

UNIVERSIDADE FEDERAL FLUMINENSE

MARQUES MOREIRA DE SOUSA

**HEURÍSTICAS PARA RESOLUÇÃO DO
PROBLEMA DO CAIXEIRO VIAJANTE COM
SELEÇÃO DE HOTÉIS**

NITERÓI

2020

UNIVERSIDADE FEDERAL FLUMINENSE

MARQUES MOREIRA DE SOUSA

HEURÍSTICAS PARA RESOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE COM SELEÇÃO DE HOTÉIS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação. Área de concentração: ALGORITMOS E OTIMIZAÇÃO.

Orientador:
LUIZ SATORU OCHI

Coorientadora:
SIMONE DE LIMA MARTINS

NITERÓI

2020

Ficha catalográfica automática - SDC/BEE
Gerada com informações fornecidas pelo autor

S725h Sousa, Marques Moreira de
 Heurísticas para Resolução do Problema do Caixeiro
 Viajante com Seleção de Hotéis / Marques Moreira de Sousa ;
 Luiz Satoru Ochi, orientador ; Simone de Lima Martins,
 coorientadora. Niterói, 2020.
 120 f.

 Tese (doutorado)-Universidade Federal Fluminense, Niterói,
 2020.

 DOI: <http://dx.doi.org/10.22409/PGC.2020.d.09158918671>

 1. Problema do caixeiro-viajante. 2. Heurística. 3.
 Mineração de dados. 4. Produção intelectual. I. Ochi, Luiz
 Satoru, orientador. II. Martins, Simone de Lima,
 coorientadora. III. Universidade Federal Fluminense. Instituto
 de Computação. IV. Título.

CDD -

MARQUES MOREIRA DE SOUSA

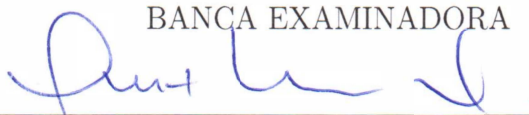
HEURÍSTICAS PARA RESOLUÇÃO DO PROBLEMA DO CAIXEIRO VIAJANTE COM SELEÇÃO DE HOTÉIS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense como requisito parcial para a obtenção do Grau de Doutor em Computação.

Área de concentração: ALGORITMOS E OTIMIZAÇÃO.

Aprovada em maio de 2020.

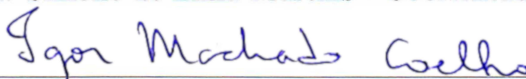
BANCA EXAMINADORA



Prof. Luiz Satoru Ochi - Orientador, UFF



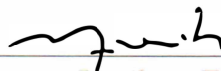
Profa. Simone de Lima Martins - Coorientadora, UFF



Prof. Igor Machado Coelho, UFF



Prof. Yuri Abitbol de Menezes Frota, UFF



Prof. Marcone Jamilson Freitas Souza, UFOP



Prof. Pedro Henrique González Silva, CEFET-RJ



Prof. Philippe Yves Paul Michelon, Université d'Avignon

Niterói

2020

Agradecimentos

Mesmo que sejam utilizadas as palavras mais bonitas e sinceras aqui, essas não seriam suficientes para expressar toda minha gratidão aos que direta ou indiretamente contribuíram para a conclusão deste trabalho.

Agradeço primeiramente aos meus pais, Marieta e José por sempre acreditarem em mim, mesmo quando eu mesmo já não acreditava. Aos meus irmãos Carlos e Marcos pelo suporte incondicional.

Agradeço à minha companheira, amiga e esposa Mariana, por compreender e auxiliar em todos os momentos de dificuldade.

Aos meus orientadores, Simone Martins e Luiz Satoru, que durante os mais de longos cinco anos advertiram, orientaram, motivaram e trabalharam, como trabalharam! Vocês são gigantes, pessoal e profissionalmente.

Aos colegas e amigos Sávio, Italo, Pedro, Edcarllos, Glaubos, Victor, Raylan, Guerine, Anderson e outros mais que contribuíram e auxiliaram prontamente em momentos de dúvidas e de descontração.

Ao IC/UFF por toda a estrutura disponibilizada. Aos demais professores do IC e membros da banca pelos ensinamentos, comentários e contribuições durante o curso e na avaliação deste trabalho.

Àqueles que não foram citados, mas que sabem que contribuíram de alguma forma em diversos momentos desta jornada nomeada doutorado.

À CAPES e ao IFSP Campus Campos do Jordão pelo apoio financeiro e suporte, sem os quais não seria possível conduzir este trabalho.

Muito obrigado a todos!

Resumo

Problemas de otimização combinatória estão cada vez mais presentes na tomada de decisão dos mais variados setores da economia. No Brasil, devido a sua extensa malha rodoviária pela qual a maioria dos produtos e serviços essenciais à população são transportados, o uso de técnicas de otimização para geração de rotas e consequente minimização dos custos operacionais, apresenta-se como uma estratégia relevante. Definir rotas por meio da representação de problemas de otimização consiste em uma alternativa para redução de custos, mantendo empresas competitivas em um mercado cada vez mais sufocado pela alta carga tributária e outros encargos que incidem sobre toda a cadeia produtiva. Nesse sentido, o Problema do Caixeiro Viajante com Seleção de Hotéis consiste na definição de uma rota que minimiza a soma dos tempos necessários para viajar entre clientes, respeitando um limite de tempo de viagem diário e escolhendo, quando necessário, hotéis para descanso entre dias consecutivos de trabalho. Sua variante com múltiplos caixeiros, além de minimizar os tempos necessários para viajar entre clientes e realizar a escolha dos hotéis intermediários, precisa construir múltiplas rotas, uma para cada caixeiro, respeitando um limite máximo de viagens para cada rota. Nesta tese, propomos heurísticas e estratégias híbridas utilizando a metaheurística Iterated Local Search (ILS), técnicas de mineração de dados e de Programação Linear Inteira. Para diversas instâncias contidas na literatura obtivemos resultados equivalentes aos melhores já conhecidos, bem como novas melhores soluções para uma parcela considerável das 1811 instâncias conhecidas para os dois problemas.

Palavras-chave: Problema do Caixeiro Viajante; Seleção de Hotéis; Heurísticas; Mineração de Padrões Frequentes.

Abstract

Combinatorial optimization problems are increasingly present in decision making in the most varied sectors of the economy. In Brazil, extensive road networks are used to transport most products and services essential to the population. Therefore the use of optimization techniques for generating routes and consequent minimization of operating costs is a relevant strategy. Defining routes through the representation of optimization problems is an alternative to reduce costs, keeping companies competitive in a market that is increasingly suffocated by high tax burden and other charges that affect the entire production chain. In this sense, the Traveling Salesperson Problem with Hotel Selection consists of defining a route that minimizes the sum of the times needed to travel between customers, respecting a daily travel time limit, and choosing when necessary hotels for rest consecutive work breaks. Its variant with multiple traveling salespersons needs to build various routes, one for each salesperson, respecting a maximum travel limit for each tour, beyond minimizing the time required to move between customers and making the choice of intermediate hotels. In this thesis, we propose heuristics and hybrid strategies using the metaheuristic Iterated Local Search and data mining and Integer Linear Programming techniques. For several instances contained in the literature, we obtained results equivalent to the best already known solutions, as well as new better solutions for some of the 1811 instances for the two problems.

Keywords: Traveling Salesperson Problem; Hotel Selection; Heuristics; Mining of Frequent Patterns.

Lista de Figuras

2.1	Movimento realizado na estrutura <i>2-opt</i>	7
2.2	Movimento realizado na estrutura <i>Or-opt</i>	7
2.3	Movimento realizado na estrutura <i>Relocate</i>	8
2.4	Movimento realizado na estrutura <i>Exchange</i>	8
2.5	Movimento realizado na estrutura <i>ChangeHotels</i>	8
2.6	Movimento realizado na estrutura <i>JoinTrips</i>	9
2.7	Representação de soluções para o PCV e PCVSH.	16
2.8	Movimento realizado na estrutura <i>InsertHotel</i>	20
2.9	Movimento realizado na estrutura <i>Shift</i>	20
2.10	Movimento realizado na estrutura <i>Swap</i>	21
3.1	Comportamento do procedimento de intensificação da perturbação.	27
3.2	Exemplo do grafo criado para obtenção de uma solução do PCVSH.	29
3.3	Relação entre porcentagem de perturbação e tamanho da instância.	33
3.4	Transformação de soluções em <i>Ids</i>	36
3.5	Aplicação da técnica FIM com <i>sup</i> = 50%.	37
5.1	Dispersão correspondente à diferenças entre o desvio padrão calculado e o valor da melhor solução alcançada.	68
5.2	Comparação de resultados para MILS com diferentes tempos computacionais.	80
6.1	<i>Gaps</i> médios agrupados por indicadores distintos.	91
6.2	Percentual de soluções ótimas encontradas.	93
6.3	<i>Gaps</i> médios agrupados por indicadores distintos.	95
6.4	Percentual de soluções ótimas encontradas.	96

Lista de Tabelas

5.1	Resultados comparativos em relação ao número de viagens, tempo computacional médio e <i>Gap</i> entre ILS-RVND com perturbação fixa e variável(parte 1).	52
5.2	Resultados comparativos em relação ao número de viagens, tempo computacional médio e <i>Gap</i> entre ILS-RVND com perturbação fixa e variável(parte 2).	52
5.3	Diferença entre soluções com todas estruturas de vizinhança e exclusão individualizada (parte 1).	53
5.4	Diferença entre soluções com todas estruturas de vizinhança e exclusão individualizada (parte 2).	53
5.5	Resultados computacionais ILS-RVND para instâncias do SET1.	55
5.6	Resultados computacionais ILS-RVND para instâncias do SET2 com 10 clientes.	56
5.7	Resultados computacionais ILS-RVND para instâncias do SET2 com 15 clientes.	56
5.8	Resultados computacionais ILS-RVND para instâncias do SET2 com 30 clientes.	56
5.9	Resultados computacionais ILS-RVND para instâncias do SET2 com 40 clientes.	57
5.10	Resultados computacionais ILS-RVND para instâncias do SET3 com 3 hotéis.	58
5.11	Resultados computacionais ILS-RVND para instâncias do SET3 com 5 hotéis.	58
5.12	Resultados computacionais ILS-RVND para instâncias do SET3 com 10 hotéis.	59
5.13	Resultados computacionais ILS-RVND para instâncias do SET4.	60

5.14	Resultados computacionais para as instâncias do SET1.	63
5.15	Resultados computacionais para as instâncias do SET2 com 10 clientes. . .	63
5.16	Resultados computacionais para as instâncias do SET2 com 15 clientes. . .	63
5.17	Resultados computacionais para as instâncias do SET2 com 30 clientes. . .	64
5.18	Resultados computacionais para as instâncias do SET2 com 40 clientes. . .	64
5.19	Resultados computacionais para instâncias do SET3 com 3 hotéis.	65
5.20	Resultados computacionais para instâncias do SET3 com 5 hotéis.	65
5.21	Resultados computacionais para instâncias do SET3 com 10 hotéis.	66
5.22	Resultados computacionais para as instâncias do SET4.	66
5.23	Resumo dos resultados para todas as instâncias propostas por [46].	67
5.24	Resultados para instâncias com 5 hotéis, 50 clientes e tempo de viagem limitado a 150.	70
5.25	Resultados para instâncias com 5 hotéis, 50 clientes e tempo de viagem limitado a 200.	70
5.26	Resultados para instâncias com 10 hotéis, 50 clientes e tempo de viagem limitado a 100.	71
5.27	Resultados para instâncias com 10 hotéis, 50 clientes e tempo de viagem limitado a 150.	71
5.28	Resultados para instâncias com 10 hotéis, 50 clientes e tempo de viagem limitado a 200.	72
5.29	Resultados para instâncias com 20 hotéis, 50 clientes e tempo de viagem limitado a 100.	72
5.30	Resultados para instâncias com 20 hotéis, 50 clientes e tempo de viagem limitado a 150.	72
5.31	Resultados para instâncias com 20 hotéis, 50 clientes e tempo de viagem limitado a 200.	73
5.32	Resultados para instâncias com 5 hotéis, 75 clientes e tempo de viagem limitado a 150.	73

5.33 Resultados para instâncias com 5 hotéis, 75 clientes e tempo de viagem limitado a 200.	73
5.34 Resultados para instâncias com 10 hotéis, 75 clientes e tempo de viagem limitado a 100.	74
5.35 Resultados para instâncias com 10 hotéis, 75 clientes e tempo de viagem limitado a 150.	74
5.36 Resultados para instâncias com 10 hotéis, 75 clientes e tempo de viagem limitado a 200.	74
5.37 Resultados para instâncias com 20 hotéis, 75 clientes e tempo de viagem limitado a 100.	75
5.38 Resultados para instâncias com 20 hotéis, 75 clientes e tempo de viagem limitado a 150.	75
5.39 Resultados para instâncias com 20 hotéis, 75 clientes e tempo de viagem limitado a 200.	75
5.40 Resultados para instâncias com 5 hotéis, 100 clientes e tempo de viagem limitado a 150.	76
5.41 Resultados para instâncias com 5 hotéis, 100 clientes e tempo de viagem limitado a 200.	76
5.42 Resultados para instâncias com 10 hotéis, 100 clientes e tempo de viagem limitado a 100.	77
5.43 Resultados para instâncias com 10 hotéis, 100 clientes e tempo de viagem limitado a 150.	77
5.44 Resultados para instâncias com 10 hotéis, 100 clientes e tempo de viagem limitado a 200.	77
5.45 Resultados para instâncias com 20 hotéis, 100 clientes e tempo de viagem limitado a 100.	78
5.46 Resultados para instâncias com 20 hotéis, 100 clientes e tempo de viagem limitado a 150.	78
5.47 Resultados para instâncias com 20 hotéis, 100 clientes e tempo de viagem limitado a 200.	78

5.48	Resumo das comparações entre o Algoritmo BCP e a MILS.	79
5.49	Comparação entre ILS-RVND com e sem procedimento de mineração de dados para o SET1 [46].	81
5.50	Comparação entre ILS-RVND com e sem procedimento de mineração de dados para o SET4 [46].	81
5.51	Comparação entre ILS-RVND com e sem procedimento de mineração de dados para todos os grupos propostos por [4].	82
5.52	Resultados obtidos pelo método híbrido para o SET1.	84
5.53	Resultados obtidos pelo método híbrido para o SET2 com 10 clientes. . .	85
5.54	Resultados obtidos pelo método híbrido para o SET2 com 15 clientes. . .	85
5.55	Resultados obtidos pelo método híbrido para o SET2 com 30 clientes. . .	86
5.56	Resultados obtidos pelo método híbrido para o SET2 com 40 clientes. . .	86
6.1	Parâmetros utilizados para gerar as instâncias em [6].	88
6.2	Média dos <i>Gaps</i> (%) encontrados pela MSRVND.	89
6.3	Média dos <i>Gaps</i> (%) encontrados por [6].	90
6.4	Média dos <i>Gaps</i> (%) para instâncias clusterizadas.	90
6.5	Média dos <i>Gaps</i> (%) para instâncias randomizadas.	90
6.6	Porcentagem de soluções ótimas encontradas pela MSRVND.	92
6.7	Porcentagem de soluções ótimas encontradas por [6].	92
6.8	Média de tempo computacional para a heurística MSRVND.	93
6.9	Média de tempo computacional para a heurística proposta por [6].	93
6.10	Média dos <i>Gaps</i> (%) encontrados pela MSRVND	94
6.11	Média dos <i>Gaps</i> (%) para instâncias clusterizadas.	94
6.12	Média dos <i>Gaps</i> (%) para instâncias randomizadas.	95
6.13	Porcentagem de soluções ótimas encontradas pela MSRVND.	96
6.14	Média de tempo computacional para a heurística MSRVND (em segundos). .	96

Lista de Abreviaturas e Siglas

BCP	: <i>Branch and Cut-and-Price</i> ;
EA-ILS	: <i>Efficient Adaptive Iterated Local Search</i> ;
EST	: Estratégia Utilizada;
FIM	: <i>Frequent Itemset Mining</i> ;
GAP	: <i>Gain Average Percentage</i> ;
GRASP	: <i>Greedy Randomized Adaptive Search Procedure</i> ;
HDM	: <i>Hybrid Dynamic programming and Memetic search</i> ;
ILS	: <i>Iterated Local Search</i> ;
ILS-RVND	: <i>ILS Random Variable Neighborhood Descent</i> ;
IRACE	: <i>Iterated Racing for Automatic Algorithm Configuration</i> ;
LC	: Lista de Clientes;
LK	: Heurística de <i>Lin-Kernighan</i> ;
LRC	: Lista Restrita de Candidatos;
MA	: <i>Memetic Algorithm</i> ;
MILS	: <i>Miner Iterated Local Search</i> ;
MSC	: Melhor Solução Conhecida;
MSRVND	: <i>Multi-Start Random Variable Neighborhood Descent</i> ;
PCV	: Problema do Caixeiro Viajante;
PCVSH	: Problema do Caixeiro Viajante com Seleção de Hotéis;
PCVM	: Problema do Caixeiro Viajante Múltiplo;
PCVMSH	: Problema do Caixeiro Viajante Múltiplo com Seleção de Hotéis;
PCVMJTSH	: PCV com Múltiplas Janelas de Tempo e Seleção de Hotéis;
PLS	: <i>Perturbation Local Search</i> ;
PRV	: Problema de Roteamento de Veículos;
VND	: <i>Variable Neighborhood Descent</i> ;
VNS	: <i>Variable Neighborhood Search</i> ;
RVND	: <i>Random Variable Neighborhood Descent</i> ;
TC	: Tempo Computacional;
TTotal	: Tempo Total Gasto;

Sumário

1	Introdução	1
1.1	Motivação	1
1.2	Objetivo da Tese	3
1.3	Organização da Tese	4
2	Descrição dos problemas	5
2.1	O Problema do Caixeiro Viajante com Seleção de Hotéis (PCVSH)	5
2.1.1	Estruturas de vizinhança	6
2.1.1.1	<i>Intra-trips</i>	6
2.1.1.2	<i>Inter-trips</i>	7
2.1.2	Revisão da literatura	9
2.1.3	Definição formal	14
2.2	O Problema do Caixeiro Viajante Múltiplo com Seleção de Hotéis (PCVMSh)	18
2.2.1	Estruturas de vizinhança	19
2.2.1.1	<i>Inter-trips</i>	19
2.2.1.2	<i>Inter-tours</i>	20
2.2.2	Revisão da literatura	21
2.2.3	Definição formal	22
3	Heurísticas para o PCVSH	25
3.1	Heurística Iterated Local Search	25
3.1.1	Solução inicial	28

3.1.2	Busca local	30
3.1.3	Estruturas de vizinhança	32
3.1.4	Perturbação	32
3.2	Hibridização com Mineração de dados	34
3.3	Hibridização com Mineração de dados com método exato	38
4	Heurística Multi-Start Local Search para o PCVMSH	40
4.1	Construção de soluções	41
4.2	Busca local	44
4.3	Estruturas de vizinhança	46
5	Experimentos Computacionais para o PCVSH	48
5.1	Resultados para instâncias propostas por [46]	48
5.1.1	Impacto da perturbação variável e das estruturas de vizinhança na heurística ILS-RVND	50
5.1.2	Resultados obtidos pela heurística ILS-RVND	53
5.1.3	Resultados obtidos pela heurística MILS	60
5.2	Resultados para instâncias propostas por [4]	68
5.3	Análise de sensibilidade da mineração de dados	80
5.4	Resultados para hibridização entre MILS e Método Exato	82
6	Experimentos Computacionais para o PCVMSH	87
6.1	Resultados para instâncias testadas em [6]	89
6.2	Resultados para instâncias não testadas em [6]	94
7	Conclusões e Trabalhos Futuros	97
7.1	Publicações	100
7.2	Trabalhos futuros	100

Capítulo 1

Introdução

1.1 Motivação

O transporte de produtos e o fornecimento de serviços são atividades cruciais para promoção do acesso a itens essenciais que são consumidos diariamente pela população, a nível mundial. Portanto, transportar é importante, mas transportar utilizando os recursos disponíveis de forma otimizada, sejam eles pessoais, energéticos ou financeiros, torna-se essencial. No Brasil, devido a sua extensão territorial e, por questões políticas e financeiras, optou-se pela utilização, prioritariamente, de rodovias. Devido aos altos custos que incidem sobre o transporte rodoviário, a pesquisa por meios que levem à economia de recursos torna-se imprescindível, pois a variação nos custos implica diretamente no repasse aos consumidores finais, afetando assim, aqueles mais vulneráveis economicamente.

A definição de trajetos que sejam otimizados do ponto de vista da distância ou do tempo percorrido pode auxiliar na redução dos custos logísticos, reduzindo assim o custo final daquele produto ou serviço que chega ao consumidor. Definir manualmente trajetos otimizados nem sempre é uma tarefa fácil, visto que o número total de combinações de trajetos possíveis pode crescer exponencialmente. Neste sentido, uma forma de otimizar tais trajetos é modelando-os como um problema de otimização combinatória.

A representação de problemas logísticos como problemas de otimização combinatória vem sendo aplicada por várias décadas. Tais representações se aplicam a problemas de estocagem, manufatura, transporte, consumo e muitas outras subáreas. O Problema do Caixeiro Viajante (PCV) é um clássico problema de otimização combinatória, e foi um dos primeiros a tratar o planejamento de trajetos, sendo até hoje amplamente estudado devido a sua importância teórica e aplicabilidade prática [2]. No PCV, o caixeiro precisa

definir um trajeto que inicia em sua residência/local de trabalho, visita um conjunto de clientes exatamente uma única vez e retorna ao seu local de origem. Neste caso, não há restrição diária de tempo ou distância percorrida. O objetivo é encontrar o melhor trajeto de atendimento aos clientes, de forma que o uso de recursos como tempo ou distância sejam minimizados.

Durante décadas de estudos, pesquisadores vêm sugerindo novas variantes para o PCV capazes de modelar características específicas de cada problema logístico. Nesse sentido, uma das variantes existentes é o Problema do Caixeiro Viajante com Seleção de Hotéis (PCVSH) [46]. No PCVSH, um determinado número de clientes precisa ser atendido, assim como no PCV. Entretanto, o problema considera uma limitação diária de trabalho para o caixeiro, logo, é necessário realizar paradas estratégicas em hotéis durante a realização de um percurso. Cada cliente só pode ser visitado uma única vez e o percurso de um novo dia de trabalho deve começar no hotel onde o percurso do dia anterior se encerrou. O objetivo a ser alcançado no PCVSH é a minimização do número total de dias necessários para o atendimento de todos os clientes e a minimização do tempo total gasto.

Por critérios de simplificação e padronização de nomenclatura, será utilizado o termo **viagem** para representar o percurso a ser realizado durante um dia de trabalho e o termo **rota** para definir o conjunto de viagens necessárias para atender a todos os clientes.

Após o surgimento do PCVSH, foi proposta uma variante conhecida como Problema do Caixeiro Viajante Múltiplo com Seleção de hotéis (PCVMSH)[6]. Tal variante compartilha características tanto do PCVSH, quanto do Problema do Caixeiro Viajante Múltiplo (PCVM)[5], que por sua vez é uma variante do clássico PCV. No PCVM, os clientes também são atendidos uma única vez, porém, existe um número previamente definido de caixeiros responsáveis por atender a todos os clientes, dividindo assim entre eles o trabalho que era realizado somente por um caixeiro no PCV. Como definido no PCVSH, a rota de um caixeiro no PCVMSH é dividida em viagens que não podem exceder um limite de tempo. O objetivo do problema é definido como sendo a construção de um conjunto de rotas limitadas por um número máximo de viagens, a serem associadas a cada um dos caixeiros disponíveis, de forma que o tempo total gasto em todas as rotas seja minimizado.

Considerando que o PCVSH pode ser facilmente transformado no PCV, simplesmente eliminando a restrição de limite de tempo de uma viagem, pode-se afirmar que o PCVSH é no mínimo tão difícil de se resolver quanto o PCV. Consequentemente, o problema base aqui tratado pode ser caracterizado como \mathcal{NP} -difícil [42]. Até o momento, não existe

uma solução computacional que consiga resolver o problema e sua variante apresentada de forma ótima considerando instâncias de tamanhos variados. A utilização de métodos exatos baseados em diferentes tipos de formulações para tentar obter a solução ótima acaba por ser impraticável na maioria dos casos, devido ao elevado tempo computacional demandado em instâncias de tamanho médio e grande. Uma forma de contornar tal limitação é utilizar métodos heurísticos, os quais permitem a obtenção de uma solução de boa qualidade, consumindo um tempo computacional aceitável para uso na prática.

O PCVMSH, por sua vez, possui complexidade no mínimo igual a seus antecessores, pois existe a limitação do número máximo de viagens por rota, fato não observado para os demais. Por não existir limite de viagens no PCVSH, a viabilidade de uma rota pode ser alcançada simplesmente adicionando mais viagens [6], enquanto para o PCVMSH essa adição ilimitada pode significar a inclusão de inviabilidade numa solução. Quanto mais restritiva for a limitação estabelecida para o número de viagens em uma rota, mais complexa se torna a criação de rotas viáveis. A troca de clientes entre rotas distintas é uma maneira simples e eficaz para lidar com a limitação do número de viagens. Entretanto, essas trocas demandam tempo computacional considerável, tempo este que no PCVSH é utilizado para explorar de forma mais abrangente o espaço de busca.

1.2 Objetivo da Tese

O objetivo principal desta tese consiste em desenvolver heurísticas para resolver o PCVSH e o PCVMSH.

Os objetivos específicos desta tese são listados a seguir:

- Apresentar uma revisão dos trabalhos existentes na literatura que tratam do PCVSH e PCVMSH;
- Propor uma prova formal da complexidade do PCVSH;
- Propor uma heurística para o PCVSH baseada nas metaheurísticas *Iterated Local Search* (ILS) e *Variable Neighborhood Descent* (VND) que seja capaz de alcançar resultados melhores que os já reportados para a maioria das instâncias propostas por [46];
- Aprimorar a capacidade de melhora das soluções da heurística proposta por meio de mineração de padrões recorrentes;

- Resolver de forma heurística os novos conjuntos de instâncias recentemente propostos para o PCVSH [4];
- Apresentar uma heurística para resolução do PCVMSH.

1.3 Organização da Tese

O restante da estrutura de apresentação deste trabalho está organizada da seguinte forma. No Capítulo 2 são apresentados o Problema do Caixeiro Viajante com Seleção de Hotéis e o Problema do Caixeiro Viajante Múltiplo com Seleção de Hotéis juntamente com um resumo de todos os trabalhos publicados acerca dos problemas; o Capítulo 3 descreve a heurística base proposta para o PCVSH juntamente com as técnicas internamente utilizadas e as estratégias de Hibridização do PCVSH utilizando mineração de padrões recorrentes e método exato; a heurística proposta para o PCVMSH e seus componentes são descritos no Capítulo 4; os experimentos computacionais relativos às heurísticas propostas para o PCVSH e seus resultados são detalhados no Capítulo 5; os resultados alcançados pela heurística desenvolvida para resolver o PCVMSH são apresentados no Capítulo 6. Por fim, concluindo o trabalho, no Capítulo 7 são apresentadas as conclusões e sugestões para trabalhos futuros.

Capítulo 2

Descrição dos problemas

Neste Capítulo serão detalhados o problema principal (PCVSH) e sua variante (PCVMSH) tratados nesta tese, apresentando uma definição formal e matemática para os problemas, bem como as soluções previamente propostas na literatura para solucioná-los. As informações aqui expostas foram parcialmente extraídas dos resultados publicados em [42, 43].

2.1 O Problema do Caixeiro Viajante com Seleção de Hotéis (PCVSH)

Como introduzido no Capítulo anterior, o PCVSH é um problema derivado do clássico PCV, sendo este último conhecido e estudado por diversos pesquisadores nas últimas décadas. De forma simples, é possível também relacionar o problema aqui estudado com outro problema clássico, o Problema de Roteamento de Veículos (PRV). Assim como no PRV, para o qual existe um depósito de onde os veículos iniciam e terminam sua jornada de trabalho, no PCVSH existe um hotel inicial de onde o caixeiro deve partir, tendo obrigatoriamente que retornar a este hotel ao término do atendimento de seus clientes.

É possível fazer uma correlação entre teoria e a prática vivenciada por empresas em relação ao PCVSH, considerando por exemplo, uma empresa que necessita entregar cargas a diferentes clientes geograficamente dispersos, sejam eles em uma mesma cidade ou cidades distintas. Tal exemplo mostra que é possível construir uma rota para um funcionário, aqui correlacionado ao caixeiro, na qual não será possível atender a todos os clientes em um mesmo dia de trabalho, obrigando então que ele faça uso de um serviço de hospedagem em algum hotel disponível nas proximidades em que seu expediente diário se encerrou.

Outras aplicações do mundo real podem ser mapeadas no PCVSH, como a definição de rotas para veículos elétricos que necessitem realizar paradas em lugares específicos para efetuar recarga, a escolha de melhores pontos de parada para caminhões que só podem trafegar durante o dia ou que necessitem de áreas específicas de parada, e outras aplicações que requerem o particionamento do serviço em vários dias de trabalho.

O PCVSH é um problema de otimização combinatória, no qual é atribuído ao caixeiro viajante a tarefa de visitar cada um dos clientes uma única vez, sendo permitido a ele a realização de paradas estratégicas em hotéis ao fim de cada jornada de trabalho. A rota do caixeiro inicia-se em um local conhecido a priori, sendo composta por visitas aos clientes e a hotéis com o intuito de garantir que o tempo de trabalho diário não extrapole um tempo máximo permitido. O problema permite que, caso um hotel seja escolhido para o caixeiro pernoitar, futuramente esse mesmo hotel possa vir a ser escolhido novamente, assim como também é permitido ao caixeiro utilizar o hotel de início/término da rota como um hotel intermediário. O objetivo do problema consiste na definição otimizada de uma rota onde prioritariamente o número de viagens utilizadas seja o menor possível e, complementarmente, o tempo total gasto incluindo tempo de viagem e visita a clientes também seja minimizado.

2.1.1 Estruturas de vizinhança

As estratégias para resolução do PCVSH a serem apresentadas no decorrer desta seção fazem uso de estruturas de vizinhança que são tradicionalmente aplicadas ao problema. Essas estruturas são classificadas em *Intra-trips* e *Inter-trips*. *Intra-trips* são estruturas que impactam apenas uma viagem de cada vez durante sua aplicação; por outro lado, as *inter-trips* alteram a composição de duas viagens distintas simultaneamente.

Em todas as figuras de descrição dos movimentos, no canto superior esquerdo é apresentada a solução de partida, seguida pelo exemplo de aplicação do movimento no canto superior direito e pela solução final obtida, centralizada abaixo.

2.1.1.1 *Intra-trips*

A estrutura *2-opt* é uma clássica estrutura de vizinhança *intra-trip* proposta por [14], onde uma solução vizinha é obtida por meio da remoção de duas arestas distintas que fazem parte de uma mesma viagem, seguido pela inversão da ordem de visita dos clientes compreendidos entre as duas arestas removidas e, por fim, reconectando novamente os

extremos que foram desconectados. No exemplo ilustrado na Figura 2.1, as arestas $(4, 10)$ e $(7, H_3)$ são removidas e as arestas $(4, 7)$ e $(10, H_3)$ são criadas, revertendo a ordem de visitação do caminho entre os clientes 10 e 7.

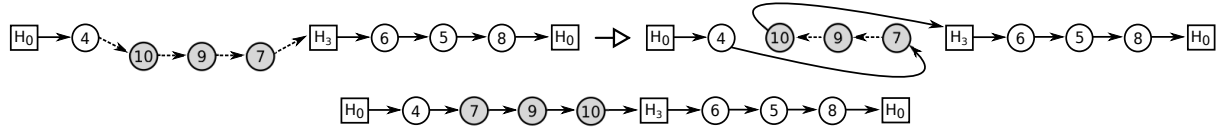


Figura 2.1: Movimento realizado na estrutura *2-opt*.

A estrutura de vizinhança *Or-opt* foi proposta por [32] e é aplicada ao PCVSH como um movimento *intra-trip*. Nesta estrutura, uma solução vizinha é obtida por meio da realocação de um conjunto de n nós adjacentes para outra posição distinta dentro da mesma viagem, onde $n \in \{1, 2, 3\}$. Os movimentos são executados sobre uma viagem por meio da aplicação desta estrutura enquanto houver melhorias. Caso contrário, os movimentos são aplicados à próxima viagem presente na solução que está sendo otimizada. A Figura 2.2, ilustra um exemplo onde $n = 2$ e os clientes 4 e 10 devem ser realocados para outra posição. As arestas $(H_0, 4)$, $(10, 9)$ e $(9, 7)$ são removidas e as arestas $(H_0, 9)$, $(9, 4)$ e $(10, 7)$ são criadas, mantendo a ordem original dos clientes envolvidos no procedimento. Todas as combinações possíveis de movimentos são exploradas e somente aquela que melhor otimiza a solução corrente é aplicada.

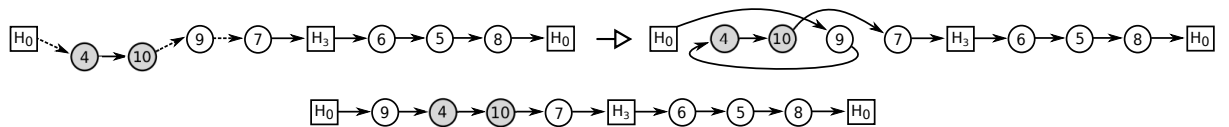
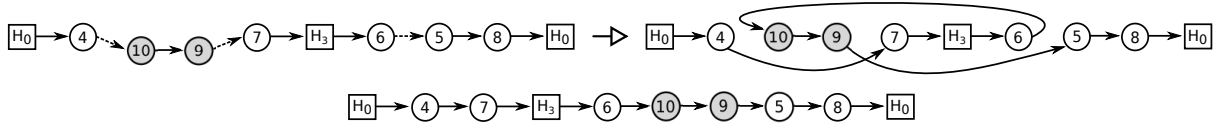


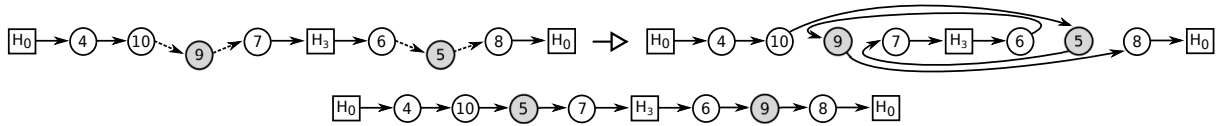
Figura 2.2: Movimento realizado na estrutura *Or-opt*.

2.1.1.2 *Inter-trips*

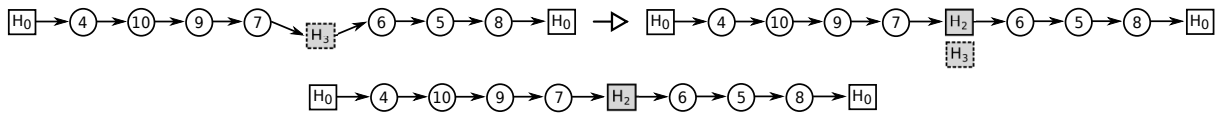
Apesar da semelhança em relação à estrutura *Or-opt*, a estrutura de vizinhança *Relocate* é classificada como *inter-trip* [9, 16, 46]. Seu funcionamento se baseia na obtenção de uma solução vizinha por meio da realocação de um grupo adjacente de clientes com tamanho $n \in \{1, 2, 3\}$, de uma viagem para outra. A principal ideia desta estrutura é possibilitar a gestão do número de clientes contido nas viagens, viabilizando assim, o alcance de viabilidade geral da solução. Um exemplo de aplicação é ilustrado na Figura 2.3, onde os clientes 10 e 9 são realocados por meio da remoção das arestas $(4, 10)$ e $(9, 7)$ da primeira viagem e $(6, 5)$ da segunda viagem. Para conclusão do procedimento, são criadas as arestas $(4, 7)$, $(6, 10)$ e $(9, 5)$, respectivamente.

Figura 2.3: Movimento realizado na estrutura *Relocate*.

Na estrutura de vizinhança *Exchange*, uma solução vizinha é obtida por meio da troca de um conjunto de n clientes adjacentes entre duas viagens distintas, onde $n \in \{1, 2, 3\}$. Exemplificando a estrutura com $n = 1$ na Figura 2.4, o nó 9 de uma viagem é trocado com o nó 5 da outra viagem. Cada movimento desta estrutura de vizinhança requer, simultaneamente, a remoção e criação de um total de oito arestas. Similar ao que é realizado em *Relocate*, apenas o melhor movimento de troca é executado.

Figura 2.4: Movimento realizado na estrutura *Exchange*.

Os autores em [9] observaram que a escolha dos hotéis intermediários que conectam viagens distintas impacta diretamente na qualidade final da solução. Desta forma, propuseram a estrutura de vizinhança *ChangeHotels* para gerar soluções vizinhas por meio da substituição de um hotel intermediário por algum outro disponível. Todas as possíveis trocas são avaliadas e a que melhor otimizar a solução corrente é efetuada. No exemplo ilustrado na Figura 2.5, o hotel intermediário H_3 é substituído pelo hotel H_2 .

Figura 2.5: Movimento realizado na estrutura *ChangeHotels*.

Com o propósito de encontrar soluções vizinhas que sejam capazes de reduzir o número de viagens necessárias para atender a todos os clientes, foi proposta a estrutura de vizinhança *JoinTrips* [46]. Seu objetivo é agrupar duas viagens que compartilham o mesmo hotel se o tempo total gasto não exceder o limite definido para uma única viagem. O exemplo da Figura 2.6 ilustra uma solução original à esquerda e a solução obtida à direita após a exclusão do hotel intermediário H_3 . Esta estrutura *inter-trip* é a única capaz de reduzir o número de viagens de uma solução em tempo de execução do algoritmo e, diretamente, favorecer o primeiro objetivo do PCVSH, que é minimizar o número de viagens.

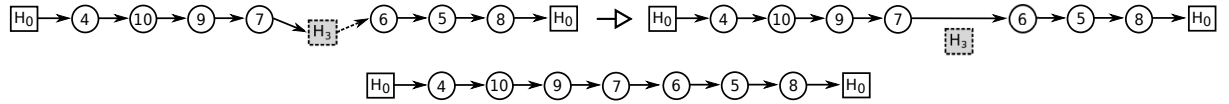


Figura 2.6: Movimento realizado na estrutura *JoinTrips*.

2.1.2 Revisão da literatura

O PCVSH foi proposto originalmente em [46] onde foram destacadas as principais características e diferenças em relação a outros problemas como o Problema de Roteamento de Veículos, Problema do Caixeiro Viajante Múltiplo, Problema de Roteamento e Alocação e Problema de Roteamento de Veículos com Múltiplos Depósitos. Dentre as contribuições apresentadas nesse primeiro trabalho, está a modelagem deste problema como um modelo de Programação Linear Inteira Mista. Tal formulação utiliza um número fixo de viagens informado no momento da aplicação do modelo. Caso a resolução do modelo não retorne uma solução viável, esse número é incrementado até se obter uma solução viável. Os autores ainda citam uma alternativa para modificação da função objetivo da formulação, onde primeiro seria minimizado o número de viagens e depois o tempo total gasto. Para prevenir a formação de subciclos, foram formuladas restrições de acordo com as formulações de *Miller-Tucker-Zemlin* aplicadas ao PCV.

Além da formulação, foram propostas duas heurísticas, as quais se diferenciam pela forma como geram a solução inicial. Ambas heurísticas buscam primeiro minimizar o número de viagens e depois o tempo total gasto. Os métodos de inicialização são:

- *Initialization I1* - Baseado no princípio do vizinho mais próximo, onde os clientes e hotéis são conectados sempre aos vizinhos geograficamente mais próximos, de forma que, ao final, cada viagem permaneça viável;
- *Initialization I2* - Resolve o problema sem considerar o limite de tempo das viagens (PCV), aplica as estruturas de vizinhança *2-Opt* e *Or-Opt* e divide a rota em viagens viáveis observando a ordem previamente definida para o PCV.

Utiliza-se um procedimento de busca local para melhoria da solução aplicando a estrutura *2-Opt*, seguida pela estrutura *Re-Opt*, que corresponde a uma realocação de clientes. A estrutura *Re-opt* agrupa os movimentos realizados nas estruturas *Or-opt* e *Relocate*. Também busca-se melhorar a solução trocando hotéis intermediários e extinguindo viagens, desde que seja possível remanejar seus clientes para alguma das outras viagens disponíveis. A melhor dentre todas as soluções encontradas é considerada. Caso não haja

melhoria da solução, o algoritmo é finalizado, caso contrário, o processo de busca local é reiniciado.

A primeira metaheurística para tratar o problema PCVSH foi apresentada em [7]. Os autores propuseram uma heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) em conjunto com um procedimento de busca local *Variable Neighborhood Descent* (VND). Foram utilizadas diferentes estruturas de vizinhança amplamente aplicadas em problemas da literatura: *2-opt*, *3-opt*, *Or-opt*, *Relocate*, *Exchange*, *ChangeHotels* e *JoinTrips*. Os resultados encontrados foram superiores àqueles contidos no trabalho previamente descrito em [46].

Posteriormente em [8] foi proposto um método para encontrar a solução exata baseado numa formulação de Programação Inteira, e foi desenvolvido um Algoritmo Memético. O método exato difere daquele apresentado em [46] por utilizar uma função objetivo ponderada de acordo com os objetivos do PCVSH e por utilizar restrições mais eficientes (*Dantzig-Fulkerson-Johnson*) para evitar que sejam formados subciclos na rota. A metaheurística proposta combina Algoritmo Genético com Busca Tabu, a qual obteve resultados ótimos para instâncias de tamanho pequeno e soluções de alta qualidade para quase todas as demais instâncias em relação aos dois primeiros trabalhos [7, 46]. A metaheurística tem como característica principal a operação em dois níveis: seleção de hotéis (Algoritmo Genético) e roteamento (Busca Tabu). Dois métodos de construção de soluções iniciais foram utilizados:

- *C1* - Define uma solução para o PCV por meio da aplicação da Heurística de *Lin-Kernighan* [24]. Depois, esta rota é dividida em viagens viáveis pelo método de divisão de *Prins* [34]. O método retorna uma rota viável para o PCVSH.
- *C2* - Inicia a construção a partir de um conjunto de viagens vazias e adiciona os clientes um a um no local onde causem o menor acréscimo no tempo total da solução. Posteriormente, aplica a estrutura de vizinhança *3-Opt* [23] que é uma variante da estrutura *2-opt*, a qual remove três arestas da viagem reconectando-as de forma a se obter uma solução distinta. Neste caso, o método não garante a viabilidade da solução.

A população inicial do Algoritmo Memético é constituída por uma solução gerada por *C1* (determinístico) e as demais geradas pelo método *C2* (estocástico). A cada iteração do Algoritmo Memético, quatro indivíduos são selecionados aleatoriamente a partir da população, comparados, e os dois melhores são escolhidos para serem os pais

da próxima geração. Um operador de cruzamento de um ponto é aplicado e dois novos descendentes são gerados. Para garantir que as novas soluções possuam diversidade, é calculada uma medida de distância entre as soluções e, enquanto essa medida não for satisfatória, mutações são feitas na prole. Quando a medida de distância é considerada aceitável, é aplicado um procedimento de Busca Tabu para melhorar as soluções. Se a nova solução atender ao critério de substituição (for melhor que a pior existente na população atual), ocorre a atualização da população. Este procedimento é realizado para as proles geradas a cada iteração do algoritmo. Na Busca Tabu, são utilizadas as seguintes estruturas de vizinhança na busca local: *3-Opt*, *Relocate*, *Exchange* e *JoinTrips*. Esses operadores já tinham sido utilizados em trabalhos anteriores envolvendo o PCVSH.

