

UNIVERSIDADE FEDERAL DE SANTA MARIA
CENTRO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

Guilherme Dhein

**PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM DEPENDÊNCIA
TEMPORAL E ESPACIAL ENTRE ROTAS DE EQUIPES DE CAMPO**

Santa Maria, RS
2016

Guilherme Dhein

**PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM DEPENDÊNCIA TEMPORAL E
ESPACIAL ENTRE ROTAS DE EQUIPES DE CAMPO**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração em Sistemas Elétricos de Potência, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Doutor em Engenharia Elétrica.**

ORIENTADOR: Prof. Ghendy Cardoso Júnior

COORIENTADOR: Prof. Olinto César Bassi de Araújo

Santa Maria, RS
2016

Ficha catalográfica elaborada através do Programa de Geração Automática da Biblioteca Central da UFSM, com os dados fornecidos pelo(a) autor(a).

Dhein, Guilherme
Problemas de roteamento de veículos com dependência temporal e espacial entre rotas de equipes de campo / Guilherme Dhein.- 2016.
147 p.; 30 cm

Orientador: Ghendy Cardoso Jr.
Coorientador: Olinto César Bassi de Araújo
Tese (doutorado) - Universidade Federal de Santa Maria, Centro de Tecnologia, Programa de Pós-Graduação em Engenharia Elétrica, RS, 2016

1. Problemas de Roteamento 2. Métrica de Dispersão 3. Algoritmo Genético 4. Algoritmo Genético com Busca Local Multiobjetivo 5. Rotas Sincronizadas I. Cardoso Jr., Ghendy II. Araújo, Olinto César Bassi de III. Título.

©2016

Todos os direitos autorais reservados a Guilherme Dhein. A reprodução de partes ou do todo deste trabalho só poderá ser feita mediante a citação da fonte.

End. Eletr.: gdhein@redes.ufsm.br

Guilherme Dhein

**PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM DEPENDÊNCIA TEMPORAL E
ESPACIAL ENTRE ROTAS DE EQUIPES DE CAMPO**

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, Área de Concentração em Sistemas Elétricos de Potência, da Universidade Federal de Santa Maria (UFSM, RS), como requisito parcial para obtenção do grau de **Doutor em Engenharia Elétrica.**

Aprovado em 26 de agosto de 2016:

Ghendy Cardoso Júnior, Dr. (UFSM)
(Presidente/Orientador)

Luciana Salete Buriol, Dra. (UFRGS - por videoconferência)

Christiano Lyra Filho, Dr. (UNICAMP - por videoconferência)

José Vicente Canto dos Santos, Dr. (UNISINOS)

Felipe Martins Müller, Dr. (UFSM)

Santa Maria, RS
2016

DEDICATÓRIA

Para Alice.

AGRADECIMENTOS

Este trabalho é o ponto culminante do curso de doutorado, e o momento provoca a reflexão sobre as pessoas que de alguma forma estiveram presentes na jornada. Algumas são tão importantes que se colocam como fundamentais para que o trabalho tenha sido realizado. A elas, o meu reconhecimento pela contribuição vai na forma de agradecimento.

Agradeço ao professor Ghendy, meu orientador, pela oportunidade, pelo apoio ao trabalho e pela confiança. Não acredito que existam muitos orientadores que permitam a liberdade na qual essa tese foi gerada.

Agradeço enormemente ao professor Olinto, meu coorientador, colega de trabalho, amigo, e grande parceiro. O seu trabalho incansável gerou para mim uma dívida enorme (e, já aviso, impagável).

Agradeço à Alice pelo amor incondicional dedicado ao pai, pela alegria que me traz com sua simples presença, e pela ajuda com seus lindos desenhos de veículos e rotas. E também por, mesmo na sua inocência, compreender as impossibilidades que as condições concomitantes de professor e aluno me impuseram. Está chegando a hora de brincarmos muito, filha linda!

Agradeço à Vilma, minha esposa, pelo amor e pelo companheirismo, e por tocar nossa vida nesse período em que, mesmo presente, estive bastante ausente.

Agradeço aos meus pais, Erica e Rivaldo, por... tudo. Acredito que a vida adulta é resultado das oportunidades que se tem na infância, e nesse sentido devo tudo isso a vocês. Tudo começou no lar cheio de amor e carinho que vocês ofereceram para os filhos e na educação de qualidade que vocês nos possibilitaram.

Agradeço aos meus irmãos, Gustavo e Daniela, pelo amor e pelo apoio. Aquela revisão no texto, aquele tempo dedicado a cuidar da Alice nas nossas impossibilidades, aquele abraço de incentivo, tudo isso representa um pedacinho desse todo.

Por fim, agradeço ao Aécio, o amigo que levo dos colegas de PPGEE, pela parceria importante em disciplinas cursadas, em dúvidas sobre o curso, naquele artigo (!) e até mesmo nos momentos de falar sobre Grêmio e Corinthians.

RESUMO

PROBLEMAS DE ROTEAMENTO DE VEÍCULOS COM DEPENDÊNCIA TEMPORAL E ESPACIAL ENTRE ROTAS DE EQUIPES DE CAMPO

AUTOR: Guilherme Dhein

ORIENTADOR: Ghendy Cardoso Júnior

COORIENTADOR: Olinto César Bassi de Araújo

Esta tese apresenta dois novos problemas de roteamento, ambos com funções objetivo voltadas para o posicionamento relativo das equipes durante o horizonte de roteamento. O posicionamento relativo resulta em uma dependência temporal e espacial entre rotas e é quantificado com uma métrica de dispersão não-linear, projetada para avaliar as distâncias instantâneas entre as equipes ao longo de um intervalo de tempo. Esta métrica permite a concepção de funções objetivo para aproximar as equipes durante a execução das rotas, quando minimizada, ou para dispersá-las, quando maximizada. Tanto a aproximação quanto a dispersão são características importantes de roteamento em algumas aplicações práticas, e dois novos problemas de otimização são propostos com esses objetivos opostos. O primeiro é uma variação do Problema de Múltiplos Caixeiros Viajantes, e seu objetivo é encontrar um conjunto de rotas em que os caixeiros viajam próximos uns dos outros, minimizando a dispersão. Um Algoritmo Genético com Busca Local é proposto para resolver o problema. Ele inclui operadores genéticos e vizinhanças especializados. Um novo conjunto de instâncias é proposto, adaptado para o novo problema de instâncias da literatura. Resultados computacionais mostram que a abordagem proposta proporciona soluções com as características desejadas de dispersão mínima. O segundo problema é um problema de roteamento de arcos biobjetivo em que as rotas devem ser construídas de modo a maximizar o lucro recolhido e o distanciamento entre as equipes. A maximização da métrica promove a dispersão das equipes durante a execução das rotas. Normalmente, os objetivos de lucro e dispersão são conflitantes, e com uma abordagem biobjetivo o tomador de decisão é capaz de avaliar a troca entre a coleta de lucros e a dispersão de equipes. Dois métodos de solução são propostos, um Algoritmo Genético Multiobjetivo e um Algoritmo Genético Multiobjetivo com Busca Local, ambos especializados para explorar as características do problema. É demonstrado, por meio de experimentos computacionais sobre um novo conjunto de instâncias, que a abordagem proposta fornece conjuntos de aproximação com as características desejadas.

Palavras-chave: Problema de Múltiplos Caixeiros Viajantes. Roteamento de Arcos. Métrica de Dispersão. Minimização da Dispersão. Maximização de Dispersão. Coleta de Prêmios. Algoritmo Genético. Algoritmo Genético com Busca Local Multiobjetivo. Rotas Sincronizadas.

ABSTRACT

VEHICLE ROUTING PROBLEMS WITH TEMPORAL AND SPATIAL DEPENDENCIES AMONG ROUTES

AUTHOR: Guilherme Dhein

ADVISOR: Ghendy Cardoso Júnior

CO-ADVISOR: Olinto César Bassi de Araújo

This thesis presents two new routing problems, both with objective functions focused on relative positioning of teams during the routing horizon. The relative positioning results in temporal and spatial dependencies among routes and is quantified with a nonlinear dispersion metric, designed to evaluate the instantaneous distances among teams over a time interval. This metric allows the design of objective functions to approximate teams during routes execution, when minimized, or disperse them, when maximized. Both approximation and dispersion are important routing characteristics in some practical applications, and two new optimization problems are proposed with these opposite objectives. The first one is a variation of the Multiple Traveling Salesman Problem, and its goal is to find a set of tours where the salesmen travel close to each other, minimizing dispersion. A Local Search Genetic Algorithm is proposed to solve the problem. It includes specialized genetic operators and neighborhoods. A new set of benchmark instances is proposed, adapted for the new problem from literature instances. Computational results show that the proposed approach provides solutions with the desired characteristics of minimal dispersion. The second problem is a bi-objective arc routing problem in which routes must be constructed in order to maximize collected profit and dispersion of teams. The maximization of the dispersion metric fosters the scattering of the teams during routing procedure. Usually, profit and dispersion objectives are conflicting, and by using a bi-objective approach the decision maker is able to choose a trade-off between collecting profits and scattering teams. Two solution methods are proposed, a Multi-objective Genetic Algorithm and a Multi-objective Genetic Local Search Algorithm, both specialized in order to exploit the characteristics of the problem. It is demonstrated, by means of computational experiments on a new set of benchmark instances, that the proposed approach provides approximation sets with the desired characteristics.

Keywords: Multiple Traveling Salesman Problem. Arc Routing. Dispersion metric. Dispersion minimization. Dispersion maximization. Profit Collection, Genetic Algorithm. Multi-objective Genetic Local Search Algorithm. Synchronized Routes.

LISTA DE FIGURAS

Figura 2.1 – Relação entre as soluções de um problema com dois objetivos de maximização.	35
Figura 2.2 – Exemplo de hipervolume.	38
Figura 2.3 – Exemplo da relação entre as soluções das fronteiras de aproximação e de referência.	39
Figura 2.4 – Demonstração da complementariedade das métricas.	40
Figura 2.5 – Troca de arcos no movimento da Busca Local 2-opt.	41
Figura 2.6 – Representação dos quatro movimentos que definem a vizinhança da Busca Local 3-opt.	41
Figura 3.1 – Diagrama que representa a distância instantânea entre dois veículos. ...	50
Figura 3.2 – Rotas e as respectivas fatias de tempo.	50
Figura 3.3 – Quatro exemplos do cálculo da métrica.	53
Figura 4.1 – Conjunto de rotas e as respectivas fatias de tempo.	57
Figura 4.2 – Conjunto \mathcal{V} de vértices de uma instância do ETSP.	59
Figura 4.3 – Ilustração de uma instância mTSP-D criada a partir de uma instância do ETSP com sete vértices.	60
Figura 4.4 – Dois veículos se afastando a partir de um par de vértices correspondentes.	61
Figura 4.5 – Características temporais de uma solução da instância de exemplo usada nas outras figuras.	62
Figura 4.6 – Os marcadores de tempo que determinam o padrão de fatias de tempo que ocorre quando é percorrido qualquer par de arestas correspondentes do trecho interno da rota.	63
Figura 4.7 – Replicação de vértices para geração do grafo do modelo matemático. ..	66
Figura 4.8 – Passo de inserção de um vértice no algoritmo construtivo.	73
Figura 4.9 – Ilustração do uso de cromossomos que privilegiam a representação da sucessão de vértices.	75
Figura 4.10 – Recombinação de soluções com o seccionamento das rotas em um instante de tempo.	76
Figura 4.11 – Cruzamento entre p_1 e p_2 para gerar o_1 e o_2	79
Figura 4.12 – Par de vértices mais custoso para uma fatia de tempo.	81
Figura 4.13 – Movimentos de inserção e troca.	82
Figura 4.14 – Movimento de ciclo com um ciclo de tamanho 3.	83
Figura 4.15 – mTSP-D e remoção de cruzamento, na solução com duas rotas e na abstração que considera a rota única.	85
Figura 4.16 – Movimento de eliminação de cruzamento aplicado a apenas uma das rotas.	86
Figura 4.17 – Conjuntos de arestas que podem ser removidas para aplicação de um movimento de redução de rotas.	87
Figura 4.18 – O movimento m2-opt, aplicado a roteamentos com duas rotas.	88
Figura 4.19 – Os quatro movimentos que definem a vizinhança m3-opt, aplicados a um roteamento com três rotas.	89
Figura 4.20 – Evolução da incumbente durante duas execuções relativas à instância eil51-m2s1.	97
Figura 4.21 – Evolução da incumbente durante a execução de melhor solução da ins-	

tância rat99-m5sA.	98
Figura 4.22 – Melhores soluções encontradas para as instâncias eil51-m2s1 e eil76-m5s1.	100
Figura 4.23 – Solução com alta dispersão momentânea para atender vértices remotos.	101
Figura 4.24 – Comparação de duas soluções para a instância eil51-m2s1.	102
Figura 4.25 – Ampliação do canto inferior esquerdo da Figura 4.22(a).	102
Figura 4.26 – Alternâncias de direção provocando cruzamentos entre rotas.	103
Figura 5.1 – Rotas e as respectivas fatias de tempo.	107
Figura 5.2 – Cruzamento aplicado a dois indivíduos com três rotas cada.	115
Figura 5.3 – $ND1$ e $ND2$ não preenchem a população.	119
Figura 5.4 – $ND1$ e $ND2$ juntos têm mais indivíduos do que $Elite$ e $F0$	120
Figura 5.5 – Arcos determinantes para a dispersão de uma fatia de tempo.	121
Figura 5.6 – Melhores e piores fronteiras de aproximação para MOGA e MOGLS de acordo com a métrica $D1_R$	126
Figura 5.7 – Comparação de duas fronteiras de referência para instâncias com diferentes variações nos valores de prêmio associado aos arcos.	127
Figura 5.8 – Fronteira de referencia para uma instância com $p_{ij} = c_{ij}$	128
Figura 5.9 – Progressão no tempo de uma solução.	128
Figura 5.10 – Progressão no tempo de uma solução com máxima dispersão.	129
Figura 5.11 – Progressão no tempo de uma solução com máximo prêmio coletado. ..	130

LISTA DE TABELAS

Tabela 4.1 – Arestas inseridas em cada movimento da vizinhança m3-opt.....	88
Tabela 4.2 – Tipos de vizinhanças exploradas durante a Busca Local.	92
Tabela 4.3 – Parâmetros do LSGA.....	93
Tabela 4.4 – Resultados Computacionais.....	95
Tabela 4.5 – Momento do surgimento da melhor solução e sua relação com o total de gerações.	99
Tabela 5.1 – Características que variam entre os conjuntos de instâncias.....	122
Tabela 5.2 – Parâmetros e valores usados nos métodos MOGA e MOGLS.....	122
Tabela 5.3 – Resultados Computacionais.....	123

LISTA DE ABREVIATURAS E SIGLAS

<i>BARP – PD</i>	Problema do Roteamento de Arcos Biobjetivo com Coleta de Prêmios e Maximização da Dispersão, sigla do inglês <i>Bi-objective Arc Routing Problem with Prize Collection and Dispersion Maximization</i>
<i>ETSP</i>	Problema de Caixeiro Viajante Euclidiano, sigla do inglês <i>Euclidean Traveling Salesman Problem</i>)
<i>LSGA</i>	Algoritmo Genético com Busca Local, sigla do inglês <i>Local Search Genetic Algorithm</i>
<i>MOGA</i>	Algoritmo Genético Multiobjetivo, sigla do inglês <i>Multi-objective Genetic Algorithm</i>
<i>MOGLS</i>	Algoritmo Genético Multiobjetivo com Busca Local, sigla do inglês <i>Multi-objective Genetic Local Search Algorithm</i> .
<i>mTSP</i>	Problema de Múltiplos Caixeiros Viajantes, sigla do inglês <i>Multiple Traveling Salesman Problem</i>
<i>mTSP – D</i>	Problema de Múltiplos Caixeiros Viajantes com Objetivo de Minimização da Dispersão, sigla do inglês <i>Multiple Traveling Salesman Problem with Minimizing Dispersion Objective</i>
<i>TSP</i>	Problema do Caixeiro Viajante, sigla do inglês <i>Traveling Salesman Problem</i>
<i>VRP</i>	Problema de Roteamento de Veículos, sigla do inglês <i>Vehicle Routing Problem</i>

SUMÁRIO

1	INTRODUÇÃO	19
1.1	OBJETIVOS	21
1.2	MOTIVAÇÃO	21
1.3	ESTRUTURA	22
2	REVISÃO BIBLIOGRÁFICA	25
2.1	PROBLEMAS DE ROTEAMENTO	26
2.2	PROBLEMAS DE ROTEAMENTO COM INTERDEPENDÊNCIA ENTRE AS ROTAS	31
2.3	PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO	34
2.4	MÉTODOS DE SOLUÇÃO	39
2.4.1	Algoritmos de Busca Local 2-opt e 3-opt	40
2.4.2	Algoritmos Genéticos	42
2.4.3	NSGA II	44
2.4.4	Método MOGLS de Armentano e Arroyo (2005)	45
2.5	RESUMO DO CAPÍTULO	46
3	MÉTRICA PARA QUANTIFICAR A DISPERSÃO ENTRE DOIS VEÍCULOS	49
3.1	SUBSTITUIÇÃO DO VETOR VELOCIDADE PELO VETOR DESLOCAMENTO NO CÁLCULO DA DISPERSÃO	51
3.2	RESUMO DO CAPÍTULO	52
4	O PROBLEMA DO MÚLTIPLO CAIXEIRO VIAJANTE COM OBJETIVO DE MINIMIZAÇÃO DA DISPERSÃO	55
4.1	DEFINIÇÃO DO PROBLEMA	56
4.1.1	Modelo Matemático	65
4.2	MÉTODO DE SOLUÇÃO	70
4.2.1	O Algoritmo Genético	70
4.2.2	Geração da População Inicial	71
4.2.3	Cromossomo e Operadores Genéticos	74
4.2.4	Busca Local	80
4.2.4.1	<i>Vizinhança de aproximação das rotas</i>	80
4.2.4.2	<i>Vizinhança de redução de rotas</i>	84
4.2.4.3	<i>Procedimento de Busca Local com as Vizinhanças Descritas</i>	90
4.3	RESULTADOS E DISCUSSÕES	92
4.3.1	Instâncias e Parâmetros	93
4.3.2	Resultados Computacionais	94
4.4	RESUMO DO CAPÍTULO	103
5	O PROBLEMA DO ROTEAMENTO DE ARCOS BI-OBJETIVO COM COLETA DE PRÊMIOS E MAXIMIZAÇÃO DA DISPERSÃO	105
5.1	DEFINIÇÃO DO PROBLEMA	106
5.2	MÉTODOS DE SOLUÇÃO	108
5.2.1	Algoritmo Genético Multiobjetivo	108
5.2.2	Algoritmo Genético Multiobjetivo com Busca Local	110
5.2.3	Geração da População Inicial	111
5.2.4	Geração de Novas Populações	112
5.2.5	Cromossomo e Operadores Genéticos	114
5.2.6	Busca Local	116

5.2.6.1	<i>Procedimento de Melhoria de Fronteira</i>	116
5.2.6.2	<i>Fronteiras Submetidas à Busca Local e Inserção de Soluções Resultantes na População</i>	118
5.2.6.3	<i>Vizinhanças</i>	120
5.3	RESULTADOS E DISCUSSÕES	121
5.3.1	Instâncias e Parâmetros	121
5.3.2	Resultados Computacionais	123
5.4	RESUMO DO CAPÍTULO	130
6	CONCLUSÃO	133
	REFERÊNCIAS BIBLIOGRÁFICAS	137
	APÊNDICE A – LISTA DE TRABALHOS ORIGINAIS	147

1 INTRODUÇÃO

Uma solução de roteamento para equipes de campo se caracteriza por um conjunto de rotas que indique os deslocamentos a serem realizados pelas equipes de modo a atenderem serviços designados. Esse problema é de natureza combinatória e se enquadra na classe de problemas conhecida como roteamento de veículos, que em geral utilizam uma função objetivo associada à minimização de algum custo referente aos deslocamentos ou à maximização de algum ganho referente aos serviços efetivamente realizados. Algumas variações consideram restrições ou objetivos que provocam a existência de uma inter-relação entre rotas, traduzida na forma de dependência mútua em que alterações em uma rota demanda alterações nas demais. No decorrer desse trabalho, as unidades executoras das rotas serão referidas indistintamente como equipes ou veículos.

Nessa tese são estudados dois problemas nos quais as rotas apresentam dependência temporal e espacial entre todas as equipes, avaliada de forma contínua durante todo o horizonte de planejamento. A dependência não se resume a arcos percorridos ou vértices visitados em sequência ou concomitantemente por diferentes equipes, mas se manifesta pela verificação constante do distanciamento relativo entre todas elas.

Para a definição dos problemas com tal característica, inicialmente é proposta uma métrica não linear, original, para quantificar a dispersão entre as equipes. Essa métrica demanda a divisão do tempo de execução das rotas em fatias de tempo, intervalos cujos limites são determinados por mudança em qualquer dos vetores velocidade que definem o deslocamento das equipes. Tais alterações nos vetores ocorrem sempre que um veículo chega ou parte de um vértice. Dentro do intervalo de uma fatia de tempo, os veículos são tomados aos pares e suas distâncias relativas avaliadas de forma contínua, no que é referido como dispersão entre dois veículos. O valor de dispersão de cada fatia de tempo é calculado considerando todos os pares de veículos no intervalo. Do somatório dos valores das fatias tem-se o valor de dispersão para o roteamento como um todo.

A partir da avaliação da dispersão, é possível estabelecer funções objetivo que promovam a construção de rotas nas quais as equipes se mantêm próximas ou afastadas. Os novos problemas propostos contemplam as duas alternativas. No primeiro, chamado Problema de Múltiplos Caixeiros Viajantes com Objetivo de Minimização da Dispersão (mTSP-D, do inglês *Multiple Traveling Salesman Problem with Minimizing Dispersion Objective*), a dispersão é minimizada. Já no segundo, chamado Problema do Roteamento de Arcos Biobjetivo com Coleta de Prêmios e Maximização da Dispersão (BARP-PD, do inglês *Bi-objective Arc Routing Problem with Prize Collection and Dispersion Maximization*) a dispersão é um dos objetivos a ser maximizado, juntamente com um objetivo de coleta de prêmios.

Os problemas propostos se unem a uma miríade de problemas de roteamento, que

apresentam uma diversidade considerável entre si. Dessa diversidade surgem outras características que diferenciam os problemas aqui propostos, além das funções objetivo antagonicas, a saber: o primeiro dos problemas é de roteamento de vértices, o outro de arcos; em um dos problemas, as rotas construídas são fechadas, no outro abertas; em um dos problemas existe restrição de turno, que limita o comprimento das rotas, no outro não; em um deles, existe tempo de serviço, ou seja, os veículos ficam parados por alguns instantes, no outro não; em um dos problemas, apenas o objetivo relativo à dispersão deve ser otimizado, no outro o objetivo relativo à dispersão é parte de uma composição biobjetivo.

O uso de objetivos baseados no posicionamento relativo das equipes, tanto de aproximação quanto de afastamento, tem aplicações relacionadas a atividades de logística realizadas em áreas hostis. Em alguns casos, é interessante que as equipes responsáveis pela atividade sejam espalhadas pelo ambiente, de modo a maximizarem a área coberta e reduzirem a probabilidade de serem percebidas. Além disso, a dispersão aumenta a robustez do roteamento, pois reduz a possibilidade de que um problema que impossibilite o prosseguimento de uma equipe afete também as demais rotas.

Por outro lado, em alguns casos a preocupação na modelagem do roteamento está na possibilidade de que as equipes sejam confrontadas com situações que exijam apoio. Nesse caso, mesmo que elas não estejam se deslocando pelos mesmos arcos e nós, por estarem associadas a rotas particulares, é interessante que sejam mantidas com relativa proximidade de modo que uma possa prestar apoio rapidamente para outra em caso de necessidade.

