluções obtidas, já que o método BPC_{PR} gerou soluções com custo médio 11% maior em relação ao custo das soluções do método BPC_{MH} . Destaca-se que o BPC com heurística primal obteve soluções com custo médio 10% e 22% maior do que o custo médio do método BPC_{MH} nos conjuntos R2 e RC2, respectivamente. Esses resultados mostram as vantagens que se podem ter com o uso de algoritmos específicos para obter boas soluções dentro de um método exato. Para o conjunto C2, o método BPC_{MH} conseguiu provar a otimalidade de uma solução a mais (C201) do que o método BPC_{PR} e na instância em que ambos os métodos provaram a otimalidade (C205), o método BPC_{MH} o fez mais rapidamente. Por outra parte, observa-se que o número médio de nós explorados por ambos métodos foi relativamente baixo em relação às instâncias do tipo 1, já que elas são mais desafiadoras para o BPC. Este resultado reforça a experiência encontrada na literatura para este tipo de instância (MUNARI; MORABITO, 2015; DESAULNIERS; LESSARD; HADJAR, 2008; DANNA; LE PAPE, 2005).

De forma complementar, as Figuras 5.3 e 5.4 mostram graficamente os custos das soluções obtidas pelos métodos para cada um dos conjuntos de instâncias. Nas figuras podem ser observados os ganhos obtidos pelo método BPC_{MH} em termos da qualidade das soluções encontradas, especialmente nos conjuntos de instâncias tipo 2.

Inst	BPC_{PR}						BPC_{MH}						BPC_{PR}/BPC_{MH}	
	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Tempo
R101	23,67*	19	1720,11	45	3	29	23,67*	19	1720,11	45	3	91	1,00	0,32
R102	20,95*	17	1530,92	38	1	11	20,95*	17	1530,92	38	1	32	1,00	0,35
R103	15,84	13	1369,92	27	21	3630	15,84	13	1368,73	27	29	3600	1,00	1,01
R104	12,61	10	1079,78	25	16	3625	12,61	10	1089,81	25	7	3601	1,00	1,01
R105	17,75	14	1534,04	36	78	3623	17,64	14	1412,52	35	58	3653	1,01	0,99
R106	14,24	11	1396,38	31	56	3617	14,33	11	1318,81	32	36	3650	0,99	0,99
R107	12,82	10	1175,59	27	29	3633	12,91	10	1112,73	28	18	3651	0,99	1,00
R108	12,81	10	1053,92	27	23	3630	11,70	9	967,00	26	8	3601	1,09	1,01
R109	14,34	11	1372,73	32	44	3628	14,23	11	1262,50	31	27	3651	1,01	0,99
R110	15,13	12	1342,08	30	42	3605	12,92	10	1180,93	28	16	3604	1,17	1,00
R111	12,81	10	1143,40	27	19	3601	12,81	10	1143,40	27	10	3641	1,00	0,99
R112	13,81	11	1091,79	27	25	3635	12,70	10	1002,31	26	15	3637	1,09	1,00
RC101	20,19*	16	1862,21	40	97	3036	20,19*	16	1862,21	40	43	2164	1,00	1,40
RC102	17,08*	13	1777,29	39	21	848	17,08*	13	1777,29	39	11	734	1,00	1,15
RC103	14,44	11	1395,11	33	34	3611	14,44	11	1387,18	33	21	3613	1,00	1,00
RC104	14,02	11	1229,65	29	23	3630	14,02	11	1247,62	29	8	3656	1,00	0,99
RC105	17,87	14	1727,73	37	67	3601	17,87	14	1718,14	37	44	3602	1,00	1,00
RC106	16,55	13	1547,93	34	57	3614	16,65	13	1529,78	35	32	3652	0,99	0,99
RC107	14,33	11	1338,18	32	25	3601	14,33	11	1338,18	32	7	3601	1,00	1,00
RC108	14,03	11	1288,03	29	14	3600	14,03	11	1286,04	29	5	3651	1,00	0,99
C101	11,08*	10	828,94	10	5	240	11,08*	10	828,94	10	3	144	1,00	1,67
C102	11,08	10	838,98	10	26	3630	11,08*	10	828,94	10	3	857	1,00	4,24
C103	11,08*	10	828,06	10	5	1832	11,08	10	828,06	10	3	3602	1,00	0,51
C104	11,08	10	832,75	10	11	3631	11,08	10	819,81	10	3	3602	1,00	1,01
C105	11,08*	10	828,94	10	17	881	11,08*	10	828,94	10	3	300	1,00	2,94
C106	11,08*	10	828,94	10	3	313	11,08*	10	828,94	10	3	197	1,00	1,59
C107	11,08*	10	828,94	10	7	718	11,08*	10	828,94	10	3	192	1,00	3,75
C108	11,08*	10	828,94	10	9	1566	11,08*	10	828,94	10	3	319	1,00	4,91
C109	11,08	10	827,38	10	8	3600	11,08*	10	827,26	10	3	1344	1,00	2,68
Média	14,31	11,66	25,34	25,34	27,10	2697,22	14,16	11,52	1196,72	25,24	14,69	2470	1,01	1,46

Tabela 5.8: Método BPC_{PR} vs BPC_{MH} nas instâncias do tipo 1.

Inst	BPC_{PR}						BPC_{MH}						BPC_{PR}/BPC_{MH}	
	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Tempo
R201	5,36	4	1634,43	12	2	3631	5,03	4	1326,01	9	1	3600	1,07	1,01
R202	4,95	4	1498,07	8	2	3625	4,82	4	1243,20	7	3	3651	1,03	0,99
R203	4,04	3	1394,24	9	2	3623	3,70	3	988,59	6	3	3603	1,09	1,01
R204	4,01	3	1095,99	9	2	3601	3,58	3	820,35	5	0	3650	1,12	0,99
R205	4,94	4	1368,27	8	0	3630	3,81	3	1093,14	7	0	3654	1,30	0,99
R206	4,02	3	1218,48	9	0	3637	3,70	3	976,91	6	2	3651	1,09	1,00
R207	3,71	3	1100,47	6	3	3614	3,59	3	906,91	5	2	3655	1,03	0,99
R208	2,69	2	914,03	6	1	3600	2,67	2	738,89	6	2	3650	1,01	0,99
R209	4,93	4	1339,97	8	0	3642	3,79	3	948,79	7	1	3604	1,30	1,01
R210	4,03	3	1319,02	9	2	3632	3,70	3	996,61	6	2	3657	1,09	0,99
R211	3,70	3	988,09	6	2	3614	3,69	3	863,82	6	2	3656	1,00	0,99
RC201	6,68	5	1831,81	15	2	3631	5,15	4	1517,29	10	2	3652	1,30	0,99
RC202	5,37	4	1693,07	12	3	3608	5,02	4	1238,15	9	5	3653	1,07	0,99
RC203	4,96	4	1571,07	8	4	3630	3,92	3	1151,27	8	2	3652	1,27	0,99
RC204	4,93	4	1267,75	8	4	3618	3,69	3	939,29	6	4	3603	1,33	1,00
RC205	6,21	5	2094,37	10	2	3631	5,05	4	1453,53	9	2	3650	1,23	0,99
RC206	5,35	4	1529,37	12	1	3615	4,92	4	1199,99	8	2	3650	1,09	0,99
RC207	5,36	4	1624,68	12	2	3631	3,92	3	1223,37	8	2	3653	1,37	0,99
RC208	4,02	3	1191,66	9	4	3630	3,80	3	977,53	7	3	3602	1,06	1,01
C201	3,39	3	943,32	3	3	3630	3,36*	3	588,88	3	3	2027	1,01	1,79
C202	3,36	3	588,88	3	3	3600	3,36	3	588,88	3	1	3603	1,00	1,00
C203	3,36	3	585,27	3	3	3600	3,36	3	585,27	3	3	3601	1,00	1,00
C204	3,39	3	907,35	3	2	3630	3,36	3	584,49	3	3	3602	1,01	1,01
C205	3,36*	3	588,49	3	3	2501	3,36*	3	588,49	3	3	861	1,00	2,91
C206	3,38	3	757,50	3	3	3630	3,36	3	588,49	3	1	3603	1,01	1,01
C207	3,36	3	587,31	3	3	3612	3,36	3	587,31	3	3	3599	1,00	1,00
C208	3,37	3	709,95	3	2	3631	3,36	3	588,29	3	1	3601	1,00	1,01
Média	4,30	3,44	1197,89	7,41	2,22	3580,62	3,87	3,19	937,18	5,89	2,15	3468,28	1,11	1,10

Tabela 5.9: Método BPC_{PR} vs BPC_{MH} nas instâncias do tipo 2.

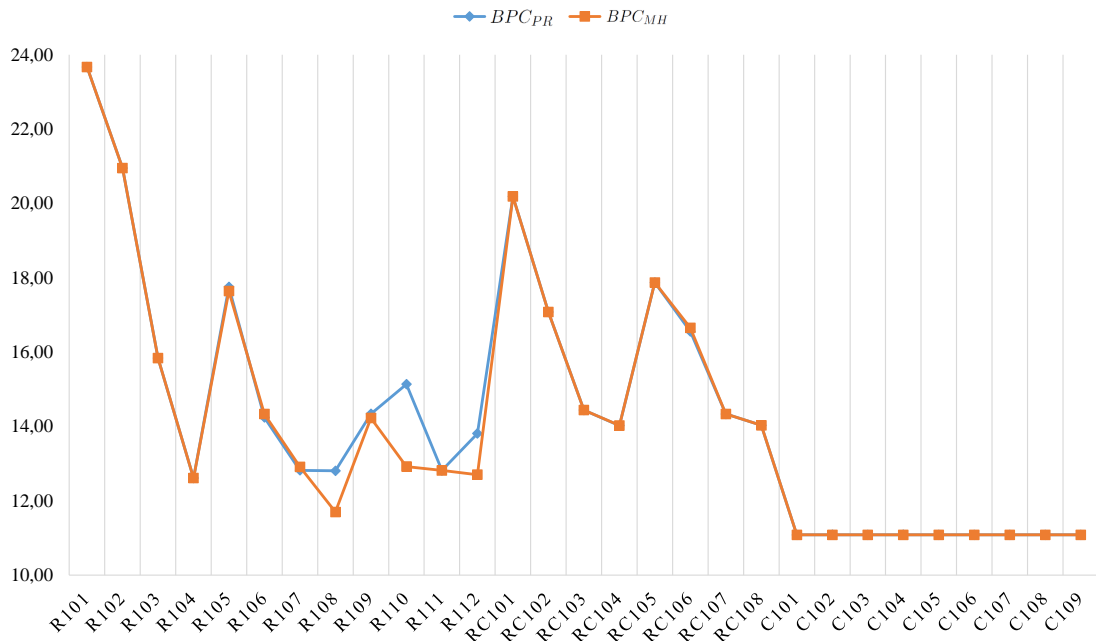


Figura 5.3: *Branch-price-and-cut* com heurística primal vs método híbrido exato nas instâncias tipo 1.

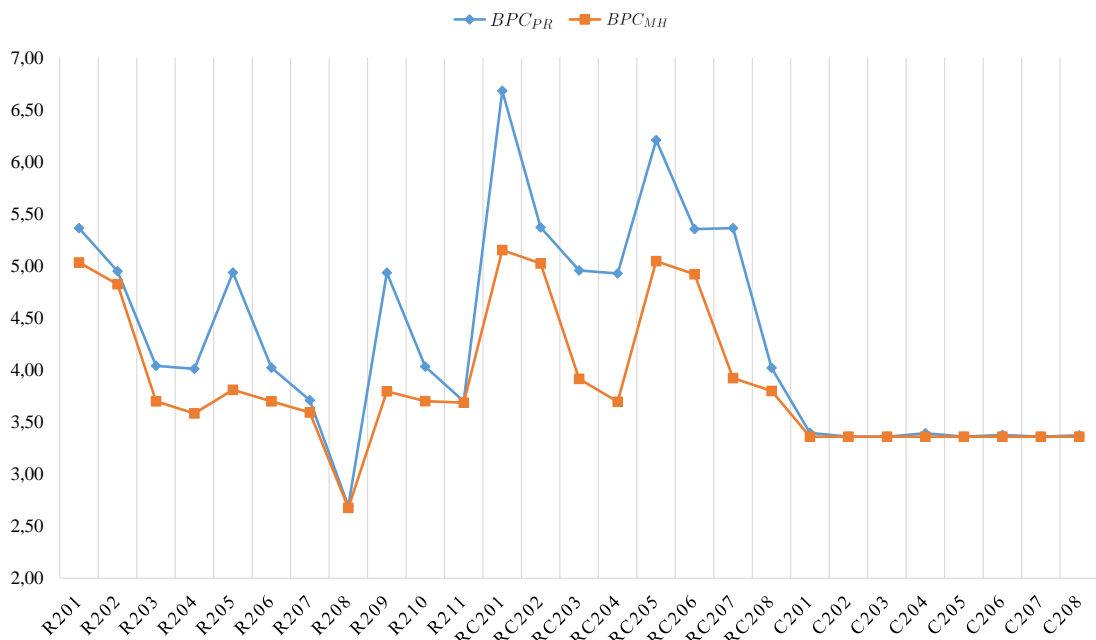


Figura 5.4: *Branch-price-and-cut* com heurística primal vs método híbrido exato nas instâncias tipo 2.