Similarmente ao trabalho desenvolvido em [8], os autores em [39] propõem uma comparação de abordagens heurísticas baseadas em Algoritmo Memético. Foram propostas duas abordagens, ambas tendo como base o Algoritmo Memético, variando a heurística utilizada para realização da busca local, sendo Busca Tabu e *Random Variable Neighborhood Descent* (RVND) [30, 44]. Duas heurísticas foram utilizadas na construção da solução inicial, a primeira de forma determinística assim como feito em [8] e a segunda estocástica utilizando uma heurística de Inserção Mais Barata, estrutura *3-Opt* e estratégia de divisão de rota baseada no algoritmo de *Dijkstra* [15]. A população de indivíduos utilizada no Algoritmo Memético foi ordenada da melhor para pior e dividida em três classes: A (20%) definida como elite, B (65%) como intermediária e C (15%) como pior. Para definição da nova população, 65% dos indivíduos são gerados a partir do cruzamento da população atual, sempre garantindo que haja um membro da elite a cada cruzamento. O restante 35% é gerado pelo procedimento de mutação da solução realizado a partir da aplicação de um movimento da estrutura *4-Opt*. Essa estrutura de vizinhança possui características similares à *2-opt*, entretanto, ocorre a geração de uma nova solução por meio da reconexão de 4 partes da solução original, derivadas da exclusão de 4 arestas distintas e não consecutivas. O procedimento Busca Tabu utilizado é baseado em [8] e o RVND adaptado de trabalhos sobre Roteamento de Veículos. As estruturas de vizinhança utilizadas foram: *2-Opt*, *Relocate*, *Exchange* e *JoinTrips*.

Em [9], é apresentada uma abordagem heurística para obter, rapidamente, soluções heurísticas de alta qualidade. A estratégia apresentada para resolver o problema é uma heurística ILS com busca local VND composta por dois operadores de perturbação da solução (clientes e hotéis) acoplados a um mecanismo de melhora da solução. A construção da solução inicial é realizada em duas etapas, conhecidas como *Order-first Split-second*. Inicialmente uma solução é gerada para o PCV através da Heurística de *Lin-Kernighan*

e depois se particiona a solução gerada utilizando um método inspirado no trabalho de *Prins* [34]. Após a criação da solução inicial de forma determinística, a cada iteração, é aplicada a perturbação de clientes enquanto for possível melhorar a solução com o VND. Quando a perturbação de clientes não surtir mais efeito, a perturbação de hotéis é aplicada também em conjunto com o VND. Além dos dois operadores de perturbação, foram utilizadas as seguintes estruturas de vizinhança no trabalho: *2-Opt*, *Or-Opt*, *Relocate*, *Exchange*, *ChangeHotels* e *JoinTrips*.

Nesse mesmo trabalho, os autores criaram uma nova formulação matemática para o PCVSH baseada em *Set-Partitioning*. Tal formulação difere da apresentada em [8] por ser baseada em viagens, ao invés de arcos, permitindo assim a adaptação para métodos de geração de colunas e *branch-and-price*.

Em [40], foi proposta uma modificação na formulação matemática apresentada em [8]. Diferentemente desse modelo proposto, onde a eliminação de subciclos era realizada com base em subconjuntos, o que gerava um número exponencial de restrições, neste modelo de Programação Linear Inteira foi utilizada uma estratégia baseada em fluxo, configurando assim um número polinomial de variáveis e restrições. Além da alteração do modelo, foi proposta uma solução heurística que combina fundamentos já utilizados nos trabalhos anteriores com uma metaheurística *Variable Neighborhood Search* (VNS) [31]. A construção da solução inicial considera o método *Order-first Split-second* apresentado em [9], porém, com uma variação no processo de divisão da rota em viagens viáveis, quando considera uma adaptação do algoritmo de *Dijkstra* [15]. Foi utilizada uma perturbação que realiza trocas dos clientes de uma rota, escolhendo o cliente a ser trocado e a nova posição de destino de forma aleatória. Em relação às estruturas de vizinhança implementadas no procedimento de busca local, foram utilizadas as mesmas citadas no trabalho desenvolvido em [9].

Em [35], os autores propuseram apenas um método de solução exata para o PCVSH. A principal contribuição foi a formulação de um modelo de Programação Linear Inteira utilizando o *Framework Tropical Mathematics*. A solução proposta somente foi aplicada a um subgrupo das instâncias disponíveis.

No trabalho [26] foi apresentada uma abordagem de programação dinâmica para encontrar a sequência ótima de hotéis a serem visitados numa rota. Também foi proposto um Algoritmo Memético com três operadores de cruzamento dedicados à recombinação das soluções, uma regra de seleção de operador de cruzamento adaptativa e um procedimento de refinamento da solução de duas fases, que alterna entre soluções viáveis e

inviáveis. O procedimento para geração de soluções iniciais para compor a população foi similar ao apresentado em [8]. O primeiro operador de cruzamento considerado é baseado na troca de uma subsequência de hotéis, enquanto que, o segundo e terceiro operadores trocam subsequências de hotéis e clientes. Um operador de cruzamento é aplicado de acordo com uma probabilidade que é calculada considerando a quantidade de vezes que aquele operador foi utilizado.

O procedimento de refinamento da solução é composto por oito estruturas de vizinhanças, que apesar de terem nomes distintos daqueles apresentados em [8], realizam o mesmo tipo de processamento da solução, que são: *Single insertion*, *Two insertion*, *Three insertion*, *Single swap*, *Two swap*, *Three swap*, *Intra-trip 3-Opt* e *Join-trips*. Como são permitidas soluções inviáveis, que extrapolam o tempo máximo de uma viagem, foi utilizada uma regra similar à que é comumente encontrada em uma metaheurística Busca Tabu, a qual proíbe determinados passos que foram realizados recentemente, garantindo assim que a heurística consiga escapar de ótimos locais. Como critérios para renovação da população, foram utilizados dois métodos simples: (i) se a nova solução for diversa o suficiente das contidas na população atual e se for melhor que a pior conhecida, então ela substitui a pior solução, e (ii) se a sequência de hotéis da nova solução for idêntica à sequência de uma solução contida na população e esta nova solução possuir uma qualidade melhor, então ela substitui a solução da população. A execução do processo só é finalizada quando não há melhorias das soluções viáveis durante uma dada quantidade de iterações. Os resultados obtidos superaram a maioria dos resultados apresentados em todos os trabalhos prévios.

Barbosa e Uchoa [4] propuseram em seu trabalho sobre o PCVSH uma formulação de Programação Inteira com *Set Partitioning* e um algoritmo *Branch-Cut-and-Price* utilizando um algoritmo de *labeling* para a resolução do problema de *pricing*. Complementarmente, para resolução do problema, as restrições de eliminação de subciclo, *2-Path* e *Subset Row Cuts* foram separadas. O algoritmo proposto foi aplicado às instâncias criadas por [46] e alcançou bons resultados para aquelas que possuem até 225 clientes, o que até então nenhum trabalho na literatura tinha conseguido por meio do uso de métodos exatos. Com a justificativa de que as instâncias de [46] por vezes não refletiam condições encontradas no mundo real, um novo conjunto de 240 instâncias foi apresentado e resultados ótimos foram obtidos para algumas instâncias de até 100 clientes.

Com o intuito de sintetizar todos os trabalhos que abordam o PCVSH, realizamos um estudo de cunho exploratório apresentado em [43]. Tal estudo revelou que apesar

do PCVSH ser um problema relativamente recente, já existem vários trabalhos na literatura que apresentam soluções heurísticas e exatas na busca por soluções de qualidade. Concluimos, de acordo com os critérios de qualidade definidos, que o trabalho publicado por [26] apresenta maior completude e qualidade. Também verificamos, dentre outros resultados, que a geração da solução inicial para o problema possui um impacto direto na qualidade da solução e que a correta escolha da ordem de visita dos hotéis durante a rota apresenta relação direta com a qualidade da solução final.

Em [3], aborda-se o Problema do Caixeiro Viajante com Múltiplas Janelas de Tempo e Seleção de Hotéis (PCVMJTSH) que é um problema mais complexo e realístico que o PCVSH, por considerar além da seleção de hotéis, janelas de tempo para que os clientes possam ser atendidos. Ao calcular o custo da solução para o PCVMJTSH, são levados em consideração os custos da viagem, salários de funcionários (incluindo custos inerentes a possíveis horas extras), taxas de hospedagem em hotéis e taxas de penalidade por possível omissão de atendimento a clientes. Por ser o primeiro trabalho a tratar tal variante, apresenta contribuições singulares, como a criação de instâncias artificiais e reais, proposição de modelagem matemática e uma heurística. A modelagem matemática consiste em um modelo de Programação Linear Inteira Mista capaz de resolver instâncias consideradas de pequeno porte. A heurística cria uma solução inicial com base na heurística de Inserção Mais Barata, combinada com a técnica de Otimalidade de Pareto, com o intuito de definir qual o melhor local a se inserir um cliente/hotel na rota, respeitando as restrições do problema. O trabalho estabelece dois critérios para definir a ordem de visita dos clientes e hotéis na rota, respeitando as janelas de tempo e as regras de trabalho associadas ao problema, que podem aumentar o custo/tempo total de uma rota. Para aprimoramento da solução é utilizada apenas a estrutura de vizinhança *2-Opt*. Apesar de se tratar de uma variante, a solução proposta nesse trabalho também pode ser aplicada às instâncias do PCVSH, obtendo soluções competitivas em relação aos primeiros trabalhos que abordaram o tema.

2.1.3 Definição formal

Considerando que os objetivos do PCVSH são, em ordem prioritária, minimizar o número de viagens necessárias para atender a todos os clientes, e de forma secundária, minimizar o tempo total gasto, apresenta-se formalmente uma representação do problema.

Considere um grafo $G = (V, E)$, onde $V = H \cup C$, sendo H um grupo não vazio dos hotéis disponíveis para serem utilizados e C o grupo de clientes que devem ser visitados

exatamente uma vez. Cada uma das arestas é definida por $E = \{(i, j) | i, j \in V, i \neq j\}$ onde (i, j) representa uma conexão entre dois clientes, dois hotéis ou um cliente e um hotel. Para cada cliente $i \in C$ existe um tempo de visita τ_i associado e um tempo de visita $\tau_i = 0$, para cada hotel $i \in H$. O tempo a ser gasto c_{ij} para viajar da facilidade i para j é calculado por meio da distância euclidiana para cada uma das combinações de pares existente (por critérios de representatividade, o tempo de viagem é associado à distância). A rota deve obrigatoriamente iniciar em um hotel específico ($i = 0, i \in H$), sendo possível que esse hotel de partida seja utilizado como um hotel intermediário para conectar duas viagens. Caso seja necessário o uso de algum outro hotel ($i \neq 0, i \in H$), este pode ser utilizado para viabilizar o atendimento aos clientes. Para os casos em que um mesmo hotel esteja estrategicamente posicionado de forma a privilegiar um dado percurso, existe a possibilidade desse hotel ser utilizado mais de uma vez em uma mesma rota, demonstrando que uma solução para o PCVSH não pode ser expressa como um ciclo simples. De forma complementar, cada viagem deve iniciar e finalizar em um dos hotéis disponíveis. O tempo total viajado, incluindo os tempos de visita aos clientes, não pode exceder uma constante L pré-definida. Por fim, uma viagem deve iniciar a partir do hotel onde a viagem prévia finalizou. Neste contexto, o objetivo do PCVSH é diminuir prioritariamente o número total de viagens e em seguida o tempo total gasto. Os autores em [4, 8, 9, 40, 46] propuseram formulações matemáticas para o PCVSH.

A Figura 2.7 ilustra duas rotas. A primeira rota ilustrada na Figura 2.7 (a) foi construída para o PCV, a qual inicia o percurso no depósito, visita todos os clientes, e finaliza onde a rota se iniciou. A segunda, mostrada na Figura 2.7 (b) ilustra uma solução para o PCVSH contendo 4 viagens. Na figura, as viagens foram coloridas em tons diferentes para uma melhor diferenciação. A primeira viagem inicia no hotel H_0 , visita três clientes (8, 7, e 9) e finaliza no hotel H_3 . A segunda viagem inicia de onde a viagem prévia terminou e finaliza no hotel H_0 , após visitar dois clientes (10 e 11). Na terceira viagem, o percurso é iniciado do hotel H_0 e finalizado no H_5 . Por fim, a quarta e última viagem do caixeiro inicia no hotel H_5 , seguindo com a visita aos clientes 4, 5, e 6, quando a visita aos clientes é concluída e o retorno ao hotel H_0 é realizado. Este exemplo mostra que um hotel disponível pode ser usado mais de uma vez e que, em alguns casos, hotéis podem não ser utilizados devido à possibilidade de se atender os clientes sem a necessidade de realizar paradas, ou seja, sem violação da restrição que limita o tempo de uma viagem. Adicionalmente, se o caixeiro decidir inverter a ordem de visita aos clientes, iniciando do final da rota em direção ao início, o mesmo tempo total gasto será obtido.

A seguir é apresentada em detalhes a Formulação de Programação Inteira proposta

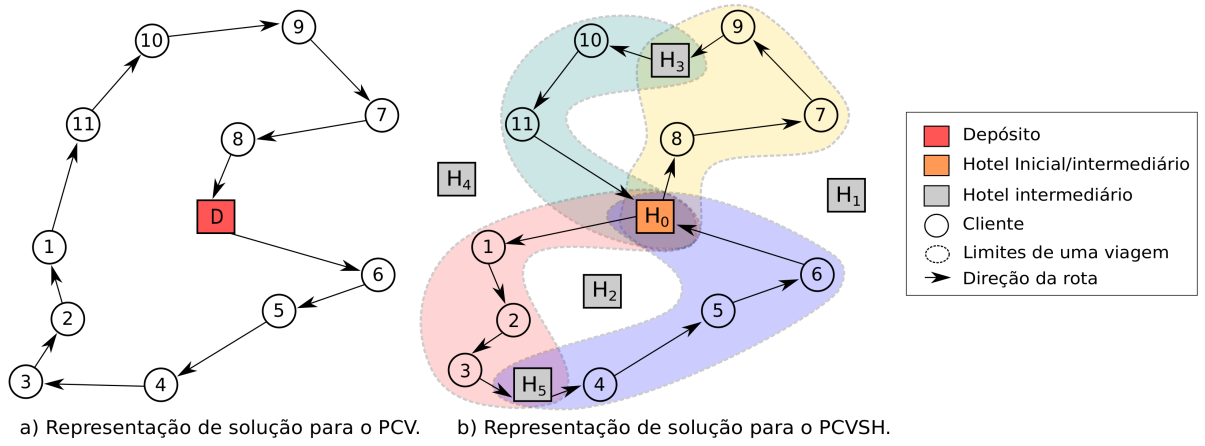


Figura 2.7: Representação de soluções para o PCV e PCVSH.

por [8]. Por se tratar de um problema de minimização, para priorizar rotas que contenham um menor número de viagens, uma constante M de valor grande o suficiente foi utilizada na função objetivo. Assim, a multiplicação do número de viagens por esta constante garante que rotas com menor número de viagens serão priorizadas.

A variável binária x_{ij}^d recebe o valor 1 se na d -ésima viagem, uma visita a um cliente ou hotel i é seguida pela visita a um cliente ou hotel j , ou o valor 0, caso contrário. A variável binária y^d recebe o valor 1 se na viagem d no mínimo um cliente ou hotel é visitado ou 0, caso contrário. Assim, y^d recebe o valor zero se não for necessária uma viagem no dia d . As variáveis y^d e x_{ij}^d são usadas na função objetivo, respectivamente, para minimizar o número de viagens e o tempo total gasto. Por fim, a constante D representa o número máximo de viagens contidas na solução e foi definida pela solução encontrada na heurística proposta em [8].

$$\min \quad M \sum_{d=1}^D y^d + \sum_{d=1}^D \left(\sum_{(i,j) \in E} c_{ij} x_{ij}^d \right) \quad (2.1)$$

$$s.a. \quad \sum_{d=1}^D \sum_{i \in V} x_{ij}^d = 1 \quad j \in C \quad (2.2)$$

$$\sum_{i \in V} x_{ij}^d = \sum_{i \in V} x_{ji}^d \quad j \in C, d = 1, \dots, D \quad (2.3)$$

$$\sum_{h \in H} \sum_{j \in V \setminus \{h\}} x_{hj}^d = y^d \quad d = 1, \dots, D \quad (2.4)$$

$$\sum_{h \in H} \sum_{i \in V \setminus \{h\}} x_{ih}^d = y^d \quad d = 1, \dots, D \quad (2.5)$$

$$\sum_{(i,j) \in E} (c_{ij} + \tau_j) x_{ij}^d \leq L \quad d = 1, \dots, D \quad (2.6)$$

$$\sum_{j \in V \setminus \{0\}} x_{0j}^1 = 1 \quad (2.7)$$

$$\sum_{i \in V \setminus \{0\}} x_{i0}^d \geq y^d - y^{d+1} \quad d = 1, \dots, D-1 \quad (2.8)$$

$$\sum_{i \in V} x_{ih}^d + y^d \geq \sum_{i \in V} x_{hi}^{d+1} + y^{d+1} \quad h \in H, d = 1, \dots, D-1 \quad (2.9)$$

$$\sum_{i \in V} x_{ih}^d - \sum_{i \in V} x_{hi}^{d+1} \leq 1 - y^{d+1} \quad h \in H, d = 1, \dots, D-1 \quad (2.10)$$

$$x_{ij}^d \leq y^d \quad (i, j) \in E, d = 1, \dots, D \quad (2.11)$$

$$y^d \geq y^{d+1} \quad d = 1, \dots, D-1 \quad (2.12)$$

$$\sum_{i \in \kappa} \sum_{j \in \kappa \setminus \{i\}} x_{ij}^d \leq |\kappa| - 1 \quad \kappa \subset C, 2 \leq |\kappa| \leq |C| - 1, d = 1, \dots, D \quad (2.13)$$

$$x_{ij}^d \in \{0, 1\} \quad (i, j) \in E, d = 1, \dots, D \quad (2.14)$$

$$y^d \in \{0, 1\} \quad d = 1, \dots, D \quad (2.15)$$

Considerando o modelo matemático, a função objetivo (2.1) visa minimizar o número de viagens e o tempo total gasto. As restrições (2.2) garantem que cada cliente será visitado exatamente uma vez. As restrições (2.3) garantem que haja conectividade entre cada viagem contida na rota, e as restrições (2.4) e (2.5) garantem que cada viagem inicia e termina em um dos $|H|$ hotéis disponíveis. As restrições (2.6) impõem um limite ao tempo total de uma viagem, que inclui o tempo gasto para percorrer o caminho e tempos de visita. As restrições (2.7) e (2.8) definem que a rota deve iniciar e terminar no hotel 0. As restrições (2.9) e (2.10) indicam que se uma viagem termina em um dado hotel, então a próxima viagem deve, obrigatoriamente, iniciar neste hotel. As restrições (2.11) definem que uma viagem está sendo utilizada se, e somente se, acontece no mínimo a visita a um cliente ou a um hotel naquela jornada de trabalho. Restrições (2.12) garantem que as viagens serão realizadas em dias consecutivos, iniciando-se no primeiro dia. As restrições (2.13) são aplicadas às viagens e garantem que não serão formados subciclos desconexos da origem. Por fim, as restrições (2.14) e (2.15) indicam o domínio das variáveis.

Embora [9] tenha apresentado uma intuição da razão do PCVSH ser um problema \mathcal{NP} -difícil, uma prova formal não foi apresentada. Esta prova formal é demonstrada a seguir.

Teorema 1. O Problema do Caixeiro Viajante com Seleção de Hotéis é \mathcal{NP} -Difícil.

Demonstração. Para provar que o PCVSH é \mathcal{NP} -difícil, é preciso mostrar que o $\text{PCV} \leq_P$

PCVSH, uma vez que o PCV é conhecidamente \mathcal{NP} -difícil[2, 22]. No PCV, uma instância consiste em um grafo completo $G' = (V', E')$, onde V' é o grupo de vértices que precisam ser visitados e d_{ij} a distância euclidiana associada a cada aresta $(i, j) \in E'$. Considerando como base uma instância do PCV, é possível construir uma instância para o PCVSH como se segue:

- atribua $V' \setminus \{v_0\}$ como sendo o grupo de clientes C e defina o grupo de hotéis como sendo $H = \{v_0\}$;
- atribua E' como sendo o grupo de arestas E ;
- defina $c_{ij} = d_{ij}$ para cada aresta $(i, j) \in E$;
- defina que o tempo total gasto em uma viagem não exceda $L = \infty$;
- defina o tempo de visitaç o $\tau_i = 0$, para todo $i \in C \cup H$.

Uma vez que essa transforma  o pode ser concluída em tempo polinomial e a resolu  o do PCVSH e do PCV retornem a mesma solu  o considerando a mesma inst  ncia, pode-se concluir que o PCVSH   no m  nimo t  o dif  cil de ser resolvido quanto o PCV. \square

2.2 O Problema do Caixeiro Viajante M  ltiplo com Sele  o de Hot  is (PCVMSH)

De forma similar ao PCV, no PCVM o objetivo   alcan  ar um caminho de custo m  nimo que visite todos os clientes exatamente uma  nica vez, utilizando m  ltiplos caixeiros. No PCVSH, o objetivo   visitar todos os clientes no menor n  mero de dias e minimizar o tempo total gasto, realizando paradas em hot  is nos casos em que n  o seja poss  vel atender a todos os clientes em uma  nica jornada de trabalho. Da combina  o destas duas variantes, originou-se o PCVMSH que considera um n  mero m  ximo de jornadas a serem realizadas por cada um dos m  ltiplos caixeiros e tem como objetivo encontrar um conjunto de rotas, cada uma iniciando e finalizando em um local comum (hotel ou dep  sito), de forma que os clientes sejam visitados uma  nica vez e o tempo total necess  rio para percorrer todas as rotas seja minimizado.

Dentre as aplica  es pr  ticas do PCVMSH vale citar o problema em que caixeiros precisam entregar uma grande quantidade de encomendas, problema de transportadores

de cargas perecíveis que precisam atender a um conjunto grande de clientes geograficamente dispersos e o problema de entregas de mercadorias realizado por Veículos Aéreos Não Tripulados que precisam ser recarregados periodicamente.

Ao herdar características relativas à estrutura de dois problemas que são \mathcal{NP} -Difíceis, a complexidade do PCVMSH é no mínimo equivalente à de seus pares. Mesmo considerando uma jornada de trabalho infinita e um número mínimo possível de caixeiros, o que em tese facilitaria a resolução do problema, a complexidade seria equivalente à apresentada pelo PCV [9].

É possível identificar que a dificuldade de resolução do PCVMSH é proporcional à quantidade de recursos com que o problema está lidando, tais como, número de clientes que precisam ser visitados, quantidade de caixeiros e quantidade de hotéis disponíveis. Estes recursos, sem dúvida, interferem diretamente na qualidade da solução e no tempo computacional necessário.

Resolver o PCVMSH de forma ótima é uma tarefa extremamente árdua. Logo, a utilização de métodos exatos fica limitada na maioria das vezes pelo tamanho da instância. Uma alternativa interessante é o uso de métodos heurísticos com a finalidade de obter soluções viáveis de alta qualidade em um tempo computacional aceitável.

2.2.1 Estruturas de vizinhança

Devido ao PCVMSH ter sido apresentado recentemente na literatura, as estruturas de vizinhança utilizadas para tratar o problema são derivadas de trabalhos prévios acerca do PCVSH e Problema de Roteamento de Veículos. Além das estruturas de vizinhança definidas para o PCVSH na Seção 2.1, é apresentada mais uma estrutura *inter-trip* utilizada para otimizar as soluções do PCVMSH. Adicionalmente, por trabalhar com múltiplas rotas, estruturas de vizinhança capazes de realizar trocas entre rotas são fundamentais. Essas estruturas são definidas como *inter-tours*.

2.2.1.1 *Inter-trips*

Utilizando-se a estrutura *InsertHotel* [6], obtém-se uma solução vizinha inserindo um hotel intermediário na rota caso esta inserção contribua positivamente para reduzir a inviabilidade da solução, ou seja, reduza ou até mesmo anule a quantia que excede o tempo limite para se percorrer uma viagem. O movimento respeita o número máximo de viagens para um caixeiro. A Figura 2.8 ilustra o comportamento da estrutura, onde a

rota inicial apresentada possui apenas uma viagem, e a solução vizinha, após a inserção do hotel intermediário, possui duas viagens.

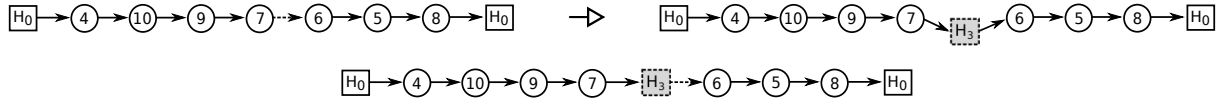


Figura 2.8: Movimento realizado na estrutura *InsertHotel*.

2.2.1.2 *Inter-tours*

Os métodos construtivos utilizados dificilmente alocam cada cliente na rota considerada ideal para garantir o melhor valor possível para a solução. Devido à aleatoriedade presente nesses métodos, é necessário utilizar procedimentos para realocação de clientes não somente internamente a uma rota, mas também entre rotas distintas. Diante desta necessidade de movimentar clientes entre rotas, foram utilizadas duas estruturas de vizinhança que exploram diferentes formas de realocar os clientes.

A estrutura *Shift* [33] atua de forma semelhante à estrutura *Relocate*, porém neste caso, a realocação de clientes é realizada entre rotas. Considera um conjunto k de clientes para serem realocados, testando $k \in \{1, 2, 3\}$. Para cada um dos movimentos possíveis utilizando os valores de k , apenas aquele que melhor otimiza a solução atual é aplicado. Caso não haja melhoria, nenhum movimento é aplicado à solução. Um exemplo dessa estrutura é ilustrado na Figura 2.9 onde são apresentadas duas rotas, uma para cada caixeiro. Utilizando $k = 1$, um cliente contido na primeira viagem da rota do primeiro caixeiro é movido para a segunda viagem da rota do segundo caixeiro.

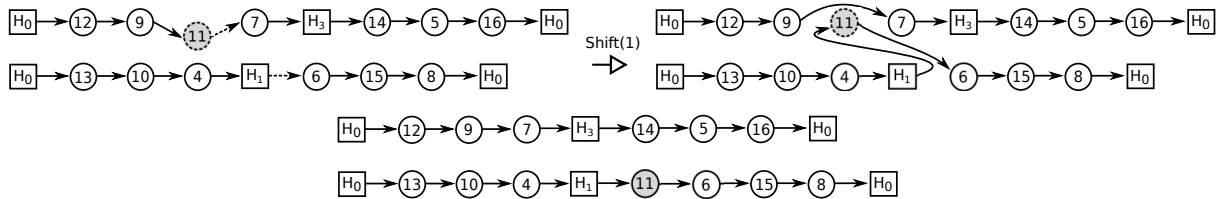


Figura 2.9: Movimento realizado na estrutura *Shift*.

Similar à estrutura *Exchange*, na estrutura de vizinhança *Swap* [33] a troca é realizada entre rotas e não entre viagens. Esta estrutura envolve um número maior de possibilidades de trocas de clientes, uma vez que testa movimentos trocando (3, 3), (3, 2), (2, 2), (2, 1) e (1, 1) clientes. De forma similar às estruturas anteriores, apenas o movimento que guie a solução corrente em direção à melhora mais efetiva será aplicado. Também não é permitida a aplicação de um movimento que não melhore a solução. Um exemplo de

aplicação da estrutura é descrito na Figura 2.10, na qual são expostas duas rotas e, a partir de um valor de troca $(2, 2)$, dois clientes contidos na segunda viagem da rota superior são trocados com dois clientes da segunda viagem da rota inferior.

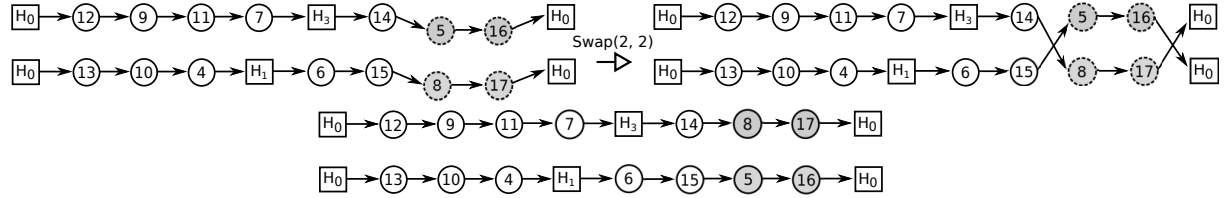


Figura 2.10: Movimento realizado na estrutura *Swap*.

2.2.2 Revisão da literatura

O PCVMSH foi definido em [6] sendo correlacionado a problemas clássicos da literatura com os quais compartilha características. Neste trabalho foi desenvolvida uma formulação matemática do tipo *set-partitioning* para descrever o problema, juntamente com a disponibilização de funções objetivo alternativas as quais permitiriam a inclusão dos custos dos hotéis, como é considerado no PCVMJTSH e o uso de múltiplos depósitos, o que permitiria que cada caixeiro fosse associado a um hotel (ou depósito) de partida distinto.

Também foi apresentada uma heurística *Iterated Local Search* para resolvê-lo, a qual apresenta os melhores resultados conhecidos para o problema. A metaheurística proposta pelo autores possui perturbações realizadas de forma aleatória sobre os clientes e hotéis. Dois métodos foram utilizados para construção da solução inicial, com e sem balanceamento de clientes: (i) inspirado na heurística de *Christofides* [11] onde são construídas m rotas em paralelo, sendo essas otimizadas posteriormente pelo uso de *2-opt/Relocate* e particionadas por meio de um algoritmo de programação dinâmica adaptado a partir do método de *Prins* [34]; (ii) constrói-se uma solução para o PCVMSH a partir de uma solução do PCV, utilizando uma estratégia de inserção do vizinho mais próximo com otimização por meio da heurística de *Lin-Kernighan* [24] e procedimento de divisão adaptado de *Prins*. Nenhum dos métodos construtivos garante que a solução seja sempre viável para o problema. O procedimento de busca local combina uma técnica de descida acentuada (similar ao VND) com uma estratégia de oscilação entre soluções viáveis e inviáveis por meio de um cálculo de penalidade capaz de permitir soluções diversificadas. As estruturas de vizinhança utilizadas foram: *2-opt*, *Relocate*, *Exchange*, *Cross*, *ChangeHotel*, *InsertHotel* e *RemoveHotel*. A estrutura *Cross* troca as partes finais de duas viagens que compartilham o mesmo hotel intermediário, com base em um ponto de cruzamento. Além

disso, os autores propuseram um total de 1440 instâncias para o problema, clusterizadas e aleatórias, contendo de 50 a 500 clientes e de 2 a 10 caixeiros. Foram apresentados resultados somente para instâncias cuja quantidade de clientes variava entre 50 e 300.

2.2.3 Definição formal

Os autores em [6] apresentaram uma descrição do problema que é exposta em detalhes a seguir. Desconsiderando qualquer associação com a definição apresentada para o PCVSH na Subseção 2.1.3, considere um conjunto de $s + 1$ hotéis $H = \{0, \dots, s\}$ e o conjunto de n clientes $C = \{s + 1, \dots, s + n\}$. O PCVMSH é definido em um grafo $G = (V, E)$, onde $V = H \cup C$ é o conjunto de vértices no grafo e $E = \{(i, j) | i, j \in V, i \neq j\}$ é o conjunto dos possíveis arestas. Para cada cliente que deve ser visitado, está associado um tempo de visita v_i ($i \in C$), sendo considerado para cada hotel $i \in H$ um tempo de visita é igual a zero ($v_i = 0$). Complementarmente, o tempo de viagem r_{ij} e o custo associado c_{ij} a cada aresta $(i, j) \in E$ são conhecidos previamente.

Seja a variável x_{ij}^t que recebe o valor 1 se a localização j é visitada imediatamente depois da localização i na viagem t e 0 caso contrário. Considere uma viagem viável t como uma sequência de clientes representados pelo vetor (x_{ij}^t) de forma que:

- (a) uma viagem inicia e finaliza em um dos hotéis disponíveis;

$$\sum_{h \in H} \sum_{(h,j) \in E} x_{hj}^t = \sum_{h \in H} \sum_{(i,h) \in E} x_{ih}^t = 1 \quad (2.16)$$

- (b) uma viagem não contém uma sub-rota inválida;

$$\sum_{(i,j) \in E: i,j \in S} x_{ij}^t \leq |S| - 1, \forall S \subseteq C \quad (2.17)$$

- (c) o tempo de uma viagem não excede um valor máximo constante L ;

$$\sum_{(i,j) \in E} r_{ij} x_{ij}^t \leq L. \quad (2.18)$$

Seja o conjunto T de todas as viagens viáveis, para as quais todas as condições (a)-(c) previamente apresentadas sejam satisfeitas. Considere duas variáveis φ_{ht} e λ_{it} , assim como o custo da viagem γ_t associados a cada viagem $t \in T$. O custo da viagem é calculado como:

$$\gamma_t = \sum_{(i,j) \in E} c_{ij} x_{ij}^t. \quad (2.19)$$

As variáveis φ_{ht} e λ_{it} são definidas da seguinte forma:

$$\varphi_{ht} = \begin{cases} 2 & \text{se a viagem } t \text{ inicia e termina no hotel } h, \\ 1 & \text{se a viagem } t \text{ ou inicia ou termina no hotel } h, \\ 0 & \text{caso contrário,} \end{cases}$$

$$\lambda_{it} = \begin{cases} 1 & \text{se o cliente } i \text{ é visitado por uma viagem } t, \\ 0 & \text{caso contrário.} \end{cases}$$

Seja I o grupo de viagens que iniciam e terminam em diferentes hotéis (conjunto de viagens para os quais $\varphi_{ht} = 1$) e $\Delta(S)$ o conjunto de viagens que iniciam e terminam em um subgrupo específico de hotéis S ($S \subseteq H$). Finalmente, seja $\Psi(S)$ o grupo de viagens que iniciam em um hotel em S e terminam em um outro hotel que não está em S , ou vice-versa.

Seja também P um grupo de m caixeiros disponíveis. O objetivo é encontrar m rotas, cada uma iniciando e terminando em um mesmo hotel ($i = 0$). A duração total da rota não pode exceder um número pré-definido de dias de trabalho D .

Finalmente, seja y_{pt} uma variável binária que possui valor 1 se uma viagem t é associada ao caixeiro p e a variável w_{hp} que representa o número de vezes que o caixeiro p chega e parte de um hotel h . Fazendo uso da notação descrita acima, o PCVMSH pode ser formulado da seguinte maneira:

$$\min \quad \sum_{p \in P} \sum_{t \in T} \gamma_t y_{pt} \quad (2.20)$$

$$s.a. \quad \sum_{p \in P} \sum_{t \in T} \lambda_{it} y_{pt} = 1, \forall i \in C \quad (2.21)$$

$$\sum_{t \in T} y_{pt} \leq D, \forall p \in P \quad (2.22)$$

$$\sum_{t \in I} \varphi_{ht} y_{pt} = 2w_{hp}, \forall h \in H, \forall p \in P \quad (2.23)$$

$$\sum_{t \in \Delta(S)} y_{pt} \leq |\Delta(S)| \left(\sum_{t \in \Psi(S)} y_{pt} \right), S \subseteq H \setminus \{0\}, \forall p \in P \quad (2.24)$$

$$y_{pt} \in \{0, 1\} \quad (2.25)$$

$$w_{hp} \in \mathbb{Z}^+ \quad (2.26)$$

A equação (2.20) considera o somatório do tempo total gasto por cada caixeiro. As restrições (2.21) garantem que todos os clientes são visitados unicamente em uma viagem, enquanto as restrições (2.22) garantem que o número de viagens associado a cada caixeiro não exceda o máximo permitido. As restrições (2.23) são restrições de cardinalidade para cada hotel, enquanto as restrições (2.24) evitam sub-rotas inválidas na solução. Por fim, as restrições (2.25) e (2.26) correspondem às restrições de integralidade.

Capítulo 3

Heurísticas para o PCVSH

3.1 Heurística Iterated Local Search

A abordagem aqui proposta é baseada na metaheurística ILS [25] apresentada no Algoritmo 1. O primeiro passo a ser realizado é a criação de uma solução inicial s_0 , seguido pela aplicação do procedimento de busca local para obtenção de uma solução localmente ótima s^* . O *loop* principal do algoritmo executa até que um determinado critério de parada seja alcançado. Internamente a esse *loop*, um procedimento de perturbação aleatório é executado para possibilitar que o algoritmo escape de soluções que são consideradas ótimos locais (passo 4). Em seguida, a busca local aprimora a solução corrente com o objetivo de alcançar uma solução melhor que aquela até então reportada (passo 5). O último passo consiste em verificar se o critério de aceitação de uma solução foi alcançado, comumente avaliado com base na qualidade da solução gerada pelo procedimento de busca local. Caso o critério de aceitação seja alcançado, a solução s^* é então atualizada. Uma vez que o critério de encerramento do algoritmo é atingido, o algoritmo retorna a melhor solução encontrada, a qual pode ou não corresponder a uma solução ótima.

Algoritmo 1 *Iterated Local Search framework* [25]

```
1:  $s_0 = \text{Criar\_Solução\_Inicial}()$ 
2:  $s^* = \text{Busca\_Local}(s_0)$ 
3: repita
4:    $s' = \text{Perturbação}(s^*)$ 
5:    $s'^* = \text{Busca\_Local}(s')$ 
6:    $s^* = \text{Critério\_Aceitação}(s^*, s'^*)$ 
7: até (alcançar critério de parada)
8: retornar  $s^*$ 
```

O Algoritmo 2 descreve em detalhes a heurística ILS-RVND proposta neste trabalho.

O método proposto considera como parâmetros de entrada o tempo computacional que será utilizado como critério de parada ($time_{hdm}$) para cada instância, o número máximo de iterações (n_{pert}) a serem executadas sem que seja aplicado o procedimento de perturbação e o número de iterações (v) sem melhora na solução para aplicação da intensificação do procedimento de perturbação.

Os primeiros três passos (passos 2 a 4) são responsáveis por inicializar as variáveis $time_{ini}$ que mensura o tempo computacional gasto desde o início da execução do algoritmo, it_{pert} que controla o número de iterações sem aplicação do procedimento de perturbação e, de acordo com o tamanho de cada instância, θ que define o número de clientes a serem realocados pelo procedimento de perturbação. No passo 5, uma solução do Problema do Caixeiro Viajante é criada para a instância, a qual é dividida em viagens para gerar uma solução viável para o PCVSH. Esta solução é armazenada na variável s_0 , a qual, no próximo passo, é submetida ao procedimento de busca local para aprimoramento e geração da solução s^* , correspondente a uma solução localmente ótima e viável para o PCVSH. O procedimento de busca local é descrito na Subseção 3.1.2.

O *loop* principal do algoritmo ILS-RVND, definido entre os passos 8 e 25, é repetido enquanto o tempo computacional gasto não ultrapassar o tempo $time_{hdm}$ para cada instância. Internamente ao *loop* principal, as instruções desde o passo 9 até o 13 geram novas soluções possibilitando a exploração de regiões distintas do espaço de busca, obtendo uma nova solução a partir da solução corrente s^* . Esta nova solução pode ser obtida por meio da troca da posição de um hotel (passo 10) ou aplicando o procedimento de perturbação de clientes (passo 12). O primeiro procedimento seleciona aleatoriamente um dos hotéis intermediários da solução e define também de maneira aleatória um novo local para reinserção. O segundo procedimento seleciona clientes para serem realocados na solução conforme descrito na Subseção 3.1.4. Aplica-se o primeiro procedimento em todas iterações, exceto para aquelas que são múltiplos de it_{pert} , porque este procedimento preserva a qualidade da solução em relação a ordem de visita dos clientes nas viagens. Dessa forma, a busca local a ser executada no passo 14 é aplicada mais rapidamente do que se fosse empregado o procedimento de perturbação de clientes. A partir do passo 15, a solução que foi melhorada s'^* é avaliada utilizando o critério de aceitação proposto. Caso tal solução utilize um número menor de viagens ou este número seja igual, mas o tempo total da solução seja minimizado, então a solução corrente passa a ser s'^* , substituindo a última até então armazenada. Neste caso, o número de iterações sem melhora é reiniciado para zero e o controlador de aplicação de perturbação it_{pert} é restaurado para seu valor original (n_{pert}). Caso contrário, apenas o número de iterações executadas é incrementado.

Algoritmo 2 Heurística ILS-RVND para o PCVSH

```

1: entrada:  $time_{hdm}$ ,  $n_{pert}$ ,  $v$    saída:  $s^*$ 
2:  $time_{ini} = Temp\_Agora()$ 
3:  $it_{pert} = n_{pert}$ 
4:  $\theta = Definir\_Perturbacao()$ 
5:  $s_0 = Criar\_Solucao\_Inicial()$ 
6:  $s^* = Busca\_Local(s_0)$ 
7:  $i = 0$ 
8: enquanto ( $Temp\_Agora() - time_{ini} < time_{hdm}$ ) faça
9:   se ( $(i \bmod it_{pert}) \neq 0$ ) então
10:     $s' = Vizinho(s^*)$ 
11:   senão
12:     $s' = Perturbacao(s^*, \theta)$ 
13:   fim se
14:    $s'^* = Busca\_Local(s')$ 
15:   se ( $s'^*$  melhor que  $s^*$ ) então
16:     $s^* = s'^*$ 
17:     $i = 0$ 
18:     $it_{pert} = n_{pert} + 1$ 
19:   senão
20:     $i = i + 1$ 
21:   fim se
22:   se ( $(i \bmod v) = 0$  e  $it_{pert} > 1$ ) então
23:     $it_{pert} = it_{pert} - 1$ 
24:   fim se
25: fim enquanto

```

Finalmente, quando o número de iterações sem melhora for um múltiplo de v , o procedimento de perturbação é intensificado (passo 23), reduzindo o intervalo de iterações sem aplicação da perturbação de clientes. Na Figura 3.1 é ilustrado o comportamento adaptativo do intervalo de iterações, com $v = 10$, o qual é reestabelecido para n_{pert} sempre que ocorre uma melhoria na solução.

Ao final de todo o processo descrito no Algoritmo 2, a melhor solução encontrada é retornada, a qual pode ou não ser um ótimo global.

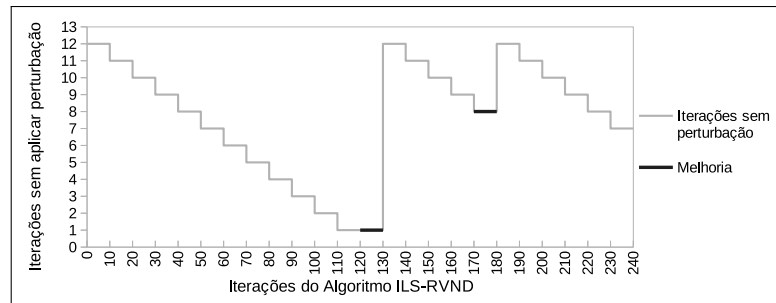


Figura 3.1: Comportamento do procedimento de intensificação da perturbação.