Ambas possibilidades podem ser percebidas em diferentes interpretações das necessidades de atividades de vigilância e de distribuição de ajuda humanitária em regiões submetidas a desastres ou conflitos militares. O agrupamento, por exemplo, é mais interessante em vigilância ostensiva em áreas violentas ou em ações humanitárias emergenciais, quando a gravidade de uma situação ainda não é totalmente conhecida e são possíveis situações que exijam o trabalho conjunto de mais de uma equipe. Já a dispersão é mais interessante em vigilância de grandes áreas e/ou não ostensiva, e na distribuição de ajuda humanitária para uma população também dispersa.

Os resultados apresentados na tese foram obtidos através de métodos baseados em Algoritmos Genéticos. Para o mTSP-D, foi implementado um Algoritmo Genético com Busca Local (LSGA, do inglês *Local Search Genetic Algorithm*), e para o BARP-PD foram implementados um Algoritmo Genético Multiobjetivo (MOGA, do inglês *Multi-objective Genetic Algorithm*) e um Algoritmo Genético Multiobjetivo com Busca Local (MOGLS, do inglês *Multi-objective Genetic Local Search Algorithm*). Especializações são propostas para tratar adequadamente características de sincronização e interdependência que são impostas pelas funções objetivo baseadas no posicionamento relativo dos veículos. É importante salientar que são apresentados conjuntos de instâncias específicos e originais para os dois problemas. Esses conjuntos de instâncias se encontram disponíveis na *Inter-*

net, juntamente com as melhores soluções encontradas nos testes computacionais.

1.1 OBJETIVOS

O objetivo geral da tese é propor uma forma de avaliar a proximidade temporal e espacial entre os veículos envolvidos em um roteamento e testar o seu uso em problemas distintos, um com o objetivo de manter os veículos próximos entre si e outro com o objetivo de dispersá-los na área definida pela instância.

Como objetivos específicos, esse trabalho visa:

- formalizar uma métrica de dispersão em que seja observado o posicionamento dos veículos em todos os instantes do roteamento, e a sua dispersão seja calculada em função das distâncias instantâneas;
- definir um problema de roteamento de veículos que tenha como objetivo manter os veículos agrupados, minimizando a dispersão;
- definir um problema de roteamento de veículos que tenha como objetivo manter os veículos espalhados pelo ambiente, maximizando a dispersão;
- investigar características específicas dos problemas decorrentes do objetivo de dispersão (seja maximização ou minimização), com destaque para o tipo específico de sincronização que é estabelecida entre os veículos e, a partir daí, conceber especializações em métodos de solução e vizinhanças para Busca Local;
- propor métodos de solução robustos para os problemas apresentados, de modo que as soluções permitam uma verificação confiável da ocorrência das relações espaciais e temporais desejadas entre os veículos;
- criar conjuntos de instâncias para cada problema, para uso na geração dos resultados da tese, com posterior divulgação para a comunidade das instâncias e dos resultados.

1.2 MOTIVAÇÃO

O trabalho teve origem em um Projeto de Pesquisa e Desenvolvimento realizado em conjunto com uma empresa concessionária de energia elétrica. No contexto da empresa, equipes são enviadas a campo para a realização de tarefas previamente agendadas

através de ordens de serviço ditas comerciais. Mas as mesmas equipes são responsáveis pelo atendimento de ordens de serviço dinâmicas, ditas emergenciais, sendo deslocadas de suas rotas planejadas de acordo com tais contingências. O tempo necessário para o início do atendimento a uma ordem emergencial é relativo à distância que as equipes estão do local de ocorrência da demanda que exige pronto atendimento.

Em reunião com pessoal ligado à concessionária para discutir características desejáveis para o roteamento comercial, foi levantada a alternativa de manter as equipes *espalhadas* pela área de cobertura. Essa sugestão implica na verificação constante das posições dos veículos e na construção de rotas que maximizem as distâncias relativas. O projeto foi finalizado sem contemplar essa alternativa, em função de outras características do problema.

Mas verificou-se posteriormente que roteamentos com uma avaliação a partir do monitoramento constante das posições relativas é uma característica importante em outras aplicações práticas, conforme já descrito. Uma mesma métrica para quantificar o distanciamento relativo oferece possibilidades de uso tanto para agrupar equipes, quando minimizada, quanto para dispersar, quando maximizada, aumentando as possibilidades de utilização.

Percebeu-se uma lacuna na literatura com relação a rotas com essas características, não sendo encontrados trabalhos que apresentassem métrica para a avaliação constante do distanciamento, e nem focados em objetivos semelhantes aos presentes nos problemas apresentados.

À utilidade e à originalidade, junta-se ainda outra motivação: o desafio do ponto de vista acadêmico. A verificação de dispersão implica em uma interdependência entre as rotas, o que é indicado na literatura como um fator de dificuldade em problemas de roteamento de múltiplas rotas.

1.3 ESTRUTURA

O presente trabalho foi organizado em cinco capítulos, contados a partir da presente introdução. No Capítulo 2 é feita uma revisão bibliográfica a respeito de problemas de roteamento de veículos, com destaque para aqueles em que se manifestam características de interdependência e sincronização entre rotas. É feita ainda uma revisão de problemas de otimização multiobjetivo, seguida de uma descrição breve dos métodos de solução que contribuíram na elaboração dos aqui implementados.

No Capítulo 3 é descrita a métrica de dispersão utilizada. É detalhada a forma como o tempo de execução das rotas é dividido em fatias de tempo em função de variações nos vetores velocidade dos veículos, e como a dispersão é calculada para cada par de veículos considerando a duração de uma fatia.

Essa métrica pode ser utilizada em problemas em que se deseja a aproximação ou a dispersão dos veículos envolvidos. O primeiro caso ocorre no problema descrito no Capítulo 4, no qual uma função objetivo de aproximação favorece soluções em que os veículos mantêm uma proximidade temporal e espacial durante a execução de suas rotas. O capítulo contém ainda a descrição do método de solução implementado e dos testes computacionais realizados. O problema biobjetivo que utiliza a métrica em um objetivo de dispersão é descrito no Capítulo 5. O problema apresenta ainda um objetivo de coleta de prêmio. A exemplo do que ocorre no capítulo anterior, são apresentados métodos de solução e resultados computacionais. Por fim, no Capítulo 6 são apresentadas as conclusões do trabalho.

2 REVISÃO BIBLIOGRÁFICA

A abstração de Leonhard Euler para o problema das Sete Pontes de Königsberg, datada de 1736, é tida com a precursora no estudo teórico de grafos (CORMEN et al., 2009). Euler associou um vértice a cada porção de terra, e uma aresta a cada ponte, obtendo um grafo. A partir daí, demonstrou formalmente a impossibilidade de construir um circuito que percorresse todas as pontes uma vez e retornasse ao ponto de partida, e estabeleceu um teorema que determina as condições para que um caminho com tais restrições (um circuito) seja possível em qualquer grafo. Em sua homenagem, um circuito que percorre todas as arestas de um grafo é chamado do Caminho Euleriano.

Formalmente, um grafo pode ser definido como um par $G = (V, A)$, em que V representa o conjunto de vértices ou nós e A representa o conjunto de arestas ou arcos que os conectam. Grafos são usados amplamente na representação de conjuntos de objetos em que pares são associados por relação binária, que estabelece conexão entre eles. No caso de problemas de roteamento de veículos, normalmente os vértices ou os arcos do grafo representam posições relacionadas a demandas a serem cumpridas.

É sobre a abstração fornecida pelo grafo que se alicerçam e são definidos os problemas de roteamento, em que se busca, dentre as configurações de rota(s) que atendem as demandas, aquela que obtém melhor avaliação frente a objetivos estabelecidos. Problemas deste tipo têm forte apelo econômico, pois estão presentes na logística de uma série de atividades ligadas a transporte de pessoas e mercadorias (rotas para transporte escolar, para coleta e entrega de encomendas, etc.), além de outros serviços que necessitam do deslocamento de pessoas e/ou veículos (rotas para leitura de consumo de energia elétrica, rotas para veículo de remoção de neve das ruas, etc.).

São apresentados na tese dois novos problemas de roteamento, e este capítulo tem por objetivo revisar a literatura pertinente a eles. Inicialmente, na Seção 2.1, é feita uma descrição genérica sobre problemas de roteamento, tanto de arcos quanto de vértices. O foco dessa seção é ilustrar como o grande conjunto de problemas de roteamento (ao qual se unem os dois problemas novos) se origina a partir de variações em objetivos ou restrições. Além disso, são apresentadas características e variações que são incorporadas aos problemas originais da tese.

Os novos problemas apresentados nos Capítulos 4 e 5 se caracterizam por múltiplas rotas que apresentam interdependência. Quando isso ocorre, a Seção 2.2 segue apresentando variações de problemas de roteamento, mas agora com o foco naqueles que se caracterizam pela ocorrência ou pela necessidade de interdependência ou sincronização entre rotas.

Em alguns casos, apenas um objetivo não é suficiente para descrever aquilo que se espera da solução. Neste caso, vários objetivos, potencialmente conflitantes, devem ser

considerados. Os problemas de roteamento multiobjetivos são abordados na Seção 2.3.

Por fim, na Seção 2.4 são revisados aqueles métodos de solução para problemas de otimização que inspiraram ou foram usados na construção daqueles implementados para a solução dos novos problemas.

2.1 PROBLEMAS DE ROTEAMENTO

O principal objetivo dessa seção é ilustrar como o roteamento de veículos se caracteriza por uma gama enorme de problemas, originados a partir de novos objetivos ou restrições que alteram ou generalizam problemas fundamentais anteriormente definidos na literatura. Enquanto as novas características aproximam modelos e rotas construídas de situações práticas, do ponto de vista acadêmico resultam em um universo vasto de problemas que demandam pesquisa para adaptação ou concepção de novos métodos de solução. É nesse contexto que se enquadra a tese, já que a partir de uma nova métrica, foram geradas novas funções objetivo que, junto a algumas outras características encontradas dentro da diversidade dos problemas de roteamento, resultaram em problemas de roteamento originais.

Não se pretende fazer uma descrição completa de tais variações, mas sim demonstrar como elas surgem nas mais diversas dimensões do contexto de roteamento, desde a definição do grafo sobre o qual as soluções são construídas, passando pelas restrições consideradas, pelas características das rotas desejadas e dos veículos disponíveis, até chegar em variações nos objetivos a otimizar. Para a visualização de um elenco exaustivo de variações e aplicações encontradas na literatura para problemas de roteamento de vértices, sugere-se a leitura de (DREXL, 2012a). Uma análise da taxonomia de problemas de roteamento pode ser encontrada em (EKSIÖGLU; VURAL; REISMAN, 2009). Em referência a formulações e aplicações de problemas de roteamentos de arcos, sugere-se a leitura de (ASSAD; GOLDEN, 1995) e da revisão da literatura publicada em duas partes em (EISELT; GENDREAU; LAPORTE, 1995a) e (EISELT; GENDREAU; LAPORTE, 1995b).

A classificação dos problemas em roteamento de arcos e roteamento de vértices é considerada há mais de 40 anos (GABBAY, 1974). No primeiro caso, é necessário que um subconjunto de arcos ou arestas do grafo (ou possivelmente todos) sejam incluídas na(s) rota(s) da solução, independente dos vértices visitados. Um exemplo clássico deste tipo de problemas é o Problema do Carteiro Chinês (??), no qual um carteiro deve perfazer o menor circuito possível, desde que todas as ruas atendidas por ele sejam percorridas pelo menos uma vez. Uma versão com múltiplas rotas é conhecida como Problema do Roteamento de Arcos Capacitado (*Capacitated Arc Routing Problem*), e é NP-difícil (WØHLK, 2008).

No segundo caso, uma solução deve garantir a visita a um certo conjunto de vértices

(possivelmente todos). Nesta categoria se inclui o clássico Problema do Caixeiro Viajante (TSP, do inglês *Traveling Salesman Problem*) (LAWLER et al., 1987), em que se busca uma rota com o menor custo de deslocamento de modo que seja um circuito (termine no vértice de início) e todos os clientes do caixeiro sejam visitados. Incluem-se também duas generalizações do TSP para múltiplas rotas, o Problema de Múltiplos Caixeiros Viajantes (mTSP, do inglês *Multiple Traveling Salesman Problem*) (BEKTAS, 2006; KARA; BEKTAS, 2006), e o Problema do Roteamento de Veículos (VRP, do inglês *Vehicle Routing Problem*) (TOTH; VIGO, 2001), versão mais genérica por considerar as características dos veículos responsáveis pela execução das rotas. O TSP é reconhecido na literatura como NP-difícil, complexidade compartilhada pelo mTSP e pelo VRP, que o generalizam (VENKATESH; SINGH, 2015; TOTH; VIGO, 2001).

Em ambos os casos, o objetivo mais comum é a redução do custo de deslocamento, em geral relacionado à distância percorrida. O Problema do Caixeiro Viajante Euclidiano (ETSP, do inglês *Euclidean Travelling Salesman Problem*) (CZUMAJ, 2014) exemplifica os problemas formulados de modo que os vértices sejam dispostos em um plano e os custos das arestas computados como as distâncias euclidianas. Trata-se obviamente de uma aproximação, mas com vantagens importantes, como a garantia da inexistência de assimetria e o respeito à desigualdade triangular, evitando a dificuldade que a inexistência dessas características traz aos métodos de solução (FLEMING; GRIFFIS; BELL, 2013; RODRÍGUEZ; RUIZ, 2012).

A construção do grafo pode associar outras grandezas às arestas, como o tempo de deslocamento, que pode ser influenciado pela velocidade possível de desempenhar em cada via (ÑAHUIS; FLORES, 2013), ou o consumo de combustível, afetado pelo grau de inclinação de cada via (TAVARES et al., 2009). Já no Problema do Caixeiro Viajante com Coleta de Prêmios (*Prize Collecting Traveling Salesman Problem*) (BALAS, 1989), as arestas são associadas a custos e prêmios, e uma restrição define o valor de prêmio mínimo a ser coletado. O objetivo do roteamento é atingir esse valor de prêmio com o mínimo custo.

É possível considerar ainda características que vão além de valores absolutos atribuídos aos caminhos, como as características de mobilidade quando o caminho é efetivamente percorrido, no que é conhecido como custo dependente do tempo (*time dependent cost*) (MALANDRAKI; DASKIN, 1992; BLACK; EGGLESE; WØHLK, 2013).

O formato das rotas também varia entre os problemas, a começar pelo seu ponto de partida. Enquanto os casos em que uma ou mais rotas partem de um mesmo vértice garagem são mais comuns, há outros com garagens múltiplas (TLILI et al., 2016; KARAKATIC; PODGORELEC, 2015; KULKARNI; TAI, 2010). Outros exemplo de diferenciação relativa aos formatos de rotas ocorrem quando elas não precisam contemplar o retorno dos veículos às garagens, sendo formadas rotas ditas abertas (LÓPEZ-SÁNCHEZ et al., 2014), ou quando são estabelecidas rotas que passam diversas vezes pela garagem, para

recarga (ou descarga) do veículo (CATTARUZZA et al., 2014).

Com relação às unidades executoras das rotas, a associação de capacidades a veículos é bastante comum, gerando problemas ditos capacitados (WØHLK, 2008; NAZIF; LEE, 2012). O aceite de variações na capacidade dos veículos possibilita a definição de problemas com frota heterogênea, embora a heterogeneidade dos veículos possa se dar ainda por outros motivos: custo de uso (consumo de combustível, manutenção diferenciada, etc.), velocidade de deslocamento, disponibilidade (alguns veículos podem ser terceirizados, não estando sempre disponíveis), equipamento transportado, etc.

A existência de compartimentos nos veículos que permitam o transporte de cargas sem o compartilhamento do espaço resulta no problema conhecido como Problema de Roteamento de Veículos Multicompartimentados (*Multicompartment VRP*) (HENKE; SPERANZA; WÄSCHER, 2015). Em casos especiais, um dos compartimentos é um reboque que pode ser separado do veículo caso alguns vértices sejam inacessíveis com o reboque acoplado. Essa variação do VRP é conhecida como Problema de Roteamento de Caminhão e Reboque (*Truck and Trailer Routing Problem*) (DERIGS; PULLMANN; VOGEL, 2013).

Além das características dos veículos, também as características dos clientes ou serviços representados nos vértices exibem diversidade. Um exemplo ocorre quando os vértices podem representar tanto tarefas de coleta como de entrega de bens. Variações de roteamento com essa característica estão sumarizados em (PARRAGH; DOERNER; HARTL, 2008a, 2008b), mas algumas podem ser destacadas: VRP com *Backhauls*, em que as rotas são construídas de modo que os pontos de entrega sejam visitados antes dos pontos de coleta (TOTH; VIGO, 1997); VRP com coleta e entrega (*Vehicle Routing Problems with Pickups and Deliveries*), em que pontos de coleta estão associados a pontos de entrega e implicam em precedência na visita (DESAULNIERS et al., 2001); e VRP com carregamento em LIFO (*VRP with LIFO loading*), quando as entregas são realizadas em ordem inversa a das coletas, de modo a facilitar o acesso às mercadorias dentro do veículo no momento da descarga (CARRABS; CORDEAU; LAPORTE, 2007). Ainda em relação ao serviço prestado, a existência de restrição quanto ao momento do atendimento levam a problemas com restrição de janelas de tempo (CORDEAU et al., 2001; LABADI; PRINS; REGHIOUI, 2008).

Existe também a possibilidade de que alguns vértices sejam apenas cobertos, e não necessariamente visitados. A cobertura se dá pela passagem de um veículo suficientemente próximo, como nos casos do Problema das Rotas de Cobertura Multiveículos (*Multi-vehicle Covering Tour Problem*) (HACHICHA et al., 2000) e do Problema de Roteamento de Arcos Suficientemente Próximo (*Close-Enough Arc Routing Problem*) (HÀ et al., 2014).

Além das características do grafo, dos vértices e dos veículos, os diferentes objetivos a serem otimizados originam também variações importantes em problemas de ro-

teamentos. Alguns dos objetivos mais encontrados na literatura, e diferentes da seminal minimização do custo total de deslocamento, serão referidos na sequência.

Um objetivo importante é a equiparação de rotas, gerando um balanceamento entre elas. Em geral, tal objetivo está associado à equiparação dos custos, pela redução da rota mais longa (NARASIMHA et al., 2013; SINGH; BAGHEL, 2008). Mas outros balanceamentos são possíveis, como pela distribuição mais uniforme da carga transportada (KRITIKOS; IOANNOU, 2010).

Quando o foco do roteamento deve ser a satisfação dos clientes representados nos vértices, é interessante ter por objetivo a redução do tempo de chegada em cada um, normalmente referido como latência. O objetivo de minimizar o somatório das latências é aplicado a roteamentos de apenas uma rota, como no Problema da Latência Mínima (*Minimum Latency Problem*) (SILVA et al., 2012), ou com mais rotas, como no Problema do Roteamento de Veículos Capacitado e Cumulativo (*Cumulative Capacitated Vehicle Routing Problem*) (RIBEIRO; LAPORTE, 2012). Outro problema focado na satisfação dos clientes (nesse caso, estudantes) é o Problema do Roteamento Urbano de Ônibus Escolares (*Urban School Bus Routing Problem*) (BOWERMAN; HALL; CALAMAI, 1995), em que se deseja definir rotas para os ônibus que minimizem a distância entre a casa de cada estudante e a parada mais próxima.

Nos problemas já descritos, as rotas são percebidas como um custo a ser minimizado enquanto demandas obrigatórias são cumpridas. Uma inversão ocorre quando o objetivo do roteamento é atrelado à coleta de prêmios associados a vértices ou arestas. Nesse caso, é possível escolher os vértices a visitar ou os arcos a percorrer de modo a maximizar o total de prêmios coletados. O Problema da Orientação (*Orienteering Problem*) e o Problema de Orientação de Equipe (*Team Orienteering Problem*) são problemas em que o objetivo de maximização de prêmio está associado ao roteamento de vértices, com uma ou mais rotas, respectivamente. Descrições detalhadas, formalização e variações com janela de tempo para os dois problemas podem ser encontradas em (VANSTEENWEGEN; SOUFFRIAU; OUDHEUSDEN, 2011).

A caracterização da maximização de prêmios e da minimização do custo de roteamento como objetivos antagônicos é detalhada em (FEILLET; DEJAX; GENDREAU, 2005b), juntamente com uma classificação de variações do TSP com prêmios. Os autores apontam a possibilidade de uma abordagem multiobjetivo, mas também apontam a possível composição de um objetivo de lucro que combina o ganho representado pelos prêmios com o custo do roteamento. Em outra composição possível, o prêmio auferido pela visita a um vértice é reduzido de acordo com a latência, ou seja, o lucro diminui com o atraso na visita ao vértice, em problema conhecido como Problema do Reparador Viajante com Lucro (*Traveling Repairman Problem with Profit*) (DEWILDE et al., 2013).

O trabalho em (MALANDRAKI; DASKIN, 1993) é normalmente referido como a primeira proposição de um problema de roteamento de arcos com coleta de prêmio, ao definir

o Problema do Carteiro Chinês com Benefício Máximo (*Maximum Benefit Chinese Postman Problem*). Variações multiveículos de roteamento de arcos com coleta de prêmios podem ser encontradas em (FEILLET; DEJAX; GENDREAU, 2005a), (ARCHETTI et al., 2010) e (ZACHARIADIS; KIRANOUDIS, 2011). Trabalhos que abordam problemas de roteamento de arco com prêmio não são abundantes na literatura, e um sumário é apresentado em (BLACK; EGGLESE; WØHLK, 2013), juntamente com uma variação do problema com dependência de tempo.

A coleta de prêmios se configura como um dos objetivos do problema de roteamento de arcos biobjetivo descrito no Capítulo 5. Mas a característica original do presente trabalho, que une os dois problemas novos propostos, é a métrica usada em objetivos que estabelecem uma interdependência entre as rotas. Por esse motivo, problemas de roteamento em que se destaca algum tipo de relação entre as rotas, especialmente características de sincronização, serão tratados na seção seguinte.

Alguns problemas se caracterizam por combinarem mais de um objetivo, e se configuram como problemas de roteamento multiobjetivo, como os descritos em (LACOMME; PRINS; SEVAUX, 2006), (GRANDINETTI et al., 2012), (SHANG et al., 2014), (BOWERMAN; HALL; CALAMAI, 1995) e (JOZEFOWIEZ; SEMET; TALBI, 2008). Objetivos conflitantes estabelecem particularidades que demandam métodos específicos de solução. Tais particularidades e métodos serão revisados na Seção 2.3.

Vale destacar ainda que o advento relativamente recente de tecnologias de comunicação móvel e de posicionamento e monitoramento deram origem aos problemas de roteamento dinâmico, em que as rotas são rearranjadas em tempo real em função de nova informação. A dinamicidade pode estar associada a diversas dimensões das instâncias, como: na inclusão de novos vértices que precisam ser visitados (CHEUNG et al., 2008; RASHIDI; FARAHANI, 2012), nas variações na demanda de cada cliente (SPLIET; GABOR; DEKKER, 2014); no número de veículos (LI; MIRCHANDANI; BORENSTEIN, 2009), no custo de deslocamento entre dois pontos (CHEUNG et al., 2008), entre outras. Uma revisão das publicações em roteamento dinâmico pode ser encontrada em (PILLAC et al., 2013).

Por fim, cabe uma referência aos métodos empregados na solução de problemas de roteamento. É possível encontrar métodos baseados em programação matemática, como os algoritmos *branch-and-bound* de (TOTH; VIGO, 1997), *branch-and-cut* de (HÀ et al., 2014) e *branch-and-price* de (DOHN; KOLIND; CLAUSEN, 2009). Uma revisão sobre métodos exatos para a solução das variações com restrições de capacidade e com janelas de tempo pode ser encontrada em (BALDACCI; MINGOZZI; ROBERTI, 2012).