Posteriormente, com a intenção de avaliar a incorporação da metaheurística ILS no método de geração de colunas, comparou-se o desempenho do método BPC_{GCH} em relação ao BPC_{MH} . As Tabelas 5.10 e 5.11 mostram os resultados obtidos por ambos os métodos dentro de uma hora de execução e uma comparação entre seus resultados. As colunas 2 a 7 mostram os resultados do método BPC_{MH} , as colunas 8 a 13 mostram os resultados do método BPC_{GCH} e as colunas 14 e 15 representam a

proporção entre o custo da solução e o tempo de execução dos dois métodos.

Para os conjuntos de instâncias tipo 1, os resultados mostram que ambos os métodos tiveram desempenhos semelhantes em termos do custo das soluções encontradas. Observa-se que o número de nós explorados é menor no método BPC_{GCH} em relação ao método BPC_{MH} . Esse resultado pode ser explicado por dois fatores principais: primeiro, as características das instâncias tipo 1 favorecem para que o desempenho do algoritmo de programação dinâmica seja adequado e, conseqüentemente, o desempenho do método híbrido exato (BPC_{MH}) também o seja. Em segundo lugar, tem-se a natureza heurística do gerador de colunas híbrido, já que a metaheurística ILS pode ser capaz de encontrar rapidamente colunas com custo reduzido negativo, mas não aquelas com o menor custo reduzido, o que faz com que a convergência do método seja mais lenta. Esse fenômeno resulta em um aumento no tempo para provar otimalidade para a maioria das instâncias cuja solução ótima é encontrada por ambos os métodos, e.g. R101, R102, C101, C105, C106, C107. Por outro lado, também observa-se casos nos quais o método que utiliza só programação dinâmica consegue provar a otimalidade e o método com gerador de colunas híbrido não o fez (RC101, RC102). Contudo, também existem instâncias em que ou o método BPC_{GCH} consegue provar mais rapidamente a otimalidade (C102) ou que só ele consegue provar a otimalidade (C103).

A respeito das instâncias tipo 2, os desempenhos dos métodos também foram semelhantes em termos da qualidade das soluções obtidas. No entanto, observa-se um incremento no número de nós que são explorados pelo método BPC_{GCH} em relação ao método BPC_{MH} . Especificamente, o número de nós explorados aumentou 83%, mostrando que de fato a metaheurística ILS desenvolvida é capaz de gerar rapidamente colunas com custo reduzido negativo enquanto que o algoritmo de programação dinâmica não é capaz de ter um desempenho adequado para esse tipo de instâncias devido ao grande número de rótulos que podem ser gerados, o que torna estas instâncias muito mais difíceis do ponto de vista computacional para o algoritmo de programação dinâmica e, por conseguinte, para o método BPC_{MH} .

Nas Figuras 5.5 e 5.6 são mostrados graficamente as soluções obtidas pelos métodos para cada um dos conjuntos de instâncias, mostrando que ambos métodos tiveram desempenhos semelhantes em termos da qualidade das soluções obtidas.

Inst	BPC_{MH}						BPC_{GCH}						BPC_{MH}/BPC_{GCH}	
	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Tempo
R101	23,67*	19	1720,11	45	3	91	23,67*	19	1720,11	45	3	97	1,00	0,94
R102	20,95*	17	1530,92	38	1	32	20,95*	17	1530,92	38	1	124	1,00	0,26
R103	15,84	13	1368,73	27	29	3600	15,84	13	1396,29	27	9	3602	1,00	1,06
R104	12,61	10	1089,81	25	7	3601	12,71	10	1064,60	26	5	3610	0,99	1,00
R105	17,64	14	1412,52	35	58	3653	17,64	14	1412,52	35	10	3603	1,00	1,01
R106	14,33	11	1318,81	32	36	3650	14,34	11	1412,16	32	9	3633	1,00	1,00
R107	12,91	10	1112,73	28	18	3651	12,81	10	1136,58	27	4	3610	1,01	1,01
R108	11,70	9	967,00	26	8	3601	11,70	9	963,03	26	8	3603	1,00	1,00
R109	14,23	11	1262,50	31	27	3651	15,13	12	1255,96	30	9	3621	0,94	1,01
R110	12,92	10	1180,93	28	16	3604	13,02	10	1186,25	29	6	3600	0,99	1,00
R111	12,81	10	1143,40	27	10	3641	12,91	10	1138,14	28	4	3665	0,99	0,99
R112	12,70	10	1002,31	26	15	3637	11,70	9	981,69	26	4	3604	1,09	1,01
RC101	20,19*	16	1862,21	40	43	2164	20,19	16	1862,21	40	9	3601	1,00	0,60
RC102	17,08*	13	1777,29	39	11	734	17,96	14	1634,69	38	6	3604	0,95	0,20
RC103	14,44	11	1387,18	33	21	3613	14,44	11	1392,99	33	4	3647	1,00	0,99
RC104	14,02	11	1247,62	29	8	3656	14,02	11	1246,30	29	4	3666	1,00	1,00
RC105	17,87	14	1718,14	37	44	3602	17,87	14	1718,14	37	6	3602	1,00	1,00
RC106	16,65	13	1529,78	35	32	3652	16,75	13	1505,91	36	4	3624	0,99	1,01
RC107	14,33	11	1338,18	32	7	3601	14,33	11	1338,18	32	4	3603	1,00	1,00
RC108	14,03	11	1286,04	29	5	3651	14,12	11	1235,29	30	6	3606	0,99	1,01
C101	11,08*	10	828,94	10	3	144	11,08*	10	828,94	10	3	163	1,00	0,88
C102	11,08*	10	828,94	10	3	857	11,08*	10	828,94	10	3	408	1,00	2,10
C103	11,08	10	828,06	10	3	3602	11,08*	10	828,06	10	3	501	1,00	7,18
C104	11,08	10	819,81	10	3	3602	11,08	10	819,81	10	3	3600	1,00	1,00
C105	11,08*	10	828,94	10	3	300	11,08*	10	828,94	10	3	367	1,00	0,82
C106	11,08*	10	828,94	10	3	197	11,08*	10	828,94	10	3	500	1,00	0,39
C107	11,08*	10	828,94	10	3	192	11,08*	10	828,94	10	7	2401	1,00	0,08
C108	11,08*	10	828,94	10	3	319	11,08	10	828,94	10	17	3604	1,00	0,09
C109	11,08*	10	827,26	10	3	1344	11,08	10	827,26	10	11	3602	1,00	0,37
Média	14,16	11,52	1196,72	25,24	14,69	2470	14,20	11,55	1192,31	25,31	5,79	2768	1,00	1,07

Tabela 5.10: Método BPC_{MH} vs BPC_{GCH} nas instâncias do tipo 1.

Inst	BPC_{MH}						BPC_{GCH}						BPC_{MH}/BPC_{GCH}	
	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Tempo
R201	5,03	4	1326,01	9	1	3600	5,03	4	1326,01	9	2	3629	1,00	0,99
R202	4,82	4	1243,20	7	3	3651	4,91	4	1147,64	8	4	3602	0,98	1,01
R203	3,70	3	988,59	6	3	3603	3,70	3	1004,02	6	3	3603	1,00	1,00
R204	3,58	3	820,35	5	0	3650	3,58	3	834,13	5	4	3621	1,00	1,01
R205	3,81	3	1093,14	7	0	3654	3,81	3	1063,06	7	2	3625	1,00	1,01
R206	3,70	3	976,91	6	2	3651	3,70	3	1000,68	6	4	3617	1,00	1,01
R207	3,59	3	906,91	5	2	3655	3,59	3	918,04	5	4	3602	1,00	1,01
R208	2,67	2	738,89	6	2	3650	2,58	2	775,09	5	6	3620	1,04	1,01
R209	3,79	3	948,79	7	1	3604	3,70	3	1047,54	6	2	3620	1,02	1,00
R210	3,70	3	996,61	6	2	3657	3,70	3	1026,55	6	2	3611	1,00	1,01
R211	3,69	3	863,82	6	2	3656	3,69	3	910,21	6	2	3603	1,00	1,01
RC201	5,15	4	1517,29	10	2	3652	5,15	4	1510,08	10	2	3628	1,00	1,01
RC202	5,02	4	1238,15	9	5	3653	4,93	4	1329,23	8	4	3600	1,02	1,01
RC203	3,92	3	1151,27	8	2	3652	3,92	3	1204,06	8	5	3608	1,00	1,01
RC204	3,69	3	939,29	6	4	3603	3,69	3	925,70	6	5	3601	1,00	1,00
RC205	5,05	4	1453,53	9	2	3650	5,05	4	1455,73	9	3	3601	1,00	1,01
RC206	4,92	4	1199,99	8	2	3650	4,92	4	1185,24	8	2	3634	1,00	1,00
RC207	3,92	3	1223,37	8	2	3653	4,02	3	1178,35	9	2	3621	0,98	1,01
RC208	3,80	3	977,53	7	3	3602	3,80	3	968,12	7	5	3632	1,00	0,99
C201	3,36*	3	588,88	3	3	2027	3,36*	3	588,88	3	3	2740	1,00	0,74
C202	3,36	3	588,88	3	1	3603	3,36	3	588,88	3	2	3611	1,00	1,00
C203	3,36	3	585,27	3	3	3601	3,36	3	585,27	3	18	3603	1,00	1,00
C204	3,36	3	584,49	3	3	3602	3,36	3	584,49	3	7	3601	1,00	1,00
C205	3,36*	3	588,49	3	3	861	3,36	3	588,49	3	4	3605	1,00	0,24
C206	3,36	3	588,49	3	1	3603	3,36	3	588,49	3	4	3608	1,00	1,00
C207	3,36	3	587,31	3	3	3599	3,36	3	587,31	3	1	3628	1,00	0,99
C208	3,36	3	588,29	3	1	3601	3,36	3	588,29	3	4	3630	1,00	0,99
Média	3,87	3,19	937,18	5,89	2,15	3468,28	3,86	3,19	944,80	5,85	3,93	3581,57	1,00	0,97

Tabela 5.11: Método BPC_{MH} vs BPC_{GCH} nas instâncias do tipo 2.