3.1.1 Solução inicial

O procedimento descrito no passo 5 do Algoritmo 2 para criar uma solução inicial é uma combinação de heurísticas clássicas e bem conhecidas. O método *order-first split-second* é empregado para criar uma rota inicial composta por uma única viagem correspondendo a uma solução do PCV. Em seguida, são utilizados os hotéis disponíveis para gerar viagens intermediárias viáveis respeitando o limite de tempo total para um dia de trabalho. Alguns trabalhos [8, 9] fizeram uso de técnicas similares para construção da solução inicial devido à simplicidade do método e o baixo custo computacional demandado.

A primeira etapa cria uma rota otimizada do PCV utilizando a heurística de *Lin-Kernighan* (LK) [24] como implementado em [2, 12]. O processo de criação nem sempre retorna diretamente uma rota viável para o PCVSH, apenas em casos específicos onde o número de clientes da instância é relativamente pequeno (por exemplo, até 15 clientes considerando as instâncias disponíveis em [46]). Na maioria dos casos, é necessário a aplicação de um procedimento de divisão, inspirado no Algoritmo de *Dijkstra* [15], para tornar uma solução do PCV viável para o PCVSH. Para isso, é criado um grafo contendo $mn + 1$ vértices, com $m = |H|$ e $n = |C|$. O primeiro nó representa o hotel de partida e é rotulado com o índice 1. O restante dos vértices correspondem à parada em um dado hotel disponível após a visita a um ou mais clientes na ordem em que aparecem na rota PCV. Esses vértices são rotulados com uma sequência de números, de forma que, ao realizar operações matemáticas de divisão/módulo sobre o rótulo, é possível identificar o hotel que terá de ser inserido e a posição deste em relação à ordem de clientes definida na rota PCV. Para cada vértice do grafo, é verificada a possibilidade de adição de arestas entre vértices subsequentes, de forma que uma aresta somente será adicionada entre dois vértices se a soma dos tempos totais, considerando tempos de visita e tempos totais percorridos, for inferior ao limite máximo de tempo para uma viagem do PCVSH. Ao inserir uma aresta, seu peso é definido como a soma dos tempos necessários calculados mais um peso constante $M = 1000000$, peso este que contribuirá com a geração de soluções do PCVSH com menor número de viagens.

Um exemplo de construção do grafo é apresentado na Figura 3.2, considerando as variáveis $m = 5$ e $n = 2$. Cada vértice do grafo corresponde a um respectivo hotel que foi precedido pelo cliente i . As arestas contidas no grafo representam a saída do caixeiro a partir de um dado hotel, visita a uma sequência de clientes e parada em outro hotel. Somente os caminhos que não excedem o limite de tempo de uma viagem receberam arestas no grafo. No exemplo, ao sair do vértice V^1 que representa H_0 , não é possível

chegar diretamente ao vértice de destino V^{10} , sendo necessária a realização de uma parada no vértice V^7 , o qual representa o hotel H_1 , precedido pela sequência $\{0, 2, 4, 6\}$. Caso haja mais de um caminho entre a origem e o destino, o caminho mínimo deve ser escolhido. Para este exemplo, o caminho no grafo que gera o menor tempo total da rota é definido por $\{V_1, V_7, V_{10}\}$, com V^1 a $V^7 = \{H_0, 2, 4, 6, H_1\}$, e V^7 a $V^{10} = \{H_1, 5, 3, H_0\}$.

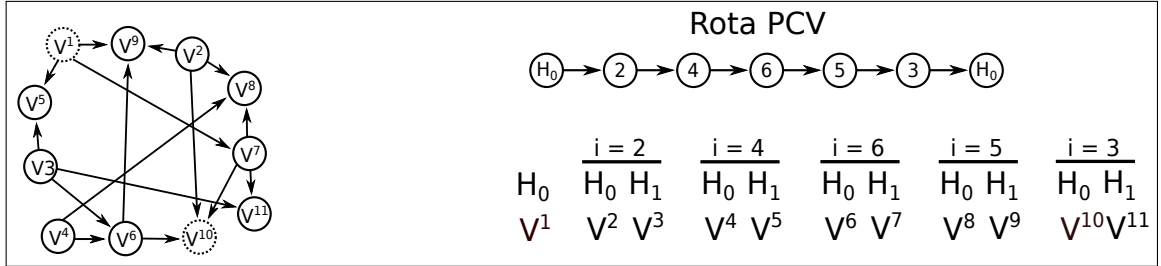


Figura 3.2: Exemplo do grafo criado para obtenção de uma solução do PCVSH.

Após a construção do grafo, um procedimento baseado no algoritmo de *Dijkstra* é aplicado para encontrar o caminho mínimo entre os vértices que correspondem ao hotel inicial (V^1) e o hotel final ($V^{(mn+1)-(m-1)}$). Como foi inserido no peso de cada aresta a constante M , o caminho mínimo definido pelo Algoritmo de *Dijkstra* prioriza a utilização do menor número de viagens necessárias para atender a todos os clientes. Desta forma, é realizado simultaneamente a minimização do número de arestas utilizadas (correspondem ao número de viagens) e o tempo total da rota.

A viabilidade da solução gerada para o PCVSH depende da construção otimizada da rota para o PCV. Caso essa rota seja de baixa qualidade, a aplicação do procedimento inspirado em *Dijkstra* pode não ser capaz de encontrar um caminho viável que respeite a restrição de tempo das viagens.

Técnicas mais simples de construção da solução inicial foram testadas, tanto para construção da rota do PCV, quanto para definição da ordem de visita dos clientes na composição das viagens. Estratégias como inclusão dos clientes na rota de maneira aleatória ou inserção mais barata foram descartadas, pois houve dificuldade de se encontrar soluções iniciais viáveis para todas as instâncias. Adicionalmente, as soluções viáveis geradas eram comumente de baixa qualidade, implicando assim em tempos de processamento mais altos durante o procedimento de busca local.

3.1.2 Busca local

O método de busca local utilizado para aprimorar a qualidade das soluções do PCVSH, conhecido como RVND, é bastante semelhante a heurística VND utilizada em [9]. A principal diferença é que o método proposto é baseado em um procedimento de descida de vizinhança variável aleatório [33, 44], o qual consiste na definição aleatória da ordem de aplicação das múltiplas estruturas de vizinhança consideradas, ao invés de uma ordem fixa que é comumente definida a priori.

O Algoritmo 3 apresenta a estrutura principal deste procedimento. Inicialmente, é recebida uma solução que pode ou não ser viável, pois os procedimentos para criação de uma solução vizinha ou perturbada quase sempre resultam em soluções inviáveis quanto ao tempo máximo das viagens. As estruturas de vizinhança exploram movimentos capazes de gerar melhores soluções e, ao final, a melhor solução alcançada é retornada, que pode ser viável ou não.

Este método de escolha aleatória da ordem de aplicação das estruturas de vizinhança funciona bem, pois como constatado em [44], é difícil definir uma ordem de aplicação das estruturas de vizinhança que seja eficiente para todas as instâncias. Desta forma, utilizando uma ordem aleatória consegue-se em média melhores resultados considerando as mais diversas características das instâncias conhecidas para o problema.

Algoritmo 3 Busca local utilizando Descida de vizinhança variável aleatória

```

1: entrada:  $s'$    saída:  $s^*$ 
2:  $Ordem\_Aplicação()$ 
3:  $k = 1$ 
4:  $s^* = s'$ 
5: enquanto ( $k \leq 4$ ) faça
6:    $s'^* = Busca(N_{DF[k]}, s^*)$ 
7:   se ( $s'^*$  é melhor que  $s^*$ ) então
8:      $s^* = s'^*$ 
9:    $k = k + 1$ 
10: senão
11:    $k = k + 1$ 
12: fim se
13: fim enquanto

```

Para aprimoramento das soluções, o RVND utiliza diferentes estruturas de vizinhança, sendo 2 estruturas *intra-trips* e 4 *inter-trips*, baseadas em estruturas previamente utilizadas nos trabalhos de [8, 26, 46].

A primeira função a ser executada é $Ordem_Aplicação()$, que de forma aleatória,

seleciona a ordem em que serão executadas as 4 estruturas de vizinhança *inter-trip* na busca local executada na iteração corrente do ILS. O *loop* principal do algoritmo, entre os passos 5 e 13, itera aplicando as estruturas de vizinhança enquanto houver alguma melhora da solução corrente.

No passo 6, o procedimento *Busca* detalhado no Algoritmo 4 iterativamente seleciona uma solução vizinha s a partir de uma estrutura de vizinhança específica N_k e calcula o valor da função $F(s)$ definida pela Equação (3.1) para determinar se essa solução vizinha é melhor que a solução corrente. A Equação (3.1) soma os tempos totais de todas as s_D viagens (primeiro termo) e penaliza soluções com viagens inviáveis (segundo termo), ou seja, viagens com duração maior que o limite máximo de tempo permitido L . Ao se definir a constante $M = 10000$, prioriza-se a minimização da inviabilidade da solução e, quando, não houver inviabilidade, a quantidade de viagens.

$$F(s) = \sum_{d=1}^{s_D} (M + \text{tempo}_d) + M \sum_{d=1}^{s_D} \max(0, \text{tempo}_d - L) \quad (3.1)$$

Se o procedimento encontrar uma solução melhor que a atual, serão executadas estruturas de vizinhança *intra-trips* (*2-Opt* e *Or-opt*) para potencializar a melhora da solução como mostrado no passo 6 do Algoritmo 4.

De forma similar a [16], o referido procedimento de busca local utiliza a estratégia de *Best Improvement* que exaustivamente explora todas as soluções vizinhas utilizando uma dada estrutura de vizinhança e retorna a solução que possui o menor tempo total da rota.

Algoritmo 4 Procedimento de busca para exploração de estrutura de vizinhança.

```

1: entrada:  $N_k$ ,  $s'$    saída:  $s''$ 
2:  $s'' = s'$ 
3: enquanto (melhora for encontrada) faça
4:    $s'^* = \operatorname{argmin}_{s \in N_k(s'')} F(s)$ 
5:   se ( $s'^*$  é melhor que  $s''$ ) então
6:      $s'' = \text{otimize } s'^* \text{ com } 2\text{-opt e } Or\text{-opt}$ 
7:   fim se
8: fim enquanto

```

Após a conclusão dos passos descritos no Algoritmo 4, se uma solução vizinha encontrada durante o procedimento representar uma melhora, então a atual melhor solução é substituída e o RVND reinicia o processo retornando à primeira estrutura de vizinhança. Caso contrário, a próxima estrutura de vizinhança *inter-trip* é explorada.

3.1.3 Estruturas de vizinhança

Considerando que uma solução do PCVSH sempre inicia e finaliza no hotel h_0 e, assumindo que o procedimento de perturbação (ver Subseção 3.1.4) pode gerar soluções inviáveis, a aplicação de um movimento baseado nas estruturas de vizinhança será aceito somente se reduz a quantidade de inviabilidade calculada pela Equação (3.1) ou se otimiza ainda mais uma solução já considerada viável.

Para abranger diversos tipos de movimentos capazes de otimizar uma solução, foram utilizadas estruturas de vizinhança comprovadamente eficazes para o problema, as quais são classificadas como *intra-trip* (*2-opt* e *Or-opt*) ou *inter-trip* (*Relocate*, *Exchange*, *ChangeHotels* e *JoinTrips*). Essas estruturas foram descritas em detalhes na Seção 2.1.

Quando uma solução é definida utilizando apenas uma única viagem, não é possível utilizar as estruturas de vizinhança *inter-trip*. Logo, se o Algoritmo 4 detecta uma solução com uma única viagem no passo 4, internamente, a estrutura de vizinhança *Relocate* é substituída pela *2-opt* e a *Exchange* pela *Or-opt*. Como as estruturas *ChangeHotels* e *JoinTrips* também não possuem utilidade prática neste caso, elas são desconsideradas.

3.1.4 Perturbação

O procedimento de perturbação utilizado no ILS-RVND altera a solução corrente modificando as posições dos clientes, assim como feito em outros trabalhos da literatura [8, 9, 40]. Majoritariamente, autores utilizam estratégias para a perturbação que aleatoriamente selecionam clientes e suas novas posições. Alguns outros trabalhos realizam a aplicação de movimentos específicos de trocas utilizando estruturas de vizinhança para gerar novas soluções que possuam características distintas. Ambas as estratégias realizam alterações em um número fixo de clientes, devendo este número ser grande o suficiente para guiar a solução para outra região do espaço de busca e, ao mesmo tempo, não tão grande para garantir que as características que conferem a boa qualidade de uma solução sejam preservadas.

Para o PCVSH, utilizar uma taxa percentual alta de perturbação para instâncias com muitos clientes, pode gerar soluções muito distintas da solução base e, utilizar pequenas taxas para instâncias com pequeno número de clientes, pode não ser suficiente para alterar a solução base. Uma alternativa inteligente para contornar o problema citado é definir a taxa proporcionalmente ao número de clientes contidos em cada instância. Após analisar possíveis estratégias, verificou-se que o uso de uma Função não Linear Simétrica Sigmoidal

(4PL) criada pelo MyCurveFit¹ é capaz de gerar tais proporcionalidades e garantir resultados satisfatórios. A equação gerada considera os seguintes parâmetros: x como sendo o número de clientes da instância, $a = 41756810$, $b = 0,6469404$, $c = 1,643992 * 10^{-9}$ e $d = 0,2665922$. Quando tal função é utilizada, um percentual mais elevado (aproximadamente 20%) é utilizado para instâncias consideradas pequenas (10 clientes), garantindo que um número significativo de clientes terão suas posições alteradas. À medida em que instâncias com um número maior de clientes vão sendo abordadas (ver Figura 3.3), o percentual é recalculado podendo chegar a valores mínimos próximo de 1% (1000 clientes). Este método de definição variável da taxa foi testado e comparado com o uso de uma porcentagem fixa em [42], ficando comprovado qualitativamente sua eficácia. Os resultados dos testes são apresentados na Seção 5.1.1.

Similar ao aplicado em outros trabalhos, cada cliente e sua nova posição são selecionados aleatoriamente, até que o número de clientes a serem alterados satisfaça o limite da porcentagem calculada previamente. Um exemplo da variação de valores que a porcentagem pode assumir é ilustrado na Figura 3.3, englobando a variação desde a menor até a maior instância conhecida para o PCVSH. Para instâncias com número reduzido de clientes a taxa é limitada superiormente a 20% e, segue decrementando à medida em que o número de clientes aumenta podendo chegar à taxa mínima de 1,25%.

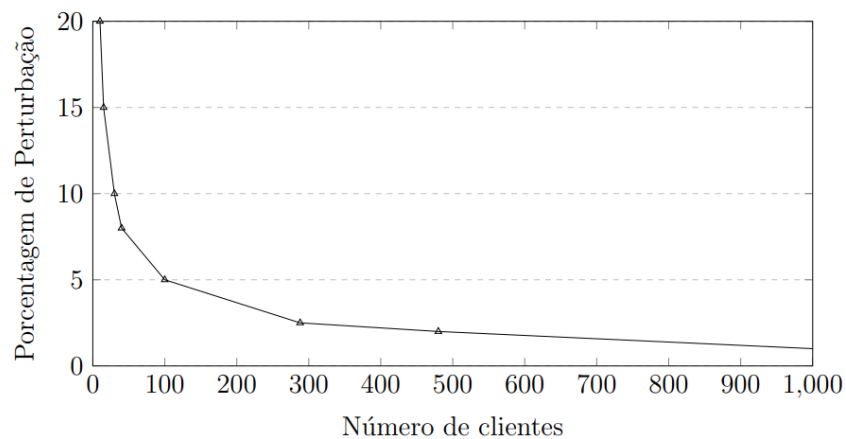


Figura 3.3: Relação entre porcentagem de perturbação e tamanho da instância.

Neste Capítulo, são apresentadas duas estratégias de hibridização da heurística ILS-RVND apresentada no capítulo anterior. Na primeira utilizam-se os padrões encontrados pela mineração de dados para tentar guiar a heurística para espaços de busca mais promissores. Na segunda utilizam-se estes padrões em conjunto com um método exato.

¹<http://mycurvefit.com/>

3.2 Hibridização com Mineração de dados

Esta estratégia de hibridização foi idealizada considerando a heurística ILS-RVND apresentada na Seção 3.1 e a aplicação de técnicas de mineração de dados como demonstrado em [17, 20].

Técnicas de mineração de dados têm sido aplicadas eficientemente na extração de padrões e regras a partir de bases de dados. Uma dessas técnicas é conhecida como Mineração de Itens Frequentes (do inglês *Frequent Itemset Mining* - FIM) [1, 19, 21] que corresponde à mineração de itens frequentemente observados numa base de dados de transações, onde cada transação é um conjunto de itens.

A hibridização da técnica FIM com outras heurísticas como GRASP e ILS tem demonstrado boa aplicabilidade, melhorando soluções para diversos problemas [17, 20, 28, 36, 37]. A ideia básica é utilizar a técnica FIM para extrair alguns padrões a partir de um *pool* de soluções de boa qualidade, que foram previamente geradas pela heurística e, então, usar os padrões minerados para construir novas soluções e guiar o processo de busca para soluções que são possivelmente melhores.

Para aplicação da técnica FIM, executa-se o procedimento de mineração descrito no Algoritmo 5, sempre após a geração de uma solução no Passo 13 do Algoritmo 2 referente à heurística ILS-RVND.

O Algoritmo 5 recebe os parâmetros *pool*, *sup*, n_{pool} e s^* . No passo 2, verifica se a solução s^* gerada pela busca local deve ser inserida no *pool* de melhores soluções que contém n melhores soluções distintas geradas pela heurística ILS-RVND. Uma solução somente é aceita no *pool*, caso seja diferente das existentes e seja melhor que a pior solução contida no *pool*. No passo 3, se durante n_{pool} iterações não houver modificações no *pool*, então uma nova solução para o PCVSH é gerada utilizando-se um padrão que foi previamente minerado, conforme explicado no Algoritmo 6. Ao final se retorna a solução gerada utilizando-se padrões ou a solução inicial gerada pela busca local.

Algoritmo 5 Inclusão do procedimento de mineração ao ILS-RVND.

```

1: entrada: pool, sup,  $n_{pool}$ ,  $s^*$    saída:  $s'^*$ 
2: Atualizar_Pool( $s^*$ )
3: se (o pool não foi atualizado durante  $n_{pool}$  iterações) então
4:    $s'^* = Usar\_Mineração\_Dados(pool, sup)$ 
5: fim se

```

O Algoritmo 6 recebe como parâmetros de entrada o *pool* contendo as melhores so-

luções distintas encontradas, o valor *sup* para o suporte mínimo e o número máximo de padrões a serem retornados (*maxPad*). Entre os passos 2 e 6, caso não haja nenhum padrão armazenado, aplica-se a técnica FIM para gerar no máximo *maxPad* novos padrões a partir do *pool* de soluções elites usando o parâmetro *sup*. A técnica FIM obtém padrões que contêm itens frequentes que aparecem percentualmente em no mínimo *sup* transações do *pool* de transações que são compostas por itens. Neste caso, uma solução do PCVSH é considerada uma transação. Como ela é uma sequência de arestas (i, j) , onde i e j podem ser hotéis ou clientes, para a aplicação da técnica FIM é necessário a utilização de um procedimento de codificação que converte a sequência de arestas em identificadores únicos [20].

Algoritmo 6 Procedimento de mineração de dados.

```

1: entrada: pool, sup, maxPad   saída:  $s'^*$ 
2: se ( $|padroes| == 0$ ) então
3:   Codificar()
4:   padroes = Executar_Minerador(pool, sup, maxPad)
5:   Decodificar(padroes)
6: fim se
7:  $s' = \text{Construir\_Solução\_Padrão}()$ 
8:  $s'^* = \text{Busca\_Local}(s')$ 

```

Considerando uma solução hipotética definida por $s = \{0, 3, 7, 4, 0\}$ a qual é representada pela sequência $0 \rightarrow 3 \rightarrow 7 \rightarrow 4 \rightarrow 0$, isto é, pelo hotel (0) e clientes (3, 7 e 4), s pode ser transformada em $s_{ids} = \{id_{0 \rightarrow 3}, id_{3 \rightarrow 7}, id_{7 \rightarrow 4}, id_{4 \rightarrow 0}\}$ que preservam a ordem de visita da solução e viabilizam o mapeamento inverso de forma trivial e direta. Depois desta codificação, assim como reportado em [20], é garantido que a ordem de visita da solução original seja preservada e que a técnica FIM possa ser aplicada para extrair os padrões a partir do *pool* de soluções elites. Complementarmente, devido a possibilidade de repetição de visita a um mesmo hotel durante o percurso, ter um identificador único é essencial para mapear arestas que chegam e saem de um mesmo hotel.

Internamente, o processo de codificação das arestas em *ids* únicos é realizado por meio do seguinte cálculo: (i) para cada posição da rota, multiplica-se o cliente ou hotel que se encontra naquela posição pela constante Z , e (ii) soma-se esse resultado com o cliente ou hotel que se encontra na posição seguinte. A constante Z possui valor igual a $(n_{clientes} + n_{hoteis} + 30)$, onde $n_{clientes}$ representa o número de clientes contidos na instância, n_{hoteis} o número de hotéis e a constante 30 é um número suficiente grande para garantir que não será gerado um mesmo número de *id* para clientes ou hotéis que se encontrem em diferentes posições da rota. Por exemplo, para uma instância com 10 clientes e 2 hotéis,

uma possível solução seria $s' = \{0, 2, 4, 3, 8, 11, 9, 10, 5, 7, 6, 0\}$. Essa solução s' seria convertida, gerando os seguintes *ids*: $s'_{ids} = \{2, 88, 171, 134, 347, 471, 388, 425, 217, 300, 252\}$. A Figura 3.4 ilustra um exemplo representativo da transformação de quatro soluções em seus *ids* correspondentes para um *pool* de tamanho $n = 4$.

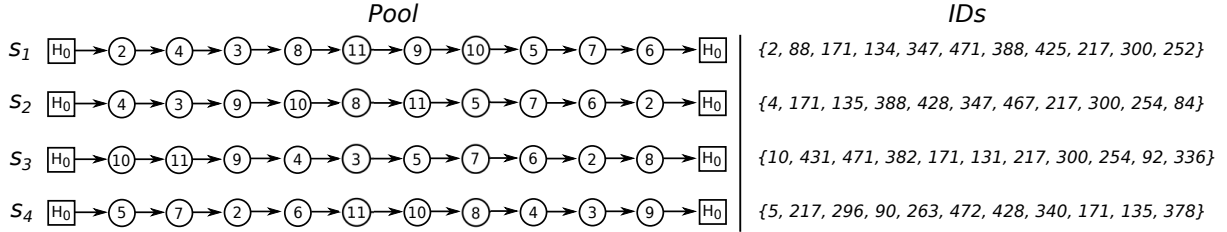


Figura 3.4: Transformação de soluções em *Ids*.

A função *Executar_Minerador()* no passo 4 do Algoritmo 6 executa o algoritmo FIM, conforme implementado em [20] e retorna no máximo *maxPad* padrões. Cada padrão retornado é composto por identificadores frequentes que estão contidos, simultaneamente, em ao menos um percentual de *sup* rotas presentes no *pool*. Então, o próximo passo é decodificar os identificadores de cada padrão armazenado na variável *padroes*, transformando-os em suas arestas correspondentes. É possível que, no mesmo padrão, duas ou mais arestas possam ser consecutivamente ligadas para criar um caminho maior. Todas essas concatenações possíveis são executadas antes da construção de uma nova solução do PCVSH a partir do padrão selecionado.

O funcionamento da técnica FIM consiste na análise dos identificadores gerados, buscando agrupar em padrões aqueles que mais aparecem nas soluções. Considerando a Figura 3.4, os identificadores 171 e 217 correspondem a arestas que estão presentes em todas as soluções, logo, formam o padrão $\{171, 217\}$. O padrão $\{171, 217, 254\}$ também pode ser extraído, pois o identificador 254 aparece juntamente com o 171 e 217 em duas soluções (s_2 e s_3). A aceitação de um padrão é condicionada ao valor do parâmetro *sup* que equivale ao suporte. Caso esse valor do suporte gere um número fracionário em relação ao número de soluções do *pool*, é considerado o número de soluções inteiras imediatamente superior. Considerando $sup = 50\%$, serão considerados padrões o conjunto de identificadores que estão presentes em duas ou mais soluções. Neste caso, os dois padrões previamente mencionados serão considerados. No caso de $sup = 75\%$, somente seriam aceitos padrões contidos simultaneamente em três ou mais soluções, inviabilizando assim o aceite do padrão $\{171, 217, 254\}$. Assim, o parâmetro *sup* funciona como um limitador determinando que um dado padrão somente será retornado, caso esteja presente em no mínimo um percentual *sup* de soluções.

Uma vez que os padrões tenham sido extraídos, o decodificador executará duas operações sobre cada *id* retornado: (i) divisão inteira pela constante Z para encontrar o primeiro vértice, e (ii) operação de resto da divisão entre o *id* e a constante Z para encontrar o segundo vértice. Na Figura 3.5, é exemplificado o retorno de dois padrões a partir da aplicação da técnica FIM aos *Ids* previamente codificados. Os padrões retornados foram $\{171, 217, 300, 347, 388\}$ e $\{171, 217, 254\}$. Aplicando as operações de decodificação ao primeiro padrão, serão identificadas as seguintes arestas $\{(4 \rightarrow 3), (5 \rightarrow 7), (7 \rightarrow 6), (8 \rightarrow 11), (9 \rightarrow 10)\}$. Como existem arestas que compartilham os mesmos vértices, essas serão concatenadas para formar caminhos a serem utilizados na construção da nova solução do PCVSH. O resultado será composto pelos seguintes caminhos $\{(4 \rightarrow 3), (5 \rightarrow 7 \rightarrow 6), (8 \rightarrow 11), (9 \rightarrow 10)\}$.

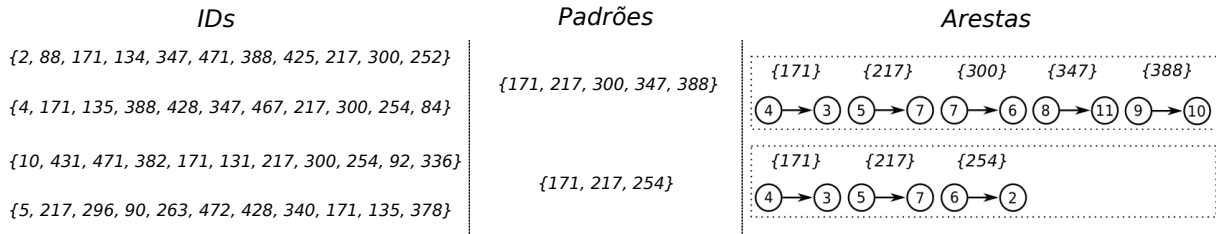


Figura 3.5: Aplicação da técnica FIM com $sup = 50\%$.

No passo 7, um padrão ainda não utilizado é selecionado para construir uma nova solução do PCVSH. Inicialmente, o método cria uma solução para o PCV inserindo iterativamente os clientes que não estão presentes no padrão considerando a posição em que representem o menor acréscimo no tempo total da rota. Após, são inseridos na rota os caminhos que foram extraídos do padrão, levando em consideração também o local de inserção que represente o menor incremento no tempo total da rota. O Algoritmo de *Dijkstra* explicado na Subseção 3.1.1 é utilizado para realizar a tentativa de divisão da solução do PCV em uma solução viável do PCVSH. Em seguida, o procedimento de busca local previamente definido na Subseção 3.1.2 busca garantir a otimização de uma solução já viável ou a redução de sua inviabilidade. Em ambos os casos, o objetivo é obter alguma melhoria em relação à solução prévia. Ao final, o procedimento retorna a solução para que o ILS-RVND continue sua execução.

No decorrer desta Tese, a hibridização da heurística ILS-RVND com a técnica FIM descrita acima, será referenciada como MILS (do inglês *Mining Iterated Local Search*).

3.3 Hibridização com Mineração de dados com método exato

Considerando o potencial de uso da técnica FIM para geração de soluções de qualidade baseadas em hibridização, este segundo método proposto considera a utilização da técnica FIM e, adicionalmente, propõe a resolução do PCVSH por meio de um método exato.

Nesta estratégia, executa-se inicialmente a heurística ILS-RVND para se obter um *pool* de soluções e encontrar uma solução heurística de boa qualidade que será utilizada como *UpperBound* no método exato. Ao final da execução do ILS-RVND, a técnica FIM é aplicada sobre as soluções que compõem o *pool*. Caso seja retornado mais que um padrão com itens frequentes, aquele padrão que possuir o maior número de identificadores será considerado para utilização no método exato.

O método exato utilizado é baseado na formulação de Programação Linear Inteira apresentada na Subseção 2.1.3. A execução do modelo exato ocorre em duas fases como apresentado no Algoritmo 7. Os parâmetros de entrada são as arestas do maior padrão obtido pela aplicação da técnica FIM e a solução retornada pela heurística. A primeira fase é composta pelos passos 2 e 3 do algoritmo. No passo 2, o modelo é criado relaxando-se a variável que especifica se há a visita consecutiva aos clientes i e j no dia d ($x_{ij}^d \in \{0, 1\}$), possibilitando que ela possua valores fracionários. No próximo passo, o modelo relaxado (m_1) é resolvido. A segunda fase inicia-se com a criação do modelo de programação inteira, permitindo que a variável x_{ij}^d somente assumam os valores 0 ou 1. Todas as variáveis x_{ij}^d que assumirem valores maiores que 0,9 na primeira fase terão seus valores fixados em 1 através da definição de restrições no modelo a ser resolvido nessa fase.

Uma vez que as arestas retornadas pela técnica FIM correspondem a arestas frequentemente utilizadas, espera-se que elas façam parte da solução ótima do PCVSH. Desta forma, as arestas correspondentes ao maior padrão retornado também são fixadas com valor 1 como restrições no modelo. O último passo da segunda fase corresponde à resolução do modelo inteiro com as restrições descritas anteriormente. A solução heurística gerada pelo ILS-RVND (*solHeuristica*) é utilizada como uma solução inicial para o algoritmo exato, caso não haja incompatibilidade entre essa solução com as variáveis já fixadas no valor 1. Caso exista incompatibilidade, o valor da solução heurística é usado como *UpperBound* para o algoritmo exato.

Algoritmo 7 Procedimento de duas fases executado no modelo exato.

- 1: **entrada:** $arestas, solHeuristica$ **saída:** s'^*
 - 2: $m_1 = Criar_Modelo_Relaxado()$
 - 3: $r_1 = Resolver_Modelo(m_1)$
 - 4: $m_2 = Criar_Modelo_Inteiro(arestas, r_1)$
 - 5: $s'^* = Resolver_Modelo(m_2, solHeuristica)$
-

Capítulo 4

Heurística Multi-Start Local Search para o PCVMSH

Neste Capítulo será apresentada a heurística proposta para lidar com as características do PCVMSH descritas na Seção 2.2, as quais deixam claro suas diferenças e dificuldades extras em relação ao PCVSH.

A heurística proposta para resolução do PCVMSH é uma combinação de técnicas clássicas utilizadas na área de otimização. A primeira técnica consiste na escolha de soluções de partida distintas ao longo da execução, chamada de *Multi-Start*, que garante a diversificação da solução provendo um mecanismo que permite alcançar diferentes ótimos locais, e eventualmente, o ótimo global. Porém, a utilização apenas de um mecanismo de diversificação não garante necessariamente bons resultados, sendo necessário utilizar uma estratégia de intensificação de busca por soluções melhores. Neste caso, a heurística em questão, denominada MSRVND, utiliza um procedimento de busca local capaz de melhorar iterativamente uma solução. Abordagens similares foram empregadas com sucesso em trabalhos que lidam com problemas de definições de rotas, bem como, roteamento de veículos [33, 44, 45].

A estrutura da heurística MSRVND é apresentada no Algoritmo 8. Inicialmente, ela recebe dois parâmetros que são o número máximo de iterações *iter_max* e o valor de α utilizado na construção de uma solução. A construção da solução inicial é realizada no passo 3. Por se tratar de um procedimento de geração de solução inicial determinístico, é utilizado uma única vez. A solução gerada passa pelo processo de otimização por meio do procedimento de busca local (passo 4). O *loop* principal do algoritmo (Passos 6 a 15) é executado por um número pré-determinado de vezes. Internamente, a cada iteração é escolhido um dos outros três possíveis construtores disponíveis para gerar uma

nova solução estocástica (passo 8) e uma nova solução é criada (passo 9) utilizando o construtor definido no passo anterior. A solução construída é então otimizada utilizando um procedimento de busca local (passo 10). Por fim, se e somente se a aplicação da busca local resultar em uma solução melhor que a solução global (s^*), esta será substituída pela solução corrente (passos 11 a 13). A execução do algoritmo é concluída com o retorno da melhor solução alcançada (s^*).

Algoritmo 8 Algoritmo *Multi-Start* com busca local RVND

```

1: entrada:  $iter\_max, \alpha$    saída:  $s^*$ 
2:  $s^* = \emptyset$ 
3: Construir_Solução( $s^*, Ins\_mais\_Barata$ )
4: RVND( $s^*$ )
5:  $i = 0$ 
6: enquanto ( $i < iter\_max$ ) faça
7:    $s' = \emptyset$ 
8:    $tipo \leftarrow rand(Ins\_Sequencial, Ins\_Paralela, Ins\_mais\_Barata\_LRC)$ 
9:   Construir_Solução( $s', tipo, \alpha$ )
10:  RVND( $s'$ )
11:  se ( $s'$  melhor que  $s^*$ ) então
12:     $s^* \leftarrow s'$ 
13:  fim se
14:   $i++$ 
15: fim enquanto

```

4.1 Construção de soluções

A fase de construção das soluções utiliza quatro heurísticas distintas para gerar soluções capazes de abranger diferentes características ideais para cada tipo de instância do problema. Três heurísticas são puramente estocásticas e geram soluções distintas a cada chamada. A outra constrói uma solução determinística, ou seja, gera sempre a mesma solução, caso seja executada várias vezes. Estes métodos construtivos já foram utilizados com sucesso em outros trabalhos para resolver o PCVMSH e problemas de roteamento de veículos [6, 33, 40].

A característica de seleção de hotéis intermediários para cada rota é determinante para a qualidade final das soluções [26]. Dado que as soluções a serem construídas podem ser pouco otimizadas devido às características particulares das heurísticas construtivas, antes de passar pelo processo de divisão, as rotas podem ser melhoradas com base na heurística de *Lin-Kernighan* [24]. Esse procedimento de otimização utiliza a implementação desenvolvida em [2, 12].

O método de divisão baseado no Algoritmo de *Dijkstra* [15] quase sempre garante que as viagens serão viáveis em relação a sua duração, entretanto, pode gerar um número de viagens que excede o máximo permitido para uma dada rota. Neste caso, a busca pela viabilidade quanto ao número de viagens nos métodos construtivos estocásticos é realizada pelo procedimento de busca local. Para o método construtivo determinístico, é realizada a aplicação de uma estratégia que realoca clientes que fazem parte das viagens excedentes, remanejando-os para as demais rotas/viagens. Estes clientes são realocados para posições em que representem o menor acréscimo para composição do tempo total da rota naquela solução. Formas distintas de lidar com a inviabilidade das soluções quanto ao número de viagens foram adotadas com o objetivo de viabilizar ao máximo a diversificação das soluções.

O primeiro método construtivo a ser utilizado pelo Algoritmo 8 é o método determinístico de Inserção Mais Barata (passo 3). Este método inicialmente cria p rotas, uma para cada caixeiro, iniciando e finalizando no hotel de partida. Insere para cada uma das rotas o cliente que menos incrementa o tempo total da solução. Com isso, garante que ao menos um cliente será atendido por cada caixeiro. Os demais clientes são inseridos um a um, na posição em que acarretem o menor acréscimo no tempo total gasto. Neste caso, é realizada uma busca exaustiva pela melhor posição, considerando todas as p rotas. As p rotas construídas passam pelo procedimento de otimização baseado na heurística de *Linkernighan* [24]. Posteriormente, são divididas em viagens e submetidas ao procedimento de realocação de clientes para reestabelecimento da viabilidade das solução em relação ao número de viagens por rota.

Os demais métodos construtivos são utilizados no passo 9 do Algoritmo 8 para produzir soluções distintas a cada iteração da heurística *Multi-Start*. O método Inserção Mais Barata com Lista Restrita de Candidatos (LRC) diverge do primeiro, pois pode gerar soluções diversas cada vez que é ativado. Inicialmente insere-se um cliente em cada rota de forma aleatória. Depois, armazenam-se os clientes em uma Lista de Clientes (LC) em ordem ascendente quanto ao acréscimo de tempo na rota que eles acarretam. Esse valor do acréscimo de tempo é recuperado pela função *val()*. Cria-se, a partir de LC, uma LRC de acordo com a Equação (4.1), que restringe o tamanho da lista de candidatos. O valor do parâmetro $\alpha \in \{0, 1\}$ especifica a porcentagem máxima de clientes da LC que farão parte da LRC. O valor de α deve ser cuidadosamente definido, pois se for muito baixo caracteriza uma escolha gulosa semelhante ao método determinístico de inserção mais barata e, no outro extremo, um valor muito alto faz com que a LRC contenha todos os clientes candidatos. Enquanto houver clientes na LC, a cada iteração é sorteado um

dos clientes pertencentes à LRC, que é adicionado em uma das rotas de modo a causar o menor acréscimo no tempo total da rota. Ao concluir a inserção, as rotas geradas pelo procedimento são divididas em viagens com base no Algoritmo de *Dijkstra* [15], assim como é feito nos demais métodos construtivos.

$$LRC = \{c \in LC | val(c) \leq \min(val(LC)) + \alpha[\max(val(LC)) - \min(val(LC))]\} \quad (4.1)$$

Objetivando aumentar a diversidade das soluções, dois métodos construtivos baseados no trabalho de Penna et al. [33] foram utilizados. Os métodos Inserção Paralela e Sequencial possuem similaridades com o método de inserção mais barata, porém, devido ao formato de distribuição dos clientes pelas rotas, escolhas aleatórias de clientes e de critérios de inserção, obtém-se um comportamento estocástico.

Nesses métodos, a escolha de qual cliente é inserido a cada iteração depende da definição prévia de qual critério de inserção será utilizado. Essa definição é feita de forma aleatória entre: (i) inserção mais barata com desconto do tempo necessário para o percurso de ida e volta entre o hotel inicial e o cliente ou (ii) menor tempo entre o hotel ou cliente prévio e o cliente a ser inserido.

No primeiro critério, para cada cliente ainda não inserido é calculado o tempo necessário para sua inserção em uma dada posição da rota. Do valor obtido por esse cálculo é subtraído um desconto correspondendo a duas vezes o tempo necessário para ir do hotel inicial até o cliente, multiplicado por um valor aleatoriamente definido para a variável γ dentro do intervalo discreto $\{0,00; 0,05; 0,10; \dots; 1,65; 1,70\}$. Esse desconto viabiliza a ocorrência de um equilíbrio no momento da escolha do cliente a ser inserido, não permitindo para os casos onde $\gamma \neq 0,00$, que aqueles clientes mais distantes do hotel de partida sejam sempre inseridos por último. Quanto maior for o valor de γ , maior será a contribuição do desconto, fazendo com que os clientes geograficamente mais distantes possam ser escolhidos mais brevemente.

O segundo critério é mais simples, pois consiste no cálculo do tempo necessário para viajar entre um cliente ou hotel que já faz parte da rota e o cliente que está sendo avaliado. O cálculo é realizado para todas as posições possíveis de inserção do cliente na rota.

Tanto para o método de Inserção Sequencial, quanto para o método de Inserção Paralela, será escolhido para inserção aquele cliente que obtiver o menor valor calculado pelo método de inserção selecionado.

No método de Inserção Sequencial, inicialmente seleciona-se de forma aleatória um critério de inserção dentre os dois disponíveis. Aloca-se aleatoriamente um cliente a cada rota e divide-se a quantidade de clientes pelo número de caixeiros para se definir o número de clientes que deve ter cada rota, de forma a obter um balanceamento quantitativo. Na primeira rota, para cada um dos clientes disponíveis para inserção, calcula-se de acordo com o critério de inserção previamente selecionado o valor para efetuar a inserção daquele cliente em cada uma das posições da rota corrente. O cliente que apresentar o menor valor calculado será inserido na posição em que o valor ocorreu. A rota seguinte somente receberá a inserção de clientes, quando o número de clientes na rota corrente tiver atingido o limite máximo previamente calculado que garante o balanceamento de clientes entre rotas.

O método de Inserção Paralela também inicia com a definição do critério de inserção e inserção aleatória de um cliente por rota. Porém, difere do anterior, pois no decorrer do processo de inserção dos clientes remanescentes, permite que a inserção ocorra em qualquer uma das rotas. Para cada cliente, é realizado o cálculo de acordo com o critério de inserção, considerando todas as posições possíveis em cada uma das rotas. O cliente é inserido naquela posição onde o valor calculado foi menor. Apesar de haver a possibilidade do método gerar soluções balanceadas em relação ao número de clientes, a tendência é que as soluções apresentem números de clientes distintos para cada rota.

4.2 Busca local

Apesar da heurística proposta apresentar quatro formas distintas de geração de soluções, esta característica isoladamente não é suficiente para garantir que soluções boas, ou até mesmo ótimas, sejam encontradas. Assim, utilizamos a estratégia de busca local RVND que foi empregada para desenvolver a heurística ILS-RVND apresentada no Capítulo 3. Devido às múltiplas rotas contidas em uma solução do PCVMSH, fazem-se necessárias alterações em relação ao Algoritmo 3 para execução da busca local. A busca local proposta itera individualmente pelas múltiplas rotas realizando a aplicação do RVND até que não seja mais possível efetuar melhorias internas a nenhuma das rotas contidas na solução. Sempre que essa limitação é alcançada, é realizada a tentativa de otimização da solução por meio da aplicação das estruturas de vizinhança capazes de mover clientes entre rotas distintas. Quando a possibilidade de realização de melhorias se esgotar, a melhor solução alcançada é retornada.

O funcionamento do procedimento de busca local RVND é exposto no Algoritmo 9. O algoritmo recebe como parâmetro a solução a ser otimizada, a qual é copiada para uma variável auxiliar que será utilizada durante a aplicação das estruturas de vizinhança (passo 2). Enquanto a solução atual puder ser melhorada pelas estruturas de vizinhança *inter-tours* (passos 3 a 17), o procedimento RVND é reexecutado. Para cada rota definida na solução corrente (passos 4 a 16), é sorteada a ordem de aplicação das estruturas de vizinhança *intra-tours* (passo 5), aplicando em seguida as estruturas de vizinhança que compõem o RVND na ordem pré-definida enquanto houver melhora (passos 7 a 15). Para cada estrutura de vizinhança, é verificado se algum de seus movimentos (passo 8) otimizará a solução atual. Caso afirmativo, a melhor solução é atualizada (passo 10) e o procedimento é reiniciado aplicando-se a primeira estrutura definida de acordo com a ordem de aplicação (passo 7 - 15). Caso contrário, a próxima estrutura na ordem é selecionada (passo 13). Ao final da aplicação do RVND, caso haja a melhoria da solução por meio da estrutura de vizinhança *Shift* ou *Swap*, o processo é reiniciado a partir do passo 3.