A complexidade dos problemas de roteamento leva à utilização de métodos heurísticos para a obtenção de soluções de boa qualidade em tempo satisfatório. E nesse contexto, a variedade de métodos de otimização existentes (e as muitas possibilidades de combinação) levam a uma miríade de métodos aplicados ao roteamento: método construtivo

seguido de Busca Local (CÁCERES-CRUZ et al., 2013), descida ou busca com vizinhanças variáveis (PIA; FILIPPI, 2006; CARRABS; CORDEAU; LAPORTE, 2007), Algoritmo Genético (KARAKATIC; PODGORELEC, 2015; NAZIF; LEE, 2012), GRASP com *path relinking* (LABADI; PRINS; REGHIOUI, 2008), algoritmo memético (CATTARUZZA et al., 2014), otimização por colônia de abelhas e por erva daninha (VENKATESH; SINGH, 2015), *iterated local search* (TLILI et al., 2016), sistema multiagentes (KULKARNI; TAI, 2010), *adaptive large neighborhood search* (RIBEIRO; LAPORTE, 2012), busca tabu (BRANDÃO, 2011), otimização por colônia de formigas (GHAFURIAN; JAVADIAN, 2011), *simulated annealing* (LIN; YU; LU, 2011), *matheuristic* (FIKAR; HIRSCH, 2015), entre outros. Uma revisão sobre métodos de solução para problemas de roteamento multiatributos pode ser encontrada em (VIDAL et al., 2013).

2.2 PROBLEMAS DE ROTEAMENTO COM INTERDEPENDÊNCIA ENTRE AS ROTAS

Em problemas com objetivos relativos a custo de deslocamento, modificações nas soluções podem ser avaliadas localmente, pela comparação dos custos dos arcos que são removidos com os custos dos arcos inseridos na solução. Independente do número e do sequenciamento das rotas, a avaliação da solução é feita pela variação de custo nos arcos modificados. Situação semelhante se dá com outros objetivos. Para um objetivo de coleta de prêmios, por exemplo, a troca de arco ou vértice pode ser avaliada pela diferença do prêmio que entra para o que sai.

Essa relativa simplicidade não é verdadeira com outros objetivos. O uso da minimização da latência como objetivo, por exemplo, tem implicações importantes na complexidade da avaliação das soluções. Nesse caso, a alteração da sequência de vértices de uma rota implica em variação na latência de todos os vértices e arestas seguintes na mesma rota.

Existem também problemas em que o objetivo ou restrições estabelecidos definem uma relação estreita entre rotas. Em casos assim, a avaliação de uma solução deixa de ser a união das avaliações das partes para se tornar uma avaliação das relações entre essas partes. Um exemplo é a busca por caminhos ou rotas dissimilares (AKGÜN; ERKUT; BATA, 2000), relacionada ao transporte de cargas perigosas ou valiosas (TALARICO; SÖRENSEN; SPRINGAEL, 2015) ou à redução da possibilidade de detecção em missões militares (THYAGARAJAN et al., 2005). A avaliação da dissimilaridade implica obviamente na consideração das relações dos diversos caminhos entre si.

Uma alternativa de avaliação de dissimilaridade, proposta na literatura, é a comparação da área *abaixo* do caminho (LOMBARD; CHURCH, 1993): dado o plano em que o grafo está representado, é calculada a área da figura formada entre um caminho e o eixo X do plano cartesiano, e a dissimilaridade entre dois caminhos é dada pela diferença

nas áreas das figuras que eles formam. Já (DELL'OLMO; GENTILI; SCOZZARI, 2005) usa uma abordagem biobjetivo e avalia a dissimilaridade entre dois caminhos gerados calculando uma área de cobertura nas laterais de cada um, e avaliando a interseção dessas áreas. Outra avaliação que considera a dissimilaridade espacial é dada em (MARTÍ; VELARDE; DUARTE, 2009), em que se considera a distância entre cada um dos vértices de um caminho até algum ponto do outro.

O trabalho (TALARICO; SÖRENSEN; SPRINGAEL, 2015) foca em rotas ao invés de caminhos, e usa como métrica de dissimilaridade o custo de arcos compartilhados entre rotas. Nesse caso, não há qualquer consideração da proximidade espacial dos arcos usados, apenas a verificação de compartilhamentos. O próprio trabalho destaca a semelhança entre seu novo problema e o problema de roteamento de veículos proposto em (NGUEVEU; PRINS; CALVO, 2010), com pequenas diferenças em restrições de compartilhamento de arcos e na função objetivo. Uma referência relevante sobre trabalhos de caminhos dissimilares e sobre as métricas usadas pode ser encontrada em (LIU; MU; YANG, 2016), juntamente com uma abordagem multiobjetivo do problema. Independente da métrica, mais simples ou mais complexa, a alteração de uma rota implica na sua comparação com as demais para a avaliação da nova solução e o cálculo da sua função objetivo.

Uma variação de problemas de roteamento com pontos de coleta e entrega de mercadoria, na qual as mercadorias podem ser transportadas em diferentes veículos, também apresenta a interdependência entre rotas (MITROVIĆ-MINIĆ; LAPORTE, 2006; QU; BARD, 2012). A transferência ocorre em pontos específicos dentro do roteamento, que funcionam como depósitos temporários. A presença dos veículos nesses pontos não é necessariamente síncrona, sendo permitido que um veículo deixe a carga para ser coletada posteriormente pelo outro. Proposta semelhante, porém voltada ao transporte de passageiros, é apresentada em (MASSON; LEHUÉDÉ; PÉTON, 2014). Nesse tipo de problema, a interdependência está ligada a restrições de ordem de chegada aos pontos de transferência, estando ligada à factibilidade das soluções.

Situações de roteamento de equipes de socorro, como bombeiros, e/ou atendimento médico, podem levar a restrição mais dura, de presença simultânea em um vértice. Isso ocorre quando um atendimento demanda as habilidades de duas ou mais equipes deferentes. Em (WEX; SCHRYEN; NEUMANN, 2013), por exemplo, é considerada a resposta a situações de emergência, algumas atendidas por equipes individuais e outras pela presença concomitante de mais de uma equipe no vértice. Também em (DOHN; KOLIND; CLAUSEN, 2009) e (AFIFI; DANG; MOUKRIM, 2013) é considerada a possibilidade de tarefas a serem realizadas simultaneamente por diferentes equipes. Em (MANKOWSKA; MEISEL; BIERWIRTH, 2014), o roteamento de equipes para atendimento médico domiciliar leva à consideração de tarefas síncronas, como o tratamento de pessoa com necessidades especiais que exigem mais de uma profissional da saúde, ou tarefas assíncronas com interdependência, como a necessidade de que se respeite um intervalo entre duas medi-

cações. Restrições temporais de precedência e sincronização entre pares de veículos são consideradas ainda em (BREDSTRÖM; RÖNNQVIST, 2008).

Uma abordagem diferente, mas também voltada ao atendimento médico domiciliar, é descrita em (FIKAR; HIRSCH, 2015). Nesse trabalho, rotas são construídas para definir os deslocamentos de veículos cuja função é transportar o pessoal responsável pelos atendimentos, mas não há vinculação entre veículo e atendente. O profissional deve ser deixado no ponto de atendimento e depois recolhido, possivelmente por veículos diferentes, e entre os dois momentos deve haver intervalo conforme o tempo necessário para o serviço, o que resulta na interdependência entre as rotas.

Sincronização em função de restrições que tornam obrigatória a presença simultânea em vértices ocorre também em problemas de roteamento de arcos, em aplicações como coleta de lixo, nos casos em que os resíduos recolhidos por veículos menores são transferidos para os maiores durante as viagens (PIA; FILIPPI, 2006; ROSA et al., 2002), ou marcação de estradas, quando os veículos de pintura precisam encontrar caminhões-tanque para abastecimento de tinta (AMAYA; LANGEVIN; TRÉPANIER, 2007, 2010; SALAZAR-AGUILAR; LANGEVIN; LAPORTE, 2013). Mesmo se tratando de roteamentos de arcos, os pontos de encontro nesses problemas se configuram como vértices. Há também problemas em que os arcos devem ser atravessados por dois ou mais veículos concomitantemente, como em rotas que envolvem veículos não-autônomos (como reboques) que devem ser puxados por autônomos (DREXL, 2013), ou na tarefa de remover neve de avenidas com pistas múltiplas (SALAZAR-AGUILAR; LANGEVIN; LAPORTE, 2012).

Uma discussão sobre problemas de roteamento com restrições de sincronização e sobre a interdependência que esse tipo de restrição acarreta nas rotas pode ser encontrada em (DREXL, 2012b). É apresentada uma ampla revisão bibliográfica focada em trabalhos sobre problemas com múltiplas restrições de sincronização, e uma classificação dos problemas define cinco tipos de sincronização: de tarefas, de operação, de movimento, de carga e de recurso. É argumentado ainda que a interdependência entre veículos tem importantes implicações na construção de métodos de solução, o que está de acordo com a argumentação do início desta seção. Não havendo tal interdependência, modificações em soluções podem ser avaliadas considerando parte da solução, seja a parte um conjunto de arcos envolvidos em trocas, sejam as rotas alteradas. Com a interdependência, modificações em uma rota têm seu efeito potencializado pelo impacto em outras, e demandam uma verificação de diversas rotas, e talvez da solução inteira.

Ressalta-se que a interdependência é abordada como resultante de uma sincronização imposta por restrições, estando ligada a questões de factibilidade. Esta limitação de escopo pode ser compreendida por ser esta a forma de sincronização mais presente na literatura. Dos trabalhos listados nessa seção, por exemplo, apenas os de caminhos e rotas dissimilares colocam uma relação esperada entre rotas na forma de objetivo. Para a

presente tese, esta segunda forma de sincronização é mais relevante, por ser a que se manifesta nos problemas propostos. É importante salientar que a complexidade se mantém. Ainda que a modificação em uma solução não tenha impacto na factibilidade (pelo menos em relação ao sincronismo), persiste a necessidade de verificar a relação com todas as outras rotas para avaliar o impacto na função objetivo.

2.3 PROBLEMAS DE OTIMIZAÇÃO MULTIOBJETIVO

Nem sempre é possível modelar um problema real considerando apenas um objetivo a ser otimizado. Muitas vezes, a configuração desejada, além de atender a certos critérios mínimos estabelecidos na forma de restrições, deve também contemplar a melhoria em objetivos diversos de forma simultânea. Quando os objetivos postos não são afins, mas pelo contrário, quando melhorias em um deles causam (potencialmente) prejuízo em outro, o problema se configura como multiobjetivo.

De forma resumida, um problema de otimização multiobjetivo pode ser descrito como o problema de encontrar a solução u^* que otimize m objetivos $\{f_1(u), f_2(u), \dots, f_m(u)\}$, satisfazendo eventuais restrições. O espaço de soluções configura-se, assim, em um espaço m -dimensional.

Havendo objetivos múltiplos e potencialmente conflitantes, é pouco provável que exista uma configuração de solução que apresente simultaneamente os valores ótimos para todas as funções objetivo a serem otimizadas. Esta característica é ilustrada na Figura 2.1, que apresenta uma configuração de um problema com dois objetivos (Obj1 e Obj2), ambos de maximização (na sequência da seção, os exemplos e relações de dominância sempre considerarão funções objetivo de maximização). Os círculos pequenos representam seis vetores (a até f) encontrados pela execução de um método de solução, colocados no plano que representa o espaço de soluções do problema.

A primeira observação importante é a que exclui qualquer possibilidade de otimalidade das soluções b , d e f . Isto se deve ao fato de existirem outras soluções com valores de função objetivo melhores nas duas dimensões do problema. Se estabelece uma relação chamada de dominância de Pareto: a solução b é dominada pela solução a , a solução d pelas soluções a e c , e a solução f pela solução c .

Não é possível verificar qualquer relação de dominância quando consideradas as soluções a , c e e . Em qualquer combinação de pares tomados dentre essas três, sempre uma apresentará valor superior em um objetivo e inferior em outro. Essas soluções são ditas incomparáveis. Como não existem na representação outras soluções que as dominem, elas compõem o conjunto de soluções não dominadas. Portanto, se considerada uma avaliação estritamente quantitativa baseada em todos os valores de função objetivo, a otimalidade deixa de ser um conceito referente a uma solução, e passa a se referir a um

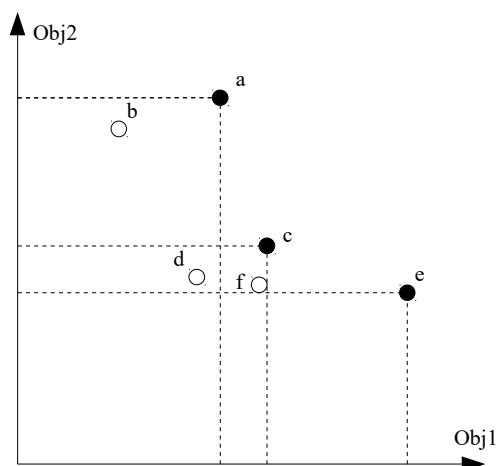


Figura 2.1 – Relação entre as soluções de um problema com dois objetivos de maximização.

conjunto, composto por todas as soluções não dominadas por outras, e conseqüentemente incomparáveis entre si.

A relação de dominância pode ser definida de maneira mais formal: dadas duas soluções u e v , diz-se que u domina v ($u \succ v$) se $\forall i \in [1..m]$ tem-se $f_i(u) \geq f_i(v)$ e se $\exists i \in [1..m]$ tal que $f_i(u) > f_i(v)$. Ou seja, u domina v se u é no mínimo tão bom quanto v em todos os objetivos e é estritamente melhor em pelo menos um. As relações de dominância ilustradas na Figura 2.1 são ainda mais evidentes, pois as soluções dominadas são inferiores em todos os objetivos, no que é conhecido como dominância estrita. Diz-se que u domina estritamente v , ou $u \succ \succ v$, se $\forall i \in [1..m]$ tem-se $f_i(u) > f_i(v)$.

Uma solução u^* é chamada de solução eficiente ou solução ótima de Pareto caso seja impossível achar, em todo o espaço de soluções, outro vetor que a domine. O conjunto de todas as soluções não dominadas do espaço de soluções (ou o conjunto de todas as soluções eficientes) é o conjunto ótimo de Pareto, também chamado de fronteira de Pareto ou fronteira verdadeira.

Em aplicações práticas, é necessária a definição de uma configuração de solução a ser executada, o que passa pela compreensão da importância de cada objetivo e a conseqüente definição da troca (*trade-off*) aceitável: quanto é possível ceder em cada objetivo de modo a obter uma solução que atenda satisfatoriamente a todos. A limitação do conjunto a uma solução pode ser feita diretamente pelo método de solução ou por um decisor humano a partir de um conjunto resultante. Três técnicas são listadas em (JAFFRÈS-RUNSER; GORCE; COMANICIU, 2008).

A primeira técnica, identificada como busca progressiva, prevê a inclusão de um decisor humano no processo de busca. A noção de importância dos objetivos e decisões sobre as trocas aceitáveis nos valores das funções objetivo são inseridas no sistema através da interação do decisor. Tal interação pode ocorrer de diversas formas, como pela

demonstração de preferências por algumas soluções em etapas intermediárias (DEB et al., 2010) ou pela definição iterativa de valores de referência para as dimensões do problema (THIELE et al., 2009). É inegável a exigência de um conhecimento do decisor sobre as características esperadas e possíveis. A vantagem principal dessa abordagem é evitar o consumo de recursos computacionais na geração, armazenamento e avaliação de soluções que certamente não serão de interesse do decisor.

A segunda forma alternativa é conhecida como busca *a priori*, e se estabelece através da combinação das diversas funções objetivo em apenas uma. A forma mais comum dessa composição é através de um somatório ponderado das funções objetivo do modelo original, com a ponderação definida pelo decisor. A função objetivo assume o formato definido na Equação (2.1), em que $\mathbf{f}(x)$ corresponde a combinação convexa dos objetivos $f_i(x)$, $i = 1, \dots, m$ ($\sum_{i=1}^m w_i = 1$).

$$\mathbf{f}(x) = \sum_{i=1}^m w_i \cdot f_i(x) \quad (2.1)$$

Resta, então, a técnica de solução *a posteriori*, construída sobre o conceito de dominância de Pareto. Como não é fornecida informação sobre preferências ou parâmetros de peso aos objetivos, a busca é feita de modo a oferecer como seu resultado um conjunto de solução não dominadas, que estabeleçam uma fronteira que se aproxime ao máximo da fronteira verdadeira. Por essas características, esta técnica não comporta a simples adaptação de métodos de objetivo único, uma vez que a busca deve focar não na obtenção de uma solução, e sim na obtenção de um conjunto. Sobre este conjunto o decisor deve, a posteriori, trabalhar para escolher a solução a ser aplicada.

A busca por um conjunto de soluções é natural para métodos populacionais, pois o avanço da busca se dá não sobre uma solução apenas, mas sobre uma população de soluções. São favorecidas a exploração concomitante de diversos pontos do espaço de soluções e a orientação da busca por métricas baseadas na verificação de dominância entre soluções, características importantes para a geração de fronteiras correntes.

Os algoritmos PAES (*Pareto Archived Evolution Strategy*) (KNOWLES; CORNE, 2000), SPEA2 (*Strength Pareto Evolutionary Algorithm II*) (ZITZLER; LAUMANN; THIELE, 2002) e NSGA-II (*Nondominated Sorting Genetic Algorithm II*) (DEB et al., 2002), contemporâneos do início do século, são demonstrativos do uso de métodos baseados em algoritmos genéticos na solução de problemas multiobjetivo. Todos eles, mas em especial o último, se tornaram referências, e apresentam estratégias e estruturas que inspiram ou são usados, completos ou em parte, até os dias de hoje. O método NSGA II será descrito em detalhes na Seção 2.4.3, juntamente com outros métodos utilizados na tese.

Scatter Search, outro método populacional, também dá origem a um método multiobjetivo genérico, chamado AbYSS (NEBRO et al., 2008). Mas existem adaptações também de métodos não populacionais, como por exemplo Busca Tabu (JAFFRÈS-RUNSER;

GORCE; COMANICIU, 2008), *Simulated Annealing* (BANDYOPADHYAY et al., 2008) (LIU et al., 2014) e GRASP com *path-relinking* (MARTÍ et al., 2015).

No caso de algoritmos populacionais, uma questão que surge é a forma de manter as soluções não dominadas encontradas. O NSGA II, por exemplo, mantém as soluções não dominadas incluídas na população, e as soluções eficientes se perpetuam naturalmente por um processo de elitismo. Mas outros métodos mantêm um conjunto separado da população (em métodos não populacionais, tal conjunto é uma necessidade natural), identificado como arquivo, para armazenar o conjunto de soluções encontradas no processo e que melhor aproxima a fronteira ótima de Pareto.

É possível observar que o tamanho potencialmente grande da fronteira de Pareto pode exigir a manutenção de arquivos de tamanho considerável. Em razão disso, alguns métodos de solução apresentam estratégias para descartar soluções não dominadas e manter o tamanho do arquivo dentro de certos limites. Mas estratégias como essas não são unanimidade. Uma série de considerações sobre problemas relacionados a restrições nos arquivos são apresentadas em (FIELDSEND; EVERSON; SINGH, 2003), como a possibilidade de remoção dos extremos e de retração na fronteira em função de clusterização, desperdício de tempo computacional com redescoberta de soluções já descartadas, ou oscilação pela inclusão na fronteira de soluções dominadas por soluções anteriormente descartadas.

Não só a capacidade de armazenamento do arquivo é fundamental. Durante um procedimento de busca, a fronteira representada no arquivo está sujeita a constantes alterações, além de um número ainda maior de comparações. Por isso, uma estrutura de dados utilizada na implementação de um arquivo deve oferecer métodos eficientes para verificar se uma nova solução é ou não dominada, inserir a nova solução no arquivo caso seja não dominada, verificar se uma nova solução domina uma ou mais soluções no arquivo e remover do arquivo soluções dominadas por nova solução. Árvores *Quad-tree* (MOSTAGHIM; TEICH; TYAGI, 2002) são uma estrutura popular, e várias outras alternativas são listadas em (ALTWAIJRY; MENAI, 2012). Problemas com apenas dois objetivos possuem propriedades especiais que permitem o armazenamento do arquivo em uma estrutura de árvore binária (BERRY; VAMPLEW, 2006), mais simples que as árvores necessárias quando o número de objetivos é maior.

É necessário ainda fazer referência às formas como um conjunto de aproximação é avaliado. A qualidade da fronteira obtida através da execução de um método de solução não pode ser mensurada apenas pela geração de elementos eficientes ou próximos. É preciso também que a fronteira apresentada tenha diversidade e representatividade coerente com a fronteira real. A literatura sobre otimização multiobjetivo apresenta diversas métricas para quantificar a qualidade da solução, e existem artigos que sumarizam algumas dessas alternativas e discutem sua qualidade, como (HELBIG; ENGELBRECHT, 2013) e (JIANG et al., 2014). Duas dessas métricas serão descritas aqui por serem as utilizadas

no Capítulo 5 para a avaliação dos conjuntos de aproximação.

A primeira forma é a avaliação através do amplamente utilizado indicador de hipervolume, introduzida no contexto multiobjetivo por (ZITZLER; THIELE, 1999). Esta métrica é dada pelo hipervolume total entre uma fronteira de aproximação não dominada e um ponto de referência, O valor representa a quantidade do espaço de objetivos que é dominado pelas soluções na fronteira, conforme a Figura 2.2, de modo que as melhores soluções apresentam valores maiores. Quando calculado para um problema biobjetivo, o indicador de hipervolume pode ser interpretado como uma área.

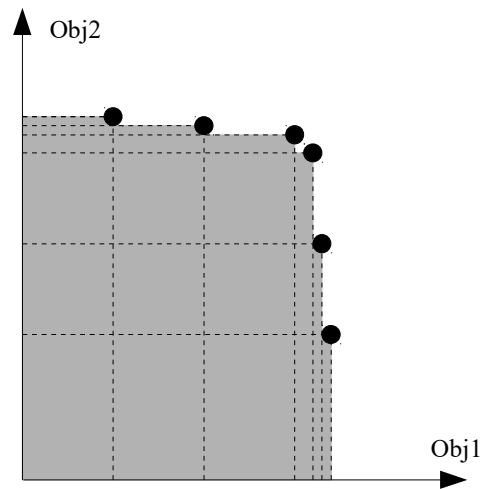


Figura 2.2 – Exemplo de hipervolume. Os pontos representam as soluções no conjunto de aproximação. O ponto de referência é (0, 0). A área cinza é o hipervolume, dado pela união das porções dominadas por cada solução.

A segunda métrica é normalmente referida como $D1_R$ (ISHIBUCHI; YOSHIDA; MURATA, 2003; KNOWLES; CORNE, 2002) e foi proposta em (CZYZAK; JASZKIEWICZ, 1998). Esta métrica mede a convergência de um conjunto de aproximação ND em relação a uma fronteira de referência Ref . Se soluções forem observadas como pontos no espaço biobjetivo, a métrica pode ser considerado como a distância euclidiana média entre cada solução em Ref a sua solução próxima em ND . Assim, valores menores para $D1_R$ indicam melhores conjuntos de soluções. O valor $D1_R$ é definido na Equação (2.2).

$$D1_R = \frac{1}{|Ref|} \cdot \sum_{x' \in Ref} \min\{d(x', x) | x \in ND\} \quad (2.2)$$

A distância entre duas soluções é dada por

$$d(x', x) = \sqrt{[f_1^*(x') - f_1^*(x)]^2 + \dots + [f_m^*(x') - f_m^*(x)]^2}$$

onde f_i^* é o valor normalizado da função objetivo i . A métrica foi usada nessa tese com a mesma normalização utilizada em (ISHIBUCHI; YOSHIDA; MURATA, 2003),

(FRAMINAN; LEISTEN, 2008) e (LEISS et al., 2011), definida pela Equação (2.3), em que os valores f_i^{min} e f_i^{max} representam, respectivamente, os valores mínimos e máximos encontrados para a função objetivo i entre as soluções em Ref .

$$f_i^*(x) = \frac{f_i(x) - f_i^{min}}{f_i^{max} - f_i^{min}} \times 100 \quad (2.3)$$

A Figura 2.3 ilustra a relação entre a fronteira de referência e a fronteira de aproximação, conforme a definição da métrica $D1_R$.