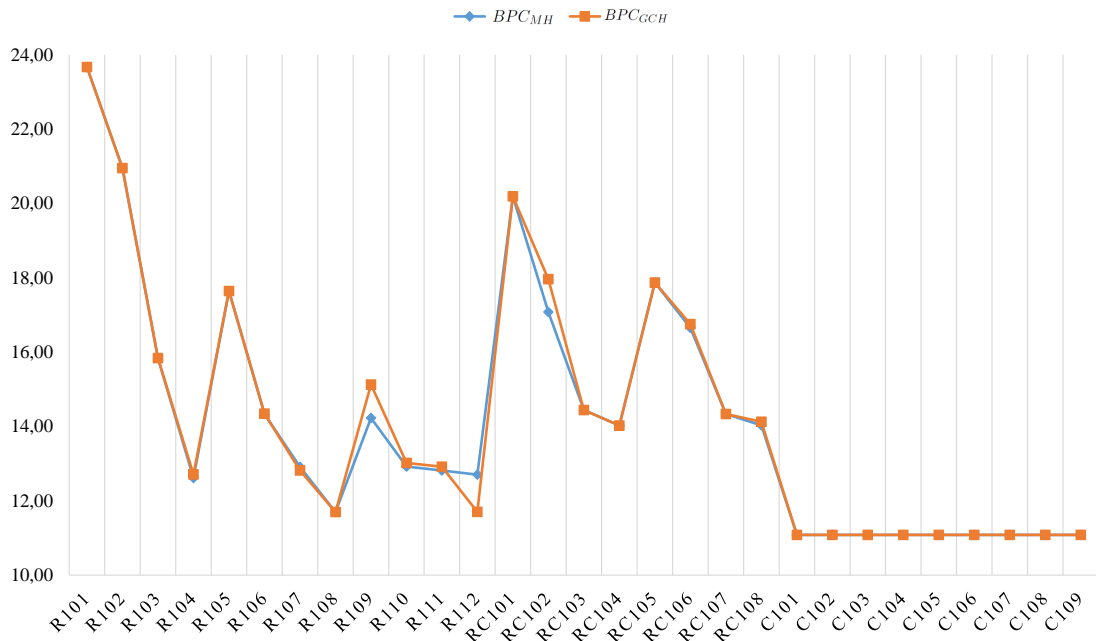


Figura 5.5: Método híbrido exato vs método híbrido com gerador de colunas híbrido nas instâncias tipo 1.

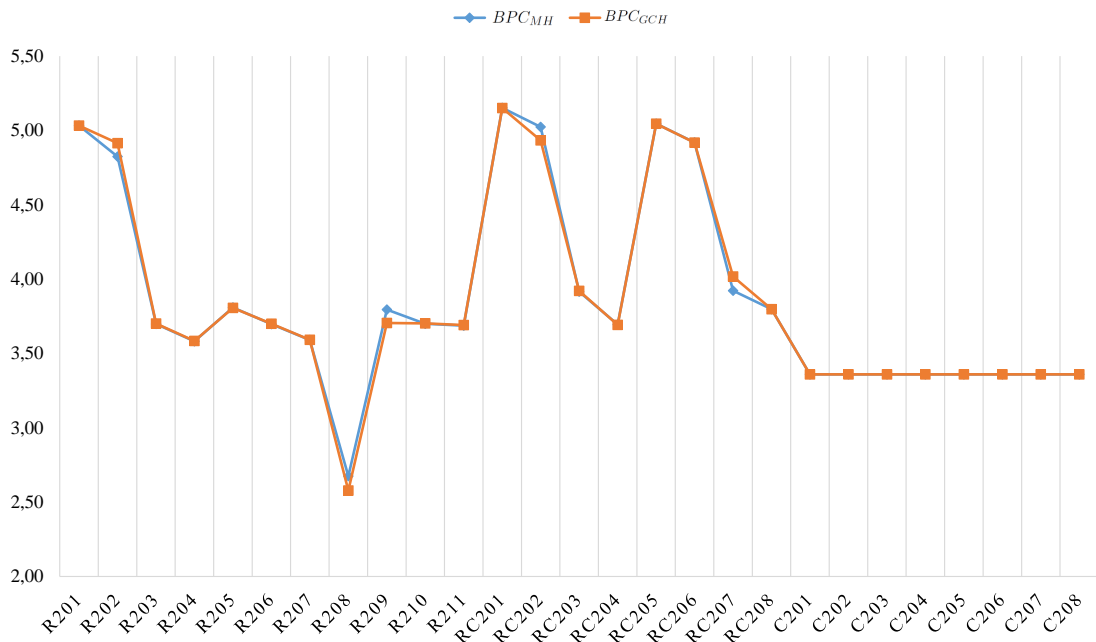


Figura 5.6: Método híbrido exato vs método híbrido com gerador de colunas híbrido nas instâncias tipo 2.

Finalmente, avaliamos e comparamos o desempenho do método híbrido heurístico que utiliza somente a metaheurística ILS para resolver o método de geração de colunas (BPC_{GC-ILS}). Seus resultados são comparados com os dos métodos BPC_{MH} e BPC_{GCH} e mostrados nas Tabelas 5.12 e 5.13. Por ser uma simplificação do método que utiliza o gerador de colunas híbrido, é de se esperar que seu desempenho seja melhor nas instâncias tipo 2 do que nas tipo 1.

Inst	BPC_{MH}						BPC_{GCH}						BPC_{GC-ILS}					
	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo
R101	23,67*	19	1720,11	45	3	91	23,67*	19	1720,11	45	3	97	23,67	19	1720,11	45	3	95
R102	20,95*	17	1530,92	38	1	32	20,95*	17	1530,92	38	1	124	20,95	17	1542,54	38	1	38
R103	15,84	13	1368,73	27	29	3600	15,84	13	1396,29	27	9	3602	15,94	13	1360,94	28	3	2527
R104	12,61	10	1089,81	25	7	3601	12,71	10	1064,60	26	5	3610	12,71	10	1086,73	26	3	714
R105	17,64	14	1412,52	35	58	3653	17,64	14	1412,52	35	10	3603	17,64	14	1412,52	35	5	2436
R106	14,33	11	1318,81	32	36	3650	14,34	11	1412,16	32	9	3633	15,13	12	1306,01	30	5	3515
R107	12,91	10	1112,73	28	18	3651	12,81	10	1136,58	27	4	3610	12,82	10	1152,76	27	3	518
R108	11,70	9	967,00	26	8	3601	11,70	9	963,03	26	8	3603	11,80	9	962,55	27	5	3602
R109	14,23	11	1262,50	31	27	3651	15,13	12	1255,96	30	9	3621	15,13	12	1254,88	30	7	3651
R110	12,92	10	1180,93	28	16	3604	13,02	10	1186,25	29	6	3600	14,02	11	1161,96	29	5	3604
R111	12,81	10	1143,40	27	10	3641	12,91	10	1138,14	28	4	3665	13,72	11	1153,04	26	8	3604
R112	12,70	10	1002,31	26	15	3637	11,70	9	981,69	26	4	3604	12,70	10	1003,50	26	5	3601
RC101	20,19*	16	1862,21	40	43	2164	20,19	16	1862,21	40	9	3601	20,28	16	1833,03	41	14	3640
RC102	17,08*	13	1777,29	39	11	734	17,96	14	1634,69	38	6	3604	17,87	14	1655,01	37	5	3603
RC103	14,44	11	1387,18	33	21	3613	14,44	11	1392,99	33	4	3647	15,34	12	1360,96	32	4	3604
RC104	14,02	11	1247,62	29	8	3656	14,02	11	1246,30	29	4	3666	14,12	11	1229,56	30	6	3607
RC105	17,87	14	1718,14	37	44	3602	17,87	14	1718,14	37	6	3602	18,17	14	1669,30	40	7	3602
RC106	16,65	13	1529,78	35	32	3652	16,75	13	1505,91	36	4	3624	16,75	13	1487,35	36	9	3412
RC107	14,33	11	1338,18	32	7	3601	14,33	11	1338,18	32	4	3603	15,23	12	1335,47	31	3	449
RC108	14,03	11	1286,04	29	5	3651	14,12	11	1235,29	30	6	3606	14,22	11	1214,62	31	10	3603
C101	11,08*	10	828,94	10	3	144	11,08*	10	828,94	10	3	163	11,08	10	828,94	10	3	122
C102	11,08*	10	828,94	10	3	857	11,08*	10	828,94	10	3	408	11,08	10	828,94	10	3	235
C103	11,08	10	828,06	10	3	3602	11,08*	10	828,06	10	3	501	11,08	10	828,06	10	3	209
C104	11,08	10	819,81	10	3	3602	11,08	10	819,81	10	3	3600	11,08	10	819,81	10	3	362
C105	11,08*	10	828,94	10	3	300	11,08*	10	828,94	10	3	367	11,08	10	828,94	10	3	160
C106	11,08*	10	828,94	10	3	197	11,08*	10	828,94	10	3	500	11,08	10	828,94	10	3	204
C107	11,08*	10	828,94	10	3	192	11,08*	10	828,94	10	7	2401	11,08	10	828,94	10	13	1467
C108	11,08*	10	828,94	10	3	319	11,08	10	828,94	10	17	3604	11,08	10	828,94	10	31	3602
C109	11,08*	10	827,26	10	3	1344	11,08	10	827,26	10	11	3602	11,08	10	827,26	10	27	3618
Média	14,16	11,52	1196,72	25,24	14,69	2470	14,20	11,55	1192,31	25,31	5,79	2768	14,41	11,76	1184,54	25,34	6,90	2186

Tabela 5.12: Soluções encontradas por todos os métodos híbridos nas instâncias do tipo 1.

Inst	<i>BPC_{MH}</i>						<i>BPC_{GCH}</i>						<i>BPC_{GC-ILS}</i>					
	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo	Custo	Veic	Dist	Entr	Nós	Tempo
R201	5,03	4	1326,01	9	1	3600	5,03	4	1326,01	9	2	3629	4,93	4	1341,94	8	4	3628
R202	4,82	4	1243,20	7	3	3651	4,91	4	1147,64	8	4	3602	4,82	4	1214,73	7	4	3607
R203	3,70	3	988,59	6	3	3603	3,70	3	1004,02	6	3	3603	3,70	3	1034,31	6	4	3602
R204	3,58	3	820,35	5	0	3650	3,58	3	834,13	5	4	3621	3,58	3	803,17	5	5	3603
R205	3,81	3	1093,14	7	0	3654	3,81	3	1063,06	7	2	3625	3,81	3	1070,32	7	2	3660
R206	3,70	3	976,91	6	2	3651	3,70	3	1000,68	6	4	3617	3,70	3	1031,18	6	4	3616
R207	3,59	3	906,91	5	2	3655	3,59	3	918,04	5	4	3602	3,59	3	920,70	5	5	3601
R208	2,67	2	738,89	6	2	3650	2,58	2	775,09	5	6	3620	2,58	2	755,22	5	4	3606
R209	3,79	3	948,79	7	1	3604	3,70	3	1047,54	6	2	3620	3,80	3	1008,25	7	4	3605
R210	3,70	3	996,61	6	2	3657	3,70	3	1026,55	6	2	3611	3,70	3	1031,66	6	4	3602
R211	3,69	3	863,82	6	2	3656	3,69	3	910,21	6	2	3603	3,69	3	887,60	6	3	3601
RC201	5,15	4	1517,29	10	2	3652	5,15	4	1510,08	10	2	3628	5,25	4	1488,94	11	4	3635
RC202	5,02	4	1238,15	9	5	3653	4,93	4	1329,23	8	4	3600	5,03	4	1259,72	9	5	3602
RC203	3,92	3	1151,27	8	2	3652	3,92	3	1204,06	8	5	3608	3,92	3	1177,08	8	5	3602
RC204	3,69	3	939,29	6	4	3603	3,69	3	925,70	6	5	3601	3,70	3	985,50	6	5	3602
RC205	5,05	4	1453,53	9	2	3650	5,05	4	1455,73	9	3	3601	5,14	4	1449,75	10	4	3631
RC206	4,92	4	1199,99	8	2	3650	4,92	4	1185,24	8	2	3634	4,92	4	1168,18	8	4	3630
RC207	3,92	3	1223,37	8	2	3653	4,02	3	1178,35	9	2	3621	4,91	4	1121,61	8	4	3604
RC208	3,80	3	977,53	7	3	3602	3,80	3	968,12	7	5	3632	3,80	3	1014,54	7	5	3602
C201	3,36*	3	588,88	3	3	2027	3,36*	3	588,88	3	3	2740	3,36	3	588,88	3	3	1798
C202	3,36	3	588,88	3	1	3603	3,36	3	588,88	3	2	3611	3,36	3	588,88	3	11	3604
C203	3,36	3	585,27	3	3	3601	3,36	3	585,27	3	18	3603	3,36	3	585,27	3	14	3603
C204	3,36	3	584,49	3	3	3602	3,36	3	584,49	3	7	3601	3,36	3	584,49	3	22	3625
C205	3,36*	3	588,49	3	3	861	3,36	3	588,49	3	4	3605	3,36	3	588,49	3	5	3604
C206	3,36	3	588,49	3	1	3603	3,36	3	588,49	3	4	3608	3,36	3	588,49	3	4	3615
C207	3,36	3	587,31	3	3	3599	3,36	3	587,31	3	1	3628	3,36	3	587,31	3	7	3611
C208	3,36	3	588,29	3	1	3601	3,36	3	588,29	3	4	3630	3,36	3	588,29	3	6	3610
Média	3,87	3,19	937,18	5,89	2,15	3468,28	3,86	3,19	944,80	5,85	3,93	3581,57	3,91	3,22	943,13	5,89	5,59	3544,75

Tabela 5.13: Soluções encontradas por todos os métodos híbridos nas instâncias do tipo 2.