Algoritmo 9 Algoritmo de Busca local RVND para o PCVMSH.

```

1: entrada:  $s^*$    saída:  $s'$ 
2:  $s' = s^*$ 
3: repita
4:   para (cada rota em  $s'$ ) faça
5:     Ordem_Aplicação()
6:      $k = 0$ 
7:     enquanto ( $k < 6$ ) faça
8:       Busca( $s', N_{DF[k]}$ )
9:       se ( $s'$  melhor que  $s^*$ ) então
10:         $s^* = s'$ 
11:         $k = 0$ 
12:       senão
13:         $k = k + 1$ 
14:       fim se
15:     fim enquanto
16:   fim para
17: até (encerrar melhorias possíveis com  $Shift(s')$  ou  $Swap(s')$ )

```

Complementarmente, a cada chamada realizada no passo 8 do Algoritmo 9, o Algoritmo 10 é executado para realizar a avaliação de cada movimento passível de ser aplicado na estrutura de vizinhança. A função de avaliação de uma dada solução resultante da aplicação de uma estrutura de vizinhança é apresentada na Equação (4.2). Consiste na soma do tempo total necessário para atender a todos os clientes presentes nas D viagens contidas em cada uma das P rotas, penalizando aquelas viagens que excederem o tempo

máximo permitido L e as rotas que utilizarem mais viagens do que o permitido. A penalização é alcançada por meio do somatório de inviabilidade acumulada nas D viagens de cada uma das rotas e/ou pelo número de viagens excedentes (*exced*) multiplicados pela constante $\omega = 10000$, a qual é suficientemente grande em relação às instâncias teste consideradas nesta tese. Caso o melhor movimento contribua para a melhora da solução, essa solução é atualizada e otimizada pela estrutura de vizinhança *intra-trip Or-opt* (passo 6).

$$F(s) = \sum_{p=1}^P \sum_{d=1}^D (\text{tempo}_{pd}) + \omega \sum_{p=1}^P \sum_{d=1}^D \max(0, \text{tempo}_{pd} - L) + \omega \sum_{p=1}^P \text{exced}_p \quad (4.2)$$

Uma vez que houver melhora da solução no passo 6, o procedimento descrito no Algoritmo 10 é reiniciado a partir do passo 3 até que não seja mais possível otimizar a solução.

Algoritmo 10 Aplicação de estruturas de vizinhança internamente ao RVND.

```

1: entrada:  $s^*$ ,  $N_k$    saída:  $s'$ 
2:  $s' = s^*$ ;
3: enquanto (melhoras forem encontradas) faça
4:    $s^* = \operatorname{argmin}_{s \in N_k(s')} F(s)$ 
5:   se ( $s^*$  melhor que  $s'$ ) então
6:      $s' = s^*$  otimizada com Or-opt
7:   fim se
8: fim enquanto

```

4.3 Estruturas de vizinhança

Internamente ao processo de busca local, foram utilizadas nove estruturas de vizinhança. Todas essas estruturas foram amplamente utilizadas com sucesso em diversas variantes do PCV [8, 9] e também do PRV [33]. Estas estruturas podem manipular clientes ou hotéis e são classificadas considerando sua localidade de aplicação: internamente a uma viagem (*intra-trip*), entre viagens distintas (*inter-trip*) e entre rotas distintas (*inter-tour*). Todas as estruturas de vizinhança utilizadas foram previamente detalhadas na Seção 2.1 e Seção 2.2.

Dentre as nove estruturas de vizinhança utilizadas, sete são aplicadas apenas internamente a uma rota específica de cada vez. De acordo com seu escopo de atuação são divididas em operações *intra-trip* (*2-opt* e *Or-opt*) e *inter-trip* (*Relocate*, *Exchange*, *InsertHotel*, *RemoveHotel* e *ChangeHotels*). A estrutura *RemoveHotel* é tecnicamente equivalente à *JoinTrips* apresentada na Seção 2.1. As duas outras estruturas de vizinhança

correspondem a movimentos *inter-tour*, realocando (*Shift*) ou trocando (*Swap*) clientes entre rotas distintas.

No momento da aplicação de cada estrutura, todas as trocas possíveis sejam elas de clientes ou de hotéis são analisadas. Para que a busca seja guiada efetivamente para uma solução vizinha, esta deve ser melhor que a solução corrente, considerando sempre a solução que mais contribua em relação à melhora da solução atual (*Best Improvement*).

Capítulo 5

Experimentos Computacionais para o PCVSH

Neste capítulo, são apresentados os resultados obtidos por meio das abordagens descritas no Capítulo 3 para resolução do PCVSH. A avaliação das heurísticas é realizada considerando conjuntos de instâncias disponíveis na literatura [4, 46].

5.1 Resultados para instâncias propostas por [46]

De forma recorrente, os trabalhos previamente publicados acerca do PCVSH, quase em sua totalidade, fazem uso dos grupos de instâncias propostos por [46] para investigar a eficiência de suas metodologias, bem como para comparar as diferentes abordagens. A proposição de novas instâncias demandaria a reprodução de algoritmos previamente propostos para fins de comparações igualitárias, o que nem sempre é possível devido à particularidades de implementação (estruturas de dados ou otimização de código específicas) que nem sempre são relatadas explicitamente na estrutura textual dos trabalhos.

As características destas 131 instâncias utilizadas são descritas a seguir:

- **SET1:** O primeiro grupo é composto por um total de 16 instâncias, sendo as seis primeiras correspondentes ao PRV com janela de tempo [38], e as outras dez inicialmente planejadas para o PRV com múltiplos depósitos e janelas de tempo [13]. As seis primeiras possuem 100 clientes, enquanto o número de clientes das demais encontra-se definido no intervalo [48, 288]. Na construção das instâncias, a janela de tempo é desconsiderada e o tempo limite de uma viagem é definido pelo tempo de fechamento da janela referente ao primeiro hotel. Seis hotéis são disponibilizados em localidades que também definem a posição de clientes (1, 11, 21, 31 e 41). Tal

definição não implica que esses clientes possam ser desconsiderados, mas sim que a visitação a determinado cliente/hotel possa ocorrer sem que necessariamente seja acrescentado um tempo decorrente do deslocamento entre tais facilidades.

- **SET2:** As instâncias contidas neste segundo grupo possuem tamanho reduzido e foram estabelecidas a partir do SET1. Os subgrupos consistem nos 10, 15, 30 ou 40 primeiros clientes extraídos a partir de 13 das 16 instâncias previamente propostas. Para estes subgrupos, apenas dois hotéis estão disponíveis, sendo o primeiro definido em uma posição aleatória, e o segundo, considerado um hotel extra, definido na mesma posição do cliente 1. Segundo [46], este grupo de instâncias foi criado para possibilitar o alcance da solução ótima por meio do uso de um solver comercial (CPLEX^{®1}), tendo assim soluções ótimas para serem comparadas àquelas obtidas pela heurística.
- **SET3:** O terceiro grupo é considerado mais complexo, possuindo instâncias com número de clientes variando entre 51 e 1002, bem como 3, 5 ou 10 hotéis disponíveis. Este grupo foi derivado de uma seleção de instâncias clássicas do PCV, logo, o tempo total de uma solução apresentada para o PCVSH é limitado ao valor associado à solução ótima do PCV. O limite de tempo para uma viagem foi definido de forma que seja possível sempre alcançar a solução do PCV, sendo os hotéis definidos para localidades onde já existe um cliente.
- **SET4:** O último grupo de instâncias utiliza 15 instâncias do SET3, associando 10 hotéis à localização exata dos clientes 1, 6, 11, 16, 21, 25, 31, 35, 41 e 45. O tempo limite de uma viagem é definido utilizando-se o tempo total da solução do PCV dividida por 5. Devido à quantidade de hotéis disponíveis e a forma como foram definidas as localidades dos hotéis, as instâncias pertencentes a este grupo são classificadas como as mais difíceis de serem resolvidas. Os melhores valores para soluções do PCVSH são limitados, assim como no SET3, pela solução ótima do PCV.

Até o presente momento, o trabalho de [26] composto por uma abordagem híbrida entre Algoritmo Memético e Programação dinâmica (HDM) e o trabalho de [8] que apresenta um Algoritmo Memético (MA) possuem a maioria dos resultados com melhor qualidade reportados para as instâncias listadas acima. Complementarmente, o trabalho de [9] possui soluções de qualidade razoável quando comparadas às citadas anteriormente, que

¹<https://www.ibm.com/br-pt/products/ilog-cplex-optimization-studio>

são alcançadas utilizando uma pequena fração do tempo computacional. Juntos, esses trabalhos alcançam as melhores soluções conhecidas em relação à qualidade e ao tempo computacional utilizado por elas.

No decorrer deste Capítulo, para definição dos melhores resultados que aparecem em negrito, são analisados em ordem de prioridade os seguintes critérios: menor número de viagens, menor *Gap* e menor tempo computacional. Em caso de empate em um critério, os critérios seguintes são avaliados sucessivamente.

5.1.1 Impacto da perturbação variável e das estruturas de vizinhança na heurística ILS-RVND

Os testes conduzidos nesta seção foram parcialmente publicados em [42] e verificam o impacto da estratégia de perturbação variável e a necessidade de serem utilizadas todas as estruturas de vizinhança pela busca local na heurística ILS-RVND descrita no Capítulo 3. O algoritmo foi codificado em C++, compilado com o compilador g++ (versão 7.5.0) usando a tag “-Ofast” e executado em uma máquina com processador Intel® Core™ i5-7200U 2.50 GHz com 8 GB de memória RAM. O sistema operacional utilizado foi o Ubuntu Linux 18.04.

Os resultados apresentados nesta subseção foram obtidos a partir da execução da heurística ILS-RVND apresentada no Capítulo 3, porém, ao invés de considerar como critério de parada um tempo computacional máximo equivalente ao de [26], utilizou-se um número máximo de iterações sem melhora (i_{max}). O outro parâmetro utilizado foi o número de iterações sem aplicação do procedimento de perturbação (n_{pert}). Por fim, foi definido o valor do parâmetro v utilizado para controlar a intensidade da aplicação do procedimento de perturbação. Os parâmetros citados foram calibrados executando 10000 iterações no IRACE (*Iterated Racing for Automatic Algorithm Configuration*) [27], utilizando 15 instâncias representativas escolhidas dentre aquelas disponibilizadas por [46]. Foram considerados intervalos de valores para i_{max} compreendidos entre [20, 40]. Para n_{pert} os valores testados foram entre [3, 7]. Os valores testados para o parâmetro v foram inteiros no intervalo [7, 12]. Após realização de testes, a melhor configuração retornada pelo IRACE foi $i_{max} = 39$, $n_{pert} = 4$ e $v = 11$.

A estratégia de perturbação utilizada por [9] considera um valor percentual fixo para alteração da solução independente das características apresentadas pelas instâncias, como por exemplo, o número de clientes. A heurística ILS-RVND utiliza um valor percentual variável definido de acordo com o número de clientes a serem atendidos em cada instância.

Para tanto, por se tratar de uma estratégia nunca testada para o PCVSH, experimentos foram realizados para demonstrar a efetividade da técnica.

A heurística ILS-RVND foi executada em duas versões, sendo uma utilizando a taxa percentual fixa definida por [9], e a outra utilizando a taxa percentual variável conforme apresentado na Subseção 3.1.4. Neste teste, ambas abordagens foram executadas 30 vezes, utilizando sementes de 1 a 30 e considerando todas as instâncias propostas por [46]. Os melhores resultados reportados para cada instância foram sintetizados e comparados com a Melhor Solução Conhecida (MSC), conforme apresentado nas Tabelas 5.1 e 5.2.

A primeira coluna nessas tabelas apresenta a origem dos resultados, onde "MSC" indica a Melhor Solução Conhecida relativa ao tempo total da rota para o PCVSH dentre aquelas publicadas na literatura, "ILS-RVND pert. variável" indica que os resultados foram obtidos por ILS-RVND com perturbação variável, e "ILS-RVND pert. fixa" indica que os resultados foram obtidos por ILS-RVND com perturbação fixa. As próximas colunas são agrupadas de três em três sub-colunas para representar os resultados por grupo de instâncias. A primeira sub-coluna " $\#V$ " traz a soma total de viagens necessárias para resolver todas as instâncias do referido grupo, a segunda " TC " indica o tempo computacional médio em segundos, e a última " Gap " exibe o desvio percentual do resultado em relação à MSC.

O Gap é calculado pela Equação (5.1) a partir dos resultados R_1 e R_2 , onde R_1 corresponde à MSC e R_2 corresponde a um dos resultados descritos anteriormente.

$$Gap = 100 \times \frac{R_1 - R_2}{R_1} \quad (5.1)$$

Quanto menos negativo é o resultado de Gap , melhor é a qualidade da solução obtida pela estratégia utilizada. Os resultados em negrito apresentados nas Tabelas 5.1 e 5.2 destacam qual estratégia obteve os melhores resultados em relação ao Gap , uma vez que as abordagens apresentam número de viagens idêntico em cada grupo. Os resultados para os subgrupos com 10 e 15 clientes pertencentes ao SET2 foram omitidos, pois não houve nenhuma diferença de qualidade da solução entre as estratégias comparadas. Para os demais grupos, o número de viagens utilizadas pela heurística ILS-RVND com ambos os tipos de perturbação foram idênticos. Os $Gaps$ obtidos pela estratégia de perturbação variável foram melhores ou iguais aos $Gaps$ obtidos pela estratégia de perturbação fixa para todos os grupos, exceto no grupo SET3_5. Em relação ao tempo computacional médio, a estratégia variável necessitou para todos os grupos, exceto para o SET2, tempo

computacional médio menor em relação à sua variante com perturbação fixa. Tempos computacionais médios para a MSC possuem grande discrepância em relação às demais estratégias, pois algumas das melhores soluções são obtidas por meio de métodos exatos, os quais demandam elevado tempo computacional.

Tabela 5.1: Resultados comparativos em relação ao número de viagens, tempo computacional médio e *Gap* entre ILS-RVND com perturbação fixa e variável(parte 1).

Estratégia	SET1			SET2 30			SET2 40		
	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)
MSC	77	3479,9	-	25	3323,2	-	31	4984,7	-
ILS-RVND pert. variável	77	2,1	-0,28	25	0,0	-1,30	32	0,0	-2,62
ILS-RVND pert. fixa	77	2,1	-0,43	25	0,0	-1,30	32	0,0	-2,62

Tabela 5.2: Resultados comparativos em relação ao número de viagens, tempo computacional médio e *Gap* entre ILS-RVND com perturbação fixa e variável(parte 2).

Estratégia	SET3 3			SET3 5			SET3 10			SET4		
	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)
MSC	64	268,9	-	97	303,4	-	169	260,5	-	92	749,8	-
ILS-RVND pert. variável	64	5,4	-0,01	96	4,0	-0,06	169	5,1	-0,28	91	53,7	-0,42
ILS-RVND pert. fixa	64	34,7	-0,02	96	39,8	-0,01	169	55,8	-0,42	91	57,7	-0,95

Após a adoção da perturbação variável, outro teste foi conduzido objetivando checar se todas as estruturas de vizinhança são realmente necessárias para otimização da solução na busca local. O experimento consiste na verificação da qualidade da solução da heurística ILS-RVND, desabilitando as estruturas de vizinhança uma a uma. Novamente, para chegar aos resultados reportados nas Tabelas 5.3 e 5.4, foram executadas todas as instâncias propostas por [46].

A estrutura das Tabelas 5.3 e 5.4 é similar à das tabelas apresentadas previamente. Na primeira coluna, indica-se a heurística que foi executada, sendo que na primeira linha executou-se a heurística com todas as estruturas de vizinhança, e nas demais linhas a heurística foi executada com a remoção da estrutura destacada.

Devido à similaridade dos resultados alcançados nos subgrupos do SET2 com 10 e 15 clientes, esses foram omitidos. Observando os resultados com base nos objetivos do PCVSH, nota-se que a heurística ILS-RVND com todas as estruturas de vizinhança obtém *Gaps* iguais ou menos negativos para quatro dos sete grupos. Não é possível determinar qual estrutura de vizinhança possui a maior influência na qualidade das soluções em relação ao tempo total gasto, pois não existe um comportamento padrão devido aos variados níveis de dificuldade encontrados nos grupos de instâncias. As variações do tempo computacional médio, apesar de importantes, somente demonstram que a estrutura de vizinhança *Relocate* demanda grande parte do tempo computacional na construção de uma solução otimizada. A versão sem a estrutura *ChangeHotels*, apesar de apresentar

soluções médias melhores para dois grupos, não pode ser simplesmente removida da heurística ILS-RVND, pois sua remoção exigiria que toda solução inicial fosse composta pela sequência ótima de hotéis a serem utilizados na rota.

As colunas em negrito demonstram quais critérios foram avaliados para definir que aquela solução é superior às demais. No caso da ocorrência de empate nos três critérios, mais de uma solução aparecerá em negrito.

Tabela 5.3: Diferença entre soluções com todas estruturas de vizinhança e exclusão individualizada (parte 1).

Estratégia	SET1			SET2 30			SET2 40		
	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)
ILS-RVND	77	2,1	-0,28	25	0,0	-1,30	32	0,0	-2,62
Sem Relocate	79	0,3	-0,72	25	0,0	-1,30	33	0,0	-2,73
Sem Exchange	78	1,1	-0,25	25	0,0	-1,30	32	0,0	-2,62
Sem JoinTrips	79	1,7	-0,24	27	0,0	-1,02	34	0,0	-2,62
Sem ChangeHotels	77	1,4	-0,27	25	0,0	-1,34	32	0,0	-2,65
Sem 2-opt	77	1,7	-0,30	25	0,0	-1,30	32	0,0	-2,64
Sem Or-opt	77	1,7	-0,30	25	0,0	-1,30	32	0,0	-2,62

Tabela 5.4: Diferença entre soluções com todas estruturas de vizinhança e exclusão individualizada (parte 2).

Estratégia	SET3 3			SET3 5			SET3 10			SET4		
	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)	#V	TC(s)	Gap(%)
ILS-RVND	64	5,4	-0,01	96	4,0	-0,06	169	5,1	-0,28	91	53,3	-0,42
Sem Relocate	65	2,0	-0,09	97	1,5	-0,09	172	1,7	-0,12	97	8,6	-5,04
Sem Exchange	64	3,8	-0,01	96	4,4	-0,01	169	3,3	-0,38	91	60,9	-0,92
Sem JoinTrips	66	11,7	-0,08	97	5,4	-0,09	172	4,9	-0,04	101	26,4	-0,11
Sem ChangeHotels	64	4,3	-0,01	96	3,9	-0,06	169	4,3	-0,31	91	59,6	-0,67
Sem 2-opt	64	6,0	-0,02	96	4,5	-0,06	169	4,7	-0,32	91	32,3	-0,60
Sem Or-opt	64	3,3	-0,04	97	4,1	-0,06	169	5,0	-0,34	91	37,1	-0,72

Por fim, o último resultado comparado é a soma total de viagens necessárias por grupo de instâncias, que possui grande relevância uma vez que de acordo com o objetivo do PCVSH, minimizar o número de viagens de uma rota é prioridade. Logo, a heurística ILS-RVND contendo todas as estruturas de vizinhança é capaz de encontrar a menor soma em relação ao número de viagens para todos os grupos de instâncias. Assim, os bons valores de *Gaps* médios obtidos juntamente com a definição de rotas com número de viagens menor corroboram com a necessidade do uso combinado de todas as estruturas de vizinhança.

5.1.2 Resultados obtidos pela heurística ILS-RVND

Esta Seção apresenta os resultados obtidos pela heurística ILS-RVND utilizando a perturbação variável e todas as estruturas de vizinhança. Cada instância foi executada 30 vezes

com sementes definidas entre 1 e 30 para os números aleatórios, e foram considerados o melhor resultado obtido e a média dentre as 30 execuções realizadas. Foram utilizados os mesmos parâmetros usados na Subseção anterior, a mesma máquina e o mesmo critério de parada, ou seja, número de iterações consecutivas sem melhora. Devido a inferioridade da capacidade de processamento da máquina utilizada para execução da heurística e para garantir uma comparação justa em relação ao tempo computacional gasto na heurística PLS, o tempo computacional reportado nas tabelas desta subseção correspondem ao tempo computacional utilizado pela heurística multiplicado por 0,5572. Essa proporção foi obtida com base no procedimento descrito em [10] utilizado para comparar testes em máquinas com capacidades computacionais distintas.

O objetivo desses testes computacionais foi priorizar o alcance de soluções de boa qualidade utilizando baixos tempos computacionais. Por isso, optamos por comparar os melhores resultados obtidos à MSC e aos resultados obtidos pela heurística PLS [9]. Dessa forma, mostramos quão distante a solução alcançada está da melhor conhecida, bem como comparamos a heurística diretamente com uma abordagem que visa puramente o alcance de soluções de qualidade utilizando pouco tempo computacional.

As Tabelas 5.5 a 5.13 trazem em sua primeira coluna o nome da instância executada. Com exceção das tabelas referentes ao grupo SET2, todas as demais trazem na segunda coluna o número de clientes que compõem a instância. As próximas três colunas indicam que os resultados foram obtidos pela MSC, heurística ILS-RVND e PLS. Para cada uma destas colunas, existem três colunas internas que mostram os resultados especificando o número de viagens necessárias para resolver a instância (“#V”), o tempo total da rota na solução (“TTotal”) e o tempo computacional em segundos (“TC”). Na heurística ILS-RVND, devido às 30 execuções que são feitas para cada instância, as tabelas também trazem uma coluna contendo o tempo total médio calculado com base nas 30 execuções (“TMed”). Para algumas instâncias, pode ocorrer do valor apresentado em TMed ser menor que o apresentado em TTotal, isso ocorre devido à soluções que utilizam um maior número de viagens, as quais tendem a ter menores tempos totais de rota. Adicionalmente, para as duas heurísticas comparadas é apresentado o desvio percentual (“Gap”) entre a abordagem e a Melhor Solução Conhecida. O valor mostrado na coluna *Gap* é calculado conforme descrito na Subseção 5.1.1, por meio da Equação (5.1).

Os resultados para o grupo SET1 são apresentados na Tabela 5.5. A heurística ILS-RVND encontra soluções melhores ou iguais a MSC para cinco instâncias, enquanto a PLS somente obtém solução equivalente à MSC em uma instância. A heurística ILS-

RVND apresenta soma do número de viagens igual em relação à MSC, pois apesar de ter utilizado um viagem extra na instância *pr05*, conseguiu reduziu o número de viagens na instância *c101*. A PLS necessitou de uma viagem extra para a instância *pr05*. O tempo computacional médio observado apresenta grande diferença em relação à MSC, pois apesar da MSC ser majoritariamente oriunda de abordagens heurísticas, para algumas instâncias a melhor solução somente foi alcançada por métodos exatos, elevando assim o tempo computacional médio reportado.

Tabela 5.5: Resultados computacionais ILS-RVND para instâncias do SET1.

Instância	N	MSC				ILS-RVND					PLS			
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TC(s)	Gap(%)
c101	100	9	9591,1	24,0		8*	9643,0	9632,1	0,1	-0,54	9	9596,9	0,1	-0,06
r101	100	8	1695,5	24,5		8	1708,2	1724,7	0,2	-0,75	8	1717,4	0,2	-1,29
rc101	100	8	1672,6	18000,0		8	1673,1	1674,9	0,2	-0,03	8	1674,3	0,2	-0,10
c201	100	3	9559,7	18000,0		3	9561,8	9565,5	0,1	-0,02	3	9563,1	0,1	-0,04
r201	100	2	1642,8	12,4		2	1642,6	1646,7	0,1	0,01	2	1648,1	0,1	-0,32
rc201	100	2	1642,7	0,1		2	1642,7	1643,4	0,1	0,00	2	1644,3	0,2	-0,10
pr01	48	2	1412,2	0,1		2	1412,2	1412,2	0,0	0,00	2	1412,2	0,0	0,00
pr02	96	3	2543,3	0,1		3	2543,3	2551,7	0,0	0,00	3	2551,3	0,2	-0,31
pr03	144	4	3404,2	48,6		4	3429,1	3435,4	0,4	-0,73	4	3421,1	0,3	-0,50
pr04	192	5	4215,3	166,2		5	4217,4	4239,3	0,3	-0,05	5	4217,4	0,6	-0,05
pr05	240	5	4948,9	340,4		6	4983,6	5002,4	5,0	-0,70	6	4974,7	1,1	-0,52
pr06	288	7	5960,9	337,6		7	5972,2	6000,9	5,6	-0,19	7	6032,0	1,6	-1,19
pr07	72	3	2069,6	18000,0		3	2070,3	2070,3	0,0	-0,03	3	2070,3	0,0	-0,03
pr08	144	4	3367,7	66,9		4	3386,6	3404,8	0,2	-0,56	4	3399,9	0,4	-0,96
pr09	216	5	4414,9	234,7		5	4451,1	4462,3	3,5	-0,82	5	4445,7	1,1	-0,70
pr10	288	7	5932,0	422,8		7	5937,6	5989,9	3,4	-0,09	7	5991,5	2,4	-1,00
Soma/Média		77		3479,9		77			1,2	-0,28	78		0,5	-0,45

As Tabelas 5.6 a 5.9 apresentam os resultados para o grupo SET2 que contém o menor número de clientes. As Tabelas 5.6 e 5.7 mostram que todos os resultados obtidos equivalem à MSC e demandaram tempos computacionais inferiores a um segundo em todas as instâncias. Para o subgrupo contendo 30 clientes, os resultados são apresentados na Tabela 5.8 e demonstram que a heurística ILS-RVND é capaz de retornar soluções com o mesmo número de viagens que a MSC para todas as instâncias. O mesmo não ocorre para a PLS, sendo necessária uma viagem a mais na instância *rc101.k30*. O tempo computacional é similar para a heurística ILS-RVND e PLS. Com exceção de duas instâncias, o valor de *Gap* é igual a 0 para as duas heurísticas, indicando que ambas atingiram o melhor valor do tempo total de rota conhecido. Na instância *rc101.k30*, PLS apresenta um valor de *Gap* significativamente melhor que ILS-RVND, mas com uma viagem a mais.

Os resultados para o último subgrupo de instâncias do SET2 são apresentados na Tabela 5.9. Comparado à MSC, a heurística ILS-RVND necessita de uma viagem a mais para retornar uma solução na instância *rc101*, enquanto a PLS necessita duas viagens extras. Os tempos computacionais são similares entre as duas abordagens e o *Gap* médio ligeiramente favorável à heurística proposta ILS-RVND. Dentre as 47 soluções ótimas

Tabela 5.6: Resultados computacionais ILS-RVND para instâncias do SET2 com 10 clientes.

Instância	MSC			ILS-RVND					PLS			
	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TC(s)	Gap(%)
c101.k10	1	955,1	<1	1	955,1	955,1	<1	0,00	1	955,1	<1	0,00
r101.k10	2	272,8	<1	2	272,8	272,8	<1	0,00	2	272,8	<1	0,00
rc101.k10	1	237,5	<1	1	237,5	237,5	<1	0,00	1	237,5	<1	0,00
pr01.k10	1	426,6	<1	1	426,6	426,6	<1	0,00	1	426,6	<1	0,00
pr02.k10	1	661,9	<1	1	661,9	661,9	<1	0,00	1	661,9	<1	0,00
pr03.k10	1	553,3	<1	1	553,3	553,3	<1	0,00	1	553,3	<1	0,00
pr04.k10	1	476,4	<1	1	476,4	476,4	<1	0,00	1	476,4	<1	0,00
pr05.k10	1	528,9	<1	1	528,9	528,9	<1	0,00	1	528,9	<1	0,00
pr06.k10	1	597,4	<1	1	597,4	597,4	<1	0,00	1	597,4	<1	0,00
pr07.k10	1	670,2	<1	1	670,2	670,2	<1	0,00	1	670,2	<1	0,00
pr08.k10	1	573,4	<1	1	573,4	573,4	<1	0,00	1	573,4	<1	0,00
pr09.k10	1	645,5	<1	1	645,5	645,5	<1	0,00	1	645,5	<1	0,00
pr10.k10	1	461,5	<1	1	461,5	461,5	<1	0,00	1	461,5	<1	0,00
Soma/Média	14		<1	14			<1	0,00	14		<1	0,00

Tabela 5.7: Resultados computacionais ILS-RVND para instâncias do SET2 com 15 clientes.

Instância	MSC			ILS-RVND					PLS			
	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TC(s)	Gap(%)
c101.k15	2	1452,2	<1	2	1452,2	1452,2	<1	0,00	2	1452,2	<1	0,00
r101.k15	2	379,8	<1	2	379,8	380,1	<1	0,00	2	379,8	<1	0,00
rc101.k15	2	303,2	<1	2	303,2	303,2	<1	0,00	2	303,2	<1	0,00
pr01.k15	1	590,4	<1	1	590,4	590,4	<1	0,00	1	590,4	<1	0,00
pr02.k15	1	745,6	<1	1	745,6	745,6	<1	0,00	1	745,6	<1	0,00
pr03.k15	1	632,9	<1	1	632,9	632,9	<1	0,00	1	632,9	<1	0,00
pr04.k15	1	683,4	<1	1	683,4	683,4	<1	0,00	1	683,4	<1	0,00
pr05.k15	1	621,2	<1	1	621,2	621,2	<1	0,00	1	621,2	<1	0,00
pr06.k15	1	685,2	<1	1	685,2	685,2	<1	0,00	1	685,2	<1	0,00
pr07.k15	1	795,3	<1	1	795,3	795,3	<1	0,00	1	795,3	<1	0,00
pr08.k15	1	707,2	<1	1	707,2	707,2	<1	0,00	1	707,2	<1	0,00
pr09.k15	1	771,7	<1	1	771,7	771,7	<1	0,00	1	771,7	<1	0,00
pr10.k15	1	611,9	<1	1	611,9	611,9	<1	0,00	1	611,9	<1	0,00
Soma/Média	16		<1	16			<1	0,00	16		<1	0,00

Tabela 5.8: Resultados computacionais ILS-RVND para instâncias do SET2 com 30 clientes.

Instância	MSC			ILS-RVND					PLS			
	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TC(s)	Gap(%)
c101.k30	3	2829,4	21600,0	3	2863,2	2863,6	<1	-1,19	3	2863,6	<1	-1,21
r101.k30	3	655,2	<1	3	655,2	655,4	<1	0,00	3	655,2	<1	0,00
rc101.k30	3	610,0	21600,0	3	705,5	697,2	<1	-15,66	4	683,8	<1	-12,10
pr01.k30	1	964,8	<1	1	964,8	964,8	<1	0,00	1	964,8	<1	0,00
pr02.k30	2	1078,3	<1	2	1078,3	1080,4	<1	0,00	2	1078,3	<1	0,00
pr03.k30	1	952,5	<1	1	952,5	952,5	<1	0,00	1	952,5	<1	0,00
pr04.k30	2	1091,6	<1	2	1091,6	1091,6	<1	0,00	2	1091,6	<1	0,00
pr05.k30	1	924,7	<1	1	924,7	924,7	<1	0,00	1	924,7	<1	0,00
pr06.k30	2	1063,2	<1	2	1063,2	1063,2	<1	0,00	2	1063,2	<1	0,00
pr07.k30	2	1130,4	<1	2	1130,4	1130,4	<1	0,00	2	1130,4	<1	0,00
pr08.k30	2	1006,2	<1	2	1006,2	1006,2	<1	0,00	2	1006,2	<1	0,00
pr09.k30	2	1091,4	<1	2	1091,4	1091,4	<1	0,00	2	1091,4	<1	0,00
pr10.k30	1	918,9	<1	1	918,9	918,9	<1	0,00	1	918,9	<1	0,00
Soma/Média	25		3323,2	25			<1	-1,30	26		<1	-1,02

conhecidas para as instâncias contidas no SET2, a heurística ILS-RVND foi capaz de obter todas elas, enquanto a PLS obteve uma a menos (*pr07.k40*).

Tabela 5.9: Resultados computacionais ILS-RVND para instâncias do SET2 com 40 clientes.

ntcs.

	MSC				ILS-RVND					PLS			
Instância	#V	TTTotal	TC(s)		#V	TTTotal	TMed	TC(s)	Gap(%)	#V	TTTotal	TC(s)	Gap(%)
c101.k40	4	3817,5	21600,0		4	3866,1	3869,2	<1	-1,27	4	3867,3	<1	-1,30
r101.k40	4	842,9	21600,0		4	862,8	873,9	<1	-2,36	4	873,5	<1	-3,63
rc101.k40	3	652,1	21600,0		4	850,3	875,6	<1	-30,39	5	870,8	<1	-33,54
pr01.k40	2	1160,5	<1		2	1160,5	1160,5	<1	0,00	2	1160,5	<1	0,00
pr02.k40	2	1336,9	<1		2	1336,9	1336,9	<1	0,00	2	1336,9	<1	0,00
pr03.k40	2	1303,4	<1		2	1303,4	1303,4	<1	0,00	2	1303,4	<1	0,00
pr04.k40	2	1259,5	<1		2	1259,5	1259,5	<1	0,00	2	1259,5	<1	0,00
pr05.k40	2	1200,7	<1		2	1200,7	1200,7	<1	0,00	2	1200,7	<1	0,00
pr06.k40	2	1242,9	<1		2	1242,9	1242,9	<1	0,00	2	1242,9	<1	0,00
pr07.k40	2	1407,0	<1		2	1407,0	1410,5	<1	0,00	2	1410,3	<1	-0,23
pr08.k40	2	1222,2	<1		2	1222,2	1222,2	<1	0,00	2	1222,2	<1	0,00
pr09.k40	2	1284,4	<1		2	1284,4	1284,4	<1	0,00	2	1284,4	<1	0,00
pr10.k40	2	1200,4	<1		2	1200,4	1200,4	<1	0,00	2	1200,4	<1	0,00
Soma/Média	31		4984,7		32			<1	-2.62	33		<1	-2.98

Resultados para instâncias com uma maior variabilidade no número de clientes foram obtidos utilizando os grupos SET3 e SET4.

No grupo SET3, as mesmas instâncias são utilizadas para formar três subgrupos distintos, os quais diferem em relação ao número de hotéis disponíveis. Estas instâncias foram criadas de tal maneira que o valor ótimo do tempo total da rota é a solução do Problema do Caixeiro Viajante para a instância. Por isso, foi incluída a coluna "PCV" com esse valor nas tabelas que apresentam os resultados para este grupo. Neste caso, para cálculo do valor de *Gap* foi utilizado o valor do PCV. É importante ressaltar que nem sempre o tempo total da rota da solução MSC para o PCVSH equivale ao valor ótimo encontrado para o PCV.

Os resultados para o grupo SET3 com 3 hotéis disponíveis são apresentados na Tabela 5.10. A heurística ILS-RVND apresenta qualidade de solução muito próxima daquela alcançada pelo PCV, tendo ligeira desvantagem apenas nas instâncias *a280* e *pcb442*. Para a instância *a280* é estabelecida uma solução melhor que a MSC. A soma do número de viagens necessárias é equivalente à soma de viagens encontrada para a MSC do PCVSH. A PLS não conseguiu alcançar a solução do PCV em três instâncias, para as quais também demandou uma viagem extra em comparação com a MSC do PCVSH. Os valores observados de *Gap* médio entre as duas abordagens possuem maior divergência, sendo favorável à heurística ILS-RVND. O tempo computacional necessário para a heurística ILS-RVND foi similar ao tempo de PLS na maioria das instâncias, sendo um pouco acima do dobro somente em uma instância.

Tabela 5.10: Resultados computacionais ILS-RVND para instâncias do SET3 com 3 hotéis.

Instância	N	PCV	MSC			ILS-RVND					PLS			
			#V	TTot	TC(s)	#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TC(s)	Gap(%)
eil51	51	426	4	426	3,0	4	426	426,5	0,0	0,00	4	426	0,0	0,00
berlin52	52	7542	4	7542	3,0	4	7542	7542,0	0,0	0,00	4	7542	0,0	0,00
st70	70	675	4	675	3,1	4	675	675,0	0,0	0,00	4	675	0,0	0,00
eil76	76	538	4	538	3,1	4	538	554,6	0,0	0,00	5	556	0,0	-3,35
pr76	76	108159	4	108159	10,2	4	108159	108159,0	0,0	0,00	4	108159	0,1	0,00
kroA100	100	21282	4	21282	10,3	4	21282	21282,0	0,0	0,00	4	21282	0,1	0,00
kroC100	100	20749	4	20749	10,2	4	20749	20749,0	0,0	0,00	4	20749	0,0	0,00
kroD100	100	21294	4	21294	10,4	4	21294	21294,0	0,0	0,00	4	21294	0,1	0,00
rd100	100	7910	4	7910	10,7	4	7910	7910,0	0,0	0,00	4	7910	0,1	0,00
eil101	101	629	4	629	10,6	4	629	629,0	0,0	0,00	4	629	0,1	0,00
lin105	105	14379	4	14379	10,9	4	14379	14379,0	0,0	0,00	4	14379	0,1	0,00
ch150	150	6528	4	6528	10,7	4	6528	6547,6	0,1	0,00	4	6528	0,2	0,00
tsp225	225	3916	4	3916	10,2	4	3916	3916,0	0,3	0,00	4	3916	0,5	0,00
a280	280	2579	4	2583	235,3	4	2582	2622,4	2,1	-0,12	5	2615	0,8	-1,40
pcb442	442	50778	4	50778	693,2	4	50820	51493,7	3,9	-0,08	5	51144	3,7	-0,72
pr1002	1002	259045	4	259045	3266,8	4	259045	263576,8	41,8	0,00	4	259045	34,5	0,00
Soma/Média			64		268,9	64			3,0	-0,01	67		2,5	-0,34

O subgrupo de instâncias com 5 hotéis disponíveis tem seus resultados demonstrados na Tabela 5.11. A heurística ILS-RVND apresenta comportamento similar ao registrado para o subgrupo com 3 hotéis, alcançando um valor de *Gap* médio que confirma a semelhança entre suas soluções e aquelas observadas para o PCV. Para a instância *a280*, única a relatar um *Gap* negativo, o valor de -0,97 representa grande superioridade quando comparado ao pior valor (-5,20) obtido pela PLS na instância *eil75*. A soma total de viagens é reduzida em uma unidade em relação à reportada para a MSC do PCVSH, diferente da heurística PLS, que possui soma contabilizando uma viagem extra. O tempo computacional demandado pela heurística ILS-RVND é similar ao de PLS para todas as instâncias com exceção de duas instâncias onde é pouco mais de 1,5 vezes maior.

Tabela 5.11: Resultados computacionais ILS-RVND para instâncias do SET3 com 5 hotéis.

Instância	N	PCV	MSC			ILS-RVND					PLS			
			#V	TTot	TC(s)	#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TC(s)	Gap(%)
eil51	51	426	6	426	3,0	6	426	426,7	0,0	0,00	6	426	0,0	0,00
berlin52	52	7542	6	7542	3,0	6	7542	7542,0	0,0	0,00	6	7542	0,0	0,00
st70	70	675	6	675	3,1	6	675	675,0	0,0	0,00	6	675	0,0	0,00
eil76	76	538	6	538	3,0	6	538	565,0	0,0	0,00	6	566	0,1	-5,20
pr76	76	108159	6	108159	3,0	6	108159	108159,0	0,0	0,00	6	108159	0,1	0,00
kroA100	100	21282	6	21282	10,2	6	21282	21282,0	0,0	0,00	6	21282	0,1	0,00
kroC100	100	20749	6	20749	10,3	6	20749	20749,0	0,0	0,00	6	20749	0,1	0,00
kroD100	100	21294	6	21294	10,3	6	21294	21294,0	0,1	0,00	6	21294	0,1	0,00
rd100	100	7910	6	7910	10,3	6	7910	7910,0	0,0	0,00	6	7910	0,1	0,00
eil101	101	629	6	629	10,2	6	629	629,0	0,0	0,00	6	629	0,1	0,00
lin105	105	14379	6	14379	10,0	6	14379	14379,0	0,0	0,00	6	14379	0,1	0,00
ch150	150	6528	6	6528	10,4	6	6528	6535,8	0,1	0,00	6	6528	0,2	0,00
tsp225	225	3916	6	3916	89,8	6	3916	3916,0	0,3	0,00	6	3916	0,4	0,00
a280	280	2579	7	2639	199,1	6*	2604	2690,8	1,2	-0,97	7	2652	0,7	-2,83
pcb442	442	50778	6	50778	490,9	6	50778	51847,9	2,2	0,00	7	51087	3,1	-0,61
pr1002	1002	259045	6	259045	3987,5	6	259045	262231,0	31,1	0,00	6	259045	20,5	0,00
Soma/Média			97		303,4	96			2,2	-0,06	98		1,6	-0,54

O último subgrupo do SET3 é caracterizado pela disponibilidade de 10 hotéis em cada instância. Seus resultados são apresentados na Tabela 5.12. É possível verificar que o aumento do número de hotéis torna mais difícil a resolução das instâncias. Tanto a

heurística ILS-RVND, quanto a PLS não foram capazes de igualar a qualidade de solução do PCV para todas as instâncias. A primeira apresentou piores soluções para duas instâncias, enquanto a segunda para três instâncias. A soma total de viagens necessárias na ILS-RVND foi correspondente àquela observada para a MSC do PCVSH. Na PLS, para as três soluções piores, foi utilizada uma viagem extra, como ocorrido também na instância *berlin52* a qual apesar de possuir mesmo valor da rota PCV, não alcança a MSC para o PCVSH. Os *Gaps* médios reportados foram de -0,28 para a ILS-RVND e -0,40 para PLS. O tempo computacional para a heurística ILS-RVND foi semelhante para a maioria das instâncias com exceção das três últimas contidas neste grupo, para as quais ILS-RVND obteve um tempo computacional em média três vezes maior que PLS.

Analisando os três subgrupos de SET3, verifica-se que há uma consistência nos resultados reportados pela heurística ILS-RVND, principalmente em relação ao número de viagens necessárias. O aumento do tempo computacional ocorre de forma proporcional ao aumento do número de hotéis disponíveis.

Tabela 5.12: Resultados computacionais ILS-RVND para instâncias do SET3 com 10 hotéis.