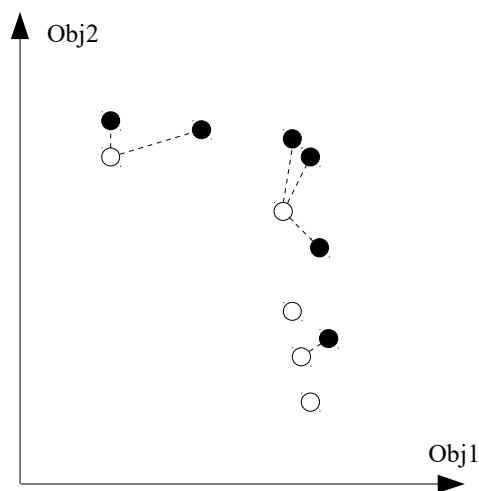


Figura 2.3 – Exemplo da relação entre as soluções das fronteiras de aproximação, representadas por círculos vazados, e de referência, representadas por círculos pretos. Cada solução da fronteira de referência é associada à solução mais próxima da fronteira de aproximação, e a distância é representada por uma linha pontilhada.

Por fim, a Figura 2.4 demonstra a importância de se utilizar mais de uma métrica na avaliação de um conjunto de aproximação. Cada um dos conjuntos apresentados leva vantagem em um dos indicadores. A primeira fronteira exibe uma melhor distribuição, enquanto a segunda cobre uma maior porção do espaço de soluções. Isso demonstra que as métricas se complementam pela avaliação de características diferentes.

2.4 MÉTODOS DE SOLUÇÃO

A presente seção abre com a descrição dos algoritmos de Busca Local 2-opt e 3-opt. Na sequência, são apresentados métodos de solução para problemas de otimização, o primeiro concebido para problemas de um objetivo, e os outros dois para problemas multiobjetivo. Todos esses métodos são importantes pois fundamentam os métodos de solução implementados para os problemas originais descritos nos próximos capítulos.

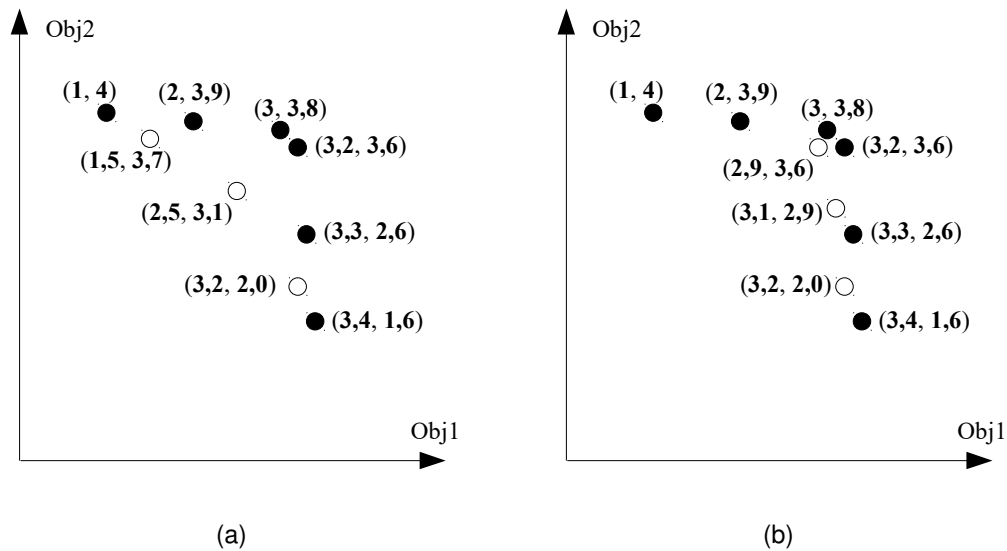


Figura 2.4 – Demonstração da complementariedade das métricas, que avaliam características diferentes dos conjuntos de aproximação. O conjunto representado em (a) tem hipervolume de 10,05 e $D1_R$ de 3,90, enquanto o conjunto representado em (b) tem hipervolume de 11,22 e $D1_R$ de 4,22.

2.4.1 Algoritmos de Busca Local 2-opt e 3-opt

Na implementação de Busca Local, uma solução incumbente é substituída iterativamente por solução melhor encontrada em sua vizinhança. A vizinhança é um conjunto de soluções que são geradas efetuando sistematicamente modificações, conhecidas como movimentos, na solução incumbente. A busca é interrompida quando a vizinhança não incluir solução que represente melhora.

Os algoritmos de Busca Local 2-opt e 3-opt seguem essa estratégia e definem movimentos que se caracterizam pela troca de arestas em uma rota. No caso do 2-opt, são removidas duas arestas, de modo que a rota é partida em dois segmentos, sendo necessária a inserção de duas novas arestas para sua recomposição em uma solução nova.

Para fins de ilustração de um movimento, será assumido que foram selecionadas para remoção as arestas (i, j) e (k, l) . Se esses vértices apareciam nessa mesma ordem na rota antes do movimento, é claro que os segmentos resultantes da partição da rota ligam o vértice j ao vértice k e o vértice l ao vértice i . Para a recomposição da rota, três alternativas podem ser consideradas: a primeira contempla a inserção das arestas (i, j) e (k, l) , o que significa recompor a rota original; a segunda contempla a inserção das arestas (i, l) e (k, j) , o que resulta em uma solução inválida pois são fechados dois ciclos independentes; e a terceira, ilustrada na Figura 2.5, contempla a inserção das arestas (i, k) e (j, l) , representando a única alternativa que resulta em rota válida e diferente da original. O movimento que gera modificação implica na inversão do sentido de um dos segmentos, o que é uma dificuldade na aplicação em roteamentos sobre grafos assimétricos.

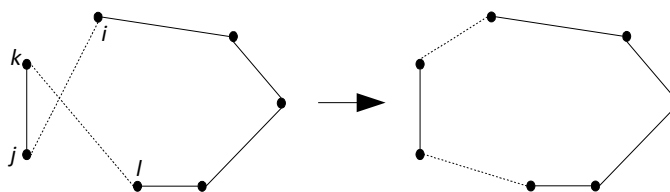


Figura 2.5 – Troca de arcos no movimento da Busca Local 2-opt.

É interessante salientar ainda a necessidade de que cada segmento gerado na partição da rota seja composto por pelo menos uma aresta, ou seja, as arestas removidas não podem ser adjacentes. Isso é importante para que haja diferença entre a rota da solução original e a rota da solução vizinha, já que se um segmento for constituído apenas por um vértice, a única forma de combinação resulta na própria rota da solução original.

A vizinhança completa do algoritmo de Busca Local 2-opt é definida quando são considerados para remoção todos os possíveis pares de arestas, desde que não adjacentes.

A vizinhança 3-opt segue o mesmo princípio, porém com a remoção de três arestas e a recomposição da rota a partir dos três segmentos resultantes. As arestas possíveis de inserir para a recomposição da rota indicam oito combinações possíveis, porém uma delas resulta em rota idêntica à original. Além disso, três outras combinações indicam a reinserção de uma das arestas e a recombinação pela inserção de apenas duas arestas novas. Essas combinações resultam em rotas que também são geradas pelos movimentos 2-opt, de modo que essa vizinhança é englobada pela 3-opt. As combinações restantes resultam nos quatro movimentos ilustrados na Figura 2.6.

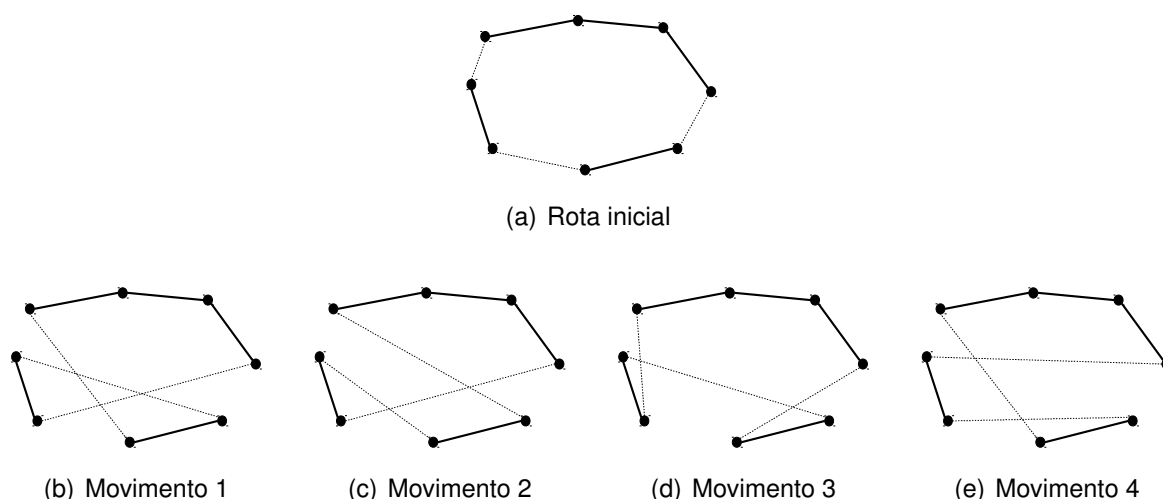


Figura 2.6 – Representação dos quatro movimentos que definem a vizinhança da Busca Local 3-opt.

É indicada ao leitor, em caso de interesse, a descrição detalhada dos algoritmos

2-opt e 3-opt disponível em (JOHNSON; MCGEOCH, 1997).

2.4.2 Algoritmos Genéticos

O nome *Algoritmos Genéticos* identifica um vasto conjunto de variações sobre um método heurístico inspirado em processos evolutivos verificados na natureza e voltado para problemas de otimização combinatória. A origem dos algoritmos evolutivos pode ser atribuída às simulações computacionais de sistemas genéticos realizadas nas décadas de 1940 e 1950, mas o trabalho de John Holland na década de 1970 (HOLLAND, 1975) é tido como pioneiro no que hoje se conhece como as metaheurísticas classificadas como Algoritmos Genéticos.

A ideia básica de um Algoritmo Genético (AG) pode ser resumida na busca por uma solução de boa qualidade a partir da evolução de uma população de soluções. Cada nova geração é construída a partir de operadores genéticos aplicados a soluções da população da geração anterior, selecionados de modo a favorecer aquelas que apresentam melhor qualidade de acordo com um valor de aptidão. O processo pode ser descrito de maneira genérica conforme o Algoritmo 1.

Algoritmo 1 Pseudocódigo do LSGA.

Inicializar uma população
 Avaliar os indivíduos da população por sua qualidade
Repita até que critério de parada seja satisfeito
 Selecionar indivíduos para reprodução
 Aplicar operadores genéticos sobre os indivíduos selecionados, gerando novos indivíduos
 Definir nova população, com indivíduos recém gerados e da população antiga
 Avaliar os indivíduos da população por sua qualidade
Fim do Repita até que critério de parada seja satisfeito

Para a implementação de um Algoritmo Genético, é necessário definir a forma como uma solução deve ser representada, estabelecendo uma abstração que codifique suas características e permita a aplicação de operadores genéticos. A forma mais comum de representação é em cromossomos implementados através de sequências de valores binários ou de outros tipos numéricos. Cada posição na sequência é chamada de gene, e o valor que ele assume é chamado de alelo.

Os operadores genéticos são os responsáveis por gerar diversidade na população, aplicando transformações a cromossomos. Em geral, são utilizados operadores de recombinação ou cruzamento e operadores de mutação. O operador de recombinação implica na seleção de dois indivíduos de uma população e na divisão de seus cromossomos em partes a serem recombinadas. Como resultado, tem-se dois novos indivíduos, compostos por porções de ambos os indivíduos originais.

Existem diversas alternativas de implementação para os operadores de recombinação. Uma opção simples é a definição do número e das posições dos pontos de cruzamento, que definem o número e a forma das divisões efetuadas em cada cromossomo. Alguns operadores mais complexos são estabelecidos para o tratamento das inactibilidades que podem ser decorrentes da recombinação de partes com origens em indivíduos diversos. Mas esses operadores em geral apresentam uma especificidade relacionada às restrições que definem a factibilidade, o que os torna próprios para um problema ou para um conjunto de problemas.

Já o operador de mutação pode ser aplicado aos indivíduos resultantes da recombinação ou a indivíduos selecionados especificamente. Em ambos os casos, a mutação se caracteriza pela alteração de um ou mais alelos.

A seleção dos indivíduos a serem submetidos aos operadores se dá de modo a favorecer a reprodução dos mais aptos. O conceito de aptidão está relacionado à qualidade da solução representada no indivíduo, geralmente considerando a função objetivo de cada problema de otimização. Duas estratégias de seleção se destacam, ambas baseadas no valor de aptidão: torneio e roleta. O torneio se caracteriza pela escolha aleatória de dois ou mais indivíduos e pela seleção daquele entre eles que possui a melhor aptidão. Já a roleta se caracteriza pela atribuição de uma probabilidade de seleção derivada da aptidão do indivíduo (melhor valor de aptidão, maior probabilidade), e a seleção é dirigida por essa probabilidade.

É interessante perceber que os operadores são aplicados sobre os cromossomos, ou seja, sobre a representação abstrata das soluções. Mas a seleção se dá sobre valores de aptidão que são calculados a partir da qualidade que as soluções representadas apresentam em relação ao problema de otimização. Por isso, procedimentos de codificação e decodificação são necessários para a representação na forma de cromossomo e para a avaliação de uma solução representada.

O procedimento iterativo do Algoritmo Genético agrupa todas essas características descritas, para evoluir populações de indivíduos por gerações sucessivas. Uma população inicial é construída com soluções geradas de forma aleatória ou através de outro método heurístico. Essa é a população corrente da primeira iteração.

Cada iteração se caracteriza por repetidas aplicações de operadores genéticos sobre indivíduos selecionados dentre os componentes da população corrente. São gerados indivíduos novos, suficientes para formar uma população que será a corrente na iteração seguinte. A forma de lidar com os indivíduos novos na formação da nova população também oferece alternativas. É possível, por exemplo, acrescentar à população corrente todos os indivíduos novos gerados e definir a população da próxima geração com aqueles de maior qualidade. Nesse caso, a perpetuação das melhores soluções entre as gerações é natural.

Outra forma passa pela simples substituição de populações, ou seja, os indivíduos

gerados pelos operadores genéticos formam a população da nova geração, e a população original é descartada. Nesse caso, torna-se necessário armazenar a solução incumbente à parte da população, ou implementar uma estratégia de elitismo que garanta que os melhores indivíduos de uma geração estejam presentes na seguinte.

A literatura de otimização combinatória apresenta um número grande de variações e aplicações de algoritmos evolutivos, e de algoritmos genéticos em particular. Para maiores detalhes, são sugeridos ao leitor os livros (HAUPT; HAUPT, 2004) e (SIMON, 2013).

2.4.3 NSGA II

O NSGA-II (DEB et al., 2002) é um Algoritmo Genético que apresentou características inovadoras e se mantém até os dias de hoje como um método importante para a solução de problemas multiobjetivo. Várias dessas características foram usadas na construção do MOGLS proposto para a solução do BARP-PD. Além disso, trata-se do MOGA implementado para comparação com os resultados do MOGLS. A opção pelo NSGA II se deu principalmente por dois motivos: por ser um algoritmo que apresenta várias etapas que são usadas no MOGLS apresentado em (ARROYO; ARMENTANO, 2005), que inspirou o método desenvolvido para o BARP-PD, e por ainda hoje ter ampla aceitação e uso (com modificações ou não) na solução de problemas multiobjetivo (SHAHRIARI, 2016; MOUSAVI et al., 2016; MARTÍNEZ-VARGAS et al., 2016).

Nos algoritmos genéticos tradicionais, para problemas com um objetivo, a avaliação dos indivíduos se dá por valores de aptidão (*fitness*), derivados diretamente dos valores de função objetivo. Esta abordagem não é adequada para a solução com técnica *a posteriori* de problemas multiobjetivos, pois, conforme descrito na Seção 2.3, a comparação não é trivial e não se deseja uma solução final, mas um conjunto. É necessária uma forma de avaliação em que a aptidão de uma solução seja obtida a partir da sua contribuição para que o conjunto de aproximação final seja representativo da fronteira verdadeira de Pareto.

É nesse ponto que o NSGA II é mais inovador, pois apresenta (i) uma forma de organização das soluções em fronteiras com níveis de dominância, (ii) uma métrica para estimar a densidade de soluções no entorno de uma solução particular, e (iii) um operador de comparação a ser usado em procedimentos de seleção.

A organização de uma população em fronteiras não dominadas tem como resultado a classificação das soluções por níveis de dominância: todas as soluções não dominadas por nenhuma outra dentre as que formam a população são consideradas de nível 0 (é a fronteira de aproximação corrente), as soluções dominadas apenas por soluções de nível 0 são consideradas de nível 1, e assim por diante. O trabalho que introduz o NSGA II apresenta um algoritmo para organizar as soluções dessa forma, chamado *Fast Nondominated Sorting Approach*. Mas (JENSEN, 2003) apresenta método mais eficiente para a

realização da classificação, que foi usado no presente trabalho e é a esse método que se faz referência sempre que essa classificação for tratada.

A segunda definição original é a medida da densidade de povoamento em torno de uma solução, considerada a fronteira (nível) da qual a solução particular faz parte. Chamada de distância de povoamento (*crowding-distance*), ela é calculada para cada solução como o somatório das distâncias entre as suas soluções vizinhas. As soluções vizinhas são as soluções anterior e posterior dentro da mesma fronteira, e devem ser encontradas para todas as dimensões do problema, de modo que a população deve ser ordenada por todas as dimensões.

O nível de dominância e a distância de povoamento são combinados em um operador de comparação usado para a seleção dos indivíduos, que é feita por torneio binário. Como a fronteira de aproximação final deve ser formada por indivíduos não dominados, na comparação de dois indivíduos quando da realização de um torneio, a preferência é dada para aquele de nível mais baixo. Havendo empate no nível, é selecionado aquele indivíduo de maior distância de povoamento, de modo a favorecer a distribuição de soluções por toda a extensão das fronteiras.

A sequência de execução do método NSGA-II o identifica claramente como um Algoritmo Genético. A cada ciclo de execução, uma população de tamanho N serve de base para a geração de uma descendência de igual tamanho. Da união dos dois conjuntos (tamanho total $2N$) são selecionados N indivíduos que formarão a população seguinte. Os indivíduos descartados são aqueles das piores fronteiras em relação aos níveis de dominância. Caso necessário, mais alguns indivíduos de pior (menor) valor de distância de povoamento são eliminados da pior fronteira remanescente, até que o descarte resulte em exatamente N elementos. O elitismo é garantido pela preservação dos indivíduos da população geradora não dominados pela população descendente, pois todos não dominados ficarão da fronteira de nível 0. Definida a nova população, o ciclo é repetido.

2.4.4 Método MOGLS de Armentano e Arroyo (2005)

O método descrito em (ARROYO; ARMENTANO, 2005) incorpora uma Busca Local a um procedimento evolutivo multiobjetivo, em um método identificado como um MOGLS.

O processo evolutivo inicia com a classificação da população de acordo com níveis de dominância, como no NSGA II. É mantido um arquivo à parte da população com todos os indivíduos não dominados encontrados no processo, e após a classificação esse arquivo é atualizado com as soluções não dominadas da fronteira de mais baixo nível. Cada indivíduo tem também a sua distância de povoamento calculada, mas o uso de seleção por roleta implica na atribuição de um valor de aptidão composto, que agrupe as duas grandezas e permita a definição de probabilidade de seleção. Para essa composição, a distância

de povoamento é normalizada e somada a um valor relativo ao nível de dominância. O intervalo entre valores atribuídos a níveis sucessivos é maior que o intervalo para o qual a distância de povoamento é normalizada, de modo que variações de nível sejam mais significativas para a aptidão de cada solução.

Uma vez definida a aptidão, a seleção por roleta é realizada, e novos indivíduos são criados por recombinação e mutação. Os novos indivíduos são avaliados de acordo com todas as funções objetivo e combinados com um conjunto de elite selecionado dentre os indivíduos do arquivo, resultando na nova população. Outra iteração do processo evolutivo é iniciada.

A Busca Local é inserida nesse processo a cada β gerações. Nessas gerações específicas, as soluções não dominadas resultantes do processo evolutivo (antes da combinação com a elite) são selecionadas como candidatas para a Busca Local. Mas só formam o conjunto S de soluções realmente submetidas à Busca Local aquelas soluções que não tenham passado anteriormente pelo processo. Tal restrição exige a manutenção de um conjunto L composto por todas as soluções já submetidas à busca, e tem por objetivo evitar a perda de recurso computacional avaliando vizinhanças já exploradas.

Selecionados os indivíduos, a Busca Local começa com a atualização da lista L , de modo a incluir os indivíduos em S . Se o número de indivíduos em S é maior que uma largura máxima para a busca, o conjunto é dividido em dois, um com indivíduos que terão a vizinhança explorada e outro com indivíduos descartados momentaneamente, mas que voltarão a ser considerados na próxima iteração da Busca Local.

Exploradas as vizinhanças, é formado um conjunto com os indivíduos não dominados encontrados. Esse conjunto é utilizado para atualizar o conjunto não dominado corrente da Busca Local e combinado com os indivíduos que foram deixados de lado anteriormente para formar o conjunto S a ser explorado em nova iteração da Busca Local. Após um número de iterações definido como parâmetro do método MOGLS, o conjunto não dominado corrente é retornado como resultado da Busca Local e inserido na população em substituição aos indivíduos que foram enviados para a Busca Local.

2.5 RESUMO DO CAPÍTULO

Este capítulo apresentou a fundamentação de literatura sobre a qual estão construídos os aspectos originais da tese, que serão descritos nos capítulos seguintes.

Inicialmente, foram listadas variações de problemas de roteamento encontradas na literatura, criadas pela adição de características novas a problemas anteriormente trabalhados. O objetivo é ilustrar como cada atividade de transporte tem suas particularidades e como a essas particularidades podem estar relacionadas a qualquer peça envolvida no procedimento. Além disso, foram apresentadas algumas variações que também foram in-

corporadas aos problemas propostos na tese e que definem a particularidade de cada um.

Na seção seguinte continuou sendo feita a apresentação de variações de problemas de roteamento, mas o foco foi colocado em um conjunto restrito de problemas em que as rotas, por diferentes razões, exibem interdependência.

Na Seção 2.3, revisou-se os conceitos, características e abordagens de solução para problemas multiobjetivo, enquanto na Seção 2.4 foram descritos alguns métodos que contribuíram para a solução dos problemas novos.

3 MÉTRICA PARA QUANTIFICAR A DISPERSÃO ENTRE DOIS VEÍCULOS

A inter-relação entre rotas, cerne dos problemas de roteamento apresentados no presente trabalho, ocorre em função do acompanhamento constante do distanciamento relativo entre os veículos envolvidos. De modo a avaliar esse distanciamento, referido como dispersão, é proposta uma métrica baseada na cálculo contínuo das distâncias instantâneas entre todos os pares de veículos.

A métrica foi definida de modo a avaliar soluções construídas sobre grafos do tipo $G = (N, A)$, em que $N = \{0, 1, \dots, n\}$ representa um conjunto de $n + 1$ nós, e A representa um conjunto de arcos que conectam nós em N . Cada arco $a_{ij} \in A$ deve ser associado um valor c_{ij} que representa o tempo necessário para que seja percorrido completamente. Os veículos, representados no conjunto $K = \{1, 2, \dots, m\}$, $m \geq 2$, partem no instante 0 do nó garagem 0 para realizar m rotas. Dado que os veículos são idênticos, assume-se que eles têm vetores de velocidade com mesma magnitude. É necessário que seja conhecida a localização dos nós, pois é a partir dessa localização que o posicionamento instantâneo dos veículos pode ser inferido. A posição de cada nó $i \in V$ é dada por um par ordenado de coordenadas (x_i, y_i) .