Os resultados mostram que o método BPC_{GC-ILS} não é capaz de equiparar o desempenho dos métodos híbridos que conservam o componente exato, especialmente para as instâncias do tipo 1 já que, como foi mencionado anteriormente, a estrutura dessas instâncias é propícia para que os métodos que incluem o algoritmo de programação dinâmica possam ter um melhor desempenho. Por outro lado, para as instâncias do tipo 2, o método BPC_{GC-ILS} é capaz de explorar heurísticamente uma maior quantidade de nós e de encontrar a melhor solução em quatro instâncias do conjunto R2 e em uma do conjunto RC2. No entanto, devido à perda da garantia de otimalidade deste método, não é possível inferir muito sobre a qualidade das soluções obtidas por ele. Em termos percentuais, o método BPC_{GC-ILS} obteve soluções com custo médio 1,77% e 1,48% maior do que as soluções para as instâncias do tipo 1 dos métodos BPC_{MH} e BPC_{GCH} , respectivamente. Para as do tipo 2, obteve soluções com custo médio 1,03% e 1,30% maior do que as dos métodos BPC_{MH} e BPC_{GCH} , respectivamente.

5.6 Fonte das soluções nos métodos híbridos

Nesta seção, estuda-se a fonte das soluções finais obtidas pelos métodos híbridos. Na Tabela 5.14 se reporta o número de vezes que cada componente do método híbrido gerou a melhor solução encontrada para execuções com tempo limite de 600 e 3600 segundos, respectivamente. Note-se que os métodos híbridos tem 5 fontes potenciais de soluções inteiras (coluna 1), a saber: (i) Solução inicial fornecida pelas metaheurísticas antes do início da execução do BPC (MH_{ini}); (ii) Solução dada pela combinação das colunas iniciais obtida pela heurística primal antes de iniciar o BPC (SP); (iii) Solução inteira no problema mestre restrito em um nó da árvore do BPC (PMR); (iv) Solução da heurística primal em um nó do BPC (IP); (v) Solução vinda do BPC e melhorada através das metaheurísticas (MH).

Fonte	BPC_{MH}		BPC_{GCH}		BPC_{GC-ILS}	
	600 s	3600 s	600 s	3600 s	600 s	3600 s
MH_{ini}	22	18	21	18	19	22
SP	0	0	0	1	1	0
PMR	0	0	0	2	0	0
IP	1	2	0	0	1	0
MH	33	36	35	35	35	34

Tabela 5.14: Fonte das soluções finais nos métodos híbridos.

Como era de se esperar, pode-se observar que, tanto para as execuções de 600 segundos quanto para as de 3600 segundos, a grande maioria das soluções provém do componente metaheurístico dos métodos, *i.e.*, as soluções inicialmente fornecidas pelas metaheurísticas e as soluções melhoradas por elas. A primeira fonte (MH_{ini}) evidencia a notável capacidade das metaheurísticas de gerar de forma rápida boas soluções para o problema, já que a execução inicial das metaheurísticas é realizada tipicamente em poucos segundos, enquanto que a segunda (MH) demonstra a efetividade do esquema de

cooperação dos métodos híbridos para gerar boas soluções.

Ainda, note que a baixa taxa de sucesso da heurística primal (SP, IP) não significa que ela não seja bem-sucedida para gerar boas soluções para o problema, mas mostra que as metaheurísticas são capazes de melhorar as soluções por ela fornecidas. Por outro lado, observe que PMR não se mostrou muito eficaz como fonte de soluções inteiras já que para ele resultar em uma solução inteira, o problema mestre restrito deveria ter suficientes desigualdades válidas ou deveria estar-se resolvendo um nó suficientemente profundo da árvore.

5.7 Comparação entre os diferentes tipos de métodos desenvolvidos

Nesta seção é feita uma comparação entre os resultados encontrados pelos métodos híbridos BPC_{MH} e BPC_{GCH} e os das abordagens metaheurísticas usadas como métodos de solução independentes. Esta comparação é feita apenas com o intuito de verificar o desempenho dos métodos híbridos em relação às abordagens metaheurísticas, pois estes métodos não são completamente comparáveis. De fato, o objetivo destes métodos híbridos é aprimorar o desempenho de um método exato para obter soluções ótimas para o problema, enquanto que as metaheurísticas são desenvolvidas para gerar boas soluções de forma rápida. Deste modo, as conclusões derivadas destes experimentos devem ser tomadas cautelosamente.

Foram feitos testes usando 600 segundos como tempo limite de execução para os métodos híbridos e seus resultados são comparados com os resultados das Tabelas 5.2 e 5.3, correspondentes aos melhores resultados das cinco execuções das abordagens metaheurísticas. O esquema de execução dos métodos híbridos foi mantido igual às execuções realizadas na Seção 5.5. Cabendo lembrar que, dentro dos híbridos, as metaheurísticas são executadas por um intervalo de tempo relativamente pequeno em relação à execução das metaheurísticas como métodos autônomos.

A Tabela 5.15 mostra os resultados agrupados para cada conjunto de instâncias, mostrando que os métodos híbridos são capazes de superar as metaheurísticas nas instâncias tipo 1. Especificamente, para o conjunto R1 o método BPC_{MH} é capaz de superar as metaheurísticas, encontrando soluções com custo médio 0,26% e 0,39% inferior ao custo da ILS e LNS, respectivamente. No conjunto RC1 os dois métodos híbridos superaram amplamente o desempenho delas, onde BPC_{MH} obteve soluções com custo médio 1,88% e 2,41% menor ao custo da ILS e LNS respectivamente, enquanto a BPC_{GCH} encontrou soluções com custo médio 0,36% e 0,90% menor ao custo da ILS e LNS respectivamente. Já para as instâncias do tipo 2, o desempenho das metaheurísticas foi superior ao dos híbridos, onde a ILS encontrou soluções com custo médio 4,98% menor ao custo dos métodos híbridos para o conjunto R2, enquanto que a LNS encontrou soluções com aproximadamente 6,43% menor custo médio que os métodos híbridos. Os resultados nas instâncias tipo 1 mostram a habilidade dos métodos híbridos para

resolver este tipo de instância, tendo em vista que as metaheurísticas foram desenvolvidas para obter boas soluções rapidamente e os métodos híbridos são capazes de superar seu desempenho. Em contrapartida, os resultados nas instâncias tipo 2 confirmam a dificuldade para os métodos tipo BPC em resolver essa classe de instância já que, como foi mencionado anteriormente, as abordagens metaheurísticas superam amplamente o desempenho dos híbridos.

Finalmente, nos conjuntos C1 e C2 todos os métodos encontraram as mesmas soluções para todas as instâncias.

Método		Conjunto de instâncias					
		R1	RC1	C1	R2	RC2	C2
ILS (melhor de 5)	Custo	15,34	16,59	11,08	3,63	4,31	3,36
	Veic	12,17	13,00	10,00	2,91	3,38	3,00
	Dist	1271,71	1482,46	827,64	993,17	1186,61	587,51
	Entr	30,50	34,38	10,00	6,18	8,13	3,00
	Tempo	600,75	601,25	600,22	604,09	603,25	601,50
LNS (melhor de 5)	Custo	15,32	16,50	11,08	3,64	4,29	3,36
	Veic	12,08	12,88	10,00	2,91	3,38	3,00
	Dist	1271,64	1492,29	827,64	998,00	1197,49	587,51
	Entr	31,08	34,75	10,00	6,27	8,00	3,00
	Tempo	602,42	603,25	600,67	610,09	609,50	602,75
BPC_{MH}	Custo	15,28	16,19	11,08	3,82	4,58	3,36
	Veic	12,08	12,63	10,00	3,09	3,63	3,00
	Dist	1255,87	1508,28	827,64	996,86	1202,44	587,51
	Entr	30,75	34,13	10,00	6,27	8,38	3,00
	Tempo	530,57	621,36	335,99	645,27	644,99	650,24
BPC_{GCH}	Custo	15,46	16,44	11,08	3,82	4,59	3,36
	Veic	12,25	12,88	10,00	3,09	3,63	3,00
	Dist	1262,74	1488,03	827,64	999,48	1232,12	587,51
	Entr	30,83	34,13	10,00	6,27	8,38	3,00
	Tempo	549,07	636,61	524,66	622,00	614,85	632,99

Tabela 5.15: Melhores resultados (agrupados) dos métodos de solução (600 segundos).

5.8 Análises adicionais

Nesta seção são feitas duas análises adicionais sobre o problema; uma delas relacionada com o cálculo dos tempos que compõem as rotas e uma segunda análise onde são explorados cenários permitindo diferentes números de entregadores a bordo. Estas análises permitem avaliar a importância de considerar a decisão do tamanho da tripulação junto com as decisões de programação e roteamento.

Inicialmente são analisadas as rotas resultantes em termos dos tempos que as compõem, identificando as proporções correspondentes a tempos de serviço, de viagem e de espera, para cada uma das classes de instâncias utilizadas em todos os testes previamente executados. Lembrando que os tempos de espera se originam quando um veículo chega no grupo de clientes antes do instante de abertura de

sua janela de tempo. Foram utilizados os resultados dos testes realizados com o método híbrido exato (BPC_{MH}) durante uma hora de execução. Os resultados são resumidos e mostrados na Tabela 5.16, mostrando que os tempos totais das rotas são dominados pelos tempos de serviço, com uma média de 46%, seguidos dos tempos de deslocamento, com uma média de 32%. Esses resultados evidenciam a necessidade de considerar o tempo de serviço como dependente do número de entregadores e, por sua vez, o número de entregadores como uma variável de decisão.

Vale destacar que esse tipo de estrutura nos tempos das rotas pode ser observado na prática, especialmente na distribuição de bebidas em centros urbanos congestionados, onde os tempos de serviço podem ocupar até 50% dos tempos totais das rotas. Esse fato é uma das principais motivações para o estudo do PRVJTME.

Note que os tempos nas soluções das instâncias do conjunto C1 e C2 são dominados pelos tempos de espera, já que nessas instâncias os grupos de clientes se encontram aglomerados e, conseqüentemente, os veículos passam mais tempo esperando do que viajando entre os locais de estacionamento dos grupos de clientes.

Conjunto	Tempos		
	Viagem	Serviço	Espera
R1	49%	46%	5%
RC1	53%	44%	3%
C1	10%	45%	45%
R2	36%	52%	12%
RC2	40%	48%	12%
C2	7%	41%	52%
Média	32%	46%	22%

Tabela 5.16: Proporções dos tempos que compõem as rotas nos diferentes tipos de instâncias.

Posteriormente, analisamos cenários onde são permitidos diferentes números de entregadores a bordo. Esta análise resulta relevante já que devido a situações que podem surgir na prática, pode ser valioso apresentar soluções considerando diferentes limites no tamanho da tripulação permitida a bordo do veículo. Para realizar os testes, se utilizou o método BPC_{MH} mudando o número máximo de entregadores permitidos a bordo entre um e quatro, ressaltando que nos trabalhos de Pureza, Morabito e Reimann (2012) e Senarclens De Grancy e Reimann (2014) foi estabelecido um limite de três entregadores a bordo devido à restrição de capacidade da cabine de um caminhão médio. Para avaliar estes cenários, foram utilizadas as 12 instâncias do conjunto de R1, impondo uma hora como tempo limite de execução. Os resultados são apresentados na Tabela 5.17.