Instância	N	PCV	MSC			ILS-RVND					PLS			
			#V	TTot	TC(s)	#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TC(s)	Gap(%)
eil51	51	426	10	426	3,0	10	426	426,9	0,0	0,00	10	426	0,0	0,00
berlin52	52	7542	8	7864	4,9	8	7870	7832,4	0,0	-4,35	9	7542	0,0	0,00
st70	70	675	10	675	3,1	10	675	675,0	0,0	0,00	10	675	0,0	0,00
eil76	76	538	11	538	3,0	11	538	574,5	0,0	0,00	12	567	0,1	-5,39
pr76	76	108159	11	108159	10,1	11	108159	108159,0	0,0	0,00	11	108159	0,1	0,00
kroA100	100	21282	11	21282	10,1	11	21282	21282,0	0,1	0,00	11	21282	0,1	0,00
kroC100	100	20749	11	20749	10,1	11	20749	20749,0	0,1	0,00	11	20749	0,1	0,00
kroD100	100	21294	11	21294	10,1	11	21294	21294,0	0,0	0,00	11	21294	0,1	0,00
rd100	100	7910	10	7910	10,1	10	7910	7910,0	0,1	0,00	10	7910	0,1	0,00
eil101	101	629	11	629	10,1	11	629	629,1	0,0	0,00	11	629	0,1	0,00
lin105	105	14379	10	14379	10,1	10	14379	14379,0	0,1	0,00	10	14379	0,1	0,00
ch150	150	6528	11	6528	10,7	11	6528	6542,7	0,1	0,00	11	6528	0,2	0,00
tsp225	225	3916	11	3916	84,9	11	3916	3916,0	0,4	0,00	11	3916	0,4	0,00
a280	280	2579	11	2585	135,2	11	2579	2658,5	3,4	0,00	12	2596	0,7	-0,66
pcb442	442	50778	11	50778	526,9	11	50875	52077,3	8,4	-0,19	12	50919	2,6	-0,28
pr1002	1002	259045	11	259045	3325,8	11	259045	267166,9	32,8	0,00	11	259045	11,9	0,00
Soma/Média			169		260,5	169			2,8	-0,28	173		1,0	-0,40

São apresentados na Tabela 5.13 os resultados obtidos pela heurística ILS-RVND para o grupo de instâncias de maior complexidade dentre aqueles propostos por [46]. O aumento da complexidade afeta diretamente a obtenção de soluções equivalentes à MSC, acarretando que a heurística ILS-RVND encontre somente uma nova melhor solução com número reduzido de viagens. Em relação ao tempo total da rota, houve redução para uma instância e igualdade em outras quatro. Essa igualdade de soluções também se repete para a PLS em três instâncias. O número de viagens necessárias para resolver todas as instâncias é reduzido em uma unidade para a ILS-RVND em relação à MSC. A PLS por sua vez, apesar de apresentar tempo computacional médio reduzido, aumenta o número de

viagens necessárias em uma unidade para três instâncias quando comparada à MSC e em quatro instâncias comparada à ILS-RVND. Um comportamento atípico é observado para as instâncias *pcb442* e *pr1002*, pois apesar da heurística ILS-RVND encontrar solução próxima à MSC para a primeira e reduzir o número de viagens na segunda, o tempo computacional foi drasticamente aumentado. O aumento do tempo computacional ocorre em menor proporção em outras duas instâncias. Esse comportamento tem relação direta com o critério de parada de iterações sem melhora, pois qualquer pequena melhora na solução reinicia o contador de iterações, elevando assim o número de iterações total, e consequentemente, o tempo computacional.

Tabela 5.13: Resultados computacionais ILS-RVND para instâncias do SET4.

Instância	N	MSC			ILS-RVND					PLS			
		#V	TTot	TC(s)	#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TC(s)	Gap(%)
eil51	51	6	429	3,9	6	431	439,0	0,0	-0,47	6	436	0,0	-1,63
berlin52	52	7	8585	6486,0	7	8642	8661,9	0,0	-0,66	7	8642	0,0	-0,66
st70	70	6	718	10,2	6	718	730,4	0,1	0,00	6	731	0,0	-1,81
eil76	76	6	539	12,6	6	545	554,9	0,1	-1,11	6	539	0,0	0,00
pr76	76	6	118061	8,4	6	118983	120912,3	0,1	-0,78	7	118719	0,1	-0,56
kroA100	100	6	22205	20,3	6	22205	22329,7	0,1	0,00	7	22044	0,1	0,73
kroC100	100	6	20933	12,2	6	20933	21343,7	0,1	0,00	6	21116	0,1	-0,87
kroD100	100	6	21464	0,1	6	21664	22021,4	0,2	-0,93	6	21464	0,1	0,00
rd100	100	6	8244	24,9	6	8377	8591,8	0,2	-1,61	7	8245	0,1	-0,01
eil101	101	6	634	23,0	6	634	639,9	0,1	0,00	6	652	0,1	-2,84
ch150	150	6	6578	56,1	6	6619	6663,2	0,2	-0,62	6	6728	0,2	-2,28
tsp225	225	6	4502	0,9	6	4433	4580,2	5,6	1,53	6	4502	0,9	0,00
a280	280	6	2645	168,6	6	2669	2729,6	6,4	-0,91	6	2658	0,9	-0,49
pcb442	442	6	54137	890,8	6	54301	55296,8	61,4	-0,30	6	55134	5,7	-1,84
pr1002	1002	7	289337	3529,5	6*	290417	295348,7	370,8	-0,37	7	290110	29,5	-0,27
Soma/Média		92		749,8	91			29,7	-0,42	95		2,5	-0,84

5.1.3 Resultados obtidos pela heurística MILS

Com o intuito de testar a efetividade das modificações realizadas sobre a heurística ILS-RVND com a introdução da técnica de mineração de dados, uma série de experimentos computacionais foi realizada com a heurística MILS. Os algoritmos foram codificados em C++, compilados com o compilador g++ (versão 7.4.0) usando a *tag* “-Ofast” e executados em uma máquina com processador Intel® Core™ i7 4770 3.4 GHz com 16 GB de memória RAM. O sistema operacional utilizado foi o Ubuntu Linux 18.04.

Cada instância foi executada 30 vezes e as sementes para geração de números aleatórios foram definidas entre 1 e 30. A melhor solução encontrada é reportada e a variabilidade das 30 execuções é analisada utilizando o desvio padrão.

A heurística MILS utiliza quatro parâmetros: (i) suporte mínimo percentual para mineração (*sup*), (ii) número de iterações sem melhora do *pool* de soluções (n_{pool}), (iii) número de iterações sem aplicação do procedimento de perturbação (n_{pert}), e (iv) número

de iterações necessárias para intensificação da perturbação (v). Os parâmetros citados foram calibrados executando 10000 iterações no IRACE (*Iterated Racing for Automatic Algorithm Configuration*) [27], utilizando 15 instâncias representativas escolhidas dentre aquelas disponibilizadas por [46]. Os valores testados para sup estavam no intervalo $[70, 90]$, para n_{pool} no intervalo $[100, 300]$, e para n_{pert} no intervalo $[5, 15]$. A melhor configuração retornada pelo IRACE foi $sup = 90$, $n_{pool} = 166$ e $n_{pert} = 12$, sendo o parâmetro v definido com valor igual a 10. Os resultados provenientes da aplicação da heurística MILS estão publicamente disponíveis em <https://bit.ly/2WbHIka>.

Os resultados obtidos por MILS são comparados com os dois algoritmos meméticos MA e HDM desenvolvidos em [8, 26], respectivamente. Os autores em [26] utilizaram um mecanismo de adequação dos tempos de CPU reportados em [8] em relação aos tempos obtidos por eles, por utilizarem máquinas com ligeira diferença de poder de processamento. Utilizamos neste trabalho o procedimento descrito em [10] para adequar os tempos de CPU das máquinas utilizadas em nossos experimentos e em [8, 26], provendo assim tempos computacionais proporcionais às capacidades das máquinas em relação ao número de cálculos de ponto flutuante por segundo, de forma a realizar uma comparação mais justa entre os algoritmos. A condição de parada da MILS foi definida com base nos tempos de CPU reescalados em relação aos do HDM e, conseqüentemente, na mesma proporção daqueles utilizados pelo MA.

Os resultados alcançados são apresentados nas Tabela 5.14 a 5.22. Em todas essas tabelas, a coluna “Instância” indica o nome da instância reportada, e a coluna “N” traz a informação da quantidade de clientes contidos na instância, com exceção das tabelas com resultados do SET2, as quais são separadas de acordo com o número de clientes (10, 15, 30 e 40). Para cada instância, o resultado obtido pelas três abordagens é apresentado, incluindo o número de viagens “#V”, o tempo total gasto na rota “TTotal” e o tempo computacional em segundos “TC”. Para a coluna da heurística MILS, também é reportado o tempo total médio das rotas “TMed” obtido a partir das 30 execuções realizadas para cada instância. Adicionalmente, para todas as instâncias, são acrescentadas duas colunas contendo número de viagens e o tempo total da solução referentes à Melhor Solução Conhecida “MSC”, solução esta obtida por meio de métodos heurísticos ou exatos [4, 8, 9, 26, 46]. Os tempos computacionais reportados nas tabelas são diferentes entre as estratégias MILS, HDM e MA, porque seguindo a metodologia de [10], o fator calculado para reescalar o tempo para MILS é de 0,83 em relação a HDM. O tempo do HDM foi reescalado por [26] em relação ao MA.

Nos grupos SET1, SET2 e SET4, a coluna $Gap(\%)$ representa a diferença percentual média entre o tempo total gasto em uma rota obtida por MILS, HDM e MA e a melhor solução reportada na literatura. Para o SET3, esse valor é calculado em relação à solução do PCVSH obtida pelas abordagens e a solução ótima do PCV reportada na literatura.

O Gap é calculado pela Equação (5.1). Para os grupos SET1, SET2 e SET4, a solução R_1 na equação corresponde ao valor da MSC, enquanto R_2 , corresponde ao valor da solução obtida por MILS, HDM ou MA, respectivamente. Um valor positivo na coluna de Gap indica que a heurística obtém uma melhor solução em relação a MSC. Relacionado ao SET3, onde o valor da solução para o PCV é conhecido, o Gap é calculado entre o tempo total obtido por uma das abordagens (R_2) e a solução do PCV (R_1). Neste caso, um valor de Gap mais próximo a zero indica uma maior qualidade da solução. Para todas as tabelas, entradas em negrito indicam os casos para os quais a MILS foi capaz de encontrar uma melhor solução do que a MSC, e consequentemente, que o HDM e MA. Asteriscos na coluna “#V” indicam que MILS reduziu o número de viagens necessárias para aquela instância.

Na Tabela 5.14 são apresentados os resultados para as instâncias do grupo SET1. É possível observar que a abordagem proposta obtém o mesmo número de viagens que HDM e MA para todas as instâncias, com exceção da instância c101, onde a MILS encontra uma solução que necessita de uma viagem a menos.

Considerando todas as instâncias que compõem o SET1, quando comparados os critérios qualitativos das soluções, a MILS alcançou 11 melhores soluções, 3 idênticas e, apenas 2 soluções piores quando comparada ao HDM. Na comparação com o MA, a MILS alcança 12 novas melhores soluções, 3 idênticas e apenas uma pior. Em relação ao valor do Gap , a heurística MILS na média consegue manter um Gap positivo de 0,02, enquanto HDM e MA atingem valores menos expressivos, sendo -0,02 e -0,11, respectivamente. Em relação à MSC, a heurística MILS estabelece 9 novas melhores soluções, iguala os resultados para outras 4 e apresenta resultado levemente inferior em 3 instâncias. Para os casos em que a solução foi inferior à MSC, o menor valor de Gap foi de -0,12.

Os resultados reportados para as instâncias do SET2 estão dispostos nas Tabelas 5.15 a 5.18. Em cada tabela, o número de clientes atendidos é idêntico para todas as 13 instâncias, não necessitando assim uma coluna específica para tal informação. Devido ao tamanho reduzido das instâncias, para o SET2 com 10 e 15 clientes, os resultados alcançados pelas heurísticas comparadas se igualam ao resultado ótimo reportado na coluna MSC. Para as instâncias do SET2 com 30 e 40 clientes, as heurísticas não foram

Tabela 5.14: Resultados computacionais para as instâncias do SET1.

Instância	MSC				MILS				HDM				MA			
	N	#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
c101	100	9	9591,1	8*	9624,4	9604,7	19,9	-0,35	9	9591,1	24,0	0,00	9	9595,6	24,8	-0,05
r101	100	8	1695,5	8	1692,5	1706,4	20,3	0,18	8	1695,5	24,5	0,00	8	1704,6	24,7	-0,54
rc101	100	8	1672,6	8	1672,6	1672,6	24,4	0,00	8	1673,4	29,4	-0,05	8	1674,1	30,3	-0,09
c201	100	3	9559,7	3	9561,8	9562,0	13,5	-0,02	3	9559,9	16,3	0,00	3	9560,0	16,6	0,00
r201	100	2	1642,8	2	1637,7	1641,8	10,3	0,31	2	1642,8	12,4	0,00	2	1643,4	11,9	-0,04
rc201	100	2	1642,7	2	1642,7	1642,7	10,5	0,00	2	1642,7	12,6	0,00	2	1642,7	12,7	0,00
pr01	48	2	1412,2	2	1412,2	1412,2	1,7	0,00	2	1412,2	2,0	0,00	2	1412,2	2,8	0,00
pr02	96	3	2543,3	3	2543,3	2548,6	15,0	0,00	3	2548,8	18,1	-0,22	3	2543,3	18,6	0,00
pr03	144	4	3404,2	4	3401,3	3422,1	40,3	0,09	4	3404,2	48,6	0,00	4	3415,1	49,8	-0,32
pr04	192	5	4215,3	5	4213,2	4228,2	138,0	0,05	5	4215,3	166,2	0,00	5	4217,4	170,7	-0,05
pr05	240	5	4948,9	5	4947,1	4966,7	282,5	0,04	5	4948,9	340,4	0,00	5	4958,7	341,7	-0,20
pr06	288	7	5960,9	7	5960,6	5983,8	280,2	0,01	7	5960,9	337,6	0,00	7	5963,1	337,5	-0,04
pr07	72	3	2069,6	3	2070,3	2070,3	10,4	-0,03	3	2070,3	12,5	-0,03	3	2070,3	13,1	-0,03
pr08	144	4	3367,7	4	3366,2	3384,4	55,5	0,04	4	3367,7	66,9	0,00	4	3372,0	66,8	-0,13
pr09	216	5	4414,9	5	4420,2	4438,5	194,8	-0,12	5	4414,9	234,7	0,00	5	4420,3	234,8	-0,12
pr10	288	7	5932,0	7	5925,0	5954,7	350,9	0,12	7	5932,0	422,8	0,00	7	5940,5	421,2	-0,14
Soma/Média		77		76			91,8	0,02	77		110,6	-0,02	77		111,1	-0,11

capazes de equiparar resultados à MSC em 5 instâncias. Adicionalmente, para a instância rc101.k40 as heurísticas além de terem tempos totais de soluções piores que a MSC, ainda necessitam de uma viagem extra para montar suas respectivas soluções. Cabe ressaltar que as heurísticas utilizam tempos computacionais não maiores que 4 segundos, enquanto na MSC foi estabelecido um tempo limite de 6 horas para execução do método exato [8]. Resumindo, todas as heurísticas comparadas alcançam os mesmos resultados relativos ao tempo total da rota e número de viagens, consequentemente, mesmos valores para os *Gaps* individual e médio.

Tabela 5.15: Resultados computacionais para as instâncias do SET2 com 10 clientes.

Instância	MSC		MILS					HDM				MA			
	#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
c101.k10	1	955,1	1	955,1	955,1	<1	0,00	1	955,1	<1	0,00	1	955,1	<1	0,00
r101.k10	2	272,8	2	272,8	272,8	<1	0,00	2	272,8	<1	0,00	2	272,8	<1	0,00
rc101.k10	1	237,5	1	237,5	237,5	<1	0,00	1	237,5	<1	0,00	1	237,5	<1	0,00
pr01.k10	1	426,6	1	426,6	426,6	<1	0,00	1	426,6	<1	0,00	1	426,6	<1	0,00
pr02.k10	1	661,9	1	661,9	661,9	<1	0,00	1	661,9	<1	0,00	1	661,9	<1	0,00
pr03.k10	1	553,3	1	553,3	553,3	<1	0,00	1	553,3	<1	0,00	1	553,3	<1	0,00
pr04.k10	1	476,4	1	476,4	476,4	<1	0,00	1	476,4	<1	0,00	1	476,4	<1	0,00
pr05.k10	1	528,9	1	528,9	528,9	<1	0,00	1	528,9	<1	0,00	1	528,9	<1	0,00
pr06.k10	1	597,4	1	597,4	597,4	<1	0,00	1	597,4	<1	0,00	1	597,4	<1	0,00
pr07.k10	1	670,2	1	670,2	670,2	<1	0,00	1	670,2	<1	0,00	1	670,2	<1	0,00
pr08.k10	1	573,4	1	573,4	573,4	<1	0,00	1	573,4	<1	0,00	1	573,4	<1	0,00
pr09.k10	1	645,5	1	645,5	645,5	<1	0,00	1	645,5	<1	0,00	1	645,5	<1	0,00
pr10.k10	1	461,5	1	461,5	461,5	<1	0,00	1	461,5	<1	0,00	1	461,5	<1	0,00
Soma/Média	14		14				0,00	14			0,00	14			0,00

Tabela 5.16: Resultados computacionais para as instâncias do SET2 com 15 clientes.

Instância	MSC		MILS					HDM				MA			
	#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
c101.k15	2	1452,2	2	1452,2	1452,2	<1	0,00	2	1452,2	<1	0,00	2	1452,2	<1	0,00
r101.k15	2	379,8	2	379,8	379,8	<1	0,00	2	379,8	<1	0,00	2	379,8	<1	0,00
rc101.k15	2	303,2	2	303,2	303,2	<1	0,00	2	303,2	<1	0,00	2	303,2	<1	0,00
pr01.k15	1	590,4	1	590,4	590,4	<1	0,00	1	590,4	<1	0,00	1	590,4	<1	0,00
pr02.k15	1	745,6	1	745,6	745,6	<1	0,00	1	745,6	<1	0,00	1	745,6	<1	0,00
pr03.k15	1	632,9	1	632,9	632,9	<1	0,00	1	632,9	<1	0,00	1	632,9	<1	0,00
pr04.k15	1	683,4	1	683,4	683,4	<1	0,00	1	683,4	<1	0,00	1	683,4	<1	0,00
pr05.k15	1	621,2	1	621,2	621,2	<1	0,00	1	621,2	<1	0,00	1	621,2	<1	0,00
pr06.k15	1	685,2	1	685,2	685,2	<1	0,00	1	685,2	<1	0,00	1	685,2	<1	0,00
pr07.k15	1	795,3	1	795,3	795,3	<1	0,00	1	795,3	<1	0,00	1	795,3	<1	0,00
pr08.k15	1	707,2	1	707,2	707,2	<1	0,00	1	707,2	<1	0,00	1	707,2	<1	0,00
pr09.k15	1	771,7	1	771,7	771,7	<1	0,00	1	771,7	<1	0,00	1	771,7	<1	0,00
pr10.k15	1	611,9	1	611,9	611,9	<1	0,00	1	611,9	<1	0,00	1	611,9	<1	0,00
Soma/Média	16		16				0,00	16			0,00	16			0,00

Os resultados para o SET3 são apresentados nas Tabelas 5.19 até 5.21, representando

Tabela 5.17: Resultados computacionais para as instâncias do SET2 com 30 clientes.

Instância	MSC		MILS					HDM				MA			
	#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
c101.k30	3	2829,4	3	2863,2	2863,2	1,2	-1,19	3	2863,2	1,4	-1,19	3	2863,2	< 1	-1,19
r101.k30	3	655,2	3	655,2	655,2	1,7	0,00	3	655,2	2,1	0,00	3	655,2	< 1	0,00
rc101.k30	3	610,0	3	705,5	704,8	1,4	-15,66	3	705,5	1,7	-15,66	3	705,5	< 1	-15,66
pr01.k30	1	964,8	1	964,8	964,8	< 1	0,00	1	964,8	< 1	0,00	1	964,8	< 1	0,00
pr02.k30	2	1078,3	2	1078,3	1078,5	< 1	0,00	2	1078,3	< 1	0,00	2	1078,3	< 1	0,00
pr03.k30	1	952,5	1	952,5	952,5	< 1	0,00	1	952,5	< 1	0,00	1	952,5	< 1	0,00
pr04.k30	2	1091,6	2	1091,6	1091,6	1,5	0,00	2	1091,6	1,8	0,00	2	1091,6	< 1	0,00
pr05.k30	1	924,7	1	924,7	924,7	< 1	0,00	1	924,7	< 1	0,00	1	924,7	< 1	0,00
pr06.k30	2	1063,2	2	1063,2	1063,2	< 1	0,00	2	1063,2	< 1	0,00	2	1063,2	< 1	0,00
pr07.k30	2	1130,4	2	1130,4	1130,4	< 1	0,00	2	1130,4	< 1	0,00	2	1130,4	< 1	0,00
pr08.k30	2	1006,2	2	1006,2	1006,2	< 1	0,00	2	1006,2	< 1	0,00	2	1006,2	< 1	0,00
pr09.k30	2	1091,4	2	1091,4	1091,4	< 1	0,00	2	1091,4	< 1	0,00	2	1091,4	< 1	0,00
pr10.k30	1	918,9	1	918,9	918,9	< 1	0,00	1	918,9	< 1	0,00	1	918,9	< 1	0,00
Soma/Média	25		25				-1,30	25			-1,30	25			-1,30

Tabela 5.18: Resultados computacionais para as instâncias do SET2 com 40 clientes.

Instância	MSC		MILS					HDM				MA			
	#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
c101.k40	4	3817,5	4	3866,1	3866,2	1,7	-1,27	4	3866,1	2,0	-1,27	4	3866,1	2,6	-1,27
r101.k40	4	842,9	4	862,8	864,2	1,3	-2,36	4	862,8	1,6	-2,36	4	862,8	2,0	-2,36
rc101.k40	3	652,1	4	850,3	850,3	2,7	-30,39	4	850,3	3,3	-30,39	4	850,3	2,2	-30,39
pr01.k40	2	1160,5	2	1160,5	1160,5	< 1	0,00	2	1160,5	< 1	0,00	2	1160,5	< 1	0,00
pr02.k40	2	1336,9	2	1336,9	1336,9	1,3	0,00	2	1336,9	1,6	0,00	2	1336,9	< 1	0,00
pr03.k40	2	1303,4	2	1303,4	1303,4	< 1	0,00	2	1303,4	< 1	0,00	2	1303,4	< 1	0,00
pr04.k40	2	1259,5	2	1259,5	1259,5	< 1	0,00	2	1259,5	< 1	0,00	2	1259,5	< 1	0,00
pr05.k40	2	1200,7	2	1200,7	1200,7	< 1	0,00	2	1200,7	< 1	0,00	2	1200,7	< 1	0,00
pr06.k40	2	1242,9	2	1242,9	1242,9	< 1	0,00	2	1242,9	< 1	0,00	2	1242,9	< 1	0,00
pr07.k40	2	1407,0	2	1407,0	1407,0	1,4	0,00	2	1407,0	1,7	0,00	2	1407,0	< 1	0,00
pr08.k40	2	1222,2	2	1222,2	1222,2	< 1	0,00	2	1222,2	< 1	0,00	2	1222,2	< 1	0,00
pr09.k40	2	1284,4	2	1284,4	1284,4	< 1	0,00	2	1284,4	< 1	0,00	2	1284,4	< 1	0,00
pr10.k40	2	1200,4	2	1200,4	1200,4	< 1	0,00	2	1200,4	< 1	0,00	2	1200,4	< 1	0,00
Soma/Média	31		32				-2,62	32			-2,62	32			-2,62

subgrupos contendo 3, 5 e 10 hotéis. Para estas instâncias, o melhor tempo total da rota é conhecido e reportado na coluna “PCV”. Todos os valores de *Gap* apresentados são relacionados ao tempo total da solução para o PCV e aquelas instâncias que possuem *Gap* igual a zero foram capazes de alcançar a melhor solução conhecida.

Para instâncias onde existem 3 hotéis disponíveis (Tabela 5.19) observou-se que apenas a heurística MILS foi capaz de encontrar tempo total de rota equivalente ao tempo total para o PCV em todas 16 instâncias que compõem o subgrupo. Adicionalmente, apenas a MILS e HDM são capazes de produzir soluções usando apenas 4 viagens para cada instância. O *Gap* médio apresentado para a MILS é igual a zero, reforçando o alcance da melhor solução conhecida, enquanto HDM e MA, obtiveram *Gaps* ligeiramente negativos, sendo -0,01 e -0,03, respectivamente.

O subgrupo que contém 5 hotéis é apresentado na Tabela 5.20. Nesse, a MILS também encontra todas as melhores soluções conhecidas para o PCV. Destaca-se também em relação ao HDM e MA a redução no número de viagens necessárias para construir as soluções, sendo menor em uma unidade para a instância *a280* em relação às duas outras heurísticas e, comparado somente ao MA, houve redução também para a instância *pr1002*. Os valores médios de *Gaps* foram 0,00 para a MILS, -0,15 para HDM, e -0,18 para MA,

corroborando assim com a melhor performance da MILS.

Para o último subgrupo do SET3, que é composto por 10 hotéis, seus resultados são apresentados na Tabela 5.21. Neste subgrupo é possível observar que quanto maior o número de hotéis disponíveis, mais difícil se torna a otimização da solução. A heurística MILS conseguiu alcançar a solução do PCV para a instância *a280*, fato que nenhuma heurística prévia tinha reportado. Em relação ao número de viagens utilizadas, as três abordagens heurísticas obtiveram um comportamento idêntico. O valor médio do *Gap* para a MILS foi levemente superior quando comparado ao HDM e mais relevante em relação ao MA. Apenas para uma instância a MILS não conseguiu solução melhor que o HDM, entretanto, o *Gap* foi de apenas -0,01. Por fim, MILS e HDM alcançaram o mesmo tempo total da solução do PCV para 14 de 16 instâncias, enquanto MA apenas para 13.

Tabela 5.19: Resultados computacionais para instâncias do SET3 com 3 hotéis.

Instância	N	PCV	MSC		MILS					HDM				MA			
			#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
eil51	51	426	4	426	4	426	426,5	2,5	0,00	4	426	3,0	0,00	4	426	3,8	0,00
berlin52	52	7542	4	7542	4	7542	7542,0	2,5	0,00	4	7542	3,0	0,00	4	7542	3,2	0,00
st70	70	675	4	675	4	675	675,0	2,6	0,00	4	675	3,1	0,00	4	675	7,0	0,00
eil76	76	538	4	538	4	538	554,2	2,6	0,00	4	538	3,1	0,00	4	538	28,5	0,00
pr76	76	108159	4	108159	4	108159	108159,0	8,5	0,00	4	108159	10,2	0,00	4	108159	15,0	0,00
kroA100	100	21282	4	21282	4	21282	21282,0	8,6	0,00	4	21282	10,3	0,00	4	21282	14,6	0,00
kroC100	100	20749	4	20749	4	20749	20749,0	8,5	0,00	4	20749	10,2	0,00	4	20749	15,5	0,00
kroD100	100	21294	4	21294	4	21294	21294,0	8,6	0,00	4	21294	10,4	0,00	4	21294	16,3	0,00
rd100	100	7910	4	7910	4	7910	7910,0	8,9	0,00	4	7910	10,7	0,00	4	7910	16,1	0,00
eil101	101	629	4	629	4	629	629,0	8,8	0,00	4	629	10,6	0,00	4	629	17,4	0,00
lin105	105	14379	4	14379	4	14379	14379,0	9,1	0,00	4	14379	10,9	0,00	4	14379	16,8	0,00
ch150	150	6528	4	6528	4	6528	6538,7	8,9	0,00	4	6528	10,7	0,00	4	6528	36,7	0,00
tsp225	225	3916	4	3916	4	3916	3916,0	8,5	0,00	4	3916	10,2	0,00	4	3916	14,5	0,00
a280	280	2579	4	2583	4	2579	2603,6	195,3	0,00	4	2583	235,3	-0,16	5	2591	235,2	-0,47
pcb442	442	50778	4	50778	4	50778	51290,3	575,4	0,00	4	50778	693,2	0,00	4	50778	692,2	0,00
pr1002	1002	259045	4	259045	4	259045	263269,4	2712,9	0,00	4	259045	3266,8	0,00	4	259045	3268,0	0,00
Soma/Média			64		64			223,2	0,00	64		268,9	-0,01	65		275,1	-0,03

Tabela 5.20: Resultados computacionais para instâncias do SET3 com 5 hotéis.

Instância	N	PCV	MSC		MILS					HDM				MA			
			#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
eil51	51	426	6	426	6	426	426,7	2,5	0,00	6	426	3,0	0,00	6	426	3,6	0,00
berlin52	52	7542	6	7542	6	7542	7542,0	2,5	0,00	6	7542	3,0	0,00	6	7542	3,7	0,00
st70	70	675	6	675	6	675	675,0	2,6	0,00	6	675	3,1	0,00	6	675	7,3	0,00
eil76	76	538	6	538	6	538	558,6	2,5	0,00	6	538	3,0	0,00	6	538	9,5	0,00
pr76	76	108159	6	108159	6	108159	108159,0	2,5	0,00	6	108159	3,0	0,00	6	108159	8,3	0,00
kroA100	100	21282	6	21282	6	21282	21282,0	8,5	0,00	6	21282	10,2	0,00	6	21282	14,9	0,00
kroC100	100	20749	6	20749	6	20749	20749,0	8,6	0,00	6	20749	10,3	0,00	6	20749	14,2	0,00
kroD100	100	21294	6	21294	6	21294	21294,0	8,6	0,00	6	21294	10,3	0,00	6	21294	15,1	0,00
rd100	100	7910	6	7910	6	7910	7910,0	8,6	0,00	6	7910	10,3	0,00	6	7910	13,9	0,00
eil101	101	629	6	629	6	629	629,0	8,5	0,00	6	629	10,2	0,00	6	629	16,7	0,00
lin105	105	14379	6	14379	6	14379	14379,0	8,3	0,00	6	14379	10,0	0,00	6	14379	15,8	0,00
ch150	150	6528	6	6528	6	6528	6532,3	8,6	0,00	6	6528	10,4	0,00	6	6528	41,4	0,00
tsp225	225	3916	6	3916	6	3916	3916,0	74,5	0,00	6	3916	89,8	0,00	6	3916	89,4	0,00
a280	280	2579	7	2639	6*	2579	2641,2	165,3	0,00	7	2639	199,1	-2,33	7	2646	198,8	-2,60
pcb442	442	50778	6	50778	6	50778	51453,5	407,5	0,00	6	50778	490,9	0,00	6	50778	497,6	0,00
pr1002	1002	259045	6	259045	6	259045	262061,5	3310,3	0,00	6	259045	3987,5	0,00	7	259774	3998,9	-0,28
Soma/Média			97		96			251,8	0,00	97		303,4	-0,15	98		309,3	-0,18

Os resultados reportados para as instâncias contidas no SET4 são apresentados na Tabela 5.22. Esse grupo é tido como o mais desafiador dentre todos os propostos por [46]. Todas as heurísticas comparadas necessitaram do mesmo número de viagens para todas as instâncias, exceto para a instância *pr1002*, onde a heurística proposta MILS reportou uma solução contendo uma viagem a menos e um menor tempo total da rota.

Tabela 5.21: Resultados computacionais para instâncias do SET3 com 10 hotéis.

Instância	N	PCV	MSC		MILS					HDM				MA			
			#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
eil51	51	426	10	426	10	426	426,7	2,5	0,00	10	426	3,0	0,00	10	426	3,3	0,00
berlin52	52	7542	8	7864	8	7864	7892,6	4,1	-4,27	8	7864	4,9	-4,27	8	7864	4,0	-4,27
st70	70	675	10	675	10	675	675,0	2,6	0,00	10	675	3,1	0,00	10	675	6,7	0,00
eil76	76	538	11	538	11	538	566,2	2,5	0,00	11	538	3,0	0,00	11	538	9,2	0,00
pr76	76	108159	11	108159	11	108159	108159,0	8,4	0,00	11	108159	10,1	0,00	11	108159	8,1	0,00
kroA100	100	21282	11	21282	11	21282	21282,0	8,4	0,00	11	21282	10,1	0,00	11	21282	14,5	0,00
kroC100	100	20749	11	20749	11	20749	20749,0	8,4	0,00	11	20749	10,1	0,00	11	20749	14,2	0,00
kroD100	100	21294	11	21294	11	21294	21294,0	8,4	0,00	11	21294	10,1	0,00	11	21294	14,5	0,00
rd100	100	7910	10	7910	10	7910	7910,0	8,4	0,00	10	7910	10,1	0,00	10	7910	14,7	0,00
eil101	101	629	11	629	11	629	629,0	8,4	0,00	11	629	10,1	0,00	11	629	15,6	0,00
lin105	105	14379	10	14379	10	14379	14379,0	8,4	0,00	10	14379	10,1	0,00	10	14379	15,7	0,00
ch150	150	6528	11	6528	11	6528	6549,4	8,9	0,00	11	6528	10,7	0,00	11	6528	36,8	0,00
tsp225	225	3916	11	3916	11	3916	3916,0	70,5	0,00	11	3916	84,9	0,00	11	3916	82,2	0,00
a280	280	2579	11	2585	11	2579	2620,2	112,2	0,00	11	2585	135,2	-0,23	11	2613	138,1	-1,32
pcb442	442	50778	11	50778	11	50785	51654,2	437,3	-0,01	11	50778	526,9	0,00	11	51774	526,7	-1,96
pr1002	1002	259045	11	259045	11	259045	265682,5	2760,5	0,00	11	259045	3325,8	0,00	11	259045	3320,8	0,00
Soma/Média			169		169			216,2	-0,27	169		260,5	-0,28	169		264,1	-0,47

Dentre as 15 instâncias, considerando o tempo total da solução, a MILS alcançou resultados melhores para 9 instâncias quando comparada ao HDM, e o HDM alcançou soluções melhores apenas em duas ocasiões (instâncias *st70* e *ch150*). O *Gap* médio alcançado pelas duas abordagens foi de 0,34 e -0,24, confirmando assim um melhor comportamento da MILS. A MILS também apresenta bons resultados quando comparada ao MA, obtendo melhoras em 10 das 15 instâncias. O *Gap* médio para essas duas heurísticas foi de 0,34 e -0,58, reforçando a superioridade da abordagem proposta.

Tabela 5.22: Resultados computacionais para as instâncias do SET4.

Instância	N	MSC		MILS					HDM				MA			
		#V	TTotal	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)	#V	TTotal	TC(s)	Gap(%)
eil51	51	6	429	6	429	435,5	3,5	0,00	6	433	4,2	-0,93	6	429	3,9	0,00
berlin52	52	7	8585	7	8642	8642,0	3,4	-0,66	7	8642	4,1	-0,66	7	8642	4,3	-0,66
st70	70	6	718	6	722	730,8	8,5	-0,56	6	718	10,2	0,00	6	723	8,7	-0,70
eil76	76	6	539	6	539	548,7	10,5	0,00	6	539	12,6	0,00	6	548	12,7	-1,67
pr76	76	6	118061	6	118061	118865,2	6,6	0,00	6	118318	8,0	-0,22	6	118061	8,4	0,00
kroA100	100	6	22205	6	21991	22401,3	16,9	0,96	6	22205	20,3	0,00	6	22343	19,7	-0,62
kroC100	100	6	20933	6	20933	21119,1	10,1	0,00	6	20933	12,2	0,00	6	20933	13,1	0,00
kroD100	100	6	21464	6	21464	21723,4	14,1	0,00	6	21548	17,0	-0,39	6	21664	17,8	-0,93
rd100	100	6	8244	6	8243	8406,9	19,7	0,01	6	8352	23,7	-1,31	6	8244	24,9	0,00
eil101	101	6	634	6	634	634,5	19,1	0,00	6	634	23,0	0,00	6	634	23,4	0,00
ch150	150	6	6578	6	6586	6654,9	46,6	-0,12	6	6578	56,1	0,00	6	6647	56,3	-1,05
tsp225	225	6	4502	6	4377	4433,2	99,9	2,78	6	4508	120,3	-0,13	6	4571	122,2	-1,53
a280	280	6	2645	6	2636	2671,4	140,0	0,34	6	2645	168,6	0,00	6	2646	163,4	-0,04
pcb442	442	6	54137	6	53452	54229,8	739,6	1,27	6	54137	890,8	0,00	6	54339	898,6	-0,37
pr1002	1002	7	289337	6*	285993	290740,8	2929,8	1,16	7	289337	3529,5	0,00	7	292690	3526,1	-1,16
Soma/Média		92		91			271,2	0,34	92		326,7	-0,24	92		326,9	-0,58

Resumindo os resultados alcançados, a Tabela 5.23 apresenta uma comparação entre as três abordagens para todos os grupos de instâncias. Para cada abordagem, a primeira linha apresenta a soma das viagens utilizadas por cada grupo, a segunda indica o número de melhores soluções encontradas, e a última linha apresenta o *Gap* médio das soluções em relação às melhores soluções reportadas na literatura. É possível observar que a MILS diminui ao menos uma viagem do total necessário, com exceção para as instâncias do SET2, considerando cada um dos grupos em comparação com HDM e MA. De forma complementar, também encontra um número maior de melhores soluções.

Considerando todas as 131 instâncias propostas por [46], a MILS encontrou 18 melho-

res soluções inéditas, obteve resultados equivalentes em outras 100 instâncias e somente teve resultados desfavoráveis em 13 instâncias.

É relevante acrescentar que os melhores desempenhos em relação aos *Gaps* entre a abordagem proposta e as duas outras heurísticas ocorreram nos grupos SET1 e SET4, os quais possuem as instâncias mais desafiadoras para o PCVSH [26]. Corroborando com essa informação, a MILS encontrou 15 das novas melhores soluções nesses dois grupos, sendo 9 no SET1 e 6 no SET4, o que demonstra a excelente performance desta abordagem heurística.

Tabela 5.23: Resumo dos resultados para todas as instâncias propostas por [46].

	SET1			SET2			SET3			SET4		
	MILS	HDM	MA	MILS	HDM	MA	MILS	HDM	MA	MILS	HDM	MA
Soma das viagens	76	77	77	87	87	87	329	330	332	91	92	92
Melhores soluções	13/16	12/16	3/16	47/52	47/52	47/52	46/48	44/48	42/48	12/15	8/15	5/15
Gap médio (%)	0,02	-0,02	-0,11	-0,98	-0,98	-0,98	-0,09	-0,15	-0,23	0,34	-0,24	-0,58

Como a heurística MILS possui característica estocástica e devido ao fato de cada instância ter sido executada 30 vezes, a robustez do algoritmo proposto foi avaliada pelo cálculo do desvio padrão dos tempos totais das rotas das soluções. Para realização do cálculo, foi obtido o desvio padrão em relação às 30 execuções. Esse resultado foi transformado em um percentual do valor da melhor solução encontrada e utilizado para montar o box plot. O box plot apresentado na Figura 5.1 ilustra o comportamento do desvio padrão observado para cada instância em seu respectivo grupo. Os resultados para o SET2 foram suprimidos, pois o desvio padrão apresentado foi igual ou muito próximo a zero para todas as instâncias.

Para o SET1, o valor percentual do desvio padrão não foi em nenhum caso superior a 0,48% (máximo obtido para a instância *pr04*), sendo a média reportada de 0,17%. A dispersão dos resultados para o SET3 com 3, 5 e 10 hotéis obteve um percentual relativamente próximo de zero, sendo o nível inferior igual a média (0,00%). Para o SET4 o qual possui as instâncias mais desafiadoras, a dispersão foi maior. Entretanto, o desvio padrão médio em relação à melhor solução reportada foi de aproximadamente 0,74%, com a máxima nunca superior a 1,32%.

Considerando o número de viagens para as 30 execuções de cada instância, houve variação em apenas 18 das 131 instâncias. Desta forma, o algoritmo possui robustez na geração de soluções com pouca variabilidade no número de viagens necessárias.

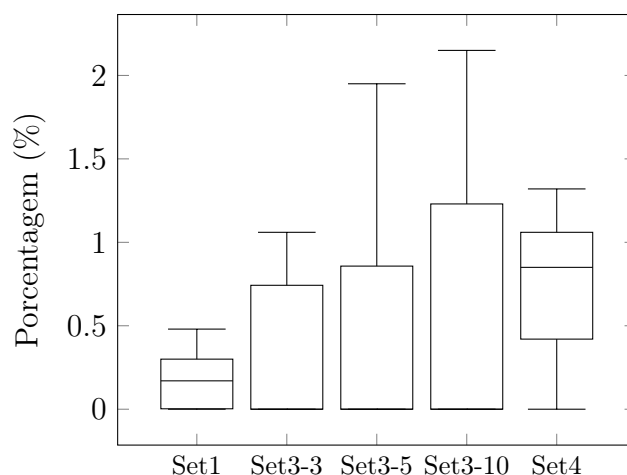


Figura 5.1: Dispersão correspondente à diferenças entre o desvio padrão calculado e o valor da melhor solução alcançada.

5.2 Resultados para instâncias propostas por [4]

Os autores em [4] desenvolveram um algoritmo *Branch-Cut and Price* (BCP) para resolver de forma exata o PCVSH. As instâncias previamente apresentadas foram utilizadas para testar o comportamento do BCP, sendo utilizado o solver comercial Gurobi² como ferramenta e as melhores soluções conhecidas na literatura como limites superiores para as soluções (*upper bounds*). Comparando com as soluções encontradas pelo HDM e MA, o BCP encontrou 3 novas melhores soluções, 95 soluções iguais às melhores encontradas por heurísticas, e 2 soluções piores. Adicionalmente, o algoritmo não foi capaz de encontrar soluções viáveis para 31 instâncias considerando o limite de tempo definido.

De acordo com [4], as instâncias propostas por [46] possuem características que limitam a aplicabilidade do BCP e também não representam fielmente cenários reais do dia a dia de empresas. Existem instâncias que possuem poucos clientes e hotéis, mas demandam muito tempo computacional para serem solucionadas pelo BCP. Também existem instâncias com muitos clientes para cada viagem, tornando impraticável a resolução do subproblema de *pricing*.

Visando resolver tais limitações, foram criadas 240 novas instâncias³ contendo clientes posicionados de forma aleatória em um plano de 100×100 . Tais instâncias foram divididas em grupos de: (i) 10 instâncias com 50 clientes, (ii) 10 instâncias com 75 clientes, e (iii) 10 instâncias com 100 clientes. Os custos associados a cada aresta são obtidos por meio da distância euclidiana entre as localidades que representam clientes e hotéis. Para

²<https://www.gurobi.com/>

³Disponíveis em: <https://bit.ly/2TMYB3c>

cada instância com diferentes quantidades de clientes, 3 subgrupos foram criados com 5, 10 e 20 hotéis. Os valores máximos que limitam o tempo das viagens foram de 150 e 200 para instâncias com 5 hotéis. Para os demais subgrupos, o tempo foi limitado a 100, 150 e 200. Para todas as instâncias, cada cliente demanda um tempo de visita igual a 10 unidades (valor que é acrescentado ao tempo total da viagem/rota).