Quando um veículo $k \in K$ percorre um arco a_{ij} , seu vetor de velocidade $\vec{v}_k = \langle v_{xk}, v_{yk} \rangle$ tem a mesma direção do vetor $\langle x_j - x_i, y_j - y_i \rangle$. O deslocamento do veículo ao longo do arco a_{ij} pode ser representado pela seguinte equação, parametrizada pelo tempo t .

$$X_k(t) = x_i + t v_{xk} \quad , \quad Y_k(t) = y_i + t v_{yk} \quad , \quad t \in [0, c_{ij}] \quad (3.1)$$

Assim, se for considerado um par de veículos k e k' viajando ao longo dos arcos a_{ij} e $a_{i'j'}$, como representado na Figura 3.1, a distância euclidiana instantânea entre eles no momento t é calculada conforme a Equação (3.2).

$$d_{kk'}(t) = \sqrt{[X_k(t) - X_{k'}(t)]^2 + [Y_k(t) - Y_{k'}(t)]^2} \quad (3.2)$$

Cada vez que o vetor de velocidade de um veículo sofre alteração, a respectiva equação de deslocamento deve ser atualizada. Os instantes em que tais alterações ocorrem estabelecem marcadores no tempo, e referimo-nos a cada intervalo entre dois marcadores, em que as equações de deslocamento permanecem inalteradas para todos os m veículos, como *fatia de tempo*. A sequência ordenada de marcadores $\mathcal{S} = \{t_\tau : \tau = 0, \dots, nf\}$ define todas as fatias de tempo, numeradas de 1 até f , sendo cada uma é delimitada por dois marcadores consecutivos $[t_{\tau-1}, t_\tau]$. A duração de cada fatia é dada por $T_\tau = t_\tau - t_{\tau-1}$. O valor nf representa o número de fatias de tempo de um roteamento. No roteamento exemplificado na Figura 3.2 existem 13 fatias. Se for considerada uma variação do problema, em que as rotas são abertas e se encerram todas ao mesmo tempo no final

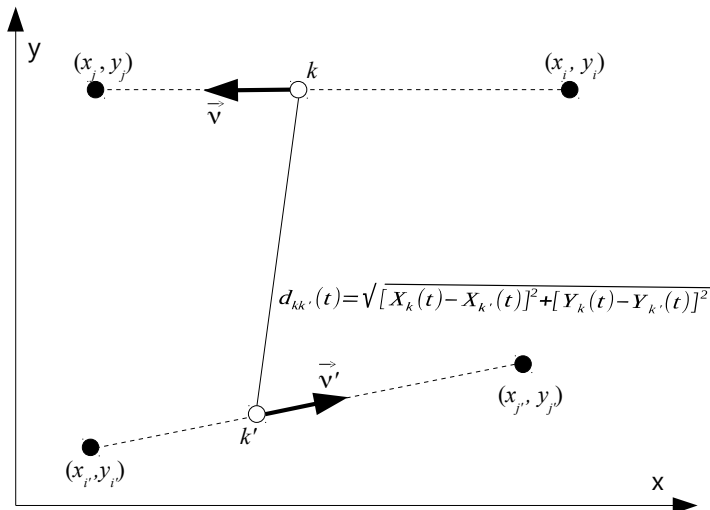


Figura 3.1 – Diagrama que representa a distância instantânea entre dois veículos.

de um turno, as m fatias que são delimitadas pelos instantes de chegada dos veículos de volta à garagem (marcadores 11, 12 e 13 na figura) seriam substituídas por apenas uma, delimitada pelo final do turno. O valor de nf , portanto, varia de acordo com a formulação do problema de roteamento.

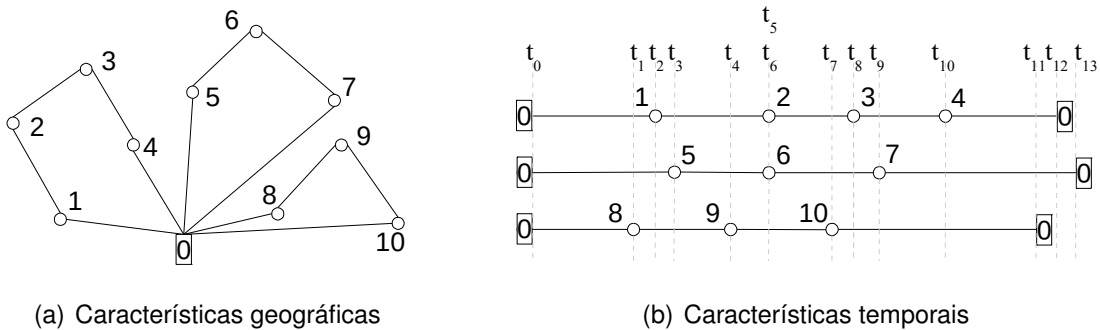


Figura 3.2 – Rotas e as respectivas fatias de tempo. Os diagramas em (a) e (b) apresentam a mesma rota em representações diferentes, a primeira considerando características geográficas, a segunda características temporais. Em (b), as linhas pontilhadas verticais identificam as fatias de tempo.

Na Figura 3.2 é possível observar ainda a ocorrência de uma fatia de tempo com duração igual a 0. Tal situação decorre da existência de dois marcadores de tempo idênticos, t_5 e t_6 , decorrentes de mudança simultânea nos vetores de velocidade de dois veículos. O marcador t_0 representa a partida da garagem.

Se for considerada uma fatia de tempo individual τ , o deslocamento de um veículo k percorrendo um arco a_{ij} durante o intervalo da fatia $[t_{\tau-1}, t_{\tau}]$ é dado pela Equação (3.3), em que $\langle v_{xk\tau}, v_{yk\tau} \rangle$ representa o vetor velocidade do veículo k , com direção $\langle x_j - x_i, y_j - y_i \rangle$, e $(X_k(t_{\tau-1}), Y_k(t_{\tau-1}))$ é a localização do veículo no início da fatia de tempo.

$$X_k(t) = X_k(t_{\tau-1}) + t \nu_{xk\tau}, \quad Y_k(t) = Y_k(t_{\tau-1}) + t \nu_{yk\tau}, \quad t \in [t_{\tau-1}, t_{\tau}] \quad (3.3)$$

Com base nas posições instantâneas de todos os veículos, é possível calcular a distância euclidiana instantânea $d_{kk'}(t)$ entre cada par k e k' . Integrando-se $d_{kk'}(t)$ ao longo do comprimento T_{τ} da fatia τ , é obtida a dispersão $D_{kk'\tau}$ entre os veículos na fatia.

$$D_{kk'\tau} = \int_0^{T_{\tau}} d_{kk'}(t) dt \quad (3.4)$$

A integral na Equação (3.4) é uma integral definida de raiz quadrada de um polinômio quadrático e uma solução analítica pode ser encontrada aplicando substituições trigonométricas.

O cálculo da métrica conforme descrito na Equação (3.4) necessita dos vetores velocidade dos dois veículos envolvidos, mas é possível deduzir uma expressão que utiliza os vetores de deslocamento. Isso pode ser demonstrado considerando os veículos k e k' , cujos vetores velocidade são dados por $\vec{v}_{k\tau} = \langle \nu_{xk\tau}, \nu_{yk\tau} \rangle$ e $\vec{v}_{k'\tau} = \langle \nu_{xk'\tau}, \nu_{yk'\tau} \rangle$ e os vetores deslocamento são $\vec{d}_{k\tau} = \langle d_{xk\tau}, d_{yk\tau} \rangle$ e $\vec{d}_{k'\tau} = \langle d_{xk'\tau}, d_{yk'\tau} \rangle$.

3.1 SUBSTITUIÇÃO DO VETOR VELOCIDADE PELO VETOR DESLOCAMENTO NO CÁLCULO DA DISPERSÃO

Dados $(x_{0k\tau}, y_{0k\tau})$ e $(x_{1k\tau}, y_{1k\tau})$, respectivamente as posições inicial e final do veículo k considerando os limites da fatia τ , os componentes do vetor deslocamento $\vec{d}_{k\tau}$ são dados por $d_{xk\tau} = x_{1k\tau} - x_{0k\tau}$ e $d_{yk\tau} = y_{1k\tau} - y_{0k\tau}$. De forma equivalente, dadas as posições inicial e final do veículo k' na fatia τ , $(x_{0k'\tau}, y_{0k'\tau})$ e $(x_{1k'\tau}, y_{1k'\tau})$, os componentes do seu vetor deslocamento $\vec{d}_{k'\tau}$ são dados por $d_{xk'\tau} = x_{1k'\tau} - x_{0k'\tau}$ e $d_{yk'\tau} = y_{1k'\tau} - y_{0k'\tau}$.

Como os veículos são homogêneos em termos de magnitude da velocidade, $|\vec{v}_{k\tau}| = |\vec{v}_{k'\tau}|$ e $|\vec{d}_{k\tau}| = |\vec{d}_{k'\tau}|$. O tempo de duração da fatia é T_{τ} .

A Equação (3.5) apresenta a integral da Equação (3.4) expandida conforme as definições para os veículos k e k' .

$$D_{kk'\tau} = \int_0^{T_{\tau}} \sqrt{[x_{0k\tau} - x_{0k'\tau} + (\nu_{xk\tau} - \nu_{xk'\tau}) \cdot t]^2 + [y_{0k\tau} - y_{0k'\tau} + (\nu_{yk\tau} - \nu_{yk'\tau}) \cdot t]^2} dt \quad (3.5)$$

Fazendo a substituição de variável $t = T_{\tau} \cdot u$, em que u é adimensional, decorre $dt = T_{\tau} du$, $t = 0 \Rightarrow u = 0$ e $t = T_{\tau} \Rightarrow u = 1$.

A partir da substituição, é possível reescrever a Equação (3.5) conforme apresentado na Equação (3.6).

$$D_{kk'\tau} = T_\tau \cdot \int_0^1 \sqrt{[x_{0k\tau} - x_{0k'\tau} + (\nu_{xk\tau} - \nu_{xk'\tau}) \cdot T_\tau \cdot u]^2 + [y_{0k\tau} - y_{0k'\tau} + (\nu_{yk\tau} - \nu_{yk'\tau}) \cdot T_\tau \cdot u]^2} du \quad (3.6)$$

A multiplicação dos vetores velocidade por T_τ resulta nos vetores de deslocamento, de modo que

$$D_{kk'\tau} = T_\tau \cdot \int_0^1 \sqrt{[x_{0k\tau} - x_{0k'\tau} + (d_{xk\tau} - d_{xk'\tau}) \cdot u]^2 + [y_{0k\tau} - y_{0k'\tau} + (d_{yk\tau} - d_{yk'\tau}) \cdot u]^2} du \quad (3.7)$$

A integral na Equação (3.7) corresponde a distância média entre os veículos no intervalo 0 e T , uma vez que a Equação (3.7) corresponde a Equação (3.4) e é possível escrever a igualdade da Equação (3.8).

$$\int_0^1 \sqrt{[X_k(u) - X_{k'}(u)]^2 + [Y_k(u) - Y_{k'}(u)]^2} du = \frac{\int_0^{T_\tau} \sqrt{[X_k(t) - X_{k'}(t)]^2 + [Y_k(t) - Y_{k'}(t)]^2} dt}{T_\tau} \quad (3.8)$$

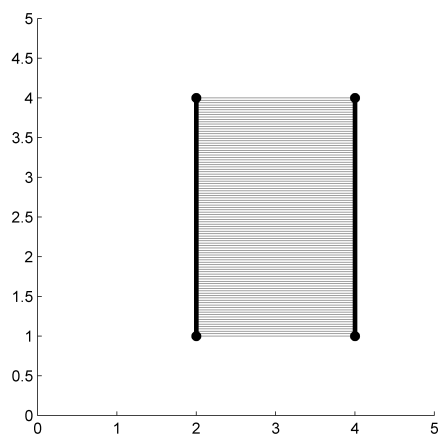
Observe que devido a dispersão entre os veículos k e k' em uma fatia de tempo ser definida em termos da distância média ponderada pelo tempo de duração da fatia, a unidade de grandeza da dispersão é dada em *unidades de distância* \times *unidades de tempo*.

O conjunto de imagens da Figura 3.3 auxilia no entendimento da métrica descrita até aqui. Os valores apresentados junto a cada figura correspondem apenas a distância média, ou seja, sem a ponderação pelo tempo de duração da fatia de tempo.

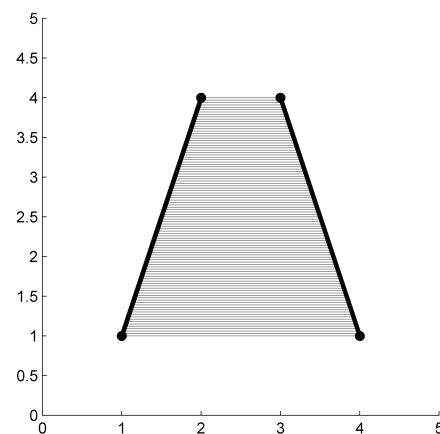
Nos problemas que serão descritos nos Capítulos 4 e 5, essa métrica é utilizada na definição de duas funções objetivo antagônicas, que favorecem a aproximação ou a dispersão entre os veículos. O detalhamento de cada uma dessas funções objetivo será feito nos respectivos capítulos.

3.2 RESUMO DO CAPÍTULO

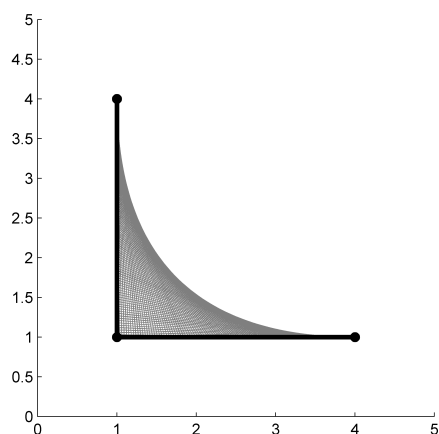
O presente capítulo apresentou uma métrica não linear de dispersão projetada para quantificar o distanciamento entre dois veículos, baseada no monitoramento constante de seu distanciamento. Foi demonstrado que o tempo de execução do roteamento precisa ser dividido em fatias de tempo, de modo que em cada fatia os veículos mantenham seus vetores velocidade inalterados. Foi demonstrado também que a métrica pode ser conside-



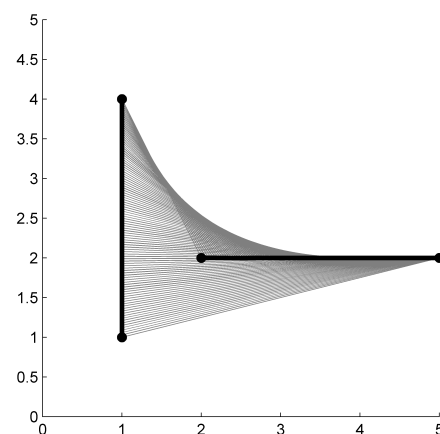
(a) De (2, 1) a (2, 4) e de (4, 1) a (4, 4);
Valor: 2



(b) De (1, 1) a (2, 4) e de (4, 1) a (3, 4);
Valor: 2



(c) De (1, 1) a (4, 1) e de (1, 4) a (1, 1);
Valor: 2,4348



(d) De (1, 4) a (1, 1) e de (2, 2) a (5, 2);
Valor: 2,7588

Figura 3.3 – Quatro exemplos do cálculo da métrica. Linhas claras representam uma discretização das distâncias instantâneas de veículos cujos deslocamentos são representados em preto. São apresentados os pontos de partida e chegada, juntamente com o resultado da integral da Equação (3.7).

rada como a distancia média entre dois veículos durante a fatia de tempo, ponderada pela duração da fatia. A métrica é usada na definição de funções objetivo nos problemas de roteamento descritos nos dois capítulos seguintes.

4 O PROBLEMA DO MÚLTIPLO CAIXEIRO VIAJANTE COM OBJETIVO DE MINIMIZAÇÃO DA DISPERSÃO

Nesse capítulo, é apresentado um novo problema de roteamento chamado Problema de Múltiplos Caixeiros Viajantes com Objetivo de Minimização da Dispersão ou mTSP-D. Nesse problema, que se constitui como variação do Problema do Múltiplo Caixeiro Viajante (mTSP, do inglês *Multi-Travelling Salesman Problem*), os deslocamentos são planejados de modo que os veículos visitem os vértices a eles atribuídos minimizando a sua distância relativa durante toda a execução das rotas.

Para isso, é necessário que as rotas construídas apresentem proximidade tanto espacial quanto temporal. Tal proximidade é avaliada através de uma constante verificação do posicionamento relativo dos veículos, e resulta em uma função objetivo construída a partir da métrica de dispersão descrita no Capítulo 3, que deve ser minimizada. Demonstra-se que o problema é *NP-difícil*, e tal complexidade levou ao uso de uma metaheurística para a busca de soluções de qualidade.

O uso do posicionamento relativo dos veículos na avaliação de soluções confere ao mTSP-D uma característica de sincronismo e interdependência entre as rotas. Na Seção 2.2, foram apresentados problemas de roteamento em que existe uma interdependência entre as rotas. Nos problemas referidos, essa necessidade é conhecida *a priori*, e considerada já no momento da construção das soluções. Mas existem situações reais em que a necessidade de colaboração só pode ser descoberta durante a execução do roteamento, pela percepção de uma situação emergencial quando da chegada a um vértice.

Um exemplo ocorre com as patrulhas policiais, inicialmente planejadas para visitarem locais de alta criminalidade, mas que podem se deparar com situações que exijam rápido reforço. A existência de outras patrulhas em visitas a locais próximos pode suprir essa exigência. Outro exemplo ocorre no envio de equipes de socorro para pontos de atendimento diversos logo após um desastre de grande magnitude. É possível que uma ou mais equipes se deparem com situações mais severas e urgentes do que previsto inicialmente, e para as quais necessitem de apoio rápido de outras.

A necessidade de reforço, especialmente em atividade policial, já foi considerada em outros trabalhos, mas com abordagens diversas, seja pelo posicionamento de helicópteros acionados em caso de emergência percebida por pessoal em terra (URK; MES; HANS, 2013), pela busca por redução da distância entre distritos de patrulha em problema de distritamento (CAMACHO-COLLADOS; LIBERATORE; ANGULO, 2015), ou pelo conceito de cobertura de *backup* (CURTIN; HAYSLETT-MCCALL; QIU, 2007), que se refere à existência de mais de uma equipe capaz de cobrir um determinado local em caso de incidente. Esses trabalhos abordam a questão do reforço a partir do posicionamento de unidades de apoio ou pela definição de áreas de atuação com proximidade ou sobreposi-

ção.

De acordo com o conhecimento do autor, a construção de soluções de roteamento em que todos os veículos atuem no cumprimento de rotas designadas, porém com a manutenção de proximidade espacial e temporal, ainda não foi pesquisada. É nesse espaço que o mTSP-D se enquadra. Busca-se rotas que distribuam os veículos de modo a atuarem como reforço para outra unidade caso necessário, possibilitando a cada uma um rápido acesso às posições correntes das demais.

O mTSP-D não apresenta restrições de sincronização, como a necessidade de que dois veículos cheguem juntos a um vértice ou percorram uma aresta de forma concomitante. A sincronização está implícita no objetivo de proximidade, e a interdependência entre veículos não se configura em uma característica determinante para a factibilidade da solução. Por outro lado, a interdependência espacial e temporal é avaliada de forma muito mais ampla que nos problemas com restrição de sincronização descritos na Seção 2.2, pois deve ser mantida constantemente e sempre entre todos os veículos. Isso potencializa o efeito que mesmo pequenas alterações sobre uma rota podem ter sobre a avaliação global da solução.

É apresentado ainda nesse capítulo um método de solução que pode ser descrito como um Algoritmo Genético com Busca Local (ou LSGA). São apresentadas especializações referentes à representação do cromossomo e aos operadores genéticos, além de vizinhanças para a Busca Local elaborados especificamente para as características do mTSP-D.

O capítulo está organizado como segue. Na Seção 4.1, o problema é descrito de forma detalhada, sendo explicitada a métrica usada para avaliar o posicionamento relativo dos veículos. Na Seção 4.2 é descrito o método de solução usado, dando ênfase aos aspectos do Algoritmo Genético e da Busca Local que são específicos do novo problema. A seguir, na Seção 4.3, são apresentados os resultados computacionais.

4.1 DEFINIÇÃO DO PROBLEMA

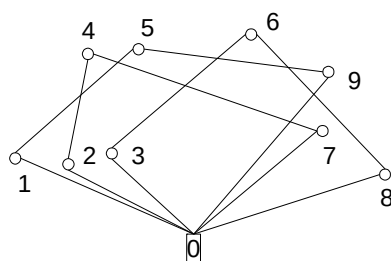
Nesse problema, é considerado um grafo completo $G = (V, E)$, no qual $V = \{0, 1, \dots, n\}$ é o conjunto de $n + 1$ vértices e $E = \{e_{ij} \mid 0 \leq i \leq n, 0 \leq j \leq n\}$ é o conjunto de arestas conectando os vértices em V .

O vértice 0 denota a garagem onde cada veículo começa e termina sua rota, enquanto os outros vértices representam pontos de interesse a serem visitados ou objetivos a serem cumpridos antes de voltar para a garagem. Uma rota é definida como a sequência de vértices visitados por um dos m veículos idênticos do conjunto $K = \{1, 2, \dots, m\}$. Cada vértice $i \in V$ tem associado a si um tempo de serviço s_i e um par ordenado de coordenadas (x_i, y_i) . Um custo c_{ij} é associado a cada aresta $e_{ij} \in E$, e representa o tempo do

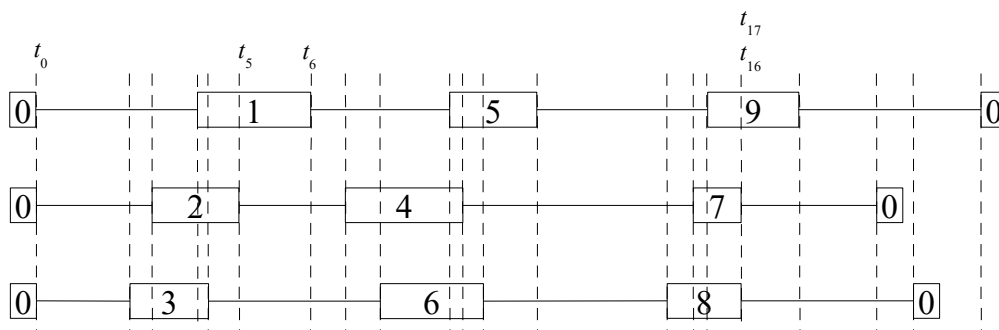
deslocamento de i para j . Como os veículos são idênticos, eles têm todos velocidade de mesma magnitude.

O mTSP-D é definido como o problema de encontrar rotas para todos os m veículos de tal modo que cada vértice é visitado apenas uma vez, enquanto se minimiza uma função objetivo de dispersão.

A Figura 4.1(a) apresenta uma representação gráfica das rotas de uma solução do problema descrito. A definição das fatias de tempo correspondentes pode ser percebida na Figura 4.1(b). É importante destacar que a existência de um tempo de serviço associado aos vértices implica em que tanto a chegada no vértice quanto a posterior partida gerem marcadores que delimitam fatias de tempo. Além disso, enquanto em serviço, um veículo tem velocidade nula.



(a) Características geográficas



(b) Características temporais

Figura 4.1 – Conjunto de rotas e as respectivas fatias de tempo.