Pode ser observado que ao aumentarmos o tamanho da tripulação permitida a bordo também se aumenta o número médio de entregadores utilizados. No entanto, esse aumento permite utilizar um número menor de veículos e, por conseguinte, permite reduzir o custo médio das soluções. Esse fato

Inst	1 entregador			Até 2 entregadores			Até 3 entregadores			Até 4 entregadores		
	Custo	Veic	Entr	Custo	Veic	Entr	Custo	Veic	Entr	Custo	Veic	Entr
R101	40,94	37	37	26,99	23	38	23,67	19	45	22,17	17	50
R102	35,41	32	32	25,27	22	31	20,95	17	38	19,75	16	36
R103	25,48	23	23	16,64	14	25	15,84	13	27	14,94	12	28
R104	22,15	20	20	14,42	12	23	12,71	10	26	12,11	9	30
R105	28,80	26	26	19,25	16	31	17,55	14	34	16,34	12	42
R106	24,38	22	22	16,64	14	25	14,33	11	32	13,63	10	35
R107	23,26	21	21	14,52	12	24	13,01	10	29	12,21	9	31
R108	22,15	20	20	14,31	12	22	11,70	9	26	10,99	8	29
R109	24,37	22	22	16,74	14	26	15,03	12	29	13,62	10	35
R110	23,27	21	21	15,73	13	26	13,92	11	28	12,31	9	32
R111	23,26	21	21	14,52	12	24	12,91	10	28	12,21	9	31
R112	22,15	20	20	14,41	12	23	11,70	9	26	11,00	8	29
Média	26,30	23,75	23,75	17,45	14,67	26,50	15,28	12,08	30,67	14,27	10,75	34,00

Tabela 5.17: Custo das soluções variando o número de entregadores permitidos a bordo.

mostra os ganhos obtidos ao considerar o número de entregadores como variável de decisão quando se trabalha em contextos com tempos de serviço longos. Para a realização desses testes foi assumido que o número total entregadores disponíveis (E) era suficientemente grande. Este pressuposto ressalta o compromisso entre os custos envolvidos, já que resulta vantajoso investir em entregadores adicionais a fim de reduzir o número de veículos utilizados (cujo custo é maior) e, conseqüentemente, reduzir o custo total de operação do sistema.

5.9 Considerações finais

Neste capítulo foi feita uma análise abrangente de todos os métodos de solução desenvolvidos para o PRVJTME. Os resultados mostram que todos os métodos são capazes de resolver o problema, sendo que os métodos híbridos se mostraram mais adequados para instâncias com janelas de tempo apertadas e veículos com capacidade pequena, enquanto que as abordagens metaheurísticas apresentaram os melhores resultados para instâncias mais folgadas. Estes comportamentos dos métodos eram esperados desde que é bem sabido que instâncias folgadas são em geral mais difíceis de serem resolvidas por métodos BPC e, conseqüentemente, por métodos híbridos baseados em BPC.

Note que os resultados encontrados podem ser considerados dependentes das instâncias utilizadas, as quais foram retiradas da literatura. No entanto, dado que se utilizaram seis grupos de instâncias, cada um com características específicas diferentes e variadas, acredita-se que um comportamento similar ao obtido deve ser observado em outros contextos.

Algumas soluções obtidas são ilustradas no Apêndice A, a fim de mostrar graficamente características dos diferentes tipos de instâncias utilizadas.

Capítulo 6

Conclusões

Nesta dissertação foi estudado o Problema de Roteamento de Veículos com Janelas de Tempo e Múltiplos Entregadores, um problema de Pesquisa Operacional que tem uma grande aplicabilidade em contextos logísticos, especialmente em situações nas quais os tempos de serviço ocupam grande parte do tempo total das rotas. Em uma primeira fase, foram desenvolvidas e apresentadas duas abordagens metaheurísticas baseadas em Busca Local Iterada e Busca em Vizinhança Grande. Posteriormente, combinaram-se as metaheurísticas com um método *branch-price-and-cut* para obter métodos híbridos de solução. Estes métodos tentam aproveitar as melhores características de cada um dos seus componentes com o propósito de gerar boas soluções para o problema.

Experimentos computacionais utilizando instâncias da literatura foram realizados. Os resultados mostraram que os métodos de solução propostos são capazes de gerar boas soluções para o problema, melhorando o desempenho de abordagens da literatura em termos da qualidade das soluções obtidas. Os resultados evidenciam que as abordagens metaheurísticas são capazes de gerar boas soluções rapidamente quando comparadas com métodos da literatura. Por outro lado, os resultados dos métodos híbridos mostraram que eles são capazes de melhorar o desempenho do *branch-price-and-cut* como método de solução independente. Os métodos híbridos se mostraram especialmente adequados para resolver instâncias com janelas de tempo apertadas e veículos com capacidade pequena, onde relativamente poucos grupos de clientes podem ser atendidos, até mesmo quando são executados durante tempos relativamente pequenos, enquanto que as abordagens metaheurísticas apresentaram os melhores resultados para instâncias mais folgadas. Esse fato concorda com experiências encontradas na literatura e evidenciam a dificuldade que métodos tipo *branch-price-and-cut* têm para resolver instâncias mais folgadas e onde muitos nós podem ser atendidos pela mesma rota.

Perspectivas para pesquisas futuras envolvem o refinamento da metaheurística usada no método de geração de colunas, já que os experimentos computacionais mostraram que ela é capaz de oferecer bons resultados. Também, podem ser desenvolvidos outros tipos de métodos, onde as funções dos híbridos

apresentados nesta dissertação sejam invertidas, *e.g.*, um método exato seja usado para melhorar o desempenho de uma heurística ou metaheurística. Também, os algoritmos aqui apresentados podem ser estendidos para tratar outras variantes do PRVJTME que incluam frota heterogênea, coleta e entrega, múltiplas rotas, múltiplos depósitos, entre outras características práticas. Uma outra frente de pesquisas futuras consiste na integração com o agrupamento de clientes, o qual foi assumido como dado de entrada nesta pesquisa. A integração da roteirização com a definição dos grupos de clientes pode fornecer uma visão mais prática do problema.

Apêndices

Apêndice A

Gráficos das soluções

Neste apêndice mostramos graficamente soluções encontradas para instâncias dos diferentes conjuntos utilizados nesta dissertação, a fim de ilustrar como elas se diferenciam e como evoluem as soluções geradas pelos métodos. Para isso, utilizamos o método BPC_{MH} em uma instância aleatória de cada conjunto. Nas figuras mostradas, o triângulo preto ao centro representa o depósito, os nós azuis representam os primeiros grupos de clientes das rotas e os pontos representam os demais grupos. Os arcos de entrada e saída do depósito foram omitidos a fim de facilitar a visualização. As figuras são mostradas a seguir.

A Figura A.1 mostra a solução para a instância C107, a qual foi fornecida inicialmente pelas metaheurísticas e permaneceu como a incumbente até que o método híbrido provou sua otimalidade. Veja que os grupos de clientes se encontram agrupados e que as rotas atendem poucos grupos (10 grupos em média).

As Figuras A.2-A.3 mostram as soluções para as instâncias R101 e RC102, respectivamente. Para ambas as instâncias, o método começa com uma solução fornecida pelas metaheurísticas, passando por uma solução da heurística primal (SP e IP) e terminando com uma posteriormente melhorada por meio das metaheurísticas. A última solução em ambas as figuras corresponde à solução ótima da respectiva instância. Note que na figura da instância R101 os grupos de clientes se encontram espalhados aleatoriamente no espaço geográfico, enquanto que a instância RC102 apresenta uma mistura de grupos de clientes aglomerados e espalhados aleatoriamente. Observe também que cada rota instâncias atende relativamente poucos grupos de clientes (entre 5 e 8 grupos, em média).

A solução encontrada para a instância C205 é mostrada na Figura A.4, a qual foi encontrada inicialmente pelas metaheurísticas e permaneceu como a incumbente até finalizar a execução do método. Na figura pode-se conferir que os grupos se encontram aglomerados no espaço geográfico e que as rotas visitam uma maior quantidade de grupos (33 em média) quando comparadas com instâncias do tipo 1.

Finalmente, as Figuras A.5-A.6 mostram as soluções para as instâncias R203 e RC205, respectiva-

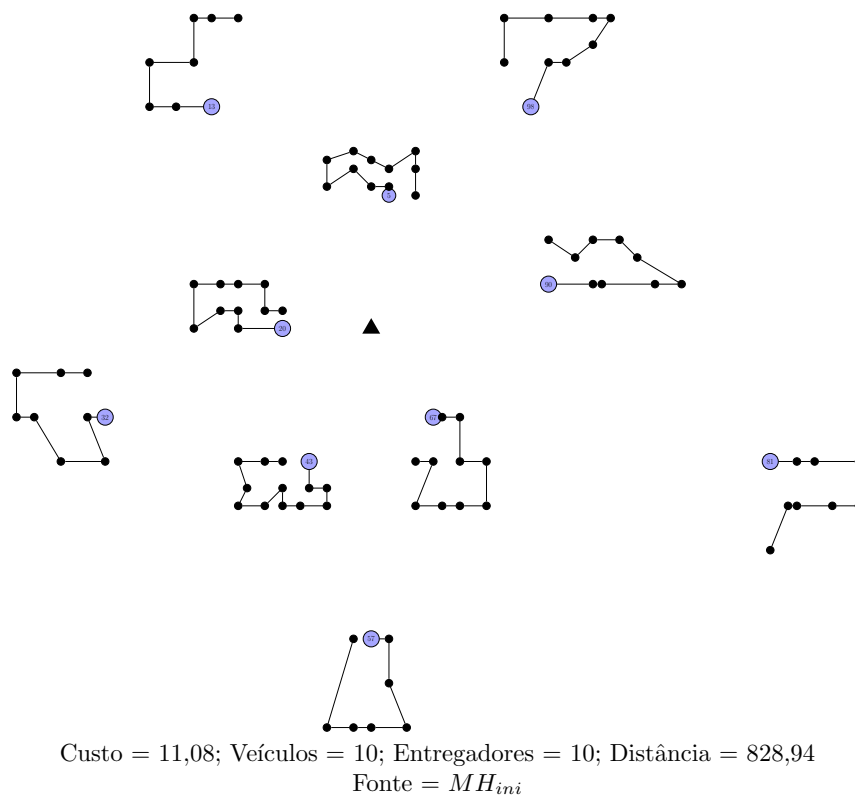
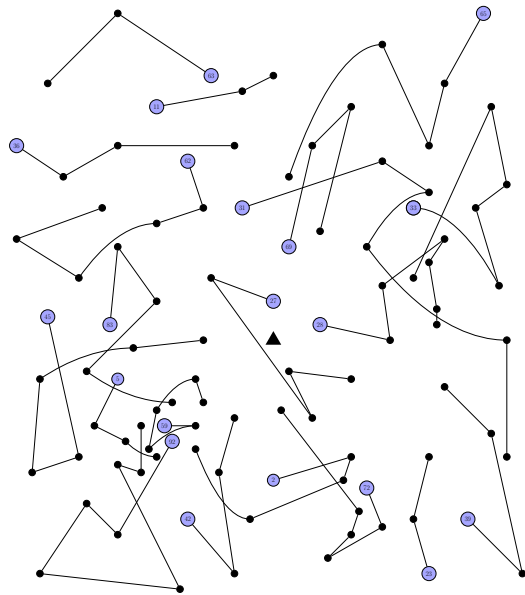
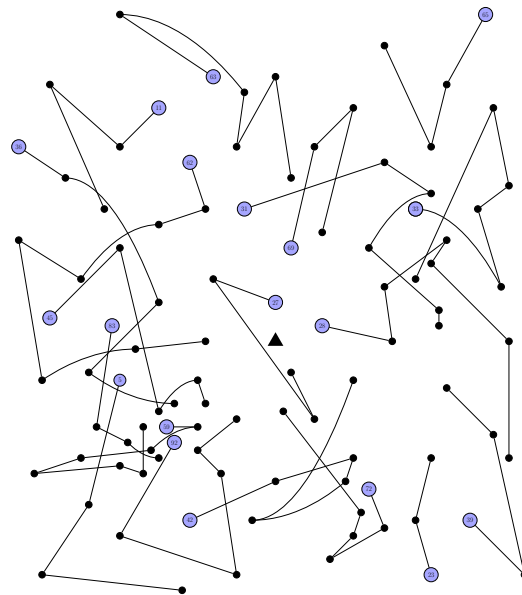


Figura A.1: Solução para a instância C107.