Os resultados encontrados para as instâncias propostas por [4] são apresentados nas Tabelas 5.24 a 5.47. Para essas tabelas, a primeira coluna apresenta o nome das instâncias referentes aquele subgrupo. Esses nomes foram padronizados com o formato $hXX_cYY_ZZ_NN$ que especifica a composição das instâncias com XX hotéis, YY clientes, ZZ limite de tempo para as viagens e NN número identificador da instância, sempre definido de 1 a 10. As quatro próximas colunas trazem os resultados reportados por [4], onde a primeira coluna informa se aquela instância foi resolvida de forma ótima ou não, a segunda coluna apresenta o número de viagens ($\#V$) necessárias para composição da rota, a terceira coluna traz o tempo total gasto na rota (TTotal) e a última o tempo computacional (TC) utilizado para execução da referida instância em segundos. Os resultados alcançados pela MILS são descritos nas próximas 8 colunas, sendo esses divididos em duas partes: (i) resultados utilizando tempos computacionais reduzidos [MILS (5, 10, 20 s)], sendo 5 segundos para instâncias com 50 clientes, 10 segundos para instâncias com 75 clientes e 20 segundos para instâncias com 100 clientes, que foram definidos com base nos tempos computacionais utilizados para instâncias de tamanhos similares dos grupos propostos por [46], e (ii) resultados utilizando tempos computacionais estendidos [MILS (20, 40, 60 s)], sendo 20 segundos para instâncias com 50 clientes, 40 segundos para instâncias com 75 clientes e 60 segundos para instâncias com 100 clientes, objetivando assim, testar o comportamento da heurística para tempos computacionais não tão limitados. Assim como na coluna do BCP, nestas duas colunas de resultados da MILS, são reportados o número de viagens necessárias ($\#V$), o tempo total gasto na solução (TTotal) e o tempo computacional. Adicionalmente, são apresentados o tempo total médio dentre as 30 execuções de cada instância (TMed) e a média de ganho percentual (Gap) extraído conforme a Equação (5.1). Para este cálculo de Gap , foi utilizada sempre a MSC reportada por [4] e, para os casos onde o algoritmo BCP não foi capaz de reportar uma solução, a coluna Gap é preenchida com um símbolo “-”.

Para as execuções das instâncias utilizando o BCP, o tempo computacional máximo foi limitado a 18000 segundos. Adicionalmente, o rótulo “Inviável” significa que para aquela instância, considerando o número de hotéis e o tempo limite disponível, não foi possível chegar a uma solução viável. Essas instâncias inviáveis, foram propositalmente criadas

por [4] para analisar que a simples adição de mais hotéis ou o incremento do limite de tempo de uma viagem poderiam ser suficientes para tornar tais instâncias viáveis. Ainda nas colunas que trazem resultados do BCP, o rótulo “Sem resultados” significa que o algoritmo proposto por [4] executou por 18000 segundos e não conseguiu alcançar nenhuma solução viável. Resultados em negrito representam que a heurística MILS obteve melhores resultados.

Os resultados para o primeiro conjunto de instâncias contendo 5 hotéis e 50 clientes são apresentados na Tabela 5.24. É possível verificar que para quatro instâncias não foi possível atender a todos os clientes respeitando o tempo limite da viagem, mas que o aumento do tempo limite de 150 para 200 foi suficiente para permitir que todas as instâncias tivessem uma solução viável como mostrado na Tabela 5.25. Desconsiderando as instâncias inviáveis da primeira tabela, o BCP foi capaz de alcançar a solução ótima para todas as instâncias, menos para *h05_c50_200_07* e *h05_c50_200_08*, as quais coincidentemente, a MILS foi capaz de melhorar utilizando o tempo de execução estendido.

Tabela 5.24: Resultados para instâncias com 5 hotéis, 50 clientes e tempo de viagem limitado a 150.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h05_c50_150_01			Inviável				Inviável					Inviável		
h05_c50_150_02	Sim	10	1323,8	8,0	10	1323,8	1333,6	5,0	0,00	10	1323,8	1332,8	20,0	0,00
h05_c50_150_03	Sim	10	1399,6	490,0	10	1399,6	1412,2	5,0	0,00	10	1399,6	1410,2	20,0	0,00
h05_c50_150_04	Sim	11	1488,8	9,0	11	1488,8	1497,6	5,0	0,00	11	1488,8	1497,1	20,0	0,00
h05_c50_150_05	Sim	10	1316,4	76,0	10	1316,4	1324,3	5,0	0,00	10	1316,4	1323,4	20,0	0,00
h05_c50_150_06			Inviável				Inviável					Inviável		
h05_c50_150_07	Sim	13	1825,0	1100,0	13	1826,1	1829,9	5,0	-0,06	13	1826,1	1829,0	20,0	-0,06
h05_c50_150_08			Inviável				Inviável					Inviável		
h05_c50_150_09	Sim	9	1222,2	1,0	9	1222,2	1222,2	5,0	0,00	9	1222,2	1222,2	20,0	0,00
h05_c50_150_10			Inviável				Inviável					Inviável		
Soma/Média		63		280,7	63			5,0	-0,01	63			20,0	-0,01

Tabela 5.25: Resultados para instâncias com 5 hotéis, 50 clientes e tempo de viagem limitado a 200.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h05_c50_200_01	Sim	7	1362,3	1966,0	7	1362,3	1367,4	5,0	0,00	7	1362,3	1365,5	20,0	0,00
h05_c50_200_02	Sim	7	1175,4	1602,0	7	1176,1	1191,4	5,0	-0,06	7	1175,4	1187,6	20,0	0,00
h05_c50_200_03	Sim	7	1168,0	30,0	7	1168,0	1169,0	5,0	0,00	7	1168,0	1169,0	20,0	0,00
h05_c50_200_04	Sim	7	1258,1	9252,0	7	1258,1	1263,5	5,0	0,00	7	1258,1	1261,8	20,0	0,00
h05_c50_200_05	Sim	6	1177,1	135,0	7	1162,5	1168,5	5,0	1,24	6	1177,1	1168,7	20,0	0,00
h05_c50_200_06	Sim	8	1398,3	385,0	8	1399,7	1406,9	5,0	-0,10	8	1399,7	1406,5	20,0	-0,10
h05_c50_200_07	Não	7	1346,9	18000,0	7	1306,0	1309,5	5,0	3,04	7	1299,9	1307,8	20,0	3,49
h05_c50_200_08	Não	7	1310,0	18000,0	7	1290,1	1301,9	5,0	1,52	7	1290,1	1298,9	20,0	1,52
h05_c50_200_09	Sim	7	1164,3	23,0	7	1164,3	1167,8	5,0	0,00	7	1164,3	1167,7	20,0	0,00
h05_c50_200_10	Sim	8	1414,6	81,0	8	1414,6	1427,0	5,0	0,00	8	1414,6	1424,0	20,0	0,00
Soma/Média		71		4947,4	72			5,0	0,56	71			20,0	0,49

Dentre as 30 instâncias que compõem os subgrupos com 10 hotéis e 50 clientes, 5 são inviáveis devido à restrição imposta pela quantidade de tempo disponível para uma viagem (100) e outras 2 não possuem valores ótimos reportados pelo BCP. Novamente, para essas onde não há ótimo conhecido, a MILS foi capaz de melhorar a solução exis-

tente (*h10_c50_150_04*) e obteve uma solução viável para outra (*h10_c50_150_07*), diferente do BCP que não havia retornado solução após os 18000 segundos. Ainda na Tabela 5.27, observa-se que o uso de tempo computacional estendido possibilitou que a MILS fosse capaz de diminuir o número de viagens em relação a sua variante com tempo reduzido, bem como alcançar o ótimo local referente a instância *h10_c50_150_01*. Neste subgrupo de instâncias, identifica-se que a MILS se comporta melhor quando há maior liberdade para alocação dos clientes nas viagens, ou seja, quando o tempo limite disponível para uma viagem vai aumentando. Essa afirmação é comprovada observando-se que para as instâncias apresentadas na Tabela 5.28, onde todas possuem ótimos reportados, a MILS utilizou número de viagens idêntico ao BCP, e alcançou a solução ótima para 8 de 10 instâncias. Para as duas instâncias onde o ótimo não foi alcançado, o *Gap* foi de apenas -0,04.

Tabela 5.26: Resultados para instâncias com 10 hotéis, 50 clientes e tempo de viagem limitado a 100.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTot	TC(s)	#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h10_c50_100_01			Inviável				Inviável					Inviável		
h10_c50_100_02	Sim	15	1334,8	1897,0	15	1337,9	1353,1	5,0	-0,23	15	1337,9	1352,2	20,0	-0,23
h10_c50_100_03	Sim	15	1297,0	82,0	15	1303,8	1307,6	5,0	-0,52	15	1303,8	1306,9	20,0	-0,52
h10_c50_100_04			Inviável				Inviável					Inviável		
h10_c50_100_05	Sim	15	1351,1	3625,0	15	1351,1	1381,3	5,0	0,00	15	1351,1	1376,3	20,0	0,00
h10_c50_100_06			Inviável				Inviável					Inviável		
h10_c50_100_07	Sim	16	1434,5	1077,0	16	1443,4	1457,3	5,0	-0,62	16	1443,4	1454,5	20,0	-0,62
h10_c50_100_08			Inviável				Inviável					Inviável		
h10_c50_100_09	Sim	14	1300,1	4,0	14	1300,1	1309,7	5,0	0,00	14	1300,1	1302,8	20,0	0,00
h10_c50_100_10			Inviável				Inviável					Inviável		
Soma/Média		75		1337,0	75			5,0	-0,28	75			20,0	-0,28

Tabela 5.27: Resultados para instâncias com 10 hotéis, 50 clientes e tempo de viagem limitado a 150.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTot	TC(s)	#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h10_c50_150_01	Sim	9	1326,2	173,0	10	1302,2	1314,4	5,0	1,81	9	1326,2	1308,0	20,0	0,00
h10_c50_150_02	Sim	9	1144,0	2208,0	9	1144,0	1153,4	5,0	0,00	9	1144,0	1151,2	20,0	0,00
h10_c50_150_03	Sim	8	1144,3	277,0	8	1149,5	1145,3	5,0	-0,45	8	1144,3	1151,0	20,0	0,00
h10_c50_150_04	Não	9	1295,5	18000,0	9	1207,6	1217,0	5,0	6,79	9	1207,6	1214,4	20,0	6,79
h10_c50_150_05	Sim	8	1184,5	937,0	9	1151,2	1157,3	5,0	2,81	9	1151,2	1157,3	20,0	2,81
h10_c50_150_06	Sim	10	1327,0	86,0	10	1328,5	1333,1	5,0	-0,11	10	1328,5	1332,6	20,0	-0,11
h10_c50_150_07		Sem resultado	18000,0		9	1176,5	1182,9	5,0	-	9	1176,5	1181,6	20,0	-
h10_c50_150_08	Sim	8	1172,6	878,0	8	1172,6	1150,6	5,0	0,00	8	1172,6	1148,5	20,0	0,00
h10_c50_150_09	Sim	8	1164,3	10185,0	9	1114,9	1116,7	5,0	4,24	9	1114,9	1116,8	20,0	4,24
h10_c50_150_10	Sim	9	1274,3	6,0	9	1274,3	1299,3	5,0	0,00	9	1274,3	1294,7	20,0	0,00
Soma/Média		78		5075,0	90			5,0	1,68	89			20,0	1,53

Aumentando significativamente o número de hotéis disponíveis nas instâncias, as Tabelas 5.29 até 5.31 apresentam resultados sem reportar nenhuma instância inviável, mesmo para o subgrupo que possui limite de tempo igual a 100 para as viagens. De forma similar ao que ocorreu nos subgrupos anteriores, para as instâncias onde o BCP não reportou solução ótima, a MILS consegue melhorar a solução até então existente. O número de viagens utilizadas pela MILS também segue a tendência dos subgrupos anteriores, tendo

Tabela 5.28: Resultados para instâncias com 10 hotéis, 50 clientes e tempo de viagem limitado a 200.

Instância		BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		Ótimo	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h10_c50_200_01	Sim	6	1152,9	8044,0	6	1152,9	1151,6	5,0	0,00	6	1152,9	1151,4	20,0	0,00	
h10_c50_200_02	Sim	6	1117,9	3157,0	6	1121,1	1130,2	5,0	-0,29	6	1117,9	1130,3	20,0	0,00	
h10_c50_200_03	Sim	6	1057,4	16,0	6	1057,4	1060,3	5,0	0,00	6	1057,4	1060,3	20,0	0,00	
h10_c50_200_04	Sim	7	1163,6	308,0	7	1167,3	1169,6	5,0	-0,32	7	1167,3	1169,6	20,0	-0,32	
h10_c50_200_05	Sim	6	1101,7	7,0	6	1101,7	1124,8	5,0	0,00	6	1101,7	1124,3	20,0	0,00	
h10_c50_200_06	Sim	6	1158,4	498,0	6	1158,4	1160,2	5,0	0,00	6	1158,4	1157,3	20,0	0,00	
h10_c50_200_07	Sim	6	1119,5	139,0	6	1119,5	1123,2	5,0	0,00	6	1119,5	1123,1	20,0	0,00	
h10_c50_200_08	Sim	6	1077,0	279,0	6	1080,8	1080,9	5,0	-0,35	6	1077,4	1080,8	20,0	-0,04	
h10_c50_200_09	Sim	6	1085,9	5577,0	6	1085,9	1086,0	5,0	0,00	6	1085,9	1086,0	20,0	0,00	
h10_c50_200_10	Sim	7	1172,0	61,0	7	1172,0	1179,0	5,0	0,00	7	1172,0	1178,5	20,0	0,00	
Soma/Média		62		1808,6	62			5,0	-0,10	62			20,0	-0,04	

maior variabilidade em relação ao BCP para instâncias que possuem limite de tempo para viagens mais restritivo. A qualidade da solução tende a ser melhor à medida que o limite de tempo para uma viagem vai aumentando, entretanto, o aumento no número de hotéis disponíveis para uma dada instância impacta negativamente a eficiência da heurística MILS.

Tabela 5.29: Resultados para instâncias com 20 hotéis, 50 clientes e tempo de viagem limitado a 100.

BCP					MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
Instância	Ótimo	#V	TTotal	TC(s)	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TMed	TC(s)	Gap(%)
h20_c50_100_01	Sim	13	1115,7	297,0	13	1124,6	1127,1	5,0	-0,80	13	1124,6	1125,5	20,0	-0,80
h20_c50_100_02	Não	13	1226,1	18000,0	13	1167,6	1182,7	5,0	4,77	13	1162,8	1179,8	20,0	5,16
h20_c50_100_03	Sim	12	1156,2	874,0	13	1106,2	1114,2	5,0	4,32	13	1106,2	1112,2	20,0	4,32
h20_c50_100_04	Sim	13	1220,0	3354,0	14	1200,4	1206,4	5,0	1,61	14	1200,4	1203,2	20,0	1,61
h20_c50_100_05	Sem resultado		18000,0		13	1219,6	1220,7	5,0	-	13	1216,5	1220,2	20,0	-
h20_c50_100_06		Sim	13	1190,2	6955,0	14	1187,3	1203,1	5,0	0,24	13	1219,2	1201,1	20,0
h20_c50_100_07	Sim	15	1389,1	2388,0	15	1394,7	1393,6	5,0	-0,40	15	1394,7	1396,9	20,0	-0,40
h20_c50_100_08	Sim	13	1164,1	2972,0	13	1175,8	1185,7	5,0	-1,01	13	1175,8	1184,9	20,0	-1,01
h20_c50_100_09	Sim	15	1349,5	438,0	15	1349,5	1357,5	5,0	0,00	15	1349,5	1355,9	20,0	0,00
h20_c50_100_10	Sim	14	1270,8	8845,0	14	1283,5	1297,2	5,0	-1,00	14	1283,5	1294,9	20,0	-1,00
Soma/Média		121		6212,3	137			5,0	0,86	136			20,0	0,61

Tabela 5.30: Resultados para instâncias com 20 hotéis, 50 clientes e tempo de viagem limitado a 150.

BCP					MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
Instância	Ótimo	#V	TTotal	TC(s)	#V	TTotal	TMed	TC(s)	Gap(%)	#V	TTotal	TMed	TC(s)	Gap(%)
h20_c50_150_01	Sim	8	1083,1	3530,0	8	1083,1	1075,0	5,0	0,00	8	1083,1	1088,5	20,0	0,00
h20_c50_150_02	Não	8	1118,7	18000,0	8	1097,3	1109,4	5,0	1,91	8	1097,3	1109,2	20,0	1,91
h20_c50_150_03	Sim	8	1052,4	4878,0	8	1052,4	1053,9	5,0	0,00	8	1052,4	1053,9	20,0	0,00
h20_c50_150_04	Sim	9	1130,1	2923,0	9	1130,1	1134,2	5,0	0,00	9	1130,1	1133,9	20,0	0,00
h20_c50_150_05	Sim	8	1115,8	634,0	8	1129,0	1131,1	5,0	-1,18	8	1129,0	1130,8	20,0	-1,18
h20_c50_150_06	Sim	8	1101,8	1042,0	8	1101,8	1105,2	5,0	0,00	8	1101,8	1104,6	20,0	0,00
h20_c50_150_07	Sim	9	1160,9	561,0	9	1162,9	1167,4	5,0	-0,17	9	1162,9	1167,3	20,0	-0,17
h20_c50_150_08	Sim	8	1086,3	1013,0	8	1089,6	1094,1	5,0	-0,30	8	1089,6	1094,0	20,0	-0,30
h20_c50_150_09	Sim	8	1170,4	825,0	9	1142,0	1142,0	5,0	2,43	9	1142,0	1142,0	20,0	2,43
h20_c50_150_10	Sim	8	1113,7	10,0	8	1113,7	1113,7	5,0	0,00	8	1113,7	1113,7	20,0	0,00
Soma/Média		82		3341,6	83			5,0	0,27	83			20,0	0,27

Nas Tabelas 5.32 e 5.33 são apresentados os resultados para instâncias contendo 5 hotéis e 75 clientes. Para esses dois subgrupos, apenas uma instância apresentou inviabilidade (*h05_c75_150_10*). Outras três instâncias não tiveram soluções viáveis reportadas pelo BCP, entretanto, a MILS apresentou soluções viáveis para essas e encontrou

Tabela 5.31: Resultados para instâncias com 20 hotéis, 50 clientes e tempo de viagem limitado a 200.

Instância	Ótimo	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h20_c50_200_01	Sim	6	1058,7	941,0		6	1058,7	1066,4	5,0	0,00	6	1058,7	1065,4	20,0	0,00
h20_c50_200_02	Sim	6	1083,7	427,0		6	1083,7	1105,0	5,0	0,00	6	1083,7	1104,3	20,0	0,00
h20_c50_200_03	Sim	6	1039,3	222,0		6	1039,3	1046,3	5,0	0,00	6	1039,3	1045,2	20,0	0,00
h20_c50_200_04	Não	6	1151,5	18000,0		6	1147,0	1145,9	5,0	0,39	6	1146,3	1147,4	20,0	0,45
h20_c50_200_05	Sim	6	1092,5	1222,0		6	1103,4	1109,6	5,0	-1,00	6	1103,4	1109,1	20,0	-1,00
h20_c50_200_06	Não	6	1117,8	18000,0		6	1091,9	1096,6	5,0	2,32	6	1091,9	1096,5	20,0	2,32
h20_c50_200_07	Sim	6	1112,6	47,0		6	1112,6	1113,9	5,0	0,00	6	1112,6	1115,0	20,0	0,00
h20_c50_200_08	Sim	6	1054,3	2785,0		6	1059,1	1059,1	5,0	-0,46	6	1059,1	1059,1	20,0	-0,46
h20_c50_200_09	Sim	6	1102,2	2466,0		6	1104,2	1106,9	5,0	-0,18	6	1103,7	1106,5	20,0	-0,14
h20_c50_200_10	Não	6	1156,7	18000,0		6	1107,1	1107,6	5,0	4,29	6	1107,1	1109,0	20,0	4,29
Soma/Média		60		6211,0		60			5,0	0,54	60			20,0	0,55

soluções melhores que o BCP para outras oito instâncias que tinham soluções ótimas desconhecidas.

Tabela 5.32: Resultados para instâncias com 5 hotéis, 75 clientes e tempo de viagem limitado a 150.

Instância	Ótimo	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h05_c75_150_01	Sim	14	1997,9	43,0		14	1997,9	2012,5	10,0	0,00	14	1997,9	2008,6	40,0	0,00
h05_c75_150_02		Sem resultado		18000,0		13	1824,1	1835,3	10,0	-	13	1824,1	1832,9	40,0	-
h05_c75_150_03		Sem resultado		18000,0		13	1778,7	1790,2	10,0	-	13	1778,7	1788,0	40,0	-
h05_c75_150_04	Sim	13	1766,7	13355,0		13	1766,7	1780,8	10,0	0,00	13	1766,7	1779,2	40,0	0,00
h05_c75_150_05	Sim	12	1666,8	5119,0		12	1668,8	1675,3	10,0	-0,12	12	1668,8	1673,7	40,0	-0,12
h05_c75_150_06	Não	14	1979,3	18000,0		14	1949,8	1954,3	10,0	1,49	14	1943,7	1953,9	40,0	1,80
h05_c75_150_07	Sim	14	1986,6	6367,0		14	1986,6	1994,1	10,0	0,00	14	1986,6	1993,5	40,0	0,00
h05_c75_150_08	Não	13	1835,6	18000,0		13	1767,1	1781,4	10,0	3,73	13	1767,1	1777,0	40,0	3,73
h05_c75_150_09	Sim	15	2072,9	150,0		15	2072,9	2095,2	10,0	0,00	15	2072,9	2091,5	40,0	0,00
h05_c75_150_10		Inviável					Inviável					Inviável			
Soma/Média		95				121			10,0		121			40,0	

Tabela 5.33: Resultados para instâncias com 5 hotéis, 75 clientes e tempo de viagem limitado a 200.

Instância	Ótimo	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h05_c75_200_01	Sim	9	1677,9	1814,0		9	1678,6	1699,4	10,0	-0,04	9	1678,6	1693,2	40,0	-0,04
h05_c75_200_02	Não	9	1749,5	18000,0		9	1630,0	1634,7	10,0	6,83	9	1630,0	1634,4	40,0	6,83
h05_c75_200_03	Não	9	1622,2	18000,0		9	1570,1	1582,6	10,0	3,21	9	1570,1	1580,0	40,0	3,21
h05_c75_200_04	Sim	9	1600,7	761,0		9	1603,7	1618,4	10,0	-0,19	9	1603,7	1616,9	40,0	-0,19
h05_c75_200_05	Não	9	1588,3	18000,0		9	1551,3	1565,2	10,0	2,33	9	1551,3	1564,4	40,0	2,33
h05_c75_200_06	Sim	9	1652,0	495,0		9	1656,8	1674,6	10,0	-0,29	9	1652,0	1670,8	40,0	0,00
h05_c75_200_07	Não	9	1709,8	18000,0		9	1700,7	1713,3	10,0	0,53	9	1699,8	1711,4	40,0	0,58
h05_c75_200_08	Não	9	1686,5	18000,0		9	1566,1	1589,8	10,0	7,14	9	1566,1	1584,0	40,0	7,14
h05_c75_200_09		Sem resultado		18000,0		9	1660,2	1681,0	10,0	-	9	1660,2	1673,6	40,0	-
h05_c75_200_10	Não	10	1938,0	18000,0		10	1873,1	1875,7	10,0	3,35	10	1871,4	1875,2	40,0	3,44
Soma/Média		82		12907,0		91			10,0	2,54	91			40,0	2,59

Resultados para subgrupos de instâncias contendo 10 hotéis e 75 clientes são detalhados nas Tabelas 5.34 a 5.36. Para o primeiro subgrupo, a restrição no tempo limite das viagens faz com que três instâncias sejam inviáveis para o BCP. Um comportamento distinto foi notado para ambas as variantes da MILS (tempo reduzido e estendido), onde para a instância *h10_c75_100_06* também foi reportada inviabilidade. Essa foi a única instância dentre todas propostas por [4] a apresentar inviabilidade, quando o BCP não apresentou. A obtenção de uma solução viável não ocorreu na heurística construtiva, nem mesmo nas próximas tentativas de otimização realizadas pelo RVND. De fato existe

uma dificuldade a mais para obtenção de uma solução viável, pois em comparação com as demais instâncias apresentadas na mesma tabela, essa foi a única a demandar um número de maior viagens. Seu valor ótimo também não foi alcançado pelo BCP. Em relação às instâncias com limite de tempo para viagens iguais a 150 e 200, a MILS apresentou muito bom desempenho em relação ao BCP, alcançando soluções viáveis para seis instâncias que não tinham soluções viáveis conhecidas. Para outras oito instâncias, a melhor solução conhecida foi superada. Outro fato relevante é o quantitativo de tempo computacional necessário para execução do BCP, que chega ao limite de 18000 segundos em 20 de 30 instâncias (considerando aquelas com inviabilidade).

Tabela 5.34: Resultados para instâncias com 10 hotéis, 75 clientes e tempo de viagem limitado a 100.

Instância	Ótimo	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h10_c75_100_01	Não	19	1776,6	18000,0		19	1730,8	1745,9	10,0	2,58	19	1726,0	1743,1	40,0	2,85
h10_c75_100_02	Sim	19	1767,5	1322,0		19	1771,0	1778,5	10,0	-0,20	19	1771,0	1777,8	40,0	-0,20
h10_c75_100_03	Sim	18	1727,4	860,0		19	1710,3	1731,5	10,0	0,99	18	1734,2	1727,4	40,0	-0,39
h10_c75_100_04	Sim	18	1728,9	8661,0		19	1703,7	1722,5	10,0	1,46	19	1703,7	1718,6	40,0	1,46
h10_c75_100_05	Não	20	1861,0	18000,0		21	1841,2	1846,7	10,0	1,06	21	1841,2	1844,0	40,0	1,06
h10_c75_100_06	Não	21	1984,4	18000,0				Inviável					Inviável		
h10_c75_100_07	Sim	19	1808,8	1409,0		20	1799,8	1819,6	10,0	0,50	20	1799,8	1814,4	40,0	0,50
h10_c75_100_08			Inviável					Inviável					Inviável		
h10_c75_100_09			Inviável					Inviável					Inviável		
h10_c75_100_10			Inviável					Inviável					Inviável		
Soma/Média		134		9464,6		117			10,0	1,06	116			40,0	0,88

Tabela 5.35: Resultados para instâncias com 10 hotéis, 75 clientes e tempo de viagem limitado a 150.

Instância	Ótimo	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h10_c75_150_01			Sem resultado	18000,0		11	1534,2	1545,1	10,0	-	11	1534,2	1544,1	40,0	-
h10_c75_150_02			Sem resultado	18000,0		11	1577,8	1562,0	10,0	-	11	1577,7	1564,0	40,0	-
h10_c75_150_03	Não	11	1599,8	18000,0		11	1522,9	1528,9	10,0	4,81	11	1522,9	1528,4	40,0	4,81
h10_c75_150_04	Sim	11	1553,1	7733,0		11	1553,1	1558,1	10,0	0,00	11	1553,1	1557,3	40,0	0,00
h10_c75_150_05	Sim	11	1580,5	3206,0		12	1563,8	1573,3	10,0	1,06	12	1563,8	1570,0	40,0	1,06
h10_c75_150_06	Não	12	1682,4	18000,0		12	1610,1	1629,5	10,0	4,30	12	1610,1	1627,5	40,0	4,30
h10_c75_150_07	Não	11	1612,2	18000,0		11	1542,2	1553,9	10,0	4,34	11	1542,2	1550,9	40,0	4,34
h10_c75_150_08	Sim	12	1615,4	12487,0		12	1619,9	1635,1	10,0	-0,28	12	1617,1	1627,1	40,0	-0,11
h10_c75_150_09	Não	13	1789,3	18000,0		13	1743,6	1761,9	10,0	2,55	13	1743,6	1757,2	40,0	2,55
h10_c75_150_10	Sim	12	1696,6	4735,0		12	1712,2	1719,6	10,0	-0,92	12	1711,6	1717,8	40,0	-0,88
Soma/Média		93		12520,1		116			10,0	1,98	116			40,0	2,01

Tabela 5.36: Resultados para instâncias com 10 hotéis, 75 clientes e tempo de viagem limitado a 200.

Instância	Ótimo	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h10_c75_200_01	Não	8	1570,3	18000,0		8	1489,3	1496,3	10,0	5,16	8	1489,3	1496,6	40,0	5,16
h10_c75_200_02			Sem resultado	18000,0		8	1499,5	1510,7	10,0	-	8	1499,5	1506,3	40,0	-
h10_c75_200_03	Sim	8	1451,6	2007,0		8	1451,8	1452,7	10,0	-0,01	8	1451,8	1452,2	40,0	-0,01
h10_c75_200_04			Sem resultado	18000,0		8	1516,0	1524,7	10,0	-	8	1516,0	1524,4	40,0	-
h10_c75_200_05	Não	8	1566,3	18000,0		8	1497,5	1497,8	10,0	4,39	8	1497,5	1497,8	40,0	4,39
h10_c75_200_06			Sem resultado	18000,0		9	1529,9	1530,4	10,0	-	9	1529,9	1530,3	40,0	-
h10_c75_200_07	Sim	8	1478,1	706,0		8	1490,5	1494,4	10,0	-0,84	8	1490,5	1506,0	40,0	-0,84
h10_c75_200_08			Sem resultado	18000,0		8	1514,8	1498,1	10,0	-	8	1514,8	1499,8	40,0	-
h10_c75_200_09	Não	9	1654,0	18000,0		9	1549,6	1559,3	10,0	6,31	9	1549,6	1557,1	40,0	6,31
h10_c75_200_10	Não	8	1558,8	18000,0		8	1535,3	1545,6	10,0	1,51	8	1535,3	1545,1	40,0	1,51
Soma/Média		49		14671,3		82			10,0	2,75	82			40,0	2,75

Como ocorrido no subgrupo de instâncias com 20 hotéis e 50 clientes, aqui no subgrupo de 20 hotéis e 75 clientes, os resultados obtidos pela MILS são de certa maneira modestos, quando comparados a outros subgrupos. Tal dificuldade também é notada para o BCP, pois das 30 instâncias o mesmo só conseguiu alcançar oito soluções ótimas. Como esperado, o tempo computacional necessário para o BCP foi relativamente alto, tendo média nunca inferior aos 13400 segundos. A variante da MILS com tempo estendido se comporta melhor em relação a sua variante com tempo reduzido, melhorando soluções para 14 instâncias, enquanto a versão com tempo reduzido só consegue melhorar soluções para oito instâncias.

Tabela 5.37: Resultados para instâncias com 20 hotéis, 75 clientes e tempo de viagem limitado a 100.

Instância	BCP					MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h20_c75_100_01	Sim	17	1622,0	199,0		18	1613,7	1628,9	10,0	0,51	18	1613,1	1626,3	40,0	0,55
h20_c75_100_02	Não	18	1706,8	18000,0		18	1674,5	1689,9	10,0	1,89	18	1673,6	1684,8	40,0	1,95
h20_c75_100_03	Não	17	1618,6	18000,0		18	1613,5	1626,6	10,0	0,32	18	1613,5	1620,2	40,0	0,32
h20_c75_100_04	Não	17	1633,3	18000,0		18	1604,7	1607,3	10,0	1,75	18	1604,7	1607,1	40,0	1,75
h20_c75_100_05		Sem resultado	18000,0			17	1537,7	1558,1	10,0	-	17	1537,7	1555,2	40,0	-
h20_c75_100_06	Não	18	1696,1	18000,0		19	1680,0	1693,9	10,0	0,95	19	1675,5	1686,6	40,0	1,21
h20_c75_100_07	Não	17	1611,9	18000,0		18	1588,7	1604,6	10,0	1,44	17	1613,3	1600,0	40,0	-0,09
h20_c75_100_08		Sem resultado	18000,0			18	1583,9	1591,8	10,0	-	18	1583,9	1591,1	40,0	-
h20_c75_100_09	Sim	17	1566,1	5846,0		17	1577,3	1574,9	10,0	-0,72	17	1577,3	1575,1	40,0	-0,72
h20_c75_100_10		Sem resultado	18000,0			17	1626,0	1627,3	10,0	-	17	1626,0	1624,9	40,0	-
Soma/Média		121		15004,5		178			10,0	0,88	177			40,0	0,71

Tabela 5.38: Resultados para instâncias com 20 hotéis, 75 clientes e tempo de viagem limitado a 150.

Instância	BCP					MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h20_c75_150_01	Não	11	1572,2	18000,0		11	1509,7	1514,9	10,0	3,98	11	1509,5	1513,5	40,0	3,99
h20_c75_150_02		Sem resultado	18000,0			11	1531,7	1532,4	10,0	-	11	1529,6	1544,9	40,0	-
h20_c75_150_03	Não	11	1476,0	18000,0		11	1473,8	1484,7	10,0	0,15	11	1470,9	1483,2	40,0	0,35
h20_c75_150_04	Não	11	1582,9	18000,0		11	1507,3	1519,5	10,0	4,78	11	1507,3	1517,5	40,0	4,78
h20_c75_150_05	Não	11	1489,6	18000,0		11	1454,8	1460,6	10,0	2,34	11	1454,8	1459,9	40,0	2,34
h20_c75_150_06	Não	11	1537,0	18004,0		11	1538,6	1539,3	10,0	-0,10	11	1538,6	1538,6	40,0	-0,10
h20_c75_150_07	Não	11	1569,3	18000,0		11	1497,9	1502,1	10,0	4,55	11	1497,7	1500,9	40,0	4,56
h20_c75_150_08	Sim	11	1447,4	3343,0		11	1447,4	1448,8	10,0	0,00	11	1447,4	1447,4	40,0	0,00
h20_c75_150_09	Sim	11	1477,4	7195,0		11	1489,5	1514,3	10,0	-0,82	11	1480,7	1505,3	40,0	-0,22
h20_c75_150_10	Não	11	1552,3	18000,0		11	1466,7	1469,0	10,0	5,51	11	1466,5	1468,3	40,0	5,53
Soma/Média		99		15454,2		110			10,0	2,26	110			40,0	2,36

Tabela 5.39: Resultados para instâncias com 20 hotéis, 75 clientes e tempo de viagem limitado a 200.

Instância	BCP					MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTot	TC(s)		#V	TTot	TMed	TC(s)	Gap(%)	#V	TTot	TMed	TC(s)	Gap(%)
h20_c75_200_01	Não	8	1475,0	18000,0		8	1475,0	1480,8	10,0	0,00	8	1475,0	1479,5	40,0	0,00
h20_c75_200_02	Não	8	1476,9	18000,0		8	1496,2	1498,4	10,0	-1,31	8	1496,2	1497,1	40,0	-1,31
h20_c75_200_03	Sim	8	1412,1	1597,0		8	1422,9	1424,7	10,0	-0,76	8	1422,9	1424,2	40,0	-0,76
h20_c75_200_04	Sim	8	1481,1	3982,0		8	1481,1	1483,3	10,0	0,00	8	1481,1	1483,2	40,0	0,00
h20_c75_200_05	Não	8	1552,9	18000,0		8	1440,2	1441,0	10,0	7,26	8	1440,2	1440,4	40,0	7,26
h20_c75_200_06	Sim	8	1482,2	16398,0		8	1499,1	1502,2	10,0	-1,14	8	1499,1	1500,9	40,0	-1,14
h20_c75_200_07		Sem resultado	18000,0			8	1458,2	1478,1	10,0	-	8	1458,2	1473,7	40,0	-
h20_c75_200_08	Sim	8	1420,6	4220,0		8	1421,4	1438,2	10,0	-0,06	8	1420,6	1436,7	40,0	0,00
h20_c75_200_09	Não	8	1548,7	18000,0		8	1453,4	1454,2	10,0	6,15	8	1453,4	1454,1	40,0	6,15
h20_c75_200_10	Não	8	1435,6	18000,0		8	1436,8	1443,7	10,0	-0,08	8	1435,6	1440,3	40,0	0,00
Soma/Média		72		13419,7		80			10,0	1,12	80			40,0	1,13

O último subgrupo de instâncias proposto por [4] é composto por 100 clientes. Para

os subgrupos que contém 5 hotéis, descritos nas Tabelas 5.40 e 5.41, a MILS apresenta excelente desempenho, tanto na versão com tempo computacional reduzido, quanto na versão com tempo estendido. Para oito instâncias, soluções viáveis são estabelecidas e para outras oito, novas melhores soluções são encontradas. O tempo computacional utilizado na versão da MILS com tempo estendido foi de 60 segundos na média, enquanto o BCP utilizou média superior a 16000 segundos.

Tabela 5.40: Resultados para instâncias com 5 hotéis, 100 clientes e tempo de viagem limitado a 150.

Instância	Ótimo	BCP			#V	MILS (5, 10, 20 s)				#V	MILS (20, 40, 60 s)			
		#V	TTotol	TC(s)		TTotol	TMed	TC(s)	Gap(%)		TTotol	TMed	TC(s)	Gap(%)
h05_c100_150_01	Não	16	2313,4	18000,0	16	2309,3	2319,3	20,0	0,18	16	2309,3	2314,6	60,0	0,18
h05_c100_150_02	Não	17	2443,6	18000,0	17	2350,0	2368,0	20,0	3,83	17	2348,2	2365,7	60,0	3,90
h05_c100_150_03		Sem resultado	18000,0		21	3085,1	3097,1	20,0	-	21	3082,4	3093,3	60,0	-
h05_c100_150_04		Sem resultado	18000,0		18	2486,2	2506,4	20,0	-	18	2486,2	2505,1	60,0	-
h05_c100_150_05	Não	19	2653,0	18000,0	19	2651,5	2678,7	20,0	0,06	19	2650,7	2674,4	60,0	0,09
h05_c100_150_06	Não	18	2480,9	18000,0	18	2481,3	2496,4	20,0	-0,02	18	2481,3	2490,9	60,0	-0,02
h05_c100_150_07	Não	17	2367,0	18000,0	17	2376,3	2406,8	20,0	-0,39	17	2370,7	2402,8	60,0	-0,16
h05_c100_150_08	Sim	16	2196,8	110,0	16	2218,0	2230,9	20,0	-0,97	16	2218,0	2230,2	60,0	-0,97
h05_c100_150_09		Sem resultado	18000,0		19	2753,8	2768,3	20,0	-	19	2753,8	2765,2	60,0	-
h05_c100_150_10		Sem resultado	18000,0		20	2769,1	2780,7	20,0	-	20	2769,1	2780,5	60,0	-
Soma/Média		103		16211,0	181			20,0	0,45	181			60,0	0,51

Tabela 5.41: Resultados para instâncias com 5 hotéis, 100 clientes e tempo de viagem limitado a 200.

Instância	Ótimo	BCP			#V	MILS (5, 10, 20 s)				#V	MILS (20, 40, 60 s)			
		#V	TTotol	TC(s)		TTotol	TMed	TC(s)	Gap(%)		TTotol	TMed	TC(s)	Gap(%)
h05_c100_200_01		Sem resultado	18000,0		11	2024,2	2038,2	20,0	-	11	2024,2	2034,7	60,0	-
h05_c100_200_02	Não	11	2090,3	18000,0	11	2076,3	2086,6	20,0	0,67	11	2076,3	2086,2	60,0	0,67
h05_c100_200_03	Sim	12	2266,1	15490,0	12	2266,1	2285,2	20,0	0,00	12	2266,1	2282,6	60,0	0,00
h05_c100_200_04	Não	11	2137,4	18000,0	11	2124,3	2134,7	20,0	0,61	11	2116,3	2133,8	60,0	0,99
h05_c100_200_05		Sem resultado	18000,0		12	2176,3	2195,0	20,0	-	12	2176,3	2185,2	60,0	-
h05_c100_200_06		Sem resultado	18000,0		11	2071,2	2097,0	20,0	-	11	2071,2	2090,7	60,0	-
h05_c100_200_07		Sem resultado	18000,0		11	2068,5	2093,4	20,0	-	11	2068,5	2089,7	60,0	-
h05_c100_200_08	Não	11	2154,1	18000,0	11	1995,7	1996,8	20,0	7,35	11	1995,7	1996,8	60,0	7,35
h05_c100_200_09	Não	12	2243,4	18000,0	12	2188,9	2200,4	20,0	2,43	12	2188,9	2199,2	60,0	2,43
h05_c100_200_10	Não	12	2222,5	18000,0	12	2199,6	2207,5	20,0	1,03	12	2199,6	2206,2	60,0	1,03
Soma/Média		69		17749,0	114			20,0	2,02	114			60,0	2,08

O subgrupo com 10 hotéis e 100 clientes possui seus resultados descritos nas Tabelas 5.42 até 5.44. No primeiro subgrupo, que possui limite de tempo para viagens igual a 100, foram reportadas 6 instâncias com impossibilidade de construção de uma solução viável. Nos três subgrupos, devido a quantidade de clientes contidos nas instâncias, o BCP não consegue reportar soluções ótimas para nenhuma instância, sendo que em 14 de 30, nem mesmo uma solução viável foi reportada. A MILS por sua vez, excluindo-se aquelas instâncias que são inviáveis por definição, alcança soluções viáveis para todas as instâncias dos subgrupos, sendo estabelecidas 21 novas melhores soluções para os subgrupos descritos.

Por fim, os resultados para os últimos subgrupos compostos por 20 hotéis e 100 clientes são detalhados nas Tabelas 5.45 a 5.47. Nesses grupos, nenhuma instância foi considerada inviável, pois como visto nos subgrupos anteriores, o aumento do número de hotéis e do

Tabela 5.42: Resultados para instâncias com 10 hotéis, 100 clientes e tempo de viagem limitado a 100.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h10_c100_100_01		Sem resultado		18000,0	25	2353,0	2370,1	20,0	-	25	2353,0	2366,1	60,0	-
h10_c100_100_02		Inviável					Inviável					Inviável		
h10_c100_100_03		Inviável					Inviável					Inviável		
h10_c100_100_04		Inviável					Inviável					Inviável		
h10_c100_100_05		Inviável					Inviável					Inviável		
h10_c100_100_06		Sem resultado		18000,0	26	2381,7	2399,5	20,0	-	26	2381,7	2393,9	60,0	-
h10_c100_100_07		Sem resultado		18000,0	30	2818,8	2832,8	20,0	-	30	2818,8	2833,6	60,0	-
h10_c100_100_08		Sem resultado		18000,0	25	2335,2	2344,9	20,0	-	25	2335,2	2342,9	60,0	-
h10_c100_100_09		Inviável					Inviável					Inviável		
h10_c100_100_10		Inviável					Inviável					Inviável		
Soma/Média				18000,0	106			20,0		106			60,0	

Tabela 5.43: Resultados para instâncias com 10 hotéis, 100 clientes e tempo de viagem limitado a 150.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h10_c100_150_01	Não	14	1993,0	18000,0	14	2000,1	1996,8	20,0	-0,36	14	1996,2	1995,6	60,0	-0,16
h10_c100_150_02		Sem resultado		18000,0	15	2105,7	2127,2	20,0	-	15	2101,7	2124,9	60,0	-
h10_c100_150_03	Não	15	2221,4	18000,0	16	2180,4	2206,6	20,0	1,85	16	2180,4	2202,0	60,0	1,85
h10_c100_150_04		Sem resultado		18000,0	16	2255,7	2273,8	20,0	-	16	2252,0	2271,9	60,0	-
h10_c100_150_05	Não	16	2283,2	18000,0	16	2242,6	2265,6	20,0	1,78	16	2237,5	2262,1	60,0	2,00
h10_c100_150_06	Não	14	2054,3	18000,0	14	2016,2	2017,1	20,0	1,85	14	2016,2	2016,5	60,0	1,85
h10_c100_150_07	Não	15	2096,7	18000,0	15	2096,8	2159,1	20,0	0,00	15	2096,8	2158,4	60,0	0,00
h10_c100_150_08		Sem resultado		18000,0	14	1998,9	1988,5	20,0	-	14	1998,9	1988,1	60,0	-
h10_c100_150_09		Sem resultado		18000,0	16	2155,3	2165,1	20,0	-	16	2155,3	2163,5	60,0	-
h10_c100_150_10	Não	15	2095,1	18000,0	15	2075,8	2090,3	20,0	0,92	15	2075,8	2089,5	60,0	0,92
Soma/Média		89		18000,0	151			20,0	1,01	151			60,0	1,08

limite de tempo das viagens facilita o alcance de soluções viáveis. A solução ótima é conhecida para apenas uma instância dos subgrupos e o BCP é capaz de encontrar apenas 11 soluções viáveis dentre o total de 30 instâncias. A heurística MILS segue a tendência vista para instâncias com menor número de clientes e aumenta o número de melhores soluções encontradas, totalizando 25 melhores soluções para estes subgrupos. Em relação ao tempo computacional demandando, o algoritmo BCP demanda para os três subgrupos com 100 clientes, o tempo máximo disponibilizado de 18000 segundos, o que demonstra sua limitação em resolver instâncias com número de clientes acima deste limiar. Por mais que exista a possibilidade de resolução de instâncias maiores com o BCP, na maioria das vezes, a solução é de baixa qualidade. Ou seja, com o aumento do número de clientes o

Tabela 5.44: Resultados para instâncias com 10 hotéis, 100 clientes e tempo de viagem limitado a 200.