Para $n+1$ vértices e m veículos existem $2n+m+1$ marcadores de tempo, sendo que $2n$ corresponde às chegadas e partidas de todos os vértices em $V \setminus \{0\}$, m corresponde às chegadas de todos os veículos na garagem ao final das rotas e 1 corresponde à partida simultânea de todos os veículos da garagem. Portanto, são $2n + m$ fatias de tempo em uma solução.

A função objetivo é concebida de modo a favorecer soluções em que os veículos se deslocam mantendo uma proximidade relativa entre si. Nesse caso, o valor de dispersão atribuído a cada fatia individualmente é determinado pelo par de veículos mais dispersos

naquele intervalo de tempo conforme a métrica de dispersão descrita no Capítulo 3. A dispersão da fatia τ é identificada por c_τ e calculada conforme descrito na Equação (4.1).

$$c_\tau = \max_{\substack{k=1, \dots, m \\ k'=k+1, \dots, m}} \int_0^{T_\tau} d_{kk'}(t) dt \quad (4.1)$$

A dispersão total de uma solução, identificada como C , é dada pelo somatório dos valores individuais de dispersão das $2n + m$ fatias, conforme a Equação (4.2).

$$C = \sum_{\tau=1}^{2n+m} c_\tau \quad (4.2)$$

Para que se obtenha soluções com proximidade entre os veículos, essa função objetivo deve ser minimizada.

A métrica de dispersão impõe um certo grau de sincronização entre os veículos, uma vez que está relacionada com as suas distâncias relativas, favorecendo soluções em que visitam juntos as mesmas regiões do espaço do grafo. Como não existe no problema a implicação de algum tipo de sincronização absoluta, com os veículos presentes concomitantemente no mesmo vértice ou aresta, a sincronização não será uma causa de infactibilidade. Mas ela está presente durante todo o andamento das rotas, e é verificada entre todos os veículos, de modo que a interdependência espacial e temporal é ampliada, e a dificuldade inerente à sincronização é percebida.

Proposição 1. *O mTSP-D é NP-Difícil.*

Demonstração. É demonstrado que uma instância do Problema de Caixeiro Viajante Euclidiano (ETSP) pode ser reduzida a uma instância do mTSP-D.

Uma instância do ETSP é dada pelo grafo $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, no qual $\mathcal{V} = \{1, \dots, n\}$ é o conjunto de vértices, $\mathcal{A} = \{a_{ij} : i, j \in \mathcal{V}, i \neq j\}$ é o conjunto de arestas e cada vértice $i \in \mathcal{V}$ tem coordenadas (x_i, y_i) . A Figura 4.2 ilustra os vértices de uma instância de exemplo.

Para a geração da instância do mTSP-D, um novo conjunto de vértices \mathcal{V}' é criado a partir de \mathcal{V} , de modo que $\mathcal{V}' = \{i + n : i \in \mathcal{V}\}$. O conjunto de arestas \mathcal{A} também é duplicado em um conjunto \mathcal{A}' . Cada vértice cópia $i + n \in \mathcal{V}'$ mantém as coordenadas do vértice original $i \in \mathcal{V}$, ou seja, $x_{i+n} = x_i$ e $y_{i+n} = y_i$. O par de vértices original e cópia é referido como vértices correspondentes, e tal denominação é mantida também para as arestas em \mathcal{A}' e \mathcal{A} .

Conforme a definição do mTSP-D, os veículos partem todos de um mesmo vértice no início das rotas, para onde retornam ao final. É necessário incluir esse vértice garagem na instância do mTSP-D, em processo que começa com a escolha arbitrária do vértice 1 como ponto inicial da rota do ETSP. Segue a criação dos vértices 0 (a própria garagem), $2n + 1$ e $2n + 2$, todos com as mesmas coordenadas dos vértices 1 e $1 + n$. Novos arcos

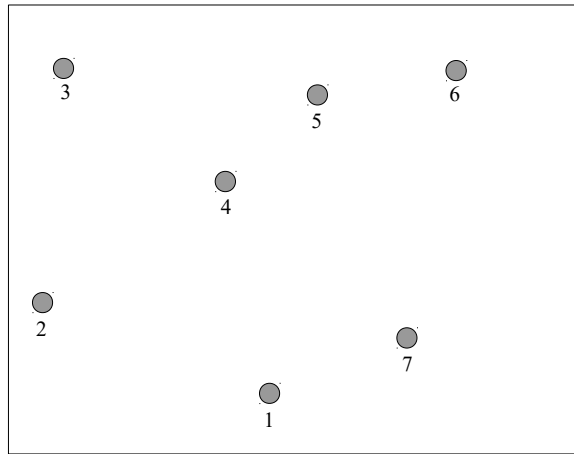


Figura 4.2 – Conjunto \mathcal{V} de vértices de uma instância do ETSP.

(dirigidos) são criados em função dos novos vértices: dois deles ligam a garagem aos vértices 1 e $1 + n$, ou seja, $a_{0,1}$ e $a_{0,n+1}$, e outros dois, $a_{2n+1,0}$ e $a_{2n+2,0}$, ligam os vértices $2n + 1$ e $2n + 2$ à garagem. São criadas ainda arestas ligando todos os vértices em \mathcal{V} ao vértice $2n + 1$, ou seja, $\{a_{i,2n+1} \forall i \in \mathcal{V}\}$, e arestas ligando todos os vértices em \mathcal{V}' ao vértice $2n + 2$, ou seja, $\{a_{i,2n+2} \forall i \in \mathcal{V}'\}$. Os arcos que incidem na garagem são definidos de tal forma que obrigatoriamente os veículos iniciam as rotas pelos vértices 1 e $1 + n$ e retornam para a garagem após visitarem os vértices $2n + 1$ e $2n + 2$.

Como resultado, o grafo referente a uma instância do mTSP-D, construído a partir de uma instância do ETSP, é dado por $G = (V, A)$, em que $V = \mathcal{V} \cup \mathcal{V}' \cup \{0, 2n + 1, 2n + 2\}$ e $A = \mathcal{A} \cup \mathcal{A}' \cup \{a_{0,1}, a_{0,1+n}, a_{2n+1,0}, a_{2n+2,0}\} \cup \{a_{i,2n+1} \forall i \in \mathcal{V}\} \cup \{a_{i,2n+2} \forall i \in \mathcal{V}'\}$. O tempo necessário para percorrer um arco é numericamente igual ao comprimento do arco e o conjunto de veículos é definido como $K = \{1, 2\}$. Dado que são dois veículos na instância e o maior vértice é $2n + 2$, uma solução para esta instância do mTSP-D terá $g = 2 \cdot (2n + 2) + 2$ fatias de tempo, ou seja, $g = 4n + 6$. A Figura 4.3 ilustra a instância do mTSP-D.

Como não existem arestas ligando vértices de \mathcal{V} e \mathcal{V}' , só é possível visitar todos os vértices de $\mathcal{V} \cup \mathcal{V}'$ se um veículo visitar todos os vértices de \mathcal{V} e outro todos os vértices de \mathcal{V}' . É possível perceber que se dois veículos partirem ao mesmo tempo para suas rotas e se mantiverem agrupados e sem atrasos entre si, a dispersão total calculada conforme a métrica do mTSP-D será nula qualquer que seja o sequenciamento dos vértices. Portanto, em uma situação como essa a obtenção de um par de rotas com a menor dispersão total (igual a 0) é trivial.

A imposição de um pequeno atraso δ na partida de um veículo em relação ao outro gera um pequeno distanciamento entre eles mesmo quando percorrendo arestas correspondentes, e resulta em dispersão não nula mesmo para veículos se deslocando agrupados. Esse atraso pode ser imposto atribuindo ao vértice 1 um tempo de serviço igual a zero

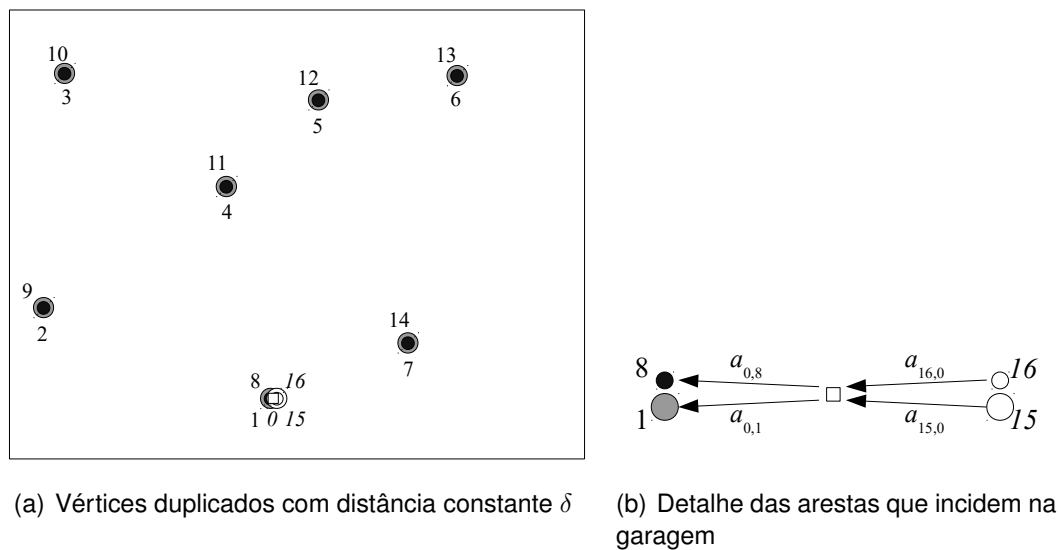


Figura 4.3 – Ilustração de uma instância mTSP-D criada a partir de uma instância do ETSP com sete vértices. Os vértices originais de \mathcal{V} são representados pelos círculos maiores e de cor mais clara, e os vértices cópia de \mathcal{V}' são representados por círculos menores e pretos. O vértice 0 é representado pelo quadrado branco, e os vértices $2n + 1$ e $2n + 2$ por círculos brancos. As figuras que representam os vértices garagem (0), de partida da garagem (1 e 8) e de retorno à garagem (15 e 16) são exibidos com deslocamento entre si para evidenciar sua existência, embora na instância tenham coordenadas idênticas.

e ao vértice $n + 1$ um tempo de serviço igual a δ , de modo que a partir desses vértices o veículo que parte de $n + 1$ estará sempre atrasado em relação ao outro. O mesmo tempo de serviço δ é atribuído a todos os demais vértices, com exceção da garagem e do vértice $2n + 2$, ou seja, $s_i = \delta \forall i \in V \setminus \{0, 1, 2n + 2\}$ e $s_i = 0 \forall i \in \{0, 1, 2n + 2\}$.

O valor de δ deve ser maior do que zero para evitar as situações em que a dispersão é nula, mas deve ser suficientemente pequeno para garantir que os veículos se mantenham agrupados na solução ótima do mTSP-D. Garantido o agrupamento constante, a forma de atribuição dos tempos de serviço faz com que o veículo mais adiantado nunca parta de um vértice antes que o outro veículo tenha chegado no vértice correspondente. De fato, essa partida se dará sempre no exato momento em que o outro veículo conclui seu deslocamento, o que garante que os veículos nunca estarão percorrendo ao mesmo tempo arcos que não sejam correspondentes. O tempo de serviço nulo no vértice $2n + 2$ resulta na eliminação do atraso, e os dois veículos encerram suas rotas com a chegada concomitante à garagem.

A definição do valor de δ é a última etapa da construção da instância do mTSP-D, sendo calculado a partir da avaliação de uma situação hipotética com $\delta = 0$. Em uma instância com essa característica, se os veículos se separarem em algum momento, ou seja, se deixarem de se deslocar agrupados, então obrigatoriamente irão se afastar a partir de dois pontos coincidentes i e $i + n$, o primeiro em direção a um vértice $l \in \mathcal{V}$, com $l \neq i$,

e o segundo em direção um vértice $j + n$, com $j \neq i$ e $j \neq l$. Essa situação é ilustrada na Figura 4.4.

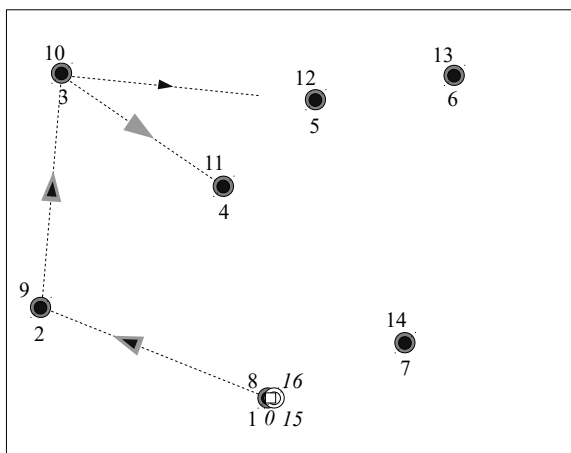


Figura 4.4 – Dois veículos se afastando a partir de um par de vértices correspondentes. Nos trechos de rotas ilustrados, os veículos se mantêm agrupados quando se deslocam dos vértices correspondentes 1 e 8 para os vértices correspondentes 2 e 9, e em seguida para os vértices correspondentes 3 e 10. Nesse ponto, eles deixam de se deslocar agrupados. Um dos veículos se desloca para o vértice 4, enquanto o outro se desloca para o vértice 12. Considerando a descrição genérica do texto, os vértices 3, 10, 4 e 12 são, respectivamente, $i, i + n, l$ e $j + n$.

Tal ocorrência acarreta em pelo menos uma dispersão não nula entre os veículos na primeira fatia de tempo após a separação, cujo valor pode ser calculado conforme descrito no Capítulo 3. Obviamente, essa separação entre os veículos pode ocorrer em qualquer momento no roteamento após os vértices correspondentes 1 e $n + 1$. É necessário avaliar os valores de dispersão para a fatia de tempo imediatamente posterior à separação considerando cada uma das possíveis combinações de i, l , e j , valor que pode ser 0 caso os três vértices sejam colineares. A menor dispersão não nula encontrada define um limitante inferior LB para a dispersão total de qualquer rota em que os veículos se afastam em algum momento.

Em uma configuração com o atraso dado por $\delta = LB/(n \cdot a_{max})$, em que a_{max} corresponde a maior aresta em A e $n \cdot a_{max}$ é um limitante superior para o comprimento da rota do ETSP, a dispersão de uma solução na qual os veículos se deslocam agrupados nunca será superior a LB . Portanto, com esse valor para o atraso, a dispersão total mínima de uma solução continua sendo obtida enquanto eles se mantêm agrupados, mantendo um paralelismo na execução de suas rotas, embora com pequeno atraso. Mas a solução deixa de ser trivial pois a ordenação das visitas passa a ser relevante.

Uma vez que a distância entre os veículos passa a ser não nula mesmo se agrupados, o comprimento da rota é determinante para definir o valor da dispersão total da solução ótima. A demonstração dessa característica determina que a solução ótima de uma instância do mTSP-D, construída da forma descrita a partir de instância do ETSP, re-

presenta também a solução ótima para essa instância original, que pode ser então obtida em tempo polinomial.

A Figura 4.5 permite observar as características temporais de uma solução obtida para a instância do mTSP-D.

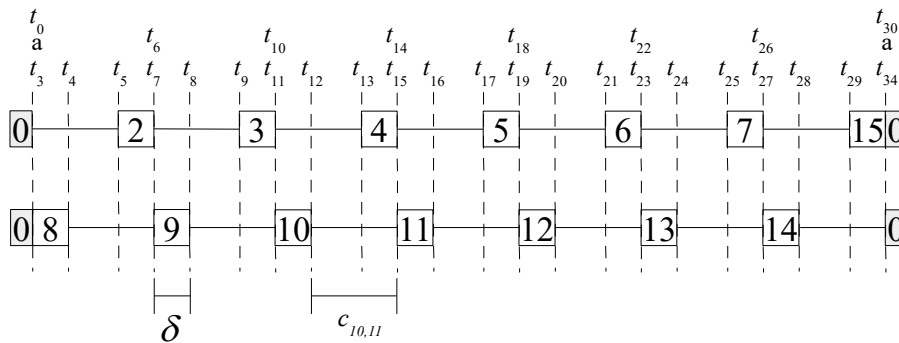


Figura 4.5 – Características temporais de uma solução da instância de exemplo usada nas outras figuras. Os tempos de deslocamento entre vértices são representados todos com a mesma dimensão para simplificar a ilustração. A solução considera que o primeiro veículo visita seus vértices na sequência 0, 1, 2, 3, 4, 5, 6, 7, 15, 0. Como os veículos se deslocam agrupados, separados apenas pelo atraso δ , o segundo veículo executa a sequência 0, 8, 9, 10, 11, 12, 13, 14, 16, 0. Por se tratar de uma instância de 17 vértices e dois veículos, existem 35 marcadores de tempo, e portanto 34 fatias. Os vértices 1 e 16 não aparecem na representação por não terem tempo de serviço, mas os marcadores de tempo relativos às partidas e chegadas nesses vértices estão representados. É possível perceber um padrão a partir da terceira fatia: o primeiro veículo parte do vértice 1, o segundo veículo parte do 8, o primeiro chega no 2 e o segundo chega no 9. A seguir, o primeiro veículo parte de vértice 1, o segundo veículo parte do 9, o primeiro chega no 3 e o segundo chega no 10. Sequência semelhante de partidas e chegadas ocorre até que o primeiro veículo parte de vértice 7, o segundo veículo parte do 14, o primeiro chega em 15 e o segundo chega em 16.

Os três primeiros marcadores de tempo representam eventos que ocorrem todos ao mesmo instante: a partida concomitante dos veículos da garagem, a chegada do primeiro ao vértice 1 e a chegada do segundo ao vértice $n + 1$. É correto afirmar, portanto, que as duas primeiras fatias de tempo terão dispersão nula.

Os quatro marcadores seguintes são definidos pelas partidas dos veículos rumo aos próximos vértices de suas rotas (geram dois marcadores devido ao desencontro causado pelo atraso) e pelas posteriores chegadas (também dois marcadores). Eles determinam um padrão, ilustrado na Figura 4.6, que se repete para todos os arcos percorridos no segmento que será chamado de *trecho interno* da rota. Esse segmento começa com o primeiro veículo partindo do vértice 1 e termina com o segundo chegando a $2n + 2$, para que ambos partam sincronizados para finalizarem suas rotas na garagem. Como esse padrão se repete para qualquer par de arcos correspondentes do trecho interno da rota, é considerado na descrição que o primeiro veículo percorre uma aresta qualquer e_{ij} , e o

segundo veículo percorre a correspondente $e_{n+i,n+j}$.

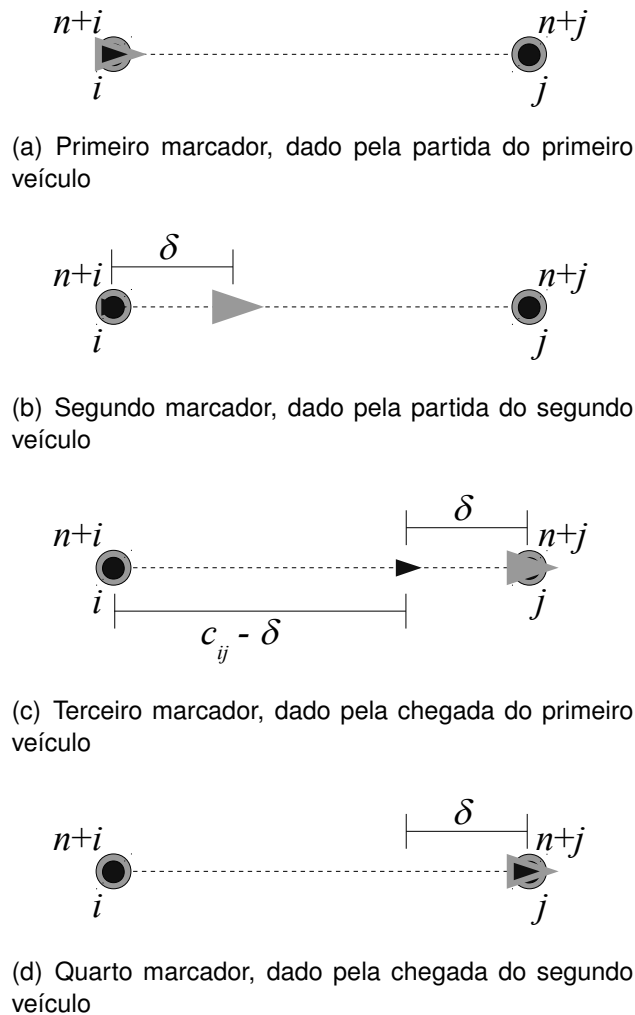


Figura 4.6 – Os marcadores de tempo que determinam o padrão de fatias de tempo que ocorre quando é percorrido qualquer par de arestas correspondentes do trecho interno da rota. A figura ilustra um caso em que o primeiro veículo se desloca entre os vértices i e j , e o segundo, por consequência, entre $n + i$ e $n + j$.

A primeira das quatro fatias de tempo determinadas por esses marcadores tem dispersão igual a zero, já que a partida do primeiro veículo se dá exatamente no mesmo momento em que o segundo veículo chega ao vértice correspondente. Conforme dito anteriormente, existe uma equivalência numérica entre a distância e o tempo para percorrê-la. Além disso, já foi demonstrado na Equação (3.8) que a dispersão de uma fatia pode ser calculada como o produto entre o tempo da fatia e a distância média entre os veículos. Desse modo, na segunda fatia, encerrada pela partida do segundo veículo, a dispersão é dada por $\delta \cdot (\delta/2)$. Segue-se uma fatia cuja duração é $c_{ij} - \delta$, finalizada pela chegada do primeiro veículo ao vértice j , e na qual a distância entre os veículos é mantida constante e igual a δ . A dispersão dessa fatia é dada por $\delta \cdot (c_{ij} - \delta)$. A última das quatro fatias de tempo é terminada pela chegada do segundo veículo ao vértice $n + j$, e apresenta

dispersão idêntica à segunda.

Portanto, para cada par de arcos correspondentes percorridos pelos dois veículos durante o trecho interno da rotas, tem-se uma dispersão dada pela soma das quatro fatias de tempo descritas, ou seja, $0 + \delta \cdot (\delta/2) + \delta \cdot (c_{ij} - \delta) + \delta \cdot (\delta/2) = c_{ij} \cdot \delta$.

Observando os marcadores de tempo do trecho interno da rota, é possível perceber a repetição desse padrão n vezes. Isso significa que, desconsiderados os deslocamentos iniciais e finais, que partem ou chegam na garagem, esse padrão de quatro fatias de tempo se repete para cada um dos pares de arestas concorrentes que serão percorridas pelos veículos.

Do total $g = 4n+6$ de fatias de uma instância, restam ainda 4 fatias por avaliar. Duas delas são delimitadas pelas partidas dos veículos a partir dos vértices $2n+1$ e $2n+2$, e as outras duas pelas chegadas na garagem. Dado que todos esses eventos são simultâneos, a dispersão nas últimas quatro fatias é nula.