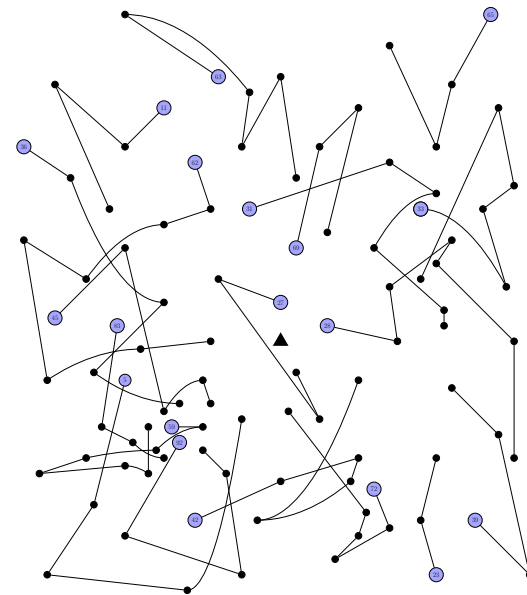
mente. Para ambas as instâncias, o método começa com uma solução fornecida pelas metaheurísticas e terminou com uma solução posteriormente melhorada através das metaheurísticas. Note as características da localização dos grupos de clientes para cada tipo de instância, *i.e.*, aleatoriamente espalhados para as tipo R e uma mistura entre aglomerado e espalhado aleatoriamente para as do tipo RC. Veja também que nestas instâncias os veículos são capazes de atender uma grande quantidade de clientes (entre 25 e 33 grupos, em média) dadas as características explicadas na Seção 5.1.



Custo = 24,47; Veículos = 20; Entregadores = 43; Distância = 1716,81
Fonte = MH_{ini}



Custo = 23,67; Veículos = 19; Entregadores = 45; Distância = 1720,40
Fonte = SP



Custo = 23,67; Veículos = 19; Entregadores = 45; Distância = 1720,11
Fonte = MH

Figura A.2: Soluções para a instância R101.

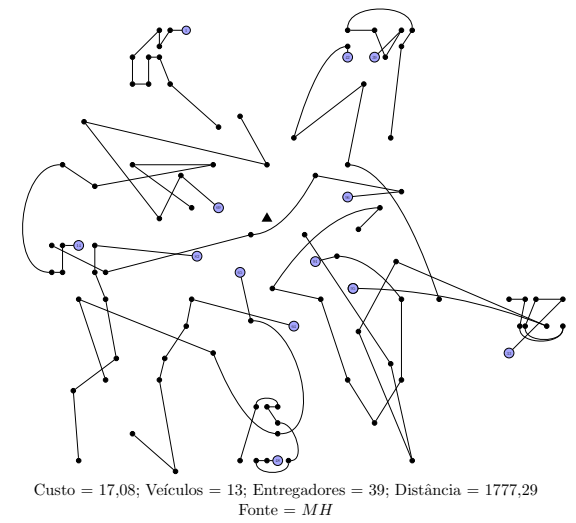
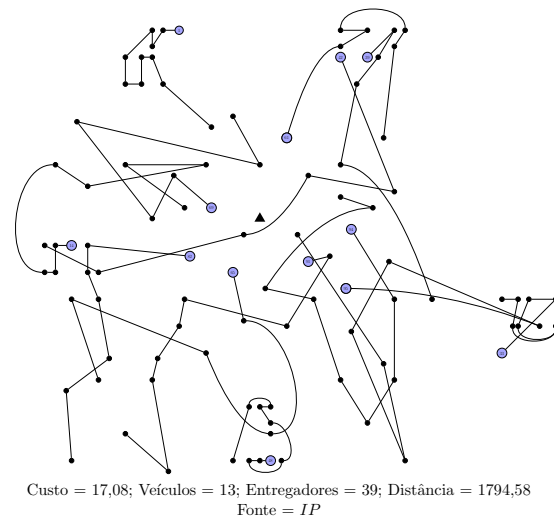
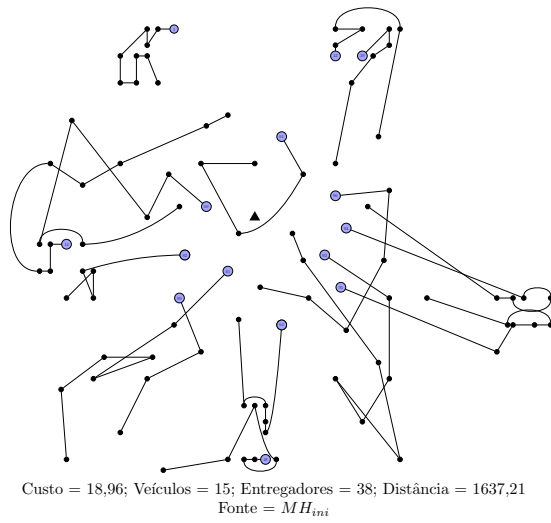
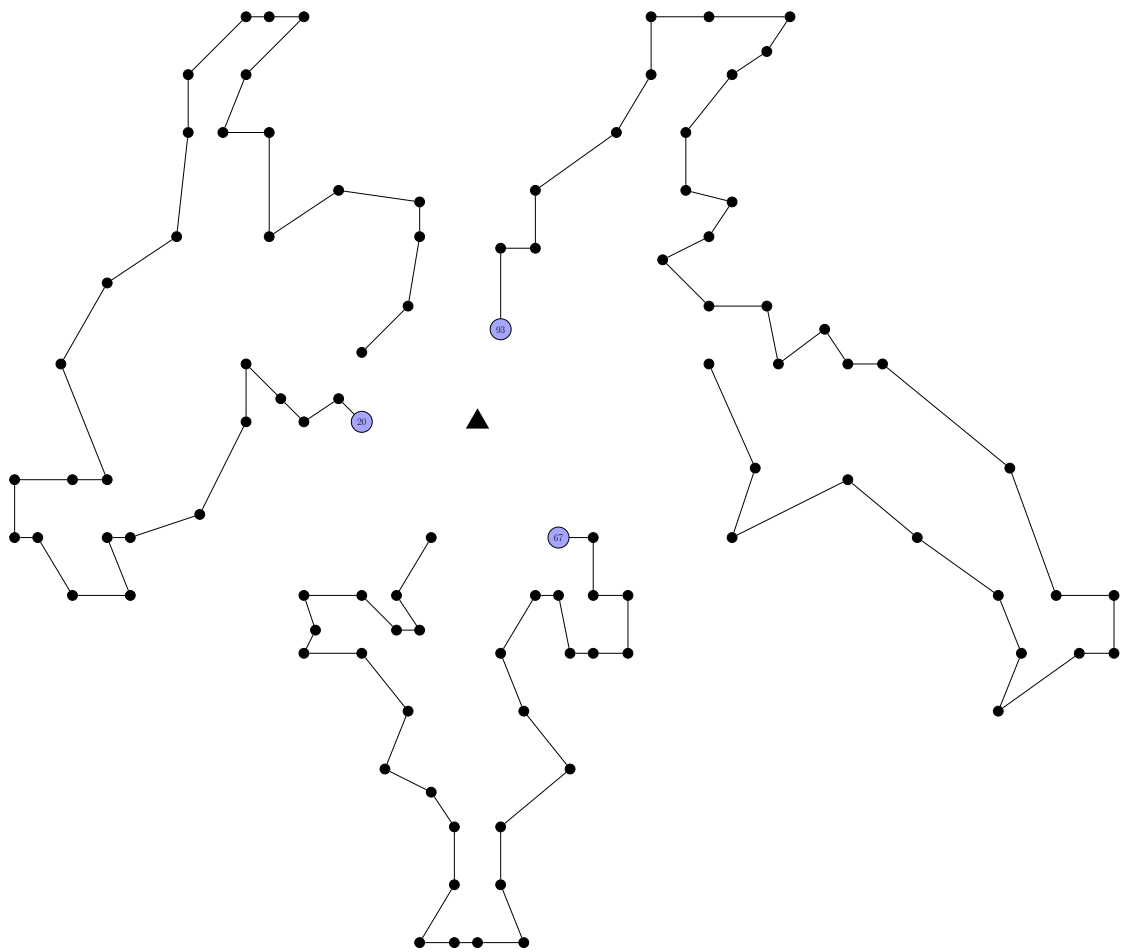
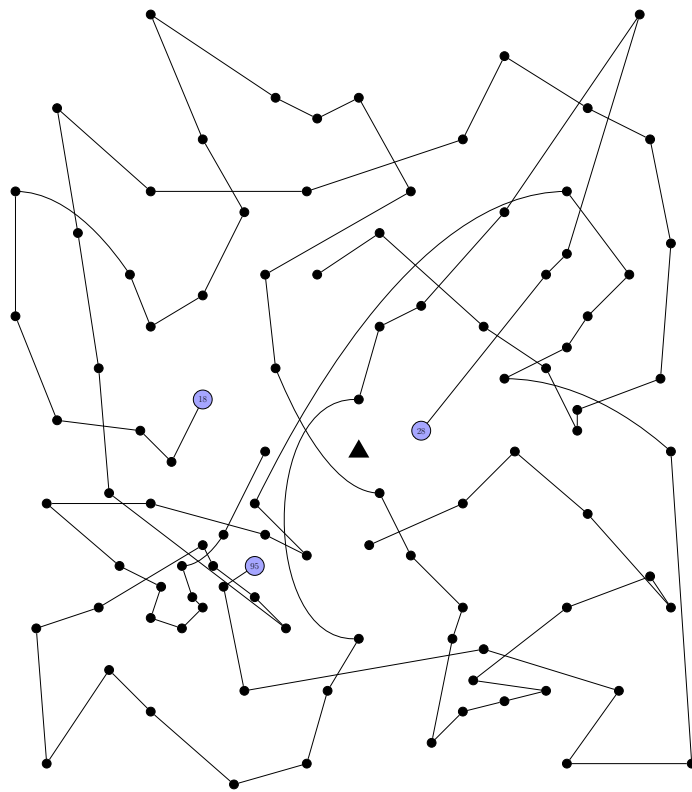


Figura A.3: Soluções para a instância RC102.

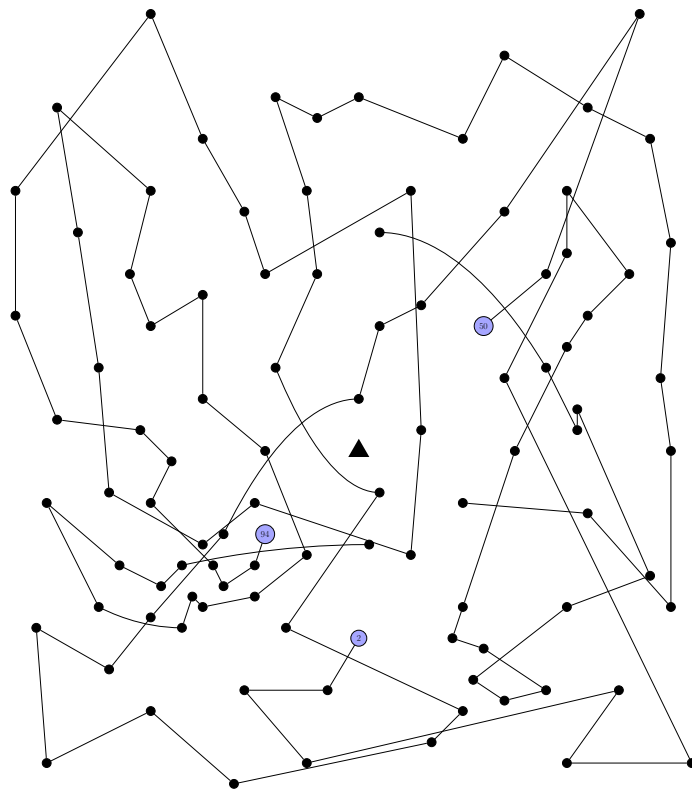


Custo = 3,36; Veículos = 3; Entregadores = 3; Distância = 588,49
 Fonte = MH_{ini}

Figura A.4: Solução para a instância C205.