Instância	BCP				MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
	Ótimo	#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h10_c100_200_01		Sem resultado		18000,0	10	1893,0	1896,8	20,0	-	10	1893,0	1896,8	60,0	-
h10_c100_200_02		Sem resultado		18000,0	10	1950,0	1957,9	20,0	-	10	1950,0	1953,0	60,0	-
h10_c100_200_03		Sem resultado		18000,0	11	2007,3	2026,0	20,0	-	11	2007,3	2024,7	60,0	-
h10_c100_200_04	Não	11	2075,5	18000,0	11	1993,9	2012,7	20,0	3,93	11	1993,9	2010,6	60,0	3,93
h10_c100_200_05	Não	11	2040,2	18000,0	11	1966,7	1985,0	20,0	3,60	11	1966,7	1983,3	60,0	3,60
h10_c100_200_06	Não	10	1908,6	18000,0	10	1897,8	1906,2	20,0	0,57	10	1897,8	1904,9	60,0	0,57
h10_c100_200_07		Sem resultado		18000,0	11	1971,6	1990,1	20,0	-	11	1957,1	1985,4	60,0	-
h10_c100_200_08		Sem resultado		18000,0	10	1906,0	1906,8	20,0	-	10	1906,0	1905,7	60,0	-
h10_c100_200_09		Sem resultado		18000,0	11	1938,1	1945,1	20,0	-	11	1938,1	1940,6	60,0	-
h10_c100_200_10	Não	10	1926,3	18000,0	10	1914,4	1926,6	20,0	0,62	10	1914,4	1924,9	60,0	0,62
Soma/Média		42		18000,0	105			20,0	2,18	105			60,0	2,18

uso de estratégias heurísticas como a MILS, tende a ser uma possibilidade inteligente para lidar com o aumento da complexidade das soluções e do espaço disponível para exploração da busca por soluções.

Tabela 5.45: Resultados para instâncias com 20 hotéis, 100 clientes e tempo de viagem limitado a 100.

Instância	Ótimo	BCP			MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h20_c100_100_01	Sim	21	1971,8	18000,0	21	2003,3	1991,2	20,0	-1,60	21	2003,3	1988,6	60,0	-1,60
h20_c100_100_02	Não	23	2221,8	18000,0	24	2115,8	2157,8	20,0	4,77	23	2155,2	2148,6	60,0	3,00
h20_c100_100_03	Não	22	2123,4	18000,0	23	2078,4	2078,6	20,0	2,12	23	2078,4	2078,5	60,0	2,12
h20_c100_100_04	Não	Sem resultado			23	2105,0	2132,7	20,0	-	23	2105,0	2127,8	60,0	-
h20_c100_100_05		24	2299,9	18000,0	24	2261,4	2280,5	20,0	1,67	24	2261,4	2277,3	60,0	1,67
h20_c100_100_06		Sem resultado			23	2080,2	2096,4	20,0	-	23	2079,2	2092,9	60,0	-
h20_c100_100_07		Sem resultado			26	2381,5	2404,3	20,0	-	26	2381,5	2399,5	60,0	-
h20_c100_100_08	Não	Sem resultado			22	2064,4	2063,7	20,0	-	22	2064,4	2062,1	60,0	-
h20_c100_100_09		22	2066,9	18000,0	22	2023,8	2050,7	20,0	2,09	22	2023,8	2045,2	60,0	2,09
h20_c100_100_10		Sem resultado			22	2050,9	2054,1	20,0	-	22	2030,6	2054,6	60,0	-
Soma/Média		112		18000,0	230			20,0	1,81	229			60,0	1,46

Tabela 5.46: Resultados para instâncias com 20 hotéis, 100 clientes e tempo de viagem limitado a 150.

Instância	Ótimo	BCP			MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h20_c100_150_01	Não	13	1908,0	18000,0	13	1868,6	1868,7	20,0	2,06	13	1860,3	1863,2	60,0	2,50
h20_c100_150_02	Não	Sem resultado			14	1931,5	1931,6	20,0	-	14	1931,5	1936,6	60,0	-
h20_c100_150_03		Sem resultado			14	1926,9	1938,3	20,0	-	14	1912,9	1936,1	60,0	-
h20_c100_150_04		Sem resultado			14	1912,1	1929,7	20,0	-	14	1909,5	1927,5	60,0	-
h20_c100_150_05		14	1951,7	18000,0	14	1925,4	1934,6	20,0	1,35	14	1925,4	1934,0	60,0	1,35
h20_c100_150_06	Não	13	1902,1	18000,0	14	1896,9	1900,0	20,0	0,27	14	1895,0	1898,0	60,0	0,37
h20_c100_150_07	Não	Sem resultado			15	1993,6	2023,9	20,0	-	15	1990,2	2018,4	60,0	-
h20_c100_150_08		Sem resultado			14	1859,8	1877,6	20,0	-	14	1859,8	1873,2	60,0	-
h20_c100_150_09	Não	13	1873,8	18000,0	13	1897,5	1883,7	20,0	-1,26	13	1892,7	1883,3	60,0	-1,01
h20_c100_150_10	Não	13	1912,1	18000,0	14	1868,9	1889,1	20,0	2,26	14	1864,8	1884,8	60,0	2,47
Soma/Média		66		18000,0	139			20,0	0,94	139			60,0	1,14

Tabela 5.47: Resultados para instâncias com 20 hotéis, 100 clientes e tempo de viagem limitado a 200.

Instância	Ótimo	BCP			MILS (5, 10, 20 s)					MILS (20, 40, 60 s)				
		#V	TTotol	TC(s)	#V	TTotol	TMed	TC(s)	Gap(%)	#V	TTotol	TMed	TC(s)	Gap(%)
h20_c100_200_01	Não	Sem resultados			10	1817,9	1835,2	20,0	-	10	1817,9	1833,1	60,0	-
h20_c100_200_02		Sem resultados			10	1859,7	1860,8	20,0	-	10	1859,7	1860,2	60,0	-
h20_c100_200_03		Sem resultados			10	1853,8	1859,0	20,0	-	10	1853,8	1856,3	60,0	-
h20_c100_200_04		Sem resultados			10	1863,8	1887,2	20,0	-	10	1863,8	1882,7	60,0	-
h20_c100_200_05		Sem resultados			10	1872,7	1879,0	20,0	-	10	1872,7	1879,4	60,0	-
h20_c100_200_06		Sem resultados			10	1813,3	1833,0	20,0	-	10	1813,2	1829,3	60,0	-
h20_c100_200_07		Sem resultados			10	1889,9	1903,9	20,0	-	10	1889,9	1901,6	60,0	-
h20_c100_200_08		Sem resultados			10	1845,5	1855,6	20,0	-	10	1838,4	1852,3	60,0	-
h20_c100_200_09		Sem resultados			10	1791,0	1829,2	20,0	-	10	1791,0	1827,3	60,0	-
h20_c100_200_10		10	1893,2	18000,0	10	1813,6	1831,7	20,0	4,20	10	1813,6	1828,2	60,0	4,20
Soma/Média		10		18000,0	100			20,00	4,20	100			60,0	4,20

Os resultados reportados pela MILS são comparados aos resultados obtidos pelo BCP e apresentados de forma resumida na Tabela 5.48. Assim como nas tabelas anteriores, a primeira coluna traz a informação acerca da composição do referido subgrupo de instâncias. A próxima coluna apresenta o número de soluções inviáveis. As três próximas colunas mostram o número de instâncias para as quais a MILS alcança melhores resultados (*Vence*), mesmos resultados (*Empata*) ou piores resultados (*Perde*) utilizando um tempo

computacional reduzido. As últimas três colunas apresentam os resultados da MILS utilizando mais tempo computacional para resolver cada instância dos subgrupos. De forma complementar, a última linha da tabela apresenta o número correspondente à soma dos resultados reportados para cada coluna.

Tabela 5.48: Resumo das comparações entre o Algoritmo BCP e a MILS.

Instâncias	Inviável	MILS (5, 10, 20 s)			MILS (20, 40, 60 s)		
		Vence	Empata	Perde	Vence	Empata	Perde
h05_c50_150	4	0	5	1	0	5	1
h05_c50_200		2	5	3	2	7	1
h10_c50_100		0	2	3	0	2	3
h10_c50_150		2	3	5	2	5	3
h10_c50_200		0	7	3	0	8	2
h20_c50_100	5	2	1	7	2	1	7
h20_c50_150		1	5	4	1	5	4
h20_c50_200		3	4	3	3	4	3
h05_c75_150		4	4	1	4	4	1
h05_c75_200		7	0	3	7	1	2
h10_c75_100	3	1	0	6	1	0	6
h10_c75_150		6	1	3	6	1	3
h10_c75_200		8	0	2	8	0	2
h20_c75_100		4	0	6	4	0	6
h20_c75_150		7	1	2	7	1	2
h20_c75_200	6	3	2	5	3	4	3
h05_c100_150		7	0	3	7	0	3
h05_c100_200		9	1	0	9	1	0
h10_c100_100		4	0	0	4	0	0
h10_c100_150		7	0	3	7	0	3
h10_c100_200	6	10	0	0	10	0	0
h20_c100_100		7	0	3	8	0	2
h20_c100_150		7	0	3	7	0	3
h20_c100_200		10	0	0	10	0	0
Soma	19	111	41	69	112	49	60

Os resultados apresentados na Tabela 5.48 mostram que a abordagem exata BCP alcança melhores resultados para instâncias com menores números de clientes, enquanto a MILS atinge seus melhores resultados à medida que o número de clientes das instâncias vai aumentando.

Com o objetivo de apresentar de forma gráfica os resultados alcançados e apresentados na Tabela 5.48, a Figura 5.2 ilustra os resultados sintetizados para cada grupo de instâncias que contém o mesmo número de clientes. Barras com legenda (5, 10, 20 s) ilustram os resultados obtidos com tempos computacionais reduzidos, enquanto barras com legenda (20, 40, 60 s) ilustram resultados para execuções com tempos computacionais

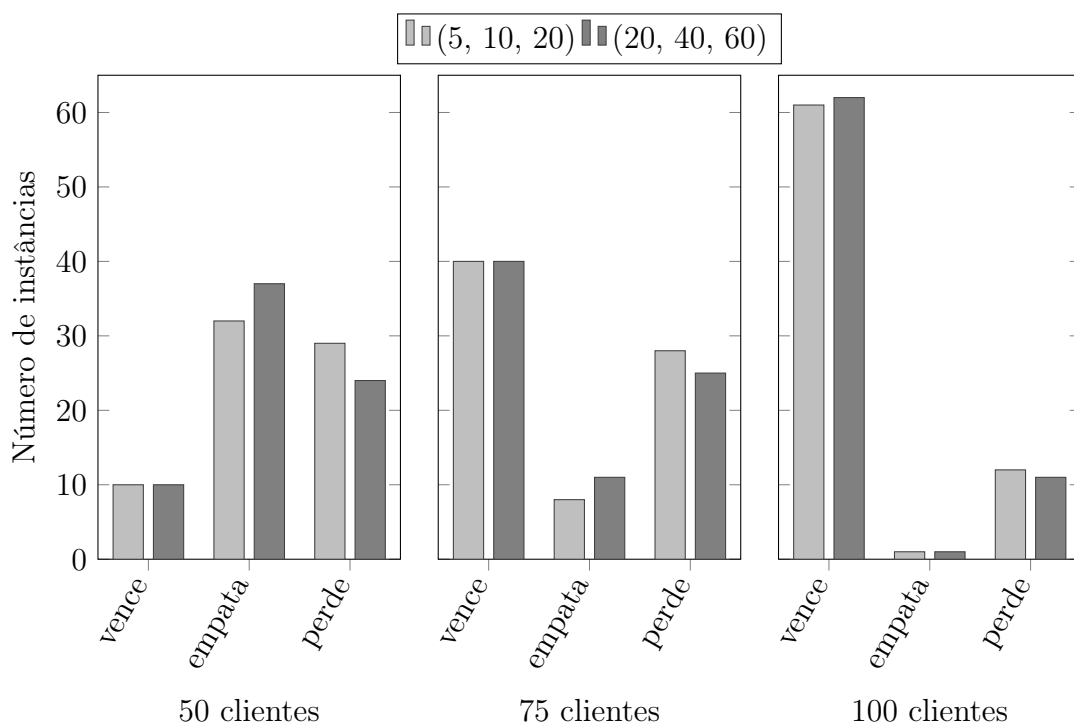


Figura 5.2: Comparação de resultados para MILS com diferentes tempos computacionais.

estendidos. Ao prover mais tempo computacional, a MILS é capaz de aumentar o número de resultados idênticos ao BCP para instâncias com 50 e 75 clientes, mantendo o mesmo número de melhores resultados reportados. Para instâncias com 100 clientes, há aumento de uma melhor solução e decremento de uma pior solução para os três grupos reportados.

Uma observação importante é que o BCP aumenta consideravelmente o uso de tempo computacional para chegar à solução reportada quando o número de clientes também aumenta. Para grande maioria das instâncias maiores, o BCP utiliza quase sempre o limite máximo de tempo computacional definido e, com frequência, não consegue reportar uma solução que seja viável para o PCVSH.

5.3 Análise de sensibilidade da mineração de dados

Nesta seção, é discutida a contribuição que a mineração de dados provê nas melhoras dos resultados obtidos pela ILS-RVND. Foram executadas todas as 371 instâncias propostas por [4, 46] com e sem o procedimento de mineração de dados. As Tabelas 5.49 a 5.51 apresentam os resultados, onde a coluna “MSC” representa as melhores soluções encontradas na literatura com seus respectivos números de viagens “#V” e o tempo total da solução “TTotal”. As próximas duas colunas apresentam resultados obtidos pelas heurísticas sem

e com o procedimento de mineração aplicado. Foram reportados somente resultados onde diferenças no número de viagens e/ou tempos totais de soluções ocorreram.

A execução das instâncias propostas por [46] obteve para ambas heurísticas os mesmos melhores resultados para o SET2 e SET3. Para instâncias do SET1 e SET4, foram obtidos os mesmos números de viagens, mas diferentes valores para o tempo total das soluções em algumas instâncias relatadas na Tabela 5.49 e Tabela 5.50. Para 3 instâncias do SET1 e 11 do SET4, a heurística sem mineração de dados obteve piores soluções, apresentando maiores tempos totais das rotas nas soluções. Portanto, para todas as 131 instâncias a ILS-RVND com mineração de dados sempre obteve desempenho similar ou melhor em relação à heurística ILS-RVND pura.

Tabela 5.49: Comparação entre ILS-RVND com e sem procedimento de mineração de dados para o SET1 [46].

Instância	MSC		ILS-RVND sem mineração de dados				ILS-RVND com mineração de dados			
	#V	TTotol	#V	TTotol	TC(s)	Gap(%)	#V	TTotol	TC(s)	Gap(%)
c101	9	9591,1	8	9632,4	19,9	-0,43	8	9624,4	19,9	-0,35
pr05	5	4948,9	5	4950,9	282,5	-0,04	5	4947,1	282,5	0,04
pr08	4	3367,7	4	3368,5	55,5	-0,02	4	3366,2	55,5	0,04

Tabela 5.50: Comparação entre ILS-RVND com e sem procedimento de mineração de dados para o SET4 [46].

Instância	MSC		ILS-RVND sem mineração de dados				ILS-RVND com mineração de dados			
	#V	TTotol	#V	TTotol	TC(s)	Gap(%)	#V	TTotol	TC(s)	Gap(%)
st70	6	718	6	731	8,5	-1,81	6	722	8,5	-0,56
eil76	6	539	6	547	10,5	-1,48	6	539	10,5	0,00
pr76	6	118061	6	118983	6,6	-0,78	6	118061	6,6	0,00
kroA100	6	22205	6	22205	16,9	0,00	6	21991	16,9	0,96
kroD100	6	21464	6	21664	14,1	-0,93	6	21464	14,1	0,00
rd100	6	8244	6	8311	19,7	-0,81	6	8243	19,7	0,01
ch150	6	6578	6	6614	46,6	-0,55	6	6586	46,6	-0,12
tsp225	6	4502	6	4483	99,9	0,42	6	4377	99,9	2,78
a280	6	2645	6	2668	140,0	-0,87	6	2636	140,0	0,34
pcb442	6	54137	6	54810	739,6	-1,24	6	53452	739,6	1,27
pr1002	7	289337	6	286838	2929,8	0,86	6	285993	2929,8	1,16

A Tabela 5.51 apresenta os resultados para instâncias propostas por [4] que durante a execução da ILS-RVND com e sem mineração de dados apresentaram diferenças significativas. Para instâncias que o Algoritmo BCP obteve uma solução viável, o *Gap* foi calculado entre a solução do BCP e a solução heurística. Caso contrário, quando o símbolo “-” é utilizado, significa que não há resultado reportado pelo BCP e o *Gap* foi calculado entre o melhor valor encontrado pelas duas heurísticas e o valor obtido por cada heurística individualmente.

Comparando a qualidade das soluções, quando aplicada a mineração de dados 18 melhores soluções foram obtidas, enquanto quando não utilizada, somente 12 foram alcança-

das. Ambas heurísticas obtiveram o mesmo número de viagens para todas as instâncias, exceto na h10_c50_150_09 para qual ILS-RVND sem mineração necessitou de uma viagem a menos. Em relação ao *Gap* médio, ambas obtiveram *Gap* positivo, entretanto, o *Gap* foi aproximadamente 9 vezes maior quando utilizado o procedimento de mineração de dados confirmando a contribuição da mineração de dados.

Tabela 5.51: Comparação entre ILS-RVND com e sem procedimento de mineração de dados para todos os grupos propostos por [4].

Instância	MSC		ILS-RVND sem mineração de dados				ILS-RVND com mineração de dados			
	#V	TTot	#V	TTot	TC(s)	Gap(%)	#V	TTot	TC(s)	Gap(%)
h10_c50_150_09	8	1164,3	8	1164,3	20,0	0,00	9	1114,9	20,0	4,24
h10_c50_150_10	9	1274,3	9	1301,6	20,0	-2,14	9	1274,3	20,0	0,00
h10_c50_200_02	6	1117,9	6	1125,0	20,0	-0,64	6	1117,9	20,0	0,00
h10_c50_200_05	6	1101,7	6	1122,7	20,0	-1,91	6	1101,7	20,0	0,00
h10_c50_200_08	6	1077,0	6	1080,8	20,0	-0,35	6	1077,4	20,0	-0,04
h20_c50_100_08	13	1164,1	13	1185,2	20,0	-1,81	13	1175,8	20,0	-1,01
h20_c50_100_10	14	1270,8	14	1283,0	20,0	-0,96	14	1283,5	20,0	-1,00
h20_c50_200_04	6	1151,5	6	1154	20,0	-0,22	6	1146,3	20,0	0,45
h05_c75_200_02	9	1749,5	9	1628,4	40,0	6,92	9	1630,0	40,0	6,83
h10_c75_150_10	12	1696,6	12	1711,2	40,0	-0,86	12	1711,6	40,0	-0,88
h10_c75_200_04	-	-	8	1520,5	40,0	-0,30	8	1516,0	40,0	0,00
h20_c75_150_09	11	1477,4	11	1500,6	40,0	-1,57	11	1480,7	40,0	-0,22
h20_c75_200_09	8	1548,7	8	1452,5	40,0	6,21	8	1453,4	40,0	6,15
h05_c100_150_05	19	2653,0	19	2651,5	60,0	0,06	19	2650,7	60,0	0,09
h05_c100_200_02	11	2090,3	11	2075,6	60,0	0,70	11	2076,3	60,0	0,67
h10_c100_100_06	-	-	26	2379,8	60,0	0,00	26	2381,7	60,0	-0,08
h10_c100_100_08	-	-	25	2335,7	60,0	-0,02	25	2335,2	60,0	0,00
h10_c100_150_01	14	1993,0	14	1993,4	60,0	-0,02	14	1996,2	60,0	-0,16
h10_c100_150_02	-	-	15	2104,5	60,0	-0,13	15	2101,7	60,0	0,00
h10_c100_150_04	-	-	16	2255,7	60,0	-0,16	16	2252,0	60,0	0,00
h10_c100_150_09	-	-	16	2156,1	60,0	-0,04	16	2155,3	60,0	0,00
h10_c100_200_01	-	-	10	1904,9	60,0	-0,63	10	1893,0	60,0	0,00
h10_c100_200_03	-	-	11	2006,7	60,0	0,00	11	2007,3	60,0	-0,03
h20_c100_100_02	23	2221,8	23	2193,9	60,0	1,26	23	2155,2	60,0	3,00
h20_c100_100_06	-	-	23	2080,3	60,0	-0,05	23	2079,2	60,0	0,00
h20_c100_100_10	-	-	22	2051,6	60,0	-1,03	22	2030,6	60,0	0,00
h20_c100_150_02	-	-	14	1929,7	60,0	0,00	14	1931,5	60,0	-0,09
h20_c100_150_07	-	-	15	1993,6	60,0	-0,17	15	1990,2	60,0	0,00
h20_c100_150_04	-	-	10	1863,7	60,0	0,00	10	1863,8	60,0	-0,01
h20_c100_150_07	-	-	10	1889,7	60,0	0,00	10	1889,9	60,0	-0,01
Média						0,07				0,60

5.4 Resultados para hibridização entre MILS e Método Exato

Nesta seção, apresentam-se os testes computacionais executados para a abordagem hibridizada utilizando a heurística ILS-RVND em conjunto com a técnica de mineração de dados FIM e resolução exata, conforme apresentada na Seção 3.3. Os parâmetros utilizados para testar essa abordagem foram os mesmos definidos por meio da calibração da heurística MILS. Com foco no fornecimento de uma solução heurística de qualidade para

o método exato, o procedimento de mineração de dados foi executado uma única vez e o padrão que possui maior tamanho foi utilizado para fixação de arestas que devem aparecer como restrições para a resolução do modelo inteiro. Foi realizada uma única execução do procedimento ILS-RVND que teve como limite de execução o tempo computacional reportado pelo HDM. O tempo computacional limite para execução do método exato foi fixado em 18000 segundos e os seguintes parâmetros foram definidos no solver: (i) *threads* = 4 e (ii) *MIP Emphasis* = 1 (ênfase em viabilidade sobre a otimalidade).

Foram executados testes computacionais apenas para o SET1 e SET2, sendo os resultados apresentados entre as Tabelas 5.52 e 5.56. Nessas tabelas, a primeira coluna traz a referência ao nome da instância executada, enquanto as colunas seguintes são compostas por subcolunas que trazem dados sobre a MSC e os resultados alcançados pela execução da heurística híbrida (ILS-RVND + Mineração de arestas + aplicação de método exato em duas fases (relaxado e inteiro)). A Tabela 5.52 traz os resultados para o grupo de instâncias SET1 e, por não ter MSC extraída de um método exato para todas as instâncias, traz a coluna “Proced.” identificando a abordagem utilizada para gerar a MSC, sendo “[H]” para Heurística e “[E]” para Método Exato. As demais tabelas que trazem os resultados para o SET2 não apresentam essa coluna, pois todas as MSCs são provenientes de um método exato. As colunas “#V” destacam o número de viagens necessárias para visitar todos os clientes, considerando o tempo total gasto descrito na coluna “TTotal”. Em relação às colunas de resultados para a hibridização proposta, “*Aminer*” e “*Arelax*” apresentam, respectivamente, as quantidades de arestas mineradas e de arestas que apresentaram no modelo relaxado valores maiores que 0.9, sendo ambas utilizadas na resolução inteira do problema. A coluna “*Bound*” apresenta o *LowerBound* alcançado para os casos em que uma solução inteira não foi reportada. A coluna “Heurística” traz o tempo total obtido pela execução da heurística ILS-RVND, enquanto a coluna “Exato” traz o tempo total referente à solução final retornada pelo método exato. A coluna *Gap* apresenta o desvio percentual entre as soluções encontradas (heurísticas ou exatas) e a MSC, calculadas de acordo com a Equação (5.1). Nas tabelas desta subseção, o símbolo “-” significa que não foi possível obter aquele dado para a execução daquela determinada instância.

Considerando os resultados apresentados na Tabela 5.52, as quatro instâncias reportadas com o símbolo “*” tiveram comportamento atípico durante a execução, sendo o processo terminado antes do tempo computacional limite, devido ao crescimento rápido do tamanho da árvore criada pelo *solver*. Para nenhuma das instâncias deste grupo, o método exato apresentou melhora em relação à solução proveniente da heurística ILS-RVND. Para instâncias onde não foi possível obter uma solução exata inteira, existiram casos

que não foi possível utilizar a solução heurística como solução inicial, porque ocorreram conflitos entre as restrições definidas pelas arestas mineradas e pelas arestas relaxadas, impossibilitando assim a viabilidade da solução heurística.

Tabela 5.52: Resultados obtidos pelo método híbrido para o SET1.

Instância	Proced.	MSC		ILS + Minerador + Exato								TC(s)	Gap(%)
		#V	TTotal	#Aminer	#Arelax	Bound	#V	Heurística	Gap(%)	#V	Exato		
c101	[H]	9	9591,1	28	0	-	8	9642,3	-0,53	8	9642,3	18265,66	-0,53
r101*	[H]	8	1695,5	17	0	-	8	1699,0	-0,21	8	1699,0	8109,47	-0,21
rc101	[E]	8	1672,6	21	0	-	8	1672,6	0,00	8	1672,6	18272,7	0,00
c201	[E]	3	9559,7	33	0	-	3	9561,8	-0,02	8	9561,8	18031,09	-0,02
r201	[H]	2	1642,8	3	1	-	2	1642,6	0,01	2	1642,6	18021,04	0,01
rc201	[H]	2	1642,7	23	2	1546,5	2	1642,7	0,00	-	-	18020,8	-
pr01	[H]	2	1412,2	11	0	-	2	1412,2	0,00	2	1412,2	455,24	0,00
pr02*	[H]	3	2543,3	47	0	2516,31	3	2548,5	-0,20	-	-	8598,07	-
pr03	[H]	4	3404,2	32	1	-	4	3428,6	-0,72	4	3428,6	18145,15	-0,72
pr04*	[H]	5	4215,3	75	0	-	5	4213,2	0,05	5	4213,2	8892,84	0,05
pr05	[H]	5	4948,9	81	0	-	6	4971,5	-0,46	6	4971,5	21778,33	-0,46
pr06	[H]	7	5960,9	42	1	-	7	5995,6	-0,58	7	5995,6	29339,21	-0,58
pr07	[E]	3	2069,6	10	0	947,9	3	2070,3	-0,03	-	-	18021,1	-
pr08	[H]	4	3367,7	16	1	-	4	3368,5	-0,02	4	3368,5	18147,3	-0,02
pr09*	[H]	5	4414,9	71	0	-	5	4442,7	-0,63	5	4442,7	10145,73	-0,63
pr10	[H]	7	5932,0	55	0	-	7	5977,2	-0,76	7	5977,2	36785,87	-0,76
Soma/Média		76					77		-0,26	66		17403,7	-0,28

Para as instâncias contidas no SET2 e seus subgrupos, os resultados apresentados nas Tabelas 5.53 a 5.56 demonstram que para os subgrupos com menor número de clientes, na maioria das instâncias, não é possível aplicar a mineração, pois o *pool* de soluções contém solução única. Novamente, foi observado que para todas as instâncias onde uma solução foi retornada pelo *solver*, essa solução já tinha sido reportada pela heurística ILS-RVND. Para as instâncias onde não foi encontrada uma solução exata, o fato de ter sido definido um *Upper Bound* igual ao valor da solução heurística, combinado com a definição de restrições baseadas nas arestas do modelo relaxado e mineração (quando for o caso), implicou em inviabilidade das soluções.

É possível verificar que a tentativa de redução do espaço de busca por meio da definição de arestas, sejam elas obtidas por meio de mineração de arestas e/ou relaxamento do modelo, pode em alguns caso facilitar e viabilizar a resolução do modelo. Por outro lado, podem significar restrições tão fortes que inviabilizam a definição de soluções, mesmo aquelas que foram previamente definidas por métodos heurísticos.

Outras variações da proposta de hibridização foram verificadas, como por exemplo, usar ou não a injeção de uma solução heurística, usar ou não *Upper Bound* heurístico, usar ou não a resolução do modelo relaxado e, uma proposta um pouco mais sofisticada, realizando a contração de arestas mineradas, onde dada uma sequência de arestas com 3 ou mais clientes consecutivos, os clientes intermediários foram suprimidos e os tempos relativos ao tempo de visita e tempo de percurso entre clientes foram somados para obter o novo tempo de percurso da aresta que liga os dois clientes mais externos. A ideia

Tabela 5.53: Resultados obtidos pelo método híbrido para o SET2 com 10 clientes.

Instância	MSC Exato		ILS + Minerador + Exato									
	#V	TTotol	#Aminer	#Arelax	Bound	#V	Heurística	Gap(%)	#V	Exato	TC(s)	Gap(%)
c101.k10	1	955,1	0	2	-	1	955,1	0,00	1	955,1	<1	0,00
r101.k10	2	272,8	3	8	-	2	272,8	0,00	2	272,8	<1	0,00
rc101.k10	1	237,5	0	3	-	1	237,5	0,00	1	237,5	7,2	0,00
pr01.k10	1	426,6	0	4	391,9	1	426,6	0,00	-	-	<1	-
pr02.k10	1	661,9	0	1	-	1	661,9	0,00	1	661,9	<1	0,00
pr03.k10	1	553,3	0	2	-	1	553,3	0,00	1	553,3	<1	0,00
pr04.k10	1	476,4	0	1	-	1	476,4	0,00	1	476,4	<1	0,00
pr05.k10	1	528,9	0	4	-	1	528,9	0,00	1	528,9	<1	0,00
pr06.k10	1	597,4	0	2	-	1	597,4	0,00	1	597,4	<1	0,00
pr07.k10	1	670,2	0	11	-	1	670,2	0,00	1	670,2	<1	0,00
pr08.k10	1	573,4	0	2	-	1	573,4	0,00	1	573,4	<1	0,00
pr09.k10	1	645,5	0	8	600,4	1	645,5	0,00	-	-	<1	-
pr10.k10	1	461,5	0	4	442,4	1	461,5	0,00	-	-	<1	-
Soma/Média	14					14		0,00	11		7,2	0,00

Tabela 5.54: Resultados obtidos pelo método híbrido para o SET2 com 15 clientes.

Instância	MSC Exato		ILS + Minerador + Exato									
	#V	TTotol	#Aminer	#Arelax	Bound	#V	Heurística	Gap(%)	#V	Exato	TC(s)	Gap(%)
c101.k15	2	1452,2	7	7	-	2	1452,2	0,00	2	1452,2	<1	0,00
r101.k15	2	379,8	0	3	-	2	379,8	0,00	2	379,8	<1	0,00
rc101.k15	2	303,2	3	10	-	2	303,2	0,00	-	-	<1	-
pr01.k15	1	590,4	0	3	-	1	590,4	0,00	-	-	<1	-
pr02.k15	1	745,6	0	1	-	1	745,6	0,00	1	745,6	<1	0,00
pr03.k15	1	632,9	0	3	-	1	632,9	0,00	1	632,9	<1	0,00
pr04.k15	1	683,4	0	11	681,7	1	683,4	0,00	-	-	<1	-
pr05.k15	1	621,2	0	5	-	1	621,2	0,00	1	621,2	<1	0,00
pr06.k15	1	685,2	0	3	-	1	685,2	0,00	1	685,2	<1	0,00
pr07.k15	1	795,3	1	7	782,1	1	795,3	0,00	-	-	<1	-
pr08.k15	1	707,2	8	0	-	1	707,2	0,00	1	707,2	<1	0,00
pr09.k15	1	771,7	0	16	744,9	1	771,7	0,00	-	-	<1	-
pr10.k15	1	611,9	0	2	-	1	611,9	0,00	1	611,9	<1	0,00
Soma/Média	16					16		0,00	10		<1	0,00

com essa contração seria reduzir o número de clientes a serem visitados, tornando a instância mais simples e, conseqüentemente, mais fácil de se resolver. Após a execução do *solver*, as arestas que foram contraídas eram expandidas para apresentação dos clientes intermediários. Os resultados alcançados com a estratégia de contração também não foram promissores, mas a estratégia pode representar um caminho a ser melhor explorado.

Apesar de não terem sido alcançados resultados promissores utilizando essa abordagem hibridizada, optou-se por apresentá-los parcialmente como forma de guiar outras pesquisas que possam surgir acerca do tema.

Tabela 5.55: Resultados obtidos pelo método híbrido para o SET2 com 30 clientes.

Instância	MSC Exato		ILS + Minerador + Exato									
	#V	TTotol	#Aminer	#Arelax	Bound	#V	Heurística	Gap(%)	#V	Exato	TC(s)	Gap(%)
c101.k30	3	2829,4	1	9	2819,60	3	2863,2	-1,19	-	-	18005,1	-
r101.k30	3	655,2	7	0	-	3	655,2	0,00	3	655,2	1229,5	0,00
rc101.k30	3	610,0	0	0	-	3	705,5	-15,66	3	705,5	18003,6	-15,66
pr01.k30	1	964,8	21	6	-	1	964,8	0,00	-	-	<1	-
pr02.k30	2	1078,3	17	3	-	2	1078,3	0,00	2	1078,3	2,3	0,00
pr03.k30	1	952,5	0	12	903,6	1	952,5	0,00	-	-	<1	-
pr04.k30	2	1091,6	19	4	1123,9	2	1091,6	0,00	-	-	1,7	-
pr05.k30	1	924,7	0	14	-	1	924,7	0,00	-	-	<1	-
pr06.k30	2	1063,2	8	0	-	2	1063,2	0,00	2	1063,2	2,9	0,00
pr07.k30	2	1130,4	0	0	-	2	1130,4	0,00	2	1130,4	43,4	0,00
pr08.k30	2	1006,2	16	3	-	2	1006,2	0,00	2	1006,2	2,0	0,00
pr09.k30	2	1091,4	5	4	-	2	1091,4	0,00	2	1091,4	2,0	0,00
pr10.k30	1	918,9	14	14	915,0	1	918,9	0,00	-	-	<1	-
Soma/Média	25					25		-1,30	16		<1	-2,24

Tabela 5.56: Resultados obtidos pelo método híbrido para o SET2 com 40 clientes.

Instância	MSC Exato		ILS + Minerador + Exato									
	#V	TTotol	#Aminer	#Arelax	Bound	#V	Heurística	Gap(%)	#V	Exato	TC(s)	Gap(%)
c101.k40	4	3817,5	3	0	-	4	3866,1	-1,27	4	3866,1	18008,2	-1,27
r101.k40	4	842,9	3	0	-	4	862,8	-2,36	4	862,8	18007,9	-2,36
rc101.k40	3	652,1	9	11	179,7	4	850,3	-30,39	-	-	18006,0	-
pr01.k40	2	1160,5	0	0	-	2	1160,5	0,00	2	1160,5	85,0	0,00
pr02.k40	2	1336,9	12	3	-	2	1336,9	0,00	2	1336,9	8,6	0,00
pr03.k40	2	1303,4	17	0	-	2	1303,4	0,00	2	1303,4	8,8	0,00
pr04.k40	2	1259,5	4	0	-	2	1259,5	0,00	2	1259,5	35,5	0,00
pr05.k40	2	1200,7	31	4	-	2	1200,7	0,00	-	-	1,2	-
pr06.k40	2	1242,9	0	4	-	2	1242,9	0,00	2	1242,9	1918,5	0,00
pr07.k40	2	1407,0	15	0	-	2	1407,0	0,00	2	1407,0	6,8	0,00
pr08.k40	2	1222,2	2	4	-	2	1222,2	0,00	2	1222,2	234,1	0,00
pr09.k40	2	1284,4	18	4	-	2	1284,4	0,00	2	1284,4	4,0	0,00
pr10.k40	2	1200,4	15	0	-	2	1200,4	0,00	2	1200,4	6,2	0,00
Soma/Média	31					32		-2,62	26		4333,1	-0,33

Capítulo 6

Experimentos Computacionais para o PCVMSH

Neste capítulo são apresentados os resultados obtidos para o PCVMSH utilizando a heurística proposta no Capítulo 4. O algoritmo foi codificado em C++, compilado com o compilador g++ (versão 7.4.0) usando a *tag* “-Ofast” e executado em uma máquina com processador Intel® Core™ i7 4770 3.4 GHz com 16 GB de memória RAM. O sistema operacional utilizado foi o Ubuntu Linux 18.04.

As instâncias disponíveis na literatura foram criadas por [6] no momento de proposição do PCVMSH. Todas as instâncias foram artificialmente criadas e são divididas em dois subgrupos: (i) clusterizadas, que possuem como característica o agrupamento dos clientes por regiões no espaço, e (ii) randomizadas, que não apresentam formas de agrupamento, sendo a escolha do local dos clientes totalmente aleatorizada. Na construção das instâncias, a localização dos clientes foi definida em um espaço de tamanho 400×400 . As instâncias geradas foram resolvidas para o PCVM e a inserção dos hotéis na rota realizada, de forma que a solução ótima para o PCVMSH seja conhecida e idêntica à solução ótima do PCVM.

Para geração das instâncias foram considerados três parâmetros: número de clientes, número de caixeiros e número máximo de viagens. Para cada combinação (três a três) de parâmetros apresentados na Tabela 6.1 e para cada um dos tipos de instâncias (clusterizadas e randomizadas), foram criadas 5 instâncias distintas totalizando 1440 instâncias.

De forma similar ao que foi realizado em trabalhos prévios [6, 41] e, buscando viabilizar a comparação dos resultados alcançados, o algoritmo MSRVND foi executado 10 vezes com números de sementes para geração dos números aleatórios definidos entre 1 e 10 para cada uma das instâncias. Os resultados apresentados consideram a média entre essas 10

Tabela 6.1: Parâmetros utilizados para gerar as instâncias em [6].

Parâmetro	Valores
Número de clientes (n)	50 - 75 - 100 - 150 - 200 - 300 - 400 - 500
Número de caixeiros (m)	2 - 3 - 4 - 5 - 6 - 7 - 8 - 9 - 10
Número máximo de viagens	5 - 10

execuções para contabilizar o tempo computacional demandado e as diferenças percentuais entre os valores referentes às soluções encontradas para o tempo total das rotas e o valor ótimo [6].

A heurística MSRVND utiliza o parâmetro α para a construção de soluções baseadas no construtor com LRC e o parâmetro número máximo de iterações ($iter_max$). Os parâmetros citados foram calibrados executando 5000 iterações no IRACE (*Iterated Racing for Automatic Algorithm Configuration*) [27], utilizando 16 instâncias representativas escolhidas de forma aleatória, considerando uma instância por subgrupo/tipo dentre aquelas disponibilizadas por [6]. Os valores testados para α compreendem o intervalo $[0,2, 0,8]$ e para $iter_max$ o intervalo $[100, 400]$. A melhor configuração retornada pelo IRACE foi $\alpha = 0,68$ e $iter_max = 353$.

A seguir são apresentadas as características dos subgrupos de instâncias e os resultados obtidos para cada um deles. Os resultados oriundos da aplicação da heurística MSRVND estão publicamente disponíveis em <https://bit.ly/3dIrDsD>. Para avaliação dos resultados obtidos, foi utilizada a métrica de ganho percentual médio (Equação (5.1)) entre a solução ótima (R_1) e o tempo total médio referente às 10 execuções da MSRVND (R_2). Essa métrica também foi utilizada para comparar os resultados encontrados pela heurística proposta por [6], entretanto, os autores fizeram uma inversão do cálculo contido no numerador da equação ($R_2 - R_1$). Tal inversão faz com que os resultados de *Gap* apresentados por [6] sejam positivos, enquanto nesta tese, eles são negativos ou no melhor caso iguais a zero (ótimo). Por não terem sido divulgados publicamente os resultados individuais para cada instância, os *Gaps* médios reportados por [6] foram multiplicados pela constante “-1” e apresentados na Tabela 6.3.

Nas Tabelas apresentadas a seguir, a primeira coluna apresenta a quantidade de clientes contidos em cada grupo de instâncias testadas, e entre a segunda e a décima coluna, apresenta-se a quantidade de caixeiros que cada instância possui.

Na última linha de cada tabela são apresentados os totais correspondentes à média dos valores agrupados pela quantidade de caixeiros disponíveis em cada subgrupo de instâncias, enquanto na última coluna são exibidas as médias dos valores observados em

relação a subgrupos de instâncias com o mesmo número de clientes.

6.1 Resultados para instâncias testadas em [6]

Nesta Seção é realizada a comparação dos resultados obtidos pela MSRVND com aqueles reportados por [6]. São consideradas apenas as instâncias com $n = \{50, 75, 100, 200, 300\}$.

Na Tabela 6.2 são apresentados os *Gaps* médios alcançados pela heurística MSRVND, obtidos por meio da comparação de seus resultados com o valor da solução ótima. Os resultados consideram tanto instâncias clusterizadas, quanto randomizadas. Verifica-se que as médias de *Gaps* alcançadas chegam a ser iguais a zero para instâncias com 2 caixeiros e 50 ou 200 clientes, como também para instâncias com 3 caixeiros e 50 clientes. À medida que o número de caixeiros e a quantidade de clientes vai aumentando, a dificuldade de alcance de soluções ótimas aumenta proporcionalmente, e isso reflete diretamente no valor do *Gap* médio. *Gaps* médios acima de -1 ficam restritos à grupos de instâncias que contém no máximo 5 caixeiros.

Os *Gaps* médios obtidos em [6] são apresentados na Tabela 6.3. Observa-se que a MSRVND consegue melhores resultados na maioria das vezes somente para grupos de instâncias contendo 2 ou 3 caixeiros. A qualidade das soluções apresentadas para a MSRVND apresenta certo balanceamento para instâncias com o mesmo número de caixeiros, enquanto na heurística desenvolvida em [6], esse balanceamento ocorre para instâncias com o mesmo número de clientes. No geral, os *Gaps* médios totais demonstram que a MSRVND ainda é por volta de três vezes menos eficiente do que a heurística proposta por [6].

Tabela 6.2: Média dos *Gaps* (%) encontrados pela MSRVND.

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	0,0	0,0	-0,3	-0,5	-1,2	-3,2	-2,3	-2,3	-3,4	-1,5
75	-0,1	-0,4	-0,7	-0,8	-1,3	-2,0	-1,9	-2,5	-2,3	-1,3
100	-0,1	-0,1	-0,5	-0,9	-2,0	-1,7	-2,0	-2,1	-2,9	-1,4
200	0,0	-0,7	-0,8	-1,0	-2,2	-2,9	-3,9	-3,3	-3,7	-2,1
300	-0,1	-0,3	-0,7	-1,5	-2,4	-3,7	-3,0	-2,6	-3,7	-2,0
Total	-0,1	-0,3	-0,6	-1,0	-1,8	-2,7	-2,6	-2,5	-3,2	-1,6

Tabela 6.3: Média dos *Gaps* (%) encontrados por [6].