Como as fatias anteriores e posteriores ao percurso do trecho interno da rota são nulas, a dispersão total de uma solução pode ser representado pelo somatório dos valores de dispersão encontrados nas n repetições do padrão de quatro fatias associadas a cada par de arestas concorrentes. Essa dispersão total é exibida na Equação 4.3, em que x_{ij} indica que a aresta e_{ij} está na solução.

$$C = \sum_{e_{ij} \in \mathcal{AU}\{a_{i,2n+1} : \forall i \in \mathcal{V}\}} c_{ij} x_{ij} \delta \quad (4.3)$$

Como δ é uma constante, é possível representar o cálculo da dispersão na forma da Equação 4.4, na qual fica evidente que a solução de mínima dispersão é aquela que minimiza o custo total de roteamento. É demonstrado, portanto, que a solução ótima do mTSP-D também é ótima para o ETSP.

$$C = \delta \sum_{e_{ij} \in \mathcal{AU}\{a_{i,2n+1} : \forall i \in \mathcal{V}\}} c_{ij} x_{ij} \quad (4.4)$$

A sequência de vértices da solução do ETSP é obtida com um processo simples: a rota em que são visitados os vértices de \mathcal{V}' é descartada, e o vértice $2n+1$ é substituído pelo vértice 1, de modo que sua rota se feche em um circuito. A criação da instância do mTSP-D demanda a criação de $n+3$ novos vértices e n^2+4n+4 novos arcos, além do cálculo do valor de δ , que exige $(n^3-3n^2+2n)/6$ passos para a comparação das combinações de vértices 3 a 3. Como todo o processo de redução pode ser executado em tempo polinomial, resta provado que o mTSP-D é *NP-Difícil*. \square

4.1.1 Modelo Matemático

Nessa seção, é apresentado um modelo matemático que descreve o mTSP-D. A obtenção de soluções ótimas através do modelo está limitada a instâncias de tamanho bastante moderado. Porém, o modelo é útil na formalização do problema, e também para orientar pesquisas sobre outros métodos de solução.

O modelo matemático é melhor compreendido se for considerado um grafo dirigido $G' = (V', A')$ construído a partir do grafo $G = (V, E)$. No grafo G' , cada vértice $i \in V \setminus \{0\}$ é duplicado em um vértice de chegada $i \in V'$ e outro de partida $i + n \in V'$. O tempo de serviço, em que um veículo se mantém parado no vértice $i \in V$, passa a ser interpretado como um deslocamento pela aresta entre i e $i + n$, com duração igual ao tempo de serviço, ou seja, $c_{i,i+n} = s_i, \forall i \in V$.

O vértice garagem é replicado em V' de forma que exista um vértice de onde partem todos os veículos, identificado como vértice 0, e vértices que representam garagens de chegada para cada um dos m veículos, numerados como $k + 2n \forall k \in K$. Os vértices de chegada na garagem definem o subconjunto VG do conjunto V' , ou seja, $VG \subset V'$.

No que se refere aos arcos, um veículo só pode partir de um vértice $i \leq n$ e se dirigir para o vértice correspondente $i + n$. De forma equivalente, um veículo só pode chegar a um vértice $i > n$ se vier de um vértice $i - n$. Além disso, uma solução não pode conter arcos que tenham como ponto de chegada o vértice 0 e nem como ponto de partida vértices $i > 2n$. Para garantir essa configuração, na construção do grafo G' são definidos como infinitos os custos de deslocamento relativos a:

- a) arcos que conectam um vértice de chegada com qualquer vértice que não seja o correspondente vértice de partida, $c_{ij} = \infty, \forall i \in V' : 0 < i \leq n, \forall j \in V' : j \neq i + n$;
- b) arcos que conectam um vértice de saída a qualquer outro de saída a um vértice de chegada diferente do correspondente vértice de chegada, $c_{ij} = \infty, \forall i \in V' : n < i \leq 2n, \forall j \in V' : n < j \leq 2n$;
- c) arcos que conectam qualquer vértice à garagem de partida, $c_{i0} = \infty, \forall i \in (V')$; e
- d) arcos que conectem uma garagem de chegada a outros vértices, $c_{ij} = \infty, \forall i \in VG, \forall j \in V' : j \leq 2n$.

A Figura 4.7(a) ilustra essa configuração de vértices e arcos.

A cardinalidade de V' é igual a $2n + m + 1$, em que $2n$ corresponde aos vértices de chegada e partida originados pela duplicação dos vértices em $V \setminus \{0\}$, m corresponde aos vértices de chegada à garagem e 1 é o vértice de partida da garagem. Dessa forma, é possível assumir que cada vértice gera apenas um marcador de tempo que delimita uma fatia de tempo.

Os conjuntos e parâmetros do modelo são:

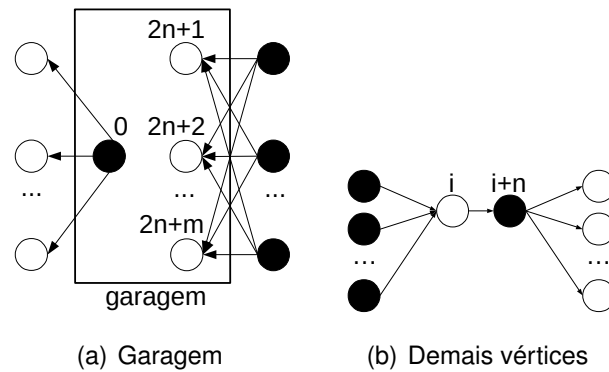


Figura 4.7 – Replicação de vértices para geração do grafo do modelo matemático. Os círculos pretos representam vértices de partida, e os brancos vértices de chegada. A garagem é replicada em um vértice de partida e m vértices de chegada, enquanto os demais vértices são replicados em um par com partida e chegada.

c_{ij}	: duração do deslocamento entre os vértices i e j
DX_{ij}	: componente horizontal do vetor deslocamento sobre o arco de i para j
DY_{ij}	: componente vertical do vetor deslocamento sobre o arco de i para j
$(X0, Y0)$: coordenadas da garagem
V'	: conjunto de vértices
VG	: conjunto de vértices garagem de chegada, $VG \subset V'$
K	: conjunto de veículos
\mathcal{S}	: conjunto dos marcadores de tempo que definem fatias de tempo
M	: um número suficientemente grande

As variáveis são definidas como:

x_{ijk}	: $\begin{cases} 1 & \text{se o vértice } j \text{ suceder o vértice } i \text{ na rota do veículo } k; \\ 0 & \text{caso contrário} \end{cases}$
$z_{i\tau}$: $\begin{cases} 1 & \text{se o instante do marcador } \tau \text{ é determinado pela chegada de um veículo no vértice } j; \\ 0 & \text{caso contrário} \end{cases}$
$w_{i,j\tau k}$: $\begin{cases} 1 & \text{se o veículo } k \text{ percorre o arco de } i \text{ a } j \text{ no instante do marcador de tempo } \tau; \\ 0 & \text{caso contrário} \end{cases}$
t_i	: tempo de chegada no vértice i
ts_τ	: tempo do marcador determinante de fatia de tempo τ
$X_{\tau k}$: posição do veículo k no instante τ em relação ao eixo x
$Y_{\tau k}$: posição do veículo k no instante τ em relação ao eixo y
u	: variável adimensional de integração

Para definição da função objetivo é utilizada a seguinte notação:

$$DDX(\tau kk') = X_{\tau-1,k} + u(X_{\tau k} - X_{\tau-1,k}) - [X_{\tau-1,k'} + u(X_{\tau k'} - X_{\tau-1,k'})]$$

e

$$DDY(\tau kk') = Y_{\tau-1,k} + u(Y_{\tau k} - Y_{\tau-1,k}) - [Y_{\tau-1,k'} + u(Y_{\tau k'} - Y_{\tau-1,k'})]$$

Modelo:

$$\text{Min} \sum_{\tau \in \mathcal{S}: \tau > 0} \left[(ts_{\tau} - ts_{\tau-1}) \max_{\substack{k=1, \dots, m \\ k'=k+1, \dots, m}} \int_0^1 \sqrt{[DDX(\tau kk')]^2 + [DDY(\tau kk')]^2} du \right] \quad (4.5)$$

s.a.

$$\sum_{j \in V'} \sum_{k \in K} x_{ijk} = 1 \quad \forall i \in V' \setminus (\{0\} \cup VG) \quad (4.6)$$

$$\sum_{i \in V'} \sum_{k \in K} x_{ijk} = 1 \quad \forall j \in V' \setminus \{0\} \quad (4.7)$$

$$\sum_{j \in V'} \sum_{k \in K} x_{0jk} = m \quad (4.8)$$

$$\sum_{i \in V'} x_{ijk} - \sum_{i \in V'} x_{jik} = 0 \quad \forall j \in V' \setminus (\{0\} \cup VG), \forall k \in K \quad (4.9)$$

$$t_0 = 0 \quad (4.10)$$

$$t_j \geq t_i + c_{ij} + \left(\sum_{k \in K} x_{ijk} - 1 \right) M \quad \forall i \in V' \setminus VG, \forall j \in V' \setminus \{0\} \quad (4.11)$$

$$t_j \leq t_i + c_{ij} + \left(1 - \sum_{k \in K} x_{ijk} \right) M \quad \forall i \in V' \setminus VG, \forall j \in V' \setminus \{0\} \quad (4.12)$$

$$ts_0 = t_0 \quad (4.13)$$

$$ts_{\tau} \leq ts_{\tau+1} \quad \forall \tau \in \mathcal{S} : \tau < |\mathcal{S}| \quad (4.14)$$

$$\sum_{j \in V': j > 0} z_{j\tau} = 1 \quad \forall \tau \in \mathcal{S} : \tau > 0 \quad (4.15)$$

$$\sum_{\tau \in \mathcal{S}: \tau > 0} z_{j\tau} = 1 \quad \forall j \in V' \setminus \{0\} \quad (4.16)$$

$$ts_{\tau} \geq t_j + (z_{j\tau} - 1)M \quad \forall j \in V' \setminus \{0\}, \forall \tau \in \mathcal{S} : \tau > 0 \quad (4.17)$$

$$ts_{\tau} \leq t_j + (1 - z_{j\tau})M \quad \forall j \in V' \setminus \{0\}, \forall \tau \in \mathcal{S} : \tau > 0 \quad (4.18)$$

$$ts_{\tau} - t_i + \left(1 - \sum_{k \in K} w_{ij\tau k} \right) M \geq 0 \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S} : \tau > 0 \quad (4.19)$$

$$t_j - ts_{\tau} + \left(1 - \sum_{k \in K} w_{ij\tau k} \right) M \geq 0 \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S} : \tau > 0 \quad (4.20)$$

$$\sum_{i \in V'} \sum_{j \in V'} w_{ij\tau k} = 1 \quad \forall \tau \in \mathcal{S} : \tau > 0, \forall k \in K \quad (4.21)$$

$$x_{ijk} \geq w_{ij\tau k} \quad \forall i \in V' \setminus VG, \forall j \in V', \quad \forall \tau \in \mathcal{S} : \tau > 0, \forall k \in K \quad (4.22)$$

$$\sum_{i \in V'} \sum_{k \in K} w_{ij\tau k} \geq z_{j\tau} \quad \forall j \in V' \setminus \{0\}, \forall \tau \in \mathcal{S} : \tau > 0 \quad (4.23)$$

$$X_{0k} = X0 \quad \forall k \in K \quad (4.24)$$

$$X_{\tau k} \geq X_{\tau-1,k} + (ts_{\tau} - ts_{\tau-1})DX_{ij} + (w_{ij\tau k} - 1)M \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S} : \tau > 0, \forall k \in K \quad (4.25)$$

$$X_{\tau k} \leq X_{\tau-1,k} + (ts_{\tau} - ts_{\tau-1})DX_{ij} + (1 - w_{ij\tau k})M \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S} : \tau > 0, \forall k \in K \quad (4.26)$$

$$Y_{0k} = Y0 \quad \forall k \in K \quad (4.27)$$

$$Y_{\tau k} \geq Y_{\tau-1,k} + (ts_{\tau} - ts_{\tau-1})DY_{ij} + (w_{ij\tau k} - 1)M \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S} : \tau > 0, \forall k \in K \quad (4.28)$$

$$Y_{\tau k} \leq Y_{\tau-1,k} + (ts_{\tau} - ts_{\tau-1})DY_{ij} + (1 - w_{ij\tau k})M \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S} : \tau > 0, \forall k \in K \quad (4.29)$$

$$x_{ijk} \in \{0, 1\} \quad \forall i \in V', \forall j \in V', \forall k \in K \quad (4.30)$$

$$t_i \geq 0 \quad \forall i \in V' \quad (4.31)$$

$$ts_{\tau} \geq 0 \quad \forall \tau \in \mathcal{S} \quad (4.32)$$

$$z_{i\tau} \in \{0, 1\} \quad \forall i \in V', \forall \tau \in \mathcal{S} \quad (4.33)$$

$$w_{ij\tau k} \in \{0, 1\} \quad \forall i \in V', \forall j \in V', \forall \tau \in \mathcal{S}, \forall k \in K \quad (4.34)$$

$$X_{\tau k} \quad \text{irrestrita} \quad \forall k \in K, \forall \tau \in \mathcal{S} \quad (4.35)$$

$$Y_{\tau k} \quad \text{irrestrita} \quad \forall k \in K, \forall \tau \in \mathcal{S} \quad (4.36)$$

A função objetivo de minimização da dispersão está representada na Equação (4.5). Para facilitar a compreensão, os conjuntos de restrições podem ser divididos em quatro blocos.

O primeiro bloco contém as restrições que definem o sequenciamento das rotas. O conjunto de restrições identificado por (4.6) indica que todos os vértices devem ter sucessor, com exceção dos vértices de garagem. O conjunto (4.7) indica que cada vértice deve ter um único predecessor. O vértice 0 deve ter m sucessores, conforme as restrições identificadas por (4.8). O conjunto de restrições (4.9) indica que cada vértice diferente de garagem tenha predecessor e sucessor na mesma rota, a restrição (4.10) define que a partida da garagem ocorre no instante 0 e os conjuntos de restrições (4.11) e (4.12) definem o instante exato de chegada em cada vértice j , além de garantirem a não formação de subciclos.

O segundo bloco de restrições define as fatias de tempo ao definir e sequenciar os marcadores de tempo que as delimitam. A restrição (4.13) define como marcador de tempo referente às partidas da garagem o instante t_0 . O conjunto (4.14) estabelece que os marcadores de tempo devem estar ordenados, e os conjuntos (4.15) e (4.16) associam um único marcador a cada vértice, e um único vértice a cada marcador. Definida essa associação, o tempo do marcador passa a ser igual ao tempo de chegada no vértice correspondente, conforme os conjuntos de restrições (4.17) e (4.18).

As restrições do terceiro bloco têm a função de definir os arcos percorridos por cada

veículo durante uma fatia de tempo. Os conjuntos de restrições (4.19) e (4.20) estabelecem que o arco sendo percorrido por cada veículo no marcador de tempo τ deve ter seu percurso iniciado em instante igual ou anterior ao marcador e terminado em instante posterior. Apenas uma aresta pode estar sendo percorrida por cada veículo em cada fatia, conforme o conjunto de restrições (4.21), a aresta sendo percorrida precisa ser parte da rota do veículo, conforme o conjunto de restrições (4.22). No caso do veículo cuja chegada a um vértice determina um marcador de tempo, tanto o arco que incide no vértice como o arco que parte do vértice atendem as restrições deste bloco. O conjunto de restrições (4.23), determina que o arco associado ao veículo que define um marcador de tempo deve ser um arco que incide no vértice.

O quarto bloco apresenta as restrições utilizadas para determinar a posição de cada veículo nos instantes definidos pelos marcadores de tempo. As restrições (4.24) e (4.27) determinam que todos os veículos estão na garagem no momento que corresponde ao primeiro marcador de tempo. Os conjuntos de restrições (4.25 - 4.26) e (4.27 - 4.29) determinam a posição de cada um dos veículos nos demais marcadores de tempo considerando a posição anterior, o tempo e a componente de deslocamento. As restrições de (4.30) a (4.36) definem o domínio das variáveis de decisão.

4.2 MÉTODO DE SOLUÇÃO

Nessa seção, são apresentados os procedimentos usados na construção das soluções. São descritos o Algoritmo Genético, com destaque para suas especializações às características do mTSP-D, um algoritmo construtivo usado para a criação da população inicial, e o procedimento de Busca Local, com destaque para as vizinhanças utilizadas.

4.2.1 O Algoritmo Genético

O Algoritmo 2 apresenta o pseudocódigo do procedimento evolutivo do LSGA utilizado para encontrar soluções de roteamento para o mTSP-D. O restante da presente seção explica este algoritmo em detalhes.

Algoritmo 2 Pseudocódigo do LSGA.

```

newPop ← ∅
curPop ← InitialPopulation(nRCL, ps,  $Nn(i) \forall i \in V$ )
LocalSearch(curPop)
g ← 1
while maxTime não é atingido do
  EvaluateIndividuals(curPop)
  Elite ← SelectElite(es, curPop)
  newPop ← newPop ∪ Elite
  while |newPop| < ps do
    Ia ← RouletteWheel(curPop)
    if GenerateRandomNumber( ∈ [0, 1) ) < cp and |newPop| < ps − 1 then
      Ib ← RouletteWheel(curPop)
      Offspring ← Crossover(Ia, Ib)
      newPop ← newPop ∪ Offspring
    else
      Ic ← Mutation(Ia)
      newPop ← newPop ∪ {Ic}
    end if
  end while
  curPop ← newPop
  newPop ← ∅
  if g mod β = 1 then
    LocalSearch(curPop)
  end if
  g ← g + 1
end while
return o melhor indivíduo em curPop

```

A população inicial $curPop = \{I_0, I_1, \dots, I_{ps-1}\}$ tem *ps* indivíduos *I*, que são gerados por um método construtivo descrito em detalhes na Seção 4.2.2. Em seguida, os *lsSize* indivíduos melhor avaliados da população são submetidos a um procedimento de Busca Local que será descrito na Seção 4.2.4. A partir dessa população, iniciam-se os

ciclos de avaliação-seleção-reprodução, que determinam as gerações do método evolutivo, encerrados quando o tempo total de execução (incluindo as etapas iniciais externas ao ciclo) atingir o tempo limite $MaxTime$.

A avaliação dos membros da população corrente é a primeira etapa de cada iteração. O valor de C na Equação (4.2) representa a dispersão de cada indivíduo da população. Mas a definição do mTSP-D determina que a dispersão deve ser minimizada, assim é necessário atribuir menor valor de aptidão (*fitness*) para indivíduos com maior dispersão, e maior valor de aptidão para indivíduos com menor dispersão. Experiências computacionais mostram que a aplicação da Equação (4.37) para o cálculo do valor de aptidão resulta em boa discriminação entre os indivíduos da população.

$$fitness(u) = \left[\left(\max_{h=1, \dots, ps} C_{I_h} \right) / C_{I_u} \right]^2 \quad (4.37)$$

A equação utiliza o indivíduo com maior dispersão para calcular a aptidão de cada indivíduo I_u , elevando ao quadrado a razão entre a máxima dispersão e C_{I_u} . O resultado orienta a seleção no LSGA.

A construção da nova população $newPop$ é iniciada pela cópia direta de uma elite formada pelos es indivíduos de melhor avaliação da população corrente $curPop$. Os demais $ps - es$ indivíduos de $newPop$ são obtidos pela aplicação de operadores genéticos. Para isso, um indivíduo I_a é selecionado de $curPop$ através do mecanismo de roleta. Com probabilidade cp , I_a será submetido a cruzamento com outro indivíduo I_b também selecionado de $curPop$ com roleta, resultando em dois novos indivíduos adicionados a $newPop$. Com probabilidade $1 - cp$, I_a é submetido a mutação, gerando o indivíduo I_c , que é adicionado a $newPop$. Nos casos em que a população atinge o tamanho $ps - 1$, o último indivíduo é gerado obrigatoriamente por mutação. Quando $newPop$ atinge o número máximo de indivíduos, é adotado como a população corrente $curPop$.

A Busca Local é custosa em termos de tempo computacional. Por isso, é aplicada a cada β gerações, começando na primeira. Finalmente, o contador de gerações g é incrementado, encerrando cada iteração do ciclo evolutivo.

4.2.2 Geração da População Inicial

A população inicial é criada com um algoritmo construtivo guloso e com aleatoriedade, cujo pseudocódigo é apresentado no Algoritmo 3. O algoritmo está organizado a partir de dois conjuntos de vértices. O conjunto Out contém os vértices por incluir nas rotas, enquanto LN contém os últimos vértices atribuídos a cada rota. Os vértices em LN podem ser interpretados como a fronteira corrente que delimita as rotas parciais. O método faz uso ainda de conjuntos $Nn(i)$, associados a cada vértice $i \in V$ e que contém

que os vértices de $V \setminus \{0, i\}$ mais próximos de i .

Algoritmo 3 Pseudocódigo do método de construção da população inicial

Input:

$nRCL$: número de candidatos a considerar na seleção aleatória

ps : tamanho da população

$Nn(i) \forall i \in V$: lista de vizinhos mais próximos de cada vértice

$Pop \leftarrow \emptyset$

while $|Pop| < ps$ **do**

 Cria um novo indivíduo I

$Out \leftarrow V$

$LN \leftarrow \emptyset$

 Seleciona aleatoriamente um vértice $i \in Out$

$Out \leftarrow Out \setminus \{i\}$

$LN \leftarrow LN \cup i$

 Inicia nova rota em I inserindo a aresta e_{0i}

$t_i \leftarrow c_{0i} + s_i$

for k **from** 2 **to** m **do**

 Seleciona aleatoriamente um vértice $j \in Out \cap Nn(i)$

$Out \leftarrow Out \setminus \{j\}$

 Inicia nova rota em I inserindo a aresta e_{0j}

$t_j \leftarrow c_{0j} + s_j$

end for**while** $|Out| > 0$ **do**

$j' \leftarrow \underset{j: j \in LN}{\operatorname{argmin}} \{t_j\}$

$eval_h \leftarrow \max_{j \in LN} \{c_{hj}\}, \forall h \in Out$

 Crie o RCL com os $nRCL$ vértices de $eval$ mais baixo

 Seleciona aleatoriamente um vértice $l \in RCL$

$Out \leftarrow Out \setminus \{l\}$

 Insere aresta $e_{j'l}$ no indivíduo I definindo l como o sucessor de j'

$LN \leftarrow LN \setminus \{j'\}$

$LN \leftarrow LN \cup \{l\}$

$t_l \leftarrow t_{j'} + c_{j'l} + s_l$

end while**for** k **from** 1 **to** m **do**

 Seleciona um vértice $j \in LN$

$LN \leftarrow LN \setminus \{j\}$

 Fecha a rota inserindo a aresta e_{j0}

end for

$Pop \leftarrow Pop \cup \{I\}$

end while

return Pop

Inicialmente, um veículo é designado para um vértice $i \in Out$ selecionado aleatoriamente, o que resulta na inclusão da aresta e_{0i} na solução, iniciando uma rota. Os demais $m - 1$ veículos são designados para vértices selecionados aleatoriamente entre aqueles que fazem parte de $Nn(i) \cap Out$, ou seja, são vértices próximos a i e não estão em rotas. Cada vértice selecionado é removido de Out , impedindo que seja selecionado mais de uma vez, e colocado em LN , uma vez que passa a ser o último em sua rota parcial. Sem-

pre que um vértice é inserido em uma rota, é calculado o instante previsto para a partida do veículo responsável pelo seu atendimento.

Nesse ponto do procedimento, cada rota contempla exatamente um vértice. A partir de então, um procedimento iterativo é iniciado para colocar em rotas, um de cada vez, os vértices ainda em *Out*.

A cada iteração, é localizado j' , o vértice em LN cujo atendimento tem previsão de encerramento mais cedo. Como esse é o mais adiantado dos vértices que marcam o fim das rotas parciais, indica que sua rota é a mais curta até então e será a rota de destino do próximo vértice incluído, escolhido como segue. Cada vértice $h \in Out$ é um potencial candidato para essa inserção, e é avaliado pela máxima distância em relação a qualquer vértice $j \in LN$. Os $nRCL$ vértices com menor valor de avaliação formam RCL , uma lista restrita de candidatos. Um vértice $l \in RCL$ é selecionado aleatoriamente, e a aresta e_{jl} é adicionada à solução. A Figura 4.8 ilustra a inclusão de um vértice, através de um exemplo com três veículos.