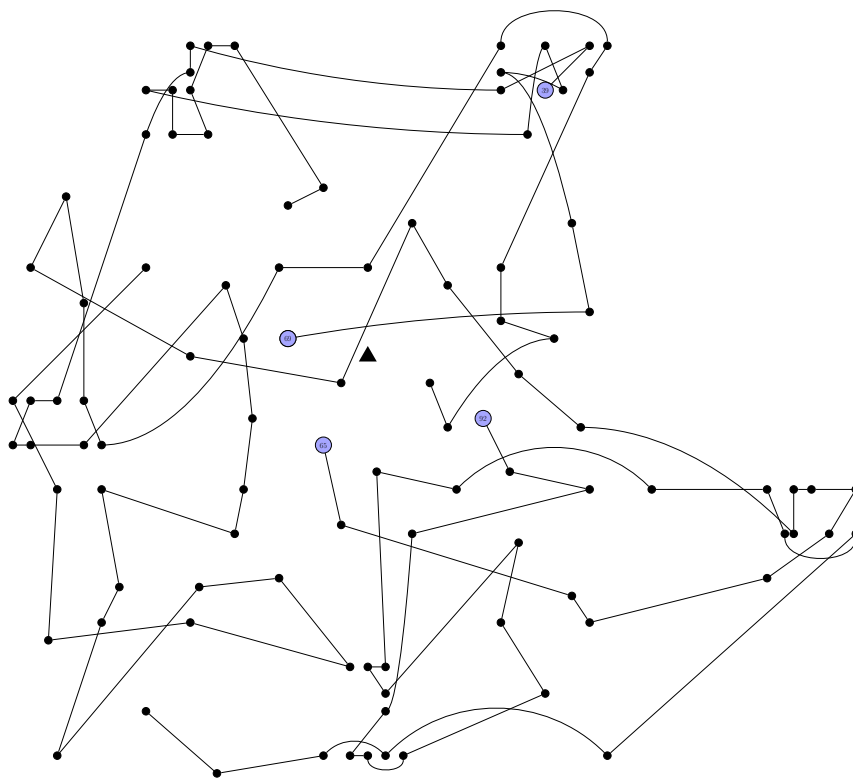


Custo = 3,80; Veículos = 3; Entregadores = 7; Distância = 1014,57
 Fonte = MH_{ini}

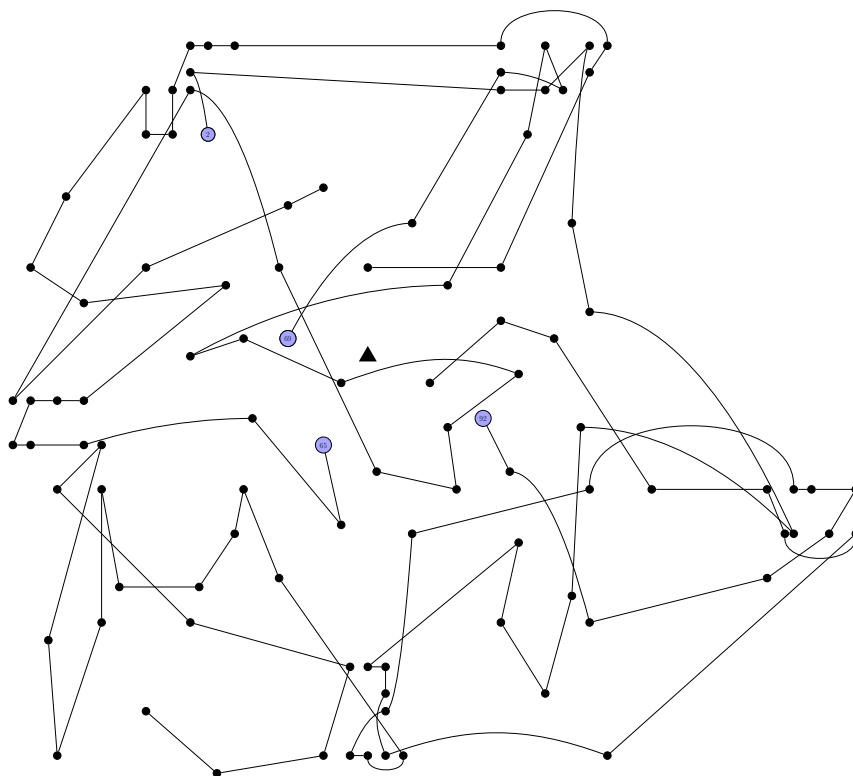


Custo = 3,71; Veículos = 3; Entregadores = 6; Distância = 1089,71
 Fonte = MH

Figura A.5: Soluções para a instância R203.



Custo = 5,24; Veículos = 4; Entregadores = 11; Distância = 1379,71
 Fonte = MH_{ini}



Custo = 5,14; Veículos = 4; Entregadores = 10; Distância = 1410,47
 Fonte = MH

Figura A.6: Soluções para a instância RC205.

Referências Bibliográficas

- ALVARENGA, G.; MATEUS, G.; DE TOMI, G. A genetic and set partitioning two-phase approach for the vehicle routing problem with time windows. *Computers & Operations Research*, v. 34, n. 6, p. 1561–1584, 2007.
- ALVELOS, F.; DE SOUSA, A.; SANTOS, D. Combining column generation and metaheuristics. *Studies in Computational Intelligence*, v. 434, p. 285–334, 2013.
- ARCHETTI, C.; SPERANZA, M. G. A survey on matheuristics for routing problems. *EURO Journal on Computational Optimization*, v. 2, n. 4, p. 223–246, 2014.
- ARCHETTI, C.; SPERANZA, M. G.; SAVELSBERGH, M. W. P. An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, v. 42, n. 1, p. 22–31, 2008.
- ARENALES, M. N.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. *Pesquisa Operacional*. [S.l.]: Elsevier, 2007.
- AZI, N.; GENDREAU, M.; POTVIN, J. Y. An adaptive large neighborhood search for a vehicle routing problem with multiple routes. *Computers & Operations Research*, v. 41, n. 1, p. 167–173, 2014.
- BALDACCI, R.; MINGOZZI, A.; ROBERTI, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. *European Journal of Operational Research*, v. 218, n. 1, p. 1–6, 2012.
- BALL, M. O. Heuristics based on mathematical programming. *Surveys in Operations Research and Management Science*, v. 16, n. 1, p. 21–38, 2011.
- BARNHART, C.; JOHNSON, E. L.; NEMHAUSER, G. L.; SAVELSBERGH, M. W. P.; VANCE, P. H. Branch-and-price: Column generation for solving huge integer programs. *Operations Research*, v. 46, p. 316–329, 1998.
- BEASLEY, J. E.; CHRISTOFIDES, N. An algorithm for the resource constrained shortest path problem. *Networks*, Wiley Subscription Services, Inc., A Wiley Company, v. 19, n. 4, p. 379–394, 1989.
- BERTRAND, J. W. M.; FRANSOO, J. C. Operations management research methodologies using quantitative modeling. *International Journal of Operations & Production Management*, v. 22, n. 2, p. 241–264, 2002.
- BOX, G. E.; ANDERSEN, S. L. Permutation theory in the derivation of robust criteria and the study of departures from assumption. *Journal of the Royal Statistical Society. Series B (Methodological)*, JSTOR, v. 17, n. 1, p. 1–34, 1955.
- BRÄYSY, O.; GENDREAU, M. Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation Science*, v. 39, p. 104–118, 2005.
- CACCHIANI, V.; HEMMELMAYR, V. C.; TRICOIRE, F. A set-covering based heuristic algorithm for the periodic vehicle routing problem. *Discrete Applied Mathematics*, v. 163, p. 53–64, 2014.

- CHABRIER, A. Vehicle routing problem with elementary shortest path based column generation. *Computers & Operations Research*, v. 33, n. 10, p. 2972 – 2990, 2006.
- CHEN, Z.-L.; XU, H. Dynamic column generation for dynamic vehicle routing with time windows. *Transportation Science*, v. 40, n. 1, p. 74–88, 2006.
- CLARKE, G.; WRIGHT, J. W. Scheduling of vehicles from a central depot to a number of delivery points. *Operations Research*, v. 12, p. 568–581, 1964.
- CPLEX, I. I. *Concert Technology version 12.1, C++ API Reference Manual*. [S.l.], 2009.
- CUERVO, D. P.; GOOS, P.; SÖRENSEN, K.; ARRÁIZ, E. An iterated local search algorithm for the vehicle routing problem with backhauls. *European Journal of Operational Research*, v. 237, n. 2, p. 454 – 464, 2014.
- DANNA, E.; LE PAPE, C. Branch-and-price heuristics: A case study on the vehicle routing problem with time windows. In: DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M. (Ed.). *Column Generation*. [S.l.]: Springer US, 2005. p. 99–129.
- DANTZIG, G. B.; RAMSER, J. H. The truck dispatching problem. *Management Science*, v. 6, p. 80–91, 1959.
- DANTZIG, G. B.; WOLFE, P. Decomposition principle for linear programs. *Operations Research*, v. 8, n. 1, p. 101–111, 1960.
- DE FRANCESCHI, R.; FISCHETTI, M.; TOTH, P. A new ilp-based refinement heuristic for vehicle routing problems. *Mathematical Programming*, v. 105, n. 2-3, p. 471–499, 2006.
- DESAULNIERS, G.; LESSARD, F.; HADJAR, A. Tabu search, partial elementarity, and generalized k-path inequalities for the vehicle routing problem with time windows. *Transportation Science*, v. 42, n. 3, p. 387–404, 2008.
- DESAULNIERS, G.; MADSEN, O. B.; ROPKE, S. The vehicle routing problem with time windows. In: TOTH, P.; VIGO, D. (Ed.). *Vehicle Routing: Problems, Methods, and Applications*. [S.l.]: SIAM, 2014. p. 119–159.
- DESROCHERS, M. *An algorithm for the shortest path problem with resource constraints*. [S.l.], 1988.
- DESIERS, J.; LÜBBECKE, M. E.; COCHRAN, J. J.; COX, L. A.; KESKINOCAK, P.; KHAROUFEH, J. P.; SMITH, J. C. Branch-price-and-cut algorithms. In: *Wiley Encyclopedia of Operations Research and Management Science*. [S.l.]: John Wiley & Sons, Inc., 2010.
- DILLENBERGER, C.; ESCUDERO, L. F.; WOLLENSAK, A.; ZHANG, W. On practical resource allocation for production planning and scheduling with period overlapping setups. *European Journal of Operational Research*, v. 75, n. 2, p. 275 – 286, 1994.
- DROR, M. Note on the complexity of the shortest-path models for column generation in VRPTW. *Operations Research*, 42, n. 5, p. 977–978, 1994.
- FEILLET, D.; DEJAX, P.; GENDREAU, M.; GUEGUEN, C. An exact algorithm for the elementary shortest path problem with resource constraints: application to some vehicle routing problems. *Networks*, v. 44, p. 216–229, 2004.
- FEO, T. A.; RESENDE, M. G. Greedy randomized adaptive search procedures. *Journal of Global Optimization*, v. 6, n. 2, p. 109–133, 1995.
- FERREIRA, V.; PUREZA, V. Some experiments with a savings heuristic and a tabu search approach for the vehicle routing problem with multiple deliverymen. *Pesquisa Operacional*, v. 32, p. 443–463, 2012.