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	0,0	-0,1	-0,3	-0,2	-0,1	-0,2	-0,4	-0,1	-0,2	-0,2
75	-0,2	-0,2	-0,2	-0,3	-0,2	-0,2	-0,2	-0,3	-0,3	-0,2
100	-0,6	-0,1	-0,2	-0,5	-0,4	-0,2	-0,4	-0,4	-0,4	-0,3
200	-0,7	-0,6	-0,5	-0,5	-0,8	-0,8	-0,8	-0,6	-0,9	-0,7
300	-1,2	-0,9	-1,0	-1,1	-1,4	-1,3	-1,1	-1,1	-1,1	-1,1
Total	-0,5	-0,4	-0,4	-0,5	-0,6	-0,5	-0,6	-0,5	-0,6	-0,5

Nas Tabelas 6.4 e 6.5 são apresentados os *Gaps* médios separados por tipo de instâncias (clusterizadas ou randomizadas). É possível verificar que para instâncias clusterizadas, a heurística MSRVND obtém melhores resultados, sendo identificados 7 subgrupos onde o *Gap* médio é igual a zero, correspondendo a resultados muito próximos ou iguais ao ótimo conhecido. Nas instâncias randomizadas, esse fator qualitativo é observado somente em 4 subgrupos. De forma geral, instâncias randomizadas tendem a ter valores para *Gaps* médios mais negativos. Esse comportamento é reforçado pela média final dos *Gaps* apresentados, sendo de -1,5 para instâncias clusterizadas e de -1,8 para randomizadas.

Tabela 6.4: Média dos *Gaps* (%) para instâncias clusterizadas.

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	0,0	0,0	0,0	-0,4	-1,1	-4,3	-2,2	-2,2	-4,1	-1,6
75	0,0	0,0	-0,1	-0,3	-1,6	-1,4	-1,0	-1,4	-1,9	-0,9
100	0,0	-0,2	-0,7	-0,4	-2,7	-1,4	-2,0	-2,0	-2,6	-1,3
200	0,0	-0,6	-0,4	-0,6	-2,4	-2,9	-3,7	-2,3	-2,8	-1,7
300	-0,2	-0,4	-1,0	-1,7	-3,0	-4,7	-2,6	-1,7	-4,1	-2,2
Total	-0,1	-0,3	-0,5	-0,7	-2,2	-2,9	-2,3	-1,9	-3,1	-1,5

Tabela 6.5: Média dos *Gaps* (%) para instâncias randomizadas.

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	0,0	0,0	-0,6	-0,7	-1,3	-2,2	-2,3	-2,3	-2,7	-1,3
75	-0,2	-0,7	-1,3	-1,3	-1,1	-2,6	-2,8	-3,6	-2,7	-1,8
100	-0,1	-0,1	-0,3	-1,4	-1,4	-1,9	-2,1	-2,1	-3,3	-1,4
200	0,0	-0,9	-1,2	-1,4	-2,1	-2,9	-4,1	-4,2	-4,6	-2,4
300	0,0	-0,1	-0,5	-1,3	-1,8	-2,7	-3,3	-3,6	-3,4	-1,9
Total	-0,1	-0,3	-0,8	-1,2	-1,5	-2,5	-2,9	-3,2	-3,3	-1,8

Os resultados apresentados nas Tabelas 6.4 e 6.5 são ilustrados na Figura 6.1. A Figura 6.1(a) apresenta a média dos *Gaps* agrupados pelo número de caixeiros e pelo tipo de instância. Quanto menor o número de caixeiros, mais fácil se torna a resolução

do problema e, conseqüentemente, o *Gap* médio reportado tende a ser mais próximo de zero. Quando os resultados são agrupados pelo número de caixeiros, o *Gap* médio tende a apresentar maior quantidade de valores negativos para as instâncias randomizadas, coincidindo com o resultado encontrado em [6]. Na Figura 6.1(b) são ilustrados os *Gaps* médios agrupados pelo número de clientes e tipo da instância. Nesta comparação, não é possível afirmar que sempre há uma tendência na forma como se comportam instâncias de diferentes tipos. Pode-se somente observar que instâncias com 75 e 300 clientes são as que apresentam maior correlação com os percentuais de soluções ótimas apresentados na Figura 6.2(b), ou seja, quanto mais próximos de 0,00 os *Gaps* estão, maior foi o percentual de soluções ótimas encontradas e vice-versa, para ambos os tipos de instâncias.

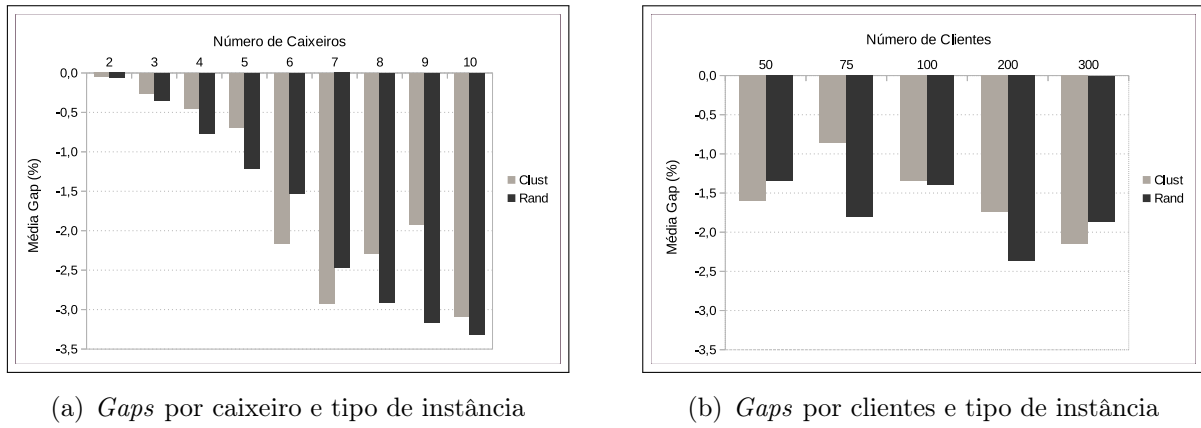


Figura 6.1: *Gaps* médios agrupados por indicadores distintos.

A capacidade de geração de soluções ótimas não é necessariamente o objetivo principal de técnicas heurísticas, entretanto, avaliar tal capacidade quando não foram desenvolvidas técnicas exatas para resolver o problema demonstra a robustez da heurística proposta. Uma vez que as instâncias foram criadas de forma a serem conhecidas soluções ótimas, a apresentação da porcentagem de soluções que alcançaram a solução ótima é mostrada na Tabela 6.6 para a heurística MSRVND e na Tabela 6.7 para a heurística de [6]. Seguindo a tendência de valores de *Gaps* médios maiores observados para [6], esperava-se que o número de soluções ótimas fosse maior, o que de fato foi confirmado por meio de um percentual geral de soluções ótimas equivalente a 52% das instâncias. Conforme apresentado na Tabela 6.6, a heurística MSRVND alcançou o percentual geral de valor ótimo para 35,6% das instâncias. Com base nesses resultados é possível afirmar que a heurística proposta por [6] alcança aproximadamente 46% a mais de soluções ótimas.

Tabela 6.6: Porcentagem de soluções ótimas encontradas pela MSRVND.

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	100,0	95,0	77,5	47,5	23,0	20,5	35,0	25,5	10,5	48,3
75	84,0	66,0	35,5	47,0	27,5	20,0	28,0	7,5	20,0	37,3
100	89,5	71,5	53,0	37,0	35,0	20,5	25,5	34,5	10,0	41,8
200	88,0	44,0	25,0	21,0	25,5	17,0	19,0	23,0	11,0	30,4
300	59,5	38,5	15,5	7,5	15,5	8,5	12,5	19,0	6,5	20,3
Total	84,2	63,0	41,3	32,0	25,3	17,3	24,0	21,9	11,6	35,6

Tabela 6.7: Porcentagem de soluções ótimas encontradas por [6].

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	96,5	94,5	85,0	80,0	87,0	86,0	71,0	86,5	77,0	84,8
75	79,0	87,0	68,0	77,0	85,0	81,0	85,5	74,0	69,5	78,4
100	73,0	69,5	68,0	51,0	53,0	65,5	61,0	58,5	48,5	60,9
200	29,5	28,5	14,5	30,5	7,5	19,5	24,5	33,5	19,5	23,1
300	21,0	13,5	11,5	24,5	6,0	16,5	12,5	8,5	1,5	12,8
Total	59,8	58,6	49,4	52,6	47,7	53,7	50,9	52,2	43,2	52,0

A Figura 6.2 ilustra a porcentagem de soluções ótimas encontradas de acordo com o número de caixeiros e número de clientes. É possível identificar um ponto interessante entre a Figura 6.1(a) e a Figura 6.2(a). Na primeira, nota-se que as instâncias randomizadas possuem *Gap* médio menor, quando comparadas às clusterizadas, o que em tese implicaria em soluções de pior qualidade. Entretanto, observa-se na segunda figura que este *Gap* médio menor não se reflete diretamente na porcentagem de soluções ótimas encontradas, a qual é superior na maioria das vezes para as instâncias randomizadas. Quando realizadas as comparações de percentuais com base no número de clientes, observa-se que os maiores percentuais de soluções ótimas para instâncias clusterizadas são alcançados quando há no máximo 75 clientes. As instâncias randomizadas por sua vez alcançam maiores percentuais de soluções ótimas para instâncias que possuem quantidade maior ou igual a 100 clientes.

As Tabelas 6.8 e 6.9 apresentam os tempos computacionais médios necessários para a execução das instâncias pela heurística MSRVND e pela heurística proposta por [6], respectivamente. De forma análoga ao formato de apresentação dos *Gaps* médios, nessas tabelas são apresentadas as médias do tempo computacional em segundos agrupadas por quantidade de caixeiros e número de clientes presentes nos subgrupos de instâncias.

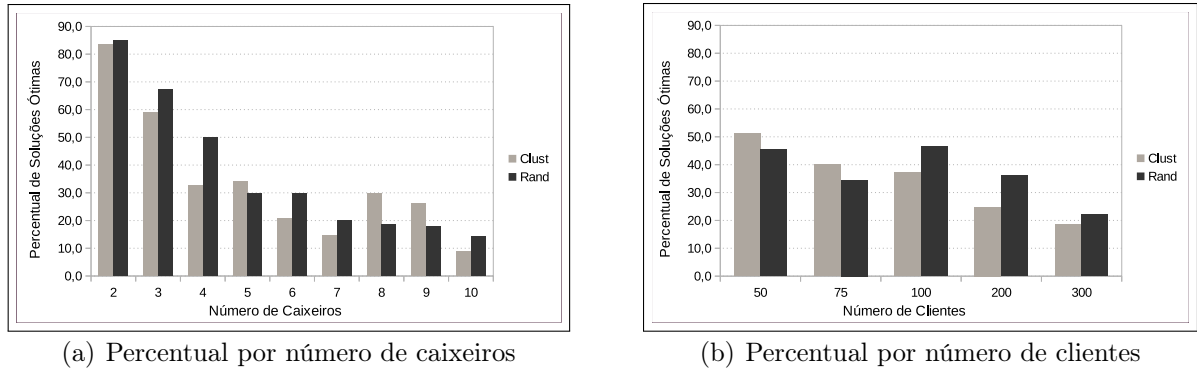


Figura 6.2: Percentual de soluções ótimas encontradas.

Tabela 6.8: Média de tempo computacional para a heurística MSRVND.

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	1,1	1,0	0,9	0,9	0,9	0,9	0,9	1,0	1,1	1,0
75	2,7	2,4	2,2	2,2	2,2	2,3	2,2	2,4	2,6	2,4
100	5,0	4,4	3,8	4,1	4,1	4,2	4,4	4,7	5,0	4,4
200	17,8	17,9	19,0	18,8	21,6	22,8	23,5	25,1	25,7	21,4
300	44,0	49,5	52,3	52,1	60,1	60,5	65,7	72,7	75,4	59,1
Total	14,1	15,0	15,6	15,6	17,8	18,1	19,3	21,2	22,0	17,6

Tabela 6.9: Média de tempo computacional para a heurística proposta por [6].

n	m									Média
	2	3	4	5	6	7	8	9	10	
50	0,8	0,9	1,2	1,1	1,2	1,3	1,2	1,1	1,2	1,1
75	1,8	1,9	1,9	2,0	2,0	2,0	2,1	2,3	2,2	2,0
100	2,8	2,8	3,3	3,5	3,6	3,4	3,5	3,8	3,9	3,4
200	14,4	14,0	13,0	12,9	13,3	13,3	12,6	12,3	13,1	13,2
300	33,4	33,4	30,5	32,4	31,1	33,7	30,3	29,8	31,4	31,8
Total	10,7	10,6	10,0	10,4	10,2	10,7	9,9	9,9	10,4	10,3

Em relação aos tempos computacionais demandados em [6], e levando em consideração que os tempos computacionais da MSRVND devem ser reescalados por uma taxa de 0,83 para termos uma comparação mais justa, verifica-se que o tempo computacional médio total utilizado pela heurística aqui proposta corresponde ao dobro do que foi utilizado na heurística de [6]. Os tempos computacionais médios chegam a ser no pior caso, quase três vezes maiores que aqueles reportados em [6] para as instâncias com $n = 300$ e $m = 10$.

Os resultados apresentados pela heurística MSRVND ainda são pouco competitivos quando comparados àqueles tidos como estado da arte. Isso reforça a necessidade de aprofundamento e refinamento das técnicas utilizadas, com o objetivo de alcançar resultados com *Gaps* médios mais próximos ou até melhores que aqueles reportados por [6]. Sugere-

tões de estudos mais aprofundados, bem como técnicas que possam ser utilizadas para melhorar os resultados aqui apresentados são citadas no próximo Capítulo como forma de trabalhos futuros.

6.2 Resultados para instâncias não testadas em [6]

Apesar dos autores em [6] terem criado instâncias para os diferentes números de clientes apresentados na Tabela 6.1, aquelas com $n = \{150, 400, 500\}$ não foram incluídas em seus testes. Os resultados apresentados a seguir consideram somente as instâncias que não possuem resultados disponíveis na literatura.

A Tabela 6.10 contém a média de *Gaps* produzidos pela heurística MSRVND em relação às soluções ótimas. Os resultados demonstram que o maior valor de *Gap* médio reportado é igual a 0,0 para o subgrupo de instâncias com $n = 50$ e $m = 2$, enquanto o menor corresponde a um *Gap* médio de -7,3 para $n = 400$ e $m = 10$.

As Tabelas 6.11 e 6.12 mostram os valores de *Gaps* médios separadamente em relação ao tipo das instâncias, randomizadas e clusterizadas. Observa-se que o menor valor (-10,8) é proveniente do subgrupo de instâncias clusterizadas. Nota-se também uma inversão no comportamento do *Gap* médio em relação àquele apresentado na Seção 6.1, onde os menores valores de *Gaps* eram reportados para instâncias do tipo randomizadas.

Tabela 6.10: Média dos *Gaps* (%) encontrados pela MSRVND

n	m									Média
	2	3	4	5	6	7	8	9	10	
150	0,0	-0,2	-0,4	-0,6	-1,6	-1,2	-2,6	-0,9	-2,3	-1,1
400	-0,3	-0,7	-1,3	-1,2	-2,7	-2,4	-4,2	-3,4	-7,3	-2,6
500	-0,5	-1,0	-1,1	-1,3	-2,1	-2,8	-2,8	-2,5	-3,2	-1,9
Total	-0,3	-0,6	-0,9	-1,0	-2,1	-2,1	-3,2	-2,2	-4,3	-1,9

Tabela 6.11: Média dos *Gaps* (%) para instâncias clusterizadas.

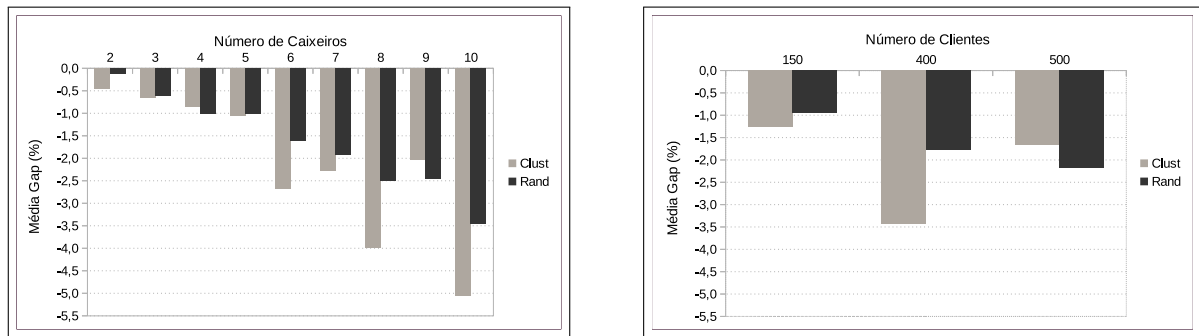
n	m									Média
	2	3	4	5	6	7	8	9	10	
150	0,0	-0,1	-0,4	-0,9	-2,4	-1,5	-3,4	-0,7	-1,9	-1,3
400	-0,6	-0,6	-1,3	-1,1	-3,5	-2,6	-5,8	-4,5	-10,8	-3,4
500	-0,8	-1,2	-0,8	-1,2	-2,1	-2,8	-2,8	-0,9	-2,5	-1,7
Total	-0,5	-0,6	-0,9	-1,0	-2,7	-2,3	-4,0	-2,0	-5,1	-2,1

Tabela 6.12: Média dos *Gaps* (%) para instâncias randomizadas.

n	m									Média
	2	3	4	5	6	7	8	9	10	
150	0,0	-0,3	-0,4	-0,2	-0,8	-0,9	-1,9	-1,1	-2,8	-0,9
400	-0,1	-0,7	-1,2	-1,3	-2,0	-2,1	-2,7	-2,2	-3,8	-1,8
500	-0,3	-0,8	-1,4	-1,4	-2,1	-2,7	-2,9	-4,1	-3,8	-2,2
Total	-0,1	-0,6	-1,0	-1,0	-1,6	-1,9	-2,5	-2,4	-3,5	-1,6

Examinando mais detalhadamente o subgrupo de instâncias clusterizadas com $n = 400$ e $m = 10$, foi identificada a instância *clust_401_10_2_5* como a responsável pela discrepância acentuada do *Gap* médio. Essa instância chega a apresentar valores de *Gaps* médios próximos a -60, enquanto seus pares em outros subgrupos, chegaram a ter nas piores avaliações valores próximos a -20. Para confirmar que não se tratava de um erro algorítmico, testes com valores aleatórios para os parâmetros da MSRVND foram conduzidos. Para esses testes, valores de *Gap* fortemente negativos também foram observados, concluindo assim que essa dificuldade ocorre devido às características particulares desta instância.

A Figura 6.3 ilustra o comportamento já apresentado previamente nas Tabela 6.11 e 6.12. O *Gap* médio cresce proporcionalmente ao número de caixeiros nas instâncias, e em grande parte dos subgrupos avaliados, a tendência é de valores percentuais menores para instâncias clusterizadas.

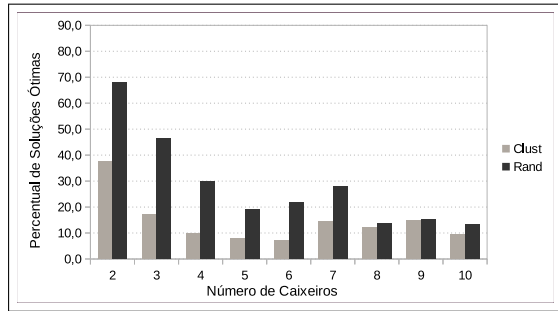
(a) *Gaps* por caixeiro e tipo de instância(b) *Gaps* por clientes e tipo de instânciaFigura 6.3: *Gaps* médios agrupados por indicadores distintos.

Os percentuais de soluções ótimas obtidos pela heurística MSRVND são apresentados de forma agrupada na Tabela 6.13 e desmembrados na Figura 6.4. É possível verificar a queda do percentual médio total de 35,6% para 21,4% em relação ao apresentado para os subgrupos de instâncias listados na Seção 6.1. Essa queda acentuada ocorreu devido ao aumento do número de clientes por instância. Observa-se ainda que existe uma diferença

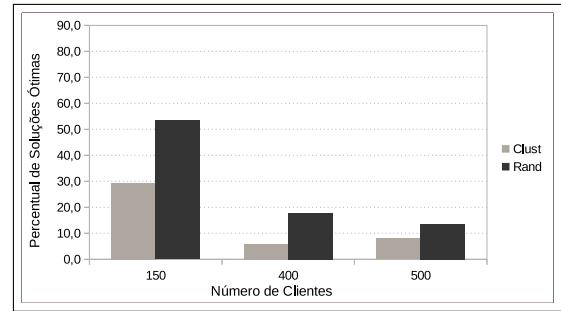
significativa entre a porcentagem de soluções ótimas retornadas entre as instâncias clusterizadas e randomizadas, sendo os maiores percentuais alcançados para instâncias deste último tipo. Na Figura 6.4(a) o percentual de soluções ótimas é comparado com relação ao número de caixeiros, enquanto na Figura 6.4(b) a comparação ocorre em relação ao número de clientes por instância. Essas comparações deixam claro que grande parte das soluções ótimas são provenientes dos subgrupos de instâncias com 150 clientes, e que, caso fossem executadas somente as instâncias com 400 e 500 clientes, o percentual médio total de soluções ótimas seria limitado à 11,3%.

Tabela 6.13: Porcentagem de soluções ótimas encontradas pela MSRVND.

n	m									Média
	2	3	4	5	6	7	8	9	10	
150	95,5	54,5	41,5	33,0	34,5	48,5	27,5	28,0	10,0	41,4
400	37,5	30,5	15,5	2,5	5,0	5,5	2,0	5,0	3,0	11,8
500	25,5	10,0	2,0	5,0	3,5	9,5	9,0	12,0	21,0	10,8
Total	52,8	31,7	19,7	13,5	14,3	21,2	12,8	15,0	11,3	21,4



(a) Percentual por número de caixeiros



(b) Percentual por número de clientes

Figura 6.4: Percentual de soluções ótimas encontradas.

Finalmente, a Tabela 6.14 contém os tempos computacionais médios requeridos pela heurística MSRVND. Os tempos computacionais médios necessários para execução dos subgrupos variam de 10,8 segundos no mínimo e de 196,2 segundos no máximo. O tempo computacional possui comportamento similar ao observado para o *Gap* médio, pois aumenta proporcionalmente em relação ao número de caixeiros e de clientes por instância.

Tabela 6.14: Média de tempo computacional para a heurística MSRVND (em segundos).

n	m									Média
	2	3	4	5	6	7	8	9	10	
150	9,9	10,5	9,4	10,1	10,1	10,8	11,2	12,4	12,6	10,8
400	84,1	87,9	93,1	104,4	114,2	119,7	123,4	138,9	151,1	113,0
500	148,0	142,3	161,4	174,9	191,5	200,8	226,4	253,9	266,3	196,2
Total	80,7	80,2	88,0	96,5	105,3	110,4	120,3	135,1	143,3	106,6

Capítulo 7

Conclusões e Trabalhos Futuros

Nesta tese foram estudados o PCVSH e o PCVMSH, que são variantes do clássico PCV. Ambos tem como principal particularidade a utilização de hotéis na composição de rotas viáveis para execução por um ou mais caixeiros viajantes. Na primeira variante, apenas um caixeiro está disponível e o objetivo é encontrar uma rota que minimize, em ordem de prioridade, o número de viagens e o tempo total gasto para atender todos os clientes. A segunda variante conta sempre com mais de um caixeiro para executar o trabalho e seu objetivo consiste basicamente na minimização do tempo total gasto para atender todos os clientes, respeitando sempre um número máximo de viagens estipulado para cada caixeiro.

Para o PCVSH, uma prova formal da complexidade do problema foi apresentada e uma heurística híbrida (MILS) foi proposta tendo como pilares o *framework* ILS e um procedimento de mineração de dados baseado na técnica FIM. A heurística MILS inicialmente utiliza a heurística de *Lin Kernighan* para criar uma solução de qualidade para o clássico PCV, aplicando em seguida, um procedimento baseado no Algoritmo de Dijkstra para transformar a rota previamente criada em uma rota/solução viável para o PCVSH. Iterativamente, no *framework* ILS é escolhida uma solução vizinha ou o procedimento de perturbação variável é aplicado. Para cada iteração, a solução corrente é otimizada utilizando um procedimento RVND composto por seis estruturas de vizinhança. Durante a execução, sempre que um determinado número de iterações é alcançado, com base no conjunto de padrões minerados, uma nova solução para o PCVSH é construída. Com base na qualidade das soluções geradas ao longo das iterações, é realizado um ajuste de intensidade no nível de perturbação.

Por serem utilizadas várias estruturas de vizinhança no RVND e pelo fato da perturbação variável ser incomum na literatura, foi conduzida uma análise do impacto para determinar a efetividade dessas técnicas para o alcance de soluções de qualidade. Foram

realizados testes com perturbação fixa e variável e os resultados indicaram que a utilização de perturbação variável proporcional ao número de clientes retorna melhores resultados. A contribuição individual de cada estrutura de vizinhança foi avaliada realizando testes computacionais com a heurística ILS-RVND usando perturbação variável. Todas as 131 instâncias propostas por [8] foram utilizadas e os testes foram realizados desabilitando uma estrutura de vizinhança por vez. Os resultados demonstraram que no geral as melhores soluções são obtidas por meio do RVND composto por todas as estruturas de vizinhança.

A heurística MILS foi executada sobre as instâncias propostas por [46] e os resultados obtidos foram comparados com as melhores soluções conhecidas e publicadas na literatura [4, 8, 9, 26]. Os experimentos demonstraram que a heurística é capaz de melhorar significativamente o número de melhores soluções conhecidas, usando tempos computacionais equivalentes aos de trabalhos que também propõem soluções heurísticas. A heurística proposta alcançou para 100 instâncias (76,34%) o mesmo tempo total da rota que os algoritmos considerados estados da arte obtiveram e, para outras 18 instâncias (13,74%), gerou soluções de melhor qualidade. Para 13 instâncias (9,92%) a qualidade foi inferior.

Para 47 instâncias onde a solução ótima do PCVSH é conhecida, a MILS encontrou o valor ótimo para todas as instâncias. Considerando os três subgrupos que contém as instâncias mais difíceis, a heurística proposta foi capaz de reduzir o número de viagens em no mínimo uma unidade quando comparado ao HDM [26] e no máximo três unidades comparado ao MA [8].

Foi proposta outra estratégia de hibridização combinando uma solução heurística obtida por uma versão simplificada do MILS e um modelo de Programação Linear Inteira. A resolução do modelo considerou a solução heurística como solução de partida e seu tempo da rota como *upperbound*. O modelo foi resolvido de forma relaxada e inteira. A estratégia utilizada foi baseada na criação de restrições para o modelo de programação inteira a partir da resolução do modelo relaxado e arestas mineradas. Devido ao grande número de clientes contidos nas instâncias, a resolução de forma inteira só alcançou soluções de qualidade iguais às aquelas inseridas como solução de partida. Em alguns casos, a solução da heurística não pôde ser utilizada como solução inicial para o modelo de programação inteira, porque era inviável em relação às restrições criadas. Nesses casos, não foi possível obter uma solução inteira.

Outro conjunto totalizando 240 instâncias proposto por [4] foi testado. Para essas instâncias, somente resultados obtidos por meio de abordagem exata estão disponíveis na literatura. A heurística MILS estabeleceu um total de 112 novas melhores soluções.

Considerando somente instâncias para as quais podem ser encontradas soluções viáveis, este número de melhores soluções alcançadas corresponde a 50,68% das instâncias. Para outras 49 instâncias (22,17%) foram encontradas soluções idênticas às reportadas por [4]. Para as demais 60 instâncias (27,15%), onde o Algoritmo exato BCP supera a heurística MILS, o valor mínimo do *Gap* foi de -2,44%. Entretanto, o tempo computacional gasto pelo BCP no pior caso foi de 6955 segundos, enquanto na heurística MILS foi de somente 20 segundos. Aumentar o tempo computacional do MILS pode reduzir o número de soluções que foram superadas pelo BCP.

Como forma de avaliar a contribuição do procedimento de mineração para o desempenho da heurística proposta, foram executados testes com todas as instâncias disponíveis na literatura, habilitando e desabilitando o processo de mineração de padrões frequentes. Observou-se que em 14 das 131 instâncias propostas por [46] e 18 das 240 propostas por [4], as melhores soluções reportadas pelo MILS não teriam sido encontradas sem o uso do procedimento de mineração. Portanto, esse experimento confirma que o procedimento de mineração contribuiu positivamente para o desempenho geral da heurística MILS.

Foi conduzido um teste para verificar se existia diferença significativa entre os resultados reportados nas melhores soluções da literatura e a MILS para ambos os conjuntos de instâncias [4, 46]. Verificamos por meio do teste de *Shapiro* que os resultados comparados não seguem uma distribuição normal. Após a aplicação do teste de *Mann-Whitney*, concluímos que apesar da relevante quantidade de melhores soluções encontradas pela heurística MILS nos dois conjuntos de instâncias, não houve diferença estatisticamente significativa em relação às soluções, sendo reportado no teste um *p-value* = 0,8604.

Para o PCVMSH foi proposta uma heurística *Multi-Start* com busca local (MSRVND). A heurística utiliza quatro construtores distintos para geração de soluções diversificadas, sendo um com característica determinística e outros 3 estocásticos. As soluções construídas correspondem a soluções para o clássico PCVM, as quais são decompostas em viagens viáveis para o PCVMSH. O procedimento constrói uma solução determinística, a qual é otimizada por meio de 8 estruturas de vizinhanças contidas na busca local baseada em RVND. As estruturas de vizinhança atuam internamente a uma viagem, entre viagens de uma mesma rota e entre rotas. Em seguida, iterativamente são geradas novas soluções por meio dos construtores estocásticos, as quais são otimizadas pelo RVND.

Os testes foram conduzidos sobre um conjunto de 1440 instâncias propostas por [6]. A heurística MSRVND obteve resultados com solução igual ou pior àqueles contidos na literatura, bem como utiliza uma maior quantidade de tempo computacional para obten-

ção de suas soluções. Os testes e comparações indicaram que a heurística estado da arte é calibrada para encontrar uma concentração de soluções ótimas de forma mais homogênea dentro de conjuntos de instâncias com mesmo número de clientes, enquanto na heurística MSRVND, essa concentração de soluções ótimas ocorre de forma mais homogênea para conjuntos de instâncias com mesmo número de caixeiros. Isso demonstra que as estruturas de vizinhança contidas na heurística proposta carecem um maior refinamento para conseguir entregar uma melhor otimização para soluções com maior número de caixeiros. Por fim, novos resultados heurísticos foram estabelecidos para instâncias que haviam sido propostas mas não resolvidas por [6].

7.1 Publicações

As publicações listadas a seguir possuem relação direta com o desenvolvimento desta tese:

1. *A Variable Neighborhood Search Heuristic for the Traveling Salesman Problem with Hotel Selection* [40];
2. Uma Heurística *Multi-Start Local Search* para o Problema do Caixeiro Viajante Múltiplo com Seleção de Hotéis [41];
3. *An Efficient Heuristic to the Traveling Salesperson Problem with Hotel Selection* [42];
4. *Traveling Salesperson Problem with Hotel Selection: a systematic review of the literature* [43];
5. *A Hybrid Iterated Local Search Heuristic for the Traveling Salesperson Problem with Hotel Selection* (submetido para *Computers & Operations Research Journal*).

7.2 Trabalhos futuros

Trabalhos futuros para o PCVSH e PCVMSH podem ser conduzidos em diferentes vertentes. Uma possibilidade é a criação de procedimentos distintos para construção de soluções baseadas em mineração de itens frequentes [20] e posterior divisão de rotas com uso de programação dinâmica [26]. Nessa proposta, novas formas de combinar padrões produzidos pela técnica FIM podem ser utilizadas em conjunto com estratégias de cruzamento e mutação comumente empregadas em algoritmo genético. As rotas geradas podem ser particionadas de forma mais eficiente com o uso de programação dinâmica.

Técnicas para redução do espaço de busca utilizando padrões minerados foram utilizadas em [29], provendo soluções de boa qualidade em menor tempo computacional. Utilizando essa abordagem, arestas obtidas por meio de padrões minerados que possam ser agrupadas seriam contraídas de forma a representar um único vértice. Com isso, a otimização a ser feita no procedimento de busca local demandaria menor tempo computacional, possibilitando que outras áreas do espaço de busca sejam exploradas.

Outra ideia é investigar outros mecanismos de hibridização combinando métodos heurísticos e exatos como em [17, 18]. Um problema poderia ser dividido em pequenos subproblemas que pudessem ser resolvidos individualmente e de forma ótima por métodos exatos. Posteriormente, as soluções ótimas obtidas para os subproblemas seriam recombinadas e re-otimizadas por procedimentos heurísticos.

Por fim, em nível de implementação, fazer uso de estruturas de dados de mais baixo nível e buscar executar estruturas de vizinhança utilizando tempos computacionais constantes. Essas estruturas de dados armazenariam informações sobre movimentos avaliados e aplicados, evitando retrabalho para aplicações de movimentos futuros.

Todas estas sugestões podem ser empregadas tanto individualmente, quanto agrupadas.

Referências

- [1] AGRAWAL, R.; SRIKANT, R. Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (1994), vol. 1215, pp. 487–499.
- [2] APPLEGATE, D. L.; BIXBY, R. E.; CHVATAL, V.; COOK, W. J. The traveling salesman problem: a computational study. Princeton University Press, 2006.
- [3] BALTZ, A.; OUALI, M. E.; JAGER, G.; SAUERLAND, V.; SRIVASTAV, A. Exact and heuristic algorithms for the travelling salesman problem with multiple time windows and hotel selection. Journal of the Operational Research Society 66 (2014), 615 – 626.
- [4] BARBOSA, L. H.; UCHOA, E. A branch-cut-and-price algorithm for the traveling salesperson problem with hotel selection. Computers & Operations Research 123 (2020), 1049 – 1086.
- [5] BEKTAS, T. The multiple traveling salesman problem: an overview of formulations and solution procedures. Omega 34, 3 (2006), 209 – 219.
- [6] CASTRO, M.; SORENSEN, K.; GOOS, P.; VANSTEENWEGEN, P. The multiple travelling salesperson problem with hotel selection. Tech. rep., University of Antwerp, Faculty of Applied Economics, BE, dec 2014.
- [7] CASTRO, M.; SÖRENSEN, K.; VANSTEENWEGEN, P.; GOOS, P. A simple grasp+vnd for the travelling salesperson problem with hotel selection. Tech. rep., University of Antwerp, Faculty of Applied Economics, BE, out 2012.
- [8] CASTRO, M.; SÖRENSEN, K.; VANSTEENWEGEN, P.; GOOS, P. A memetic algorithm for the travelling salesperson problem with hotel selection. Computers & Operations Research 40, 7 (2013), 1716 – 1728.
- [9] CASTRO, M.; SÖRENSEN, K.; VANSTEENWEGEN, P.; GOOS, P. A fast metaheuristic for the travelling salesperson problem with hotel selection. 4OR (2014), 1 – 20.
- [10] CHEN, Y.; HAO, J.-K.; GLOVER, F. A hybrid metaheuristic approach for the capacitated arc routing problem. European Journal of Operational Research 253, 1 (2016), 25–39.
- [11] CHRISTOFIDES, N.; MINGOZZI, A.; TOTH, P. The vehicle routing problem. Combinatorial optimization (1979), 315 – 338.
- [12] COOK, W. Concorde tsp solver. <http://www.math.uwaterloo.ca/tsp/index.html>. Acessado em: 20-04-2020.

- [13] CORDEAU, J. F.; LAPORTE, G.; MERCIER, A. A unified tabu search heuristic for vehicle routing problems with time windows. Journal of the Operational research society 52, 8 (2001), 928–936.
- [14] CROES, G. A. A method for solving traveling-salesman problems. Operations Research 6, 6 (1958), 791–812.
- [15] DIJKSTRA, E. W. A note on two problems in connexion with graphs. Numerische mathematik 1, 1 (1959), 269–271.
- [16] FREITAS, J. C.; PENNA, P. H. V. A variable neighborhood search for flying sidekick traveling salesman problem. International Transactions in Operational Research 27 (2018), 267–290.
- [17] GONZALEZ, P. H.; MACAMBIRA, A. F. U. S.; PINTO, R. V.; SIMONETTI, L.; MACULAN, M.; MICHELON, P. New proposals for modelling and solving the problem of covering solids using spheres of different radii. RAIRO-Operations Research (2019).
- [18] GONZALEZ, P. H.; SIMONETTI, L.; MICHELON, P.; MARTINHON, C.; SANTOS, E. A variable fixing heuristic with local branching for the fixed charge uncapacitated network design problem with user-optimal flow. Computers & Operations Research 76 (2016), 134–146.
- [19] GRAHNE, G.; ZHU, J. Efficiently using prefix-trees in mining frequent itemsets. In Frequent Itemset Mining Implementations (FIMI) (2003), vol. 90, pp. 123–132. ICDM 2003 Workshop on Frequent Itemset Mining Implementations.
- [20] GUERINE, M.; ROSSETI, I.; PLASTINO, A. Extending the hybridization of metaheuristics with data mining: Dealing with sequences. Intelligent Data Analysis 20, 5 (2016), 1133–1156.
- [21] HAN, J.; PEI, J.; YIN, Y. Mining frequent patterns without candidate generation. In ACM sigmod record (2000), vol. 29, ACM, pp. 1–12.
- [22] LEISERSON, C. E.; RIVEST, R. L.; CORMEN, T. H.; STEIN, C. Introduction to algorithms, vol. 6. MIT press Cambridge, MA, 2001.
- [23] LIN, S. Computer solutions of the traveling salesman problem. Bell System Technical Journal 44, 10 (1965), 2245 – 2269.
- [24] LIN, S.; KERNIGHAN, B. W. An effective heuristic algorithm for the traveling-salesman problem. Operations research 21, 2 (1973), 498 – 516.
- [25] LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In Handbook of metaheuristics. Springer, 2019, pp. 129–168.
- [26] LU, Y.; BENLIC, U.; WU, Q. A hybrid dynamic programming and memetic algorithm to the traveling salesman problem with hotel selection. Computers & Operations Research 90 (2018), 193 – 207.
- [27] LÓPEZ-IBÁÑEZ, M.; DUBOIS-LACOSTE, J.; PÉREZ CÁCERES, L.; STÜTZLE, T.; BIRATTARI, M. The irace package: Iterated racing for automatic algorithm configuration. Operations Research Perspectives 3 (2016), 43–58.

- [28] MAIA, M. R. H. Heurísticas híbridas com mineração de dados para o problema de roteamento de veículos com frota heterogênea. Dissertação de mestrado, Universidade Federal Fluminense, Niterói, Rio de Janeiro, nov 2015.
- [29] MAIA, M. R. H.; PLASTINO, A.; PENNA, P. H. V. Hybrid data mining heuristics for the heterogeneous fleet vehicle routing problem. RAIRO-Operations Research 52 (2018), 661–690.
- [30] MARTINS, I. C.; PINHEIRO, R. G.; PROTTI, F.; OCHI, L. S. A hybrid iterated local search and variable neighborhood descent heuristic applied to the cell formation problem. Expert Systems with Applications 42, 22 (2015), 8947 – 8955.
- [31] MLADENović, N.; HANSEN, P. Variable neighborhood search. Computers & Operations Research 24, 11 (1997), 1097 – 1100.
- [32] OR, I. Traveling salesman-type combinatorial problems and their relation to the logistics of regional blood banking. Tese de Doutorado, Northwestern University, Evanston, Illinois, June 1976.
- [33] PENNA, P. H. V.; SUBRAMANIAN, A.; OCHI, L. S. An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. Journal of Heuristics 19, 2 (2013), 201 – 232.
- [34] PRINS, C. A simple and effective evolutionary algorithm for the vehicle routing problem. Computers & Operations Research 31, 12 (2004), 1985 – 2002.
- [35] RADMANESH, M.; KUMAR, M.; NEMATI, A.; SARIM, M. Solution of traveling salesman problem with hotel selection in the framework of milp-tropical optimization. In 2016 American Control Conference (ACC) (2016), IEEE, pp. 5593 – 5598.
- [36] RIBEIRO, M. H.; PLASTINO, A.; M., S. L. Hybridization of grasp metaheuristic with data mining techniques. Journal of Mathematical Modelling and Algorithms 5, 1 (2006), 23–41.
- [37] SANTOS, L. F.; MARTINS, S. L.; PLASTINO, A. Applications of the dm-grasp heuristic: a survey. International Transactions in Operational Research 15, 4 (2008), 387–416.
- [38] SOLOMON, M. M. Algorithms for the vehicle routing and scheduling problems with time window constraints. Operations research 35, 2 (1987), 254–265.
- [39] SOUSA, M. M.; GONÇALVES, L. B. Comparação de abordagens heurísticas baseadas em algoritmo memético para o problema do caixeiro viajante com seleção de hotéis. In Proc. XLVI Simpósio Brasileiro de Pesquisa Operacional (Salvador, Brasil, set 2014), pp. 1543–1554.
- [40] SOUSA, M. M.; OCHI, L. S.; COELHO, I. M.; GONCALVES, L. B. A variable neighborhood search heuristic for the traveling salesman problem with hotel selection. In Computing Conference (CLEI), 2015 Latin American (2015), IEEE, pp. 1 – 12.

- [41] SOUSA, M. M.; OCHI, L. S.; DE LIMA MARTINS, S. Uma heurística multi-start local search para o problema do caixeiro viajante múltiplo com seleção de hotéis. In Proc. XLVIII Simpósio Brasileiro de Pesquisa Operacional (Vitória, Brasil, set 2016), pp. 1637 – 1648.
- [42] SOUSA, M. M.; OCHI, L. S.; DE LIMA MARTINS, S. An efficient heuristic to the traveling salesperson problem with hotel selection. In International Workshop on Hybrid Metaheuristics (2019), Springer, pp. 31 – 45.
- [43] SOUSA, M. M.; OCHI, L. S.; DE LIMA MARTINS, S. Traveling salesperson problem with hotel selection: a systematic review of the literature. In Simpósio Brasileiro de Sistemas de Informação (2019), ACM.
- [44] SOUZA, M. J.; COELHO, I. M.; RIBAS, S.; SANTOS, H. G.; MERSCHMANN, L. H. D. C. A hybrid heuristic algorithm for the open-pit-mining operational planning problem. European Journal of Operational Research 207, 2 (2010), 1041–1051.
- [45] SUBRAMANIAN, A.; UCHOA, E.; OCHI, L. S. A hybrid algorithm for a class of vehicle routing problems. Computers & Operations Research 40, 10 (2013), 2519 – 2531.
- [46] VANSTEENWEGEN, P.; SOUFFRIAUX, W.; SORENSEN, K. The travelling salesperson problem with hotel selection. Journal of the Operational Research Society 63, 2 (2012), 207 – 217.