- FISCHETTI, M.; LODI, A. Local branching. *Mathematical Programming*, v. 98, n. 1-3, p. 23–47, 2003.
- FRANÇA, P.; SOSA, N.; PUREZA, V. An adaptive tabu search algorithm for the capacitated clustering problem. *International Transactions in Operational Research*, v. 6, n. 6, p. 665 – 678, 1999.
- FUKASAWA, R.; LONGO, H.; LYSGAARD, J.; POGGI DE ARAGÃO, M.; REIS, M.; UCHOA, E.; WERNECK, R. F. Robust branch-and-cut-and-price for the capacitated vehicle routing problem. *Mathematical Programming*, v. 106, n. 3, p. 491–511, 2006.
- GLASS, G. V.; PECKHAM, P. D.; SANDERS, J. R. Consequences of failure to meet assumptions underlying the fixed effects analyses of variance and covariance. *Review of educational research*, JSTOR, p. 237–288, 1972.
- GONDZIO, J.; GONZÁLEZ-BREVIS, P.; MUNARI, P. New developments in the primal-dual column generation technique. *European Journal of Operational Research*, v. 224, n. 1, p. 41–51, 2013.
- HAUGE, K.; LARSEN, J.; LUSBY, R. M.; KRAPPER, E. A hybrid column generation approach for an industrial waste collection routing problem. *Computers & Industrial Engineering*, v. 71, p. 10–20, 2014.
- HONG, L. An improved Ins algorithm for real-time vehicle routing problem with time windows. *Computers & Operations Research*, v. 39, n. 2, p. 151–163, 2012.
- HUTTER, F.; HOOS, H. H.; STÜTZLE, T. Automatic algorithm configuration based on local search. *Proceedings of the Twenty-Second Conference on Artificial Intelligence*, p. 1152–1157, 2007.
- ILOG Inc. *Solver CPLEX*. 2008. (<http://www.ilog.com/products/cplex/>).
- IRNICH, S.; DESAULNIERS, G. Shortest path problems with resource constraints. In: DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M. (Ed.). *Column Generation*. [S.l.]: Springer US, 2005. p. 33–65.
- JEPSEN, M.; PETERSEN, B.; SPOORENDONK, S.; PISINGER, D. Subset-row inequalities applied to the vehicle routing problem with time windows. *Operations Research*, v. 56, n. 2, p. 497–511, 2008.
- JONCOUR, C.; MICHEL, S.; SADYKOV, R.; SVERDLOV, D.; VANDERBECK, F. Column generation based primal heuristics. *Electronic Notes in Discrete Mathematics*, v. 36, p. 695–702, 2010.
- JOURDAN, L.; BASSEUR, M.; TALBI, E.-G. Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, v. 199, n. 3, p. 620–629, 2009.
- KRAMER, R.; SUBRAMANIAN, A.; VIDAL, T.; CABRAL, L. d. A. F. A matheuristic approach for the pollution-routing problem. *European Journal of Operational Research*, v. 243, n. 2, p. 523–539, 2015.
- LIX, L. M.; KESELMAN, J. C.; KESELMAN, H. Consequences of assumption violations revisited: A quantitative review of alternatives to the one-way analysis of variance f test. *Review of educational research*, v. 66, n. 4, p. 579–619, 1996.
- LOURENÇO, H.; MARTIN, O.; STÜTZLE, T. Iterated local search. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). *Handbook of Metaheuristics*. [S.l.]: Springer US, 2003. p. 321–353.
- LOURENÇO, H.; MARTIN, O.; STÜTZLE, T. Iterated local search: Framework and applications. In: *Handbook of Metaheuristics*. [S.l.: s.n.], 2010. v. 146, p. 363–397.
- LOZANO, L.; DUQUE, D.; MEDAGLIA, A. L. An exact algorithm for the elementary shortest path problem with resource constraints. *Transportation Science*, in press, n. July, p. 1–10, 2015.
- LÜBBECKE, M.; DESROSIERS, J. Selected topics in column generation. *Operations Research*, v. 53, n. 6, p. 1007–1023, 2005.

- LÜBBECKE, M.; PUCHERT, C. Primal heuristics for branch-and-price algorithms. In: KLATTE, D.; LÜTHI, H.-J.; SCHMEDDERS, K. (Ed.). *Operations Research Proceedings 2011*. [S.l.]: Springer Berlin Heidelberg, 2012. p. 65–70.
- MANIEZZO, V.; STÜTZLE, T.; VOSS, S. (Ed.). *Matheuristics*. Boston, MA: Springer US, 2010. (Annals of Information Systems, v. 10).
- MELECHOVSKY, J. Evolutionary local search algorithm to solve the multi-compartment vehicle routing problem with time windows. *Proceedings of 30th International Conference Mathematical Methods in Economics*, p. 564–568, 2012.
- MENDOZA, J. E.; VILLEGAS, J. G. A multi-space sampling heuristic for the vehicle routing problem with stochastic demands. *Optimization Letters*, v. 7, n. 7, p. 1503–1516, 2012.
- MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. *Computers & Operations Research*, v. 24, p. 1097–1100, 1997.
- MONTGOMERY, D. C. *Design and analysis of experiments*. [S.l.]: John Wiley & Sons, 2008.
- MORABITO, R.; PUREZA, V. Modelagem e simulação. In: CAUCHICK, P. A. C. (Ed.). *Metodologia de pesquisa em engenharia de produção e gestão de operações*. Segunda edição. [S.l.]: Elsevier Editora Ltda., 2012. p. 169 – 198.
- MUNARI, P.; GONDZIO, J. Using the primal-dual interior point algorithm within the branch-price-and-cut method. *Computers & Operations Research*, v. 40, n. 8, p. 2026–2036, 2013.
- MUNARI, P.; MORABITO, R. *The vehicle routing problem with time windows and multiple deliverymen: exact solution approaches*. 2015. Working paper, Departamento de Engenharia de Produção, Universidade Federal de São Carlos.
- MUTER, I.; BIRBIL, S. I.; SAHIN, G. Combination of metaheuristic and exact algorithms for solving set covering-type optimization problems. *INFORMS Journal on Computing*, v. 22, n. 4, p. 603–619, 2010.
- NADERI, B.; RUIZ, R.; ZANDIEH, M. Algorithms for a realistic variant of flowshop scheduling. *Computers & Operations Research*, v. 37, n. 2, p. 236–246, 2010.
- NISHI, T.; IZUNO, T. Column generation heuristics for ship routing and scheduling problems in crude oil transportation with split deliveries. *Computers & Chemical Engineering*, v. 60, p. 329–338, 2014.
- OPTIMIZATION, I. G. *Gurobi Optimizer Reference Manual*. 2015. Disponível em: (<http://www.gurobi.com>).
- PARRAGH, S. N.; SCHMID, V. Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & operations research*, v. 40, n. 1, p. 490–497, 2013.
- PECIN, D.; PESSOA, A.; POGGI, M.; UCHOA, E. Improved branch-cut-and-price for capacitated vehicle routing. In: LEE, J.; VYGEN, J. (Ed.). *Integer Programming and Combinatorial Optimization*. [S.l.]: Springer International Publishing, 2014, (Lecture Notes in Computer Science). p. 393–403.
- PENNA, P.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, v. 19, p. 201–232, 2013.
- PILLAC, V.; GUÉRET, C.; MEDAGLIA, A. A parallel matheuristic for the technician routing and scheduling problem. *Optimization Letters*, v. 7, n. 7, p. 1525–1535, 2012.
- PISINGER, D.; ROPKE, S. A general heuristic for vehicle routing problems. *Computers & Operations Research*, v. 34, p. 2403–2435, 2007.

- PISINGER, D.; ROPKE, S. Large neighborhood search. In: *Handbook of Metaheuristics*. [S.l.]: Springer US, 2010. v. 146, p. 1–22.
- PUREZA, V.; FRANÇA, P. Uma abordagem adaptativa de busca tabu aplicada ao problema de roteamento de veículos. *Transportes*, v. 9, n. 2, p. 28–47, 2001.
- PUREZA, V.; MORABITO, R.; REIMANN, M. Vehicle routing with multiple deliverymen: Modeling and heuristic approaches for the vrptw. *European Journal of Operational Research*, v. 218, n. 3, p. 636–647, 2012.
- RAIDL, G. Decomposition based hybrid metaheuristics. *European Journal of Operational Research*, v. 244, n. 1, p. 66 – 76, 2015.
- RAIDL, G.; PUCHINGER, J. Combining (integer) linear programming techniques and metaheuristics for combinatorial optimization. *Hybrid Metaheuristics*, v. 62, n. 2008, p. 31–62, 2008.
- REIMANN, M.; DOERNER, K.; HARTL, R. F. Insertion based ants for vehicle routing problems with backhauls and time windows. In: DORIGO, M.; CARO, G. D.; SAMPELS, M. (Ed.). *Ant Algorithms*. [S.l.]: Springer Berlin Heidelberg, 2012. v. 2463, p. 135–148.
- RIGHINI, G.; SALANI, M. New dynamic programming algorithms for the resource constrained elementary shortest path problem. *Networks*, v. 51, n. 3, p. 155–170, 2008.
- ROCHAT, Y.; TAILLARD, É. D. Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, v. 1, n. 1, p. 147–167, 1995.
- ROPKE, S.; CORDEAU, J.-F. Branch and cut and price for the pickup and delivery problem with time windows. *Transportation Science*, v. 43, n. 3, p. 267–286, 2009.
- ROPKE, S.; PISINGER, D. An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation Science*, v. 40, p. 455–472, 2006.
- ROUSSEAU, L.-M.; GENDREAU, M.; FEILLET, D. Interior point stabilization for column generation. *Operations Research Letters*, v. 35, n. 5, p. 660–668, 2007.
- RUSSELL, R.; CHIANG, W.-C. Scatter search for the vehicle routing problem with time windows. *European Journal of Operational Research*, v. 169, n. 2, p. 606–622, 2006.
- SADYKOV, R.; VANDERBECK, F.; PESSOA, A.; TAHIRI, I.; UCHOA, E. Primal heuristics for branch-and-price. Working paper or preprint. 2015. Disponível em: (<https://hal.inria.fr/hal-01237204>).
- SALARI, M.; TOTH, P.; TRAMONTANI, A. An ilp improvement procedure for the open vehicle routing problem. *Computers & Operations Research*, v. 37, n. 12, p. 2106–2120, 2010.
- SANTOS, D.; DE SOUSA, A.; ALVELOS, F. A hybrid column generation with grasp and path relinking for the network load balancing problem. *Computers & Operations Research*, v. 40, n. 12, p. 3147–3158, 2013.
- SAVELSBERGH, M.; SOL, M. DRIVE: Dynamic routing of independent vehicles martin. *Operations research*, v. 46, n. 4, p. 474–490, 1998.
- SCHMID, V.; DOERNER, K. F.; HARTL, R. F.; SALAZAR-GONZÁLEZ, J.-J. Hybridization of very large neighborhood search for ready-mixed concrete delivery problems. *Computers & Operations Research*, v. 37, n. 3, p. 559–574, 2010.
- SENARCLENS DE GRANCY, G.; REIMANN, M. Vehicle routing problems with time windows and multiple service workers: a systematic comparison between aco and grasp. *Central European Journal of Operations Research*, 2014.

- SHAW, P. Using constraint programming and local search methods to solve vehicle routing problems. In: *Principles and Practice of Constraint Programming - CP98*. [S.l.: s.n.], 1998. p. 417–431.
- SILVA, M. M.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the split delivery vehicle routing problem. *Computers & Operations Research*, v. 53, p. 234–249, 2015.
- SOLOMON, M. Algorithms for the vehicle routing and scheduling problems with time window constraints. *Operations Research*, v. 35, n. 2, p. 254–265, 1987.
- SPLIET, R.; DESAULNIERS, G. The discrete time window assignment vehicle routing problem. *European Journal of Operational Research*, v. 244, n. 2, p. 379–391, 2015.
- SUBRAMANIAN, A.; PENNA, P. H. V.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, v. 221, n. 2, p. 285–295, 2012.
- SUBRAMANIAN, A.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for a class of vehicle routing problems. *Computers & Operations Research*, v. 40, n. 10, p. 2519–2531, 2013.
- TOTH, P.; VIGO, D. *The vehicle routing problem*. [S.l.]: SIAM, 2002. 367 p.
- VALLE, C. A.; MARTINEZ, L. C.; DA CUNHA, A. S.; MATEUS, G. R. Heuristic and exact algorithms for a min-max selective vehicle routing problem. *Computers & Operations Research*, v. 38, n. 7, p. 1054–1065, 2011.
- VANDERBECK, F. Implementing mixed integer column generation. In: DESAULNIERS, G.; DESROSIERS, J.; SOLOMON, M. M. (Ed.). *Column Generation*. [S.l.]: Springer US, 2005. p. 331–358.
- VILLEGAS, J. G.; PRINS, C.; PRODHON, C.; MEDAGLIA, A.; VELASCO, N. A matheuristic for the truck and trailer routing problem. *European Journal of Operational Research*, v. 230, n. 2, p. 231–244, 2013.
- XU, H.; CHEN, Z.-L.; RAJAGOPAL, S.; ARUNAPURAM, S. Solving a practical pickup and delivery problem. *Transportation Science*, v. 37, n. 3, p. 347–364, 2003.
- YILDIRIM, U. M.; ÇATAY, B. A parallel matheuristic for solving the vehicle routing problems. In: SOUSA, J. F.; ROSSI, R. (Ed.). *Computer-based Modelling and Optimization in Transportation SE - 35*. [S.l.]: Springer International Publishing, 2014. v. 262, p. 477–489.