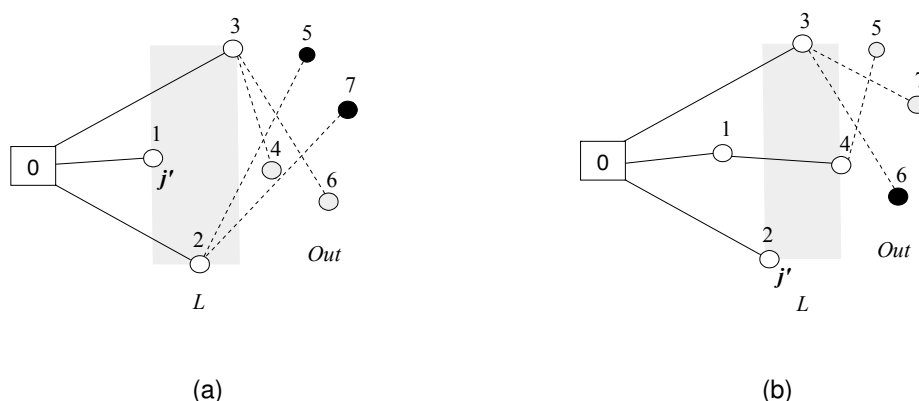


Figura 4.8 – Passo de inserção de um vértice no algoritmo construtivo. Círculos brancos representam vértices já visitados, círculos pretos e cinza são vértices ainda não visitados ($\in Out$), e a área retangular sombreada cobre os vértices $\in L$, a fronteira atual das rotas parciais. A fronteira inicial já está construída. (a) O veículo no vértice 1 (j') é o próximo a ser considerado, porque ele está atrasado em relação aos demais. Os vértices candidatos em *Out* são definidos com referência aos demais vértices em L (ou seja, 2 e 3). Cada vértice $h \in Out$ está conectado por uma linha tracejada ao mais distante vértice $j \in L \setminus \{j'\}$. O comprimento representa c_{jh} . Círculos cinza representam os vértices selecionados para fazerem parte da RCL , com $nRCL = 2$ nesse exemplo. (b) O vértice 4 foi aleatoriamente escolhido a partir da RCL apresentada em (a). A fronteira é atualizada. O veículo no vértice 2 (o novo j') é o próximo a ser encaminhado, pois está atrasado em relação aos outros. RCL é atualizado.

O objetivo de escolher como candidatos aqueles vértices de menor distância máxima é garantir que o novo vértice, que sucede j' na rota e o substitui em LN , mantenha a proximidade entre os componentes dessa fronteira. A manutenção de uma fronteira com

vértices próximos está de acordo com o objetivo de minimização de dispersão, já que os vértices colocados em LN simultaneamente devem exibir proximidade temporal na solução de roteamento.

Os vértices são incluídos, um por vez, até que $Out = \emptyset$. Nesse ponto, todas as rotas recebem uma aresta ligando seu vértice em LN à garagem.

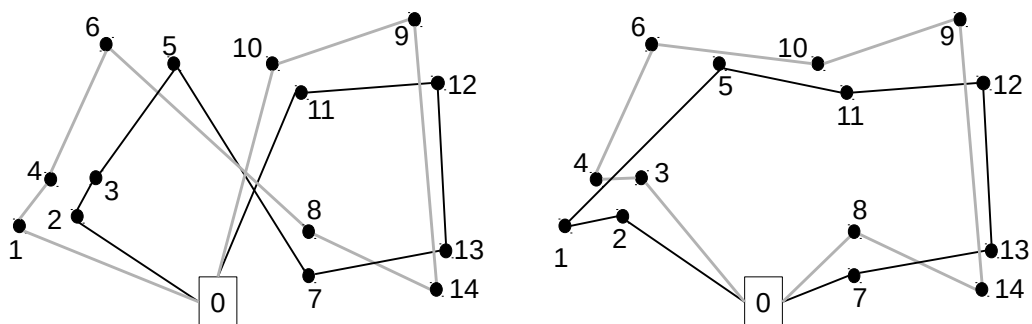
4.2.3 Cromossomo e Operadores Genéticos

Usualmente, algoritmos genéticos aplicados a problemas de roteamento de veículos fazem uso de cromossomos representados como cadeias de inteiros para a codificação de soluções. Em uma estrutura desse tipo, os números representam os vértices, e a cadeia representa a sua ordenação na solução, tornando explícita a sucessão de visitadas que define cada rota. Novas soluções são geradas a partir de antigas, através de alterações em posições definidas por índices na cadeia. Tal representação favorece a propagação, quando da aplicação de operadores genéticos, das informações referentes ao sequenciamento dos vértices e das arestas. Isso é consistente com os objetivos relacionados com o comprimento das rotas, em que deve ser privilegiada a proximidade espacial entre vértices sucessivos.

No contexto de sincronização como a que se apresenta no mTSP-D, é preciso privilegiar a propagação das informações referentes a arestas percorridas concomitantemente nas soluções de melhor qualidade, pois são as arestas concomitantes que determinam as distâncias instantâneas entre os m veículos. Ou seja, é mais importante propagar informação sobre o que acontece simultaneamente nas diversas rotas. Nesse caso, uma representação como a descrita não é favorável, conforme é possível observar na Figura 4.9.

Nas duas soluções novas ilustradas nas Figuras 4.9(d) e 4.9(e), há uma sensível degeneração de qualidade em relação às rotas originais, pois trechos de rotas de soluções distintas, sem relação temporal e espacial, são colocados para execução de forma concomitante. Para evitar isso, é interessante que trechos concomitantes das diversas rotas originais sejam propagados agrupados. Optou-se então por uma implementação alternativa dos cromossomos, que permite que os pontos de aplicação das modificações genéticas sejam definidos em função instantes de tempo, selecionados no intervalo de roteamento, e não em função de índices de posição de genes. Definido um instante de tempo, a alteração do operador genético pode ser aplicada a todas as rotas, de modo que as interrelações presentes nas partes anteriores e posteriores ao ponto de alteração sejam mantidas nas novas soluções geradas.

A Figura 4.10 ilustra a vantagem de seccionar as rotas em função de tempo, de forma a manter os segmentos concomitantes das rotas agrupados.

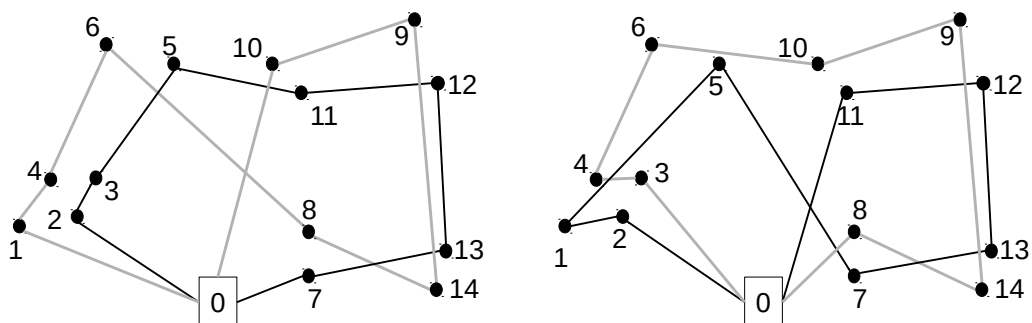


(a) Rotas da Solução Original 1

(b) Rotas da Solução Original 2

1	4	6	8	14	9	10	0	2	3	5	7	13	12	11
3	4	6	10	9	14	8	0	2	1	5	11	12	13	7
x														
1	4	6	8	14	9	10	0	2	3	5	11	12	13	7
3	4	6	10	9	14	8	0	2	1	5	7	13	12	11

(c) Cruzamento



(d) Rotas da Solução Resultante 1

(e) Rotas da Solução Resultante 2

Figura 4.9 – Ilustração do uso de cromossomos que privilegiam a representação da sucessão de vértices. Em todas soluções ilustradas, os veículos começam as rotas partindo da base para o lado esquerdo da figura. As soluções originais, representadas em (a) e (b), apresentam sincronismo, com boa relação temporal e espacial entre as rotas. Em (c) está exemplificada uma recombinação de um ponto (indicado pela linha vertical) entre essas duas soluções, representadas nos cromossomos na parte superior. Os cromossomos codificam as soluções através de cadeias de inteiros, com um marcador (ilustrado pelo valor 0, da base) indicando o fim de uma rota e o início da outra. As figuras em (d) e (e) ilustram as soluções resultantes do cruzamento, e é possível perceber que boa parte do sincronismo que conferia qualidade às soluções iniciais não foi propagada para a descendência. Nas duas novas soluções, os veículos se afastam após partirem dos vértices 5 e 6.

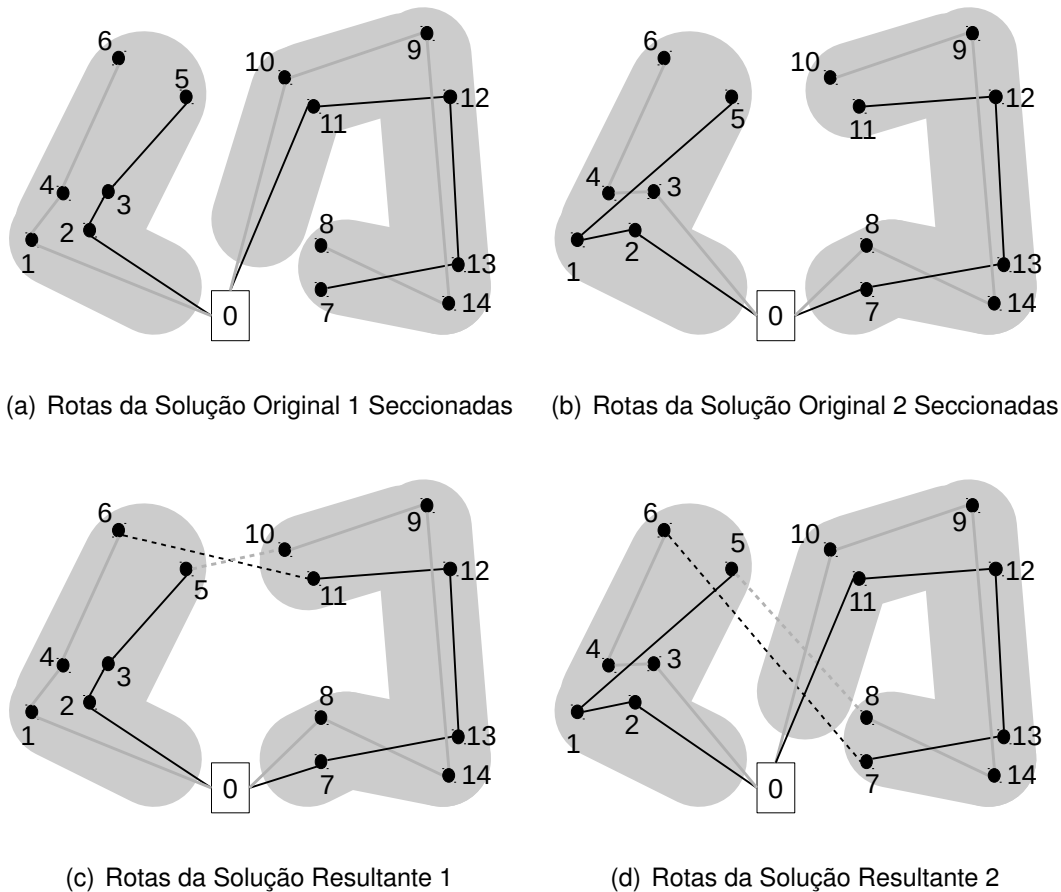


Figura 4.10 – Recombinação de soluções com o seccionamento das rotas em um instante de tempo. São consideradas como soluções iniciais as configurações ilustradas nas Figuras 4.9(a) e 4.9(b). Dado o ponto de cruzamento como um instante logo após os veículos deixarem os vértices 5 e 6, as rotas são seccionadas conforme apresentados em (a) e (b). As soluções ilustradas em (c) e (d) apresentam uma das possíveis recombinações dos segmentos de modo a gerar soluções novas. Percebe-se que as soluções resultantes são originais, distintas das soluções iniciais, mas que os segmentos concomitantes nas soluções iniciais mantém essa característica nas soluções geradas.

O cromossomo é implementado como um conjunto de rotas, cada uma em uma estrutura específica, tornando evidente o paralelismo e o sincronismo da execução de uma solução. Para a implementação de cada rota, optou-se por listas ligadas, em uma representação explícita do sequenciamento que torna mínimos os processos de codificação e decodificação. A representação encadeada simplifica os operadores genéticos, pois reduz as alterações nas rotas a ajustes de algumas referências de sequenciamento. Isso também impacta no desempenho de Busca Local, porque os movimentos são aplicados sobre a mesma estrutura e não há necessidade do procedimento de decodificação para a avaliação de uma solução.

O operador de cruzamento aplicado sobre tal cromossomo é mais complexo que o ilustrado na Figura 4.10, mas também combina as soluções de dois indivíduos pais, p_1 e p_2 , para criar duas novas soluções, o_1 e o_2 , de indivíduos filhos. O procedimento parte da definição de dois marcadores de tempo de cruzamento, ct_1 e ct_2 , que dividem todas as rotas de uma solução em três segmentos: o inicial, contendo vértices com tempo de chegada anterior ou igual a ct_1 ; o central, contendo os vértices com tempo de chegada posterior a ct_1 e anterior ou igual a ct_2 ; e o final, contendo os vértices com tempo de chegada posterior a ct_2 . O uso do tempo de chegada ao vértice como comparação determina que um vértice sendo visitado nos instantes ct_1 ou ct_2 seja colocado no segmento anterior ao ponto.

Cada rota da solução o_1 recebe o segmento inicial da rota equivalente da solução p_1 , em seguida o segmento central de uma das rotas da solução p_2 , e então o segmento final da mesma rota que contribuiu com o segmento inicial. A definição das combinações entre as rotas de p_1 com rotas de p_2 para cada nova solução se dá de forma aleatória a cada novo cruzamento.

Como os seccionamentos nas rotas de uma solução se dão em função de dois marcadores, ct_1 e ct_2 , essa operação é referida como um cruzamento de dois pontos. Mas é importante salientar que existem na verdade $2m$ pontos em que uma solução é seccionada, pois a divisão em segmentos ocorre em cada rota.

É importante notar que os segmentos de soluções diferentes podem trazer consigo referências a um mesmo vértice, gerando referência duplicada em o_1 . Ocorrências assim são contornadas descartando a segunda inclusão do vértice, e colocando-o em uma lista para inserção ao final da composição dos segmentos de o_2 . A construção de tal lista é necessária porque o direcionamento para inclusão duplicada em o_1 implica necessariamente na sua ausência do vértice em o_2 .

O descarte da segunda referência sempre que ocorre a duplicidade, com a manutenção da ocorrência mais adiantada, decorre do fato de que qualquer alteração em rota (inclusão ou retirada de vértice) afeta o sincronismo em todo o roteamento previsto para tempo posterior. Portanto, retiradas de vértices no início das rotas tendem a ser mais prejudiciais do que as realizadas no final.

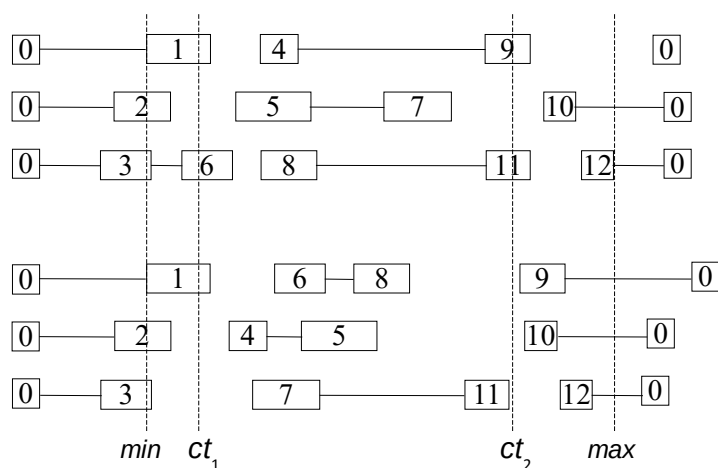
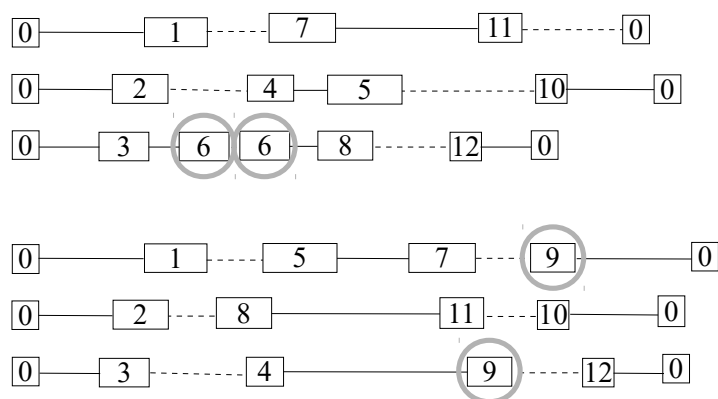
Após a composição dos segmentos da solução o_1 , é feita a reinserção dos vértices remanescentes que haviam sido direcionados em duplicidade à o_2 . A rota parcial mais curta de o_1 é identificada, e cada vértice a ser inserido é avaliado pela sua máxima distância aos vértices finais das demais rotas. O vértice de menor valor de avaliação (menor distância máxima) é colocado na rota atrasada, e o procedimento é repetido para os demais vértices que devem ser incluídos. Este procedimento é semelhante ao usado no algoritmo de geração da população inicial descrito na Seção 4.2.2, mas não faz uso de lista de candidatos e aleatoriedade, selecionando para inserção exatamente o vértice com melhor avaliação.

O procedimento para construção da solução o_2 é semelhante, alterando-se apenas a contribuição de p_1 e p_2 . De forma resumida, o operador de cruzamento pode ser descrito como a troca de segmentos centrais entre rotas selecionadas aleatoriamente. Isso garante que segmentos concomitantes em p_1 e p_2 sejam redistribuídos, formando novas rotas em o_1 e o_2 , mas tenham sua relação temporal pouco afetada.

É importante salientar ainda problemas que ocorrem pela escolha de pontos de cruzamento ruins. Uma situação indesejável ocorre quando ct_1 precede a primeira chegada a um vértice, e ct_2 é posterior ao momento em que todos os vértices já foram servidos. Neste caso, todas as rotas terão vértices apenas em seus segmentos centrais, e as trocas resultarão em indivíduos novos que representam as mesmas soluções dos indivíduos que lhes deram origem. Para evitar essa situação, ct_1 e ct_2 devem ser posteriores a um limite mínimo *min* dado pelo momento em que o último veículo chega ao primeiro vértice de sua rota, e anteriores ao instante *max* em que ocorre a partida do último vértice visitado. Esses limites garantem que o primeiro segmento de cada rota terá pelo menos um vértice. Mas não há garantia de que o segmento final não será vazio em todas. Essa situação é aceita pois equivale a um cruzamento de um ponto, em que haverá trocas entre segmentos iniciais e centrais, suficientes para gerar rotas diferentes nos novos indivíduos.

Uma segunda situação a ser evitada ocorre quando todos os segmentos centrais são vazios por não haver vértice a ser visitado no intervalo entre ct_1 e ct_2 . Para que isso se torne improvável, é definido que ct_1 e ct_2 devem estabelecer um intervalo tal que $ct_2 - ct_1 > \max_{i \in V} s_i + \max_{i,j \in V} c_{ij}$. Se esse intervalo excede o intervalo entre *min* e *max*, ct_1 e ct_2 são fixados exatamente sobre esses limites. A Figura 4.11 ilustra o procedimento descrito no cruzamento entre indivíduos que representam soluções de três rotas.

Como o operador de cruzamento apenas troca segmentos centrais, é possível que os primeiros vértices de cada rota fiquem intocados. De modo a gerar uma maior flexibilidade, optou-se por possibilitar a inversão das rotas antes de aplicar o operador de cruzamento. Selecionados dois indivíduos para a cruzamento, existe uma probabilidade de 30% de que o operador seja aplicado após a inversão no sentido de todas as rotas de p_1 e p_2 . Esse percentual foi definido empiricamente, sendo também testadas probabilidades de 10% e 50%.

(a) Indivíduos originais p_1 e p_2 segmentados

(b) Soluções inactíveis nos indivíduos resultantes do cruzamento

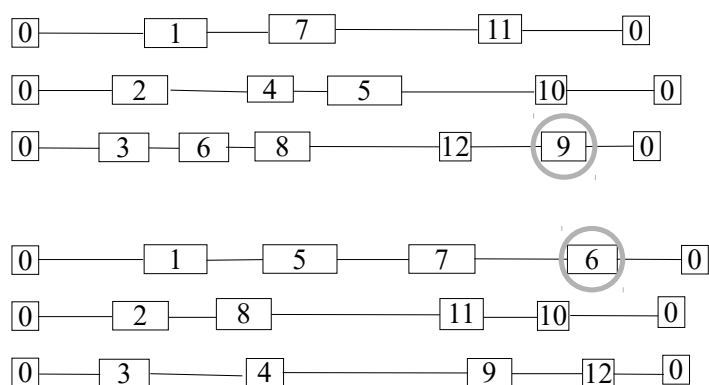
(c) Indivíduos resultantes o_1 e o_2 factíveis

Figura 4.11 – Cruzamento entre p_1 e p_2 para gerar o_1 e o_2 . (a) Rotas segmentadas nos marcadores ct_1 e ct_2 , aleatoriamente definidos entre os limites min e max . (b) Segmentos centrais são trocados para gerar o_1 e o_2 . O procedimento resultou em soluções inactíveis pela duplicidade do vértice 6 em o_1 e do vértice 9 em o_2 . (c) As soluções o_1 e o_2 são corrigidas, e resultam viáveis.

O operador de mutação é aplicado considerando quatro possíveis procedimentos, todos com igual probabilidade. As três primeiras formas de mutação têm como ponto de partida a escolha aleatória de um marcador de tempo de mutação mt entre 0 e o instante de chegada na garagem da rota mais curta. Na primeira forma, o vértice que precede mt em cada rota é trocado de posição com um de seus vizinhos do respectivo conjunto Nn . Na segunda forma, o vértice que precede mt em cada rota é colocado em posição aleatória considerando todas as rotas da solução, mas cada rota é selecionada apenas uma vez como destino. Na terceira forma, os mesmos vértices que precedem mt são removidos de suas rotas e um novo marcador de tempo é definido aleatoriamente para sua reinserção nas rotas. A definição da rota de destino de cada vértice é aleatória, mas novamente todas as rotas receberão um vértice apenas. Na quarta forma de mutação, são selecionados m vértices, um por rota, para serem reposicionados, também um por rota, em posições escolhidas também de forma aleatória. A diferença neste caso em relação aos outros três é que não há qualquer relação temporal entre os vértices envolvidos.

4.2.4 Busca Local

A observação das melhores soluções obtidas nos primeiros testes permitiu inferir duas características importantes para a minimização da dispersão: o (já referido) sincronismo entre os veículos e o comprimento das rotas. A importância desses atributos orientou a busca por esquemas de vizinhança, descritos nas subseções a seguir. Esses esquemas são utilizados em procedimento de Busca Local detalhado na Seção 4.2.4.3, que faz uso de vizinhanças variáveis (MLADENVIĆ; HANSEN, 1997).

4.2.4.1 Vizinhança de aproximação das rotas

A disposição espacial das rotas é definida pela disposição dos vértices que devem ser visitados por cada um dos veículos. Em geral, o posicionamento dos vértices no ambiente é uma informação estabelecida nas instâncias, e cabe aos métodos de solução sequenciar esses vértices, indiretamente estabelecendo os instantes de chegada e partida em cada um. Em problemas de roteamento focados na redução de custo de deslocamento, a disposição sequencial dos vértices é relevante para a função objetivo, e as rotas são formadas preferencialmente pela colocação em sequência de vértices espacialmente próximos. No caso do mTSP-D, para garantir a proximidade dos veículos, ainda mais importante é a disposição de vértices próximos para serem visitados em paralelo, de modo a garantir a proximidade temporal entre vértices espacialmente próximos.

Como exemplo, é possível considerar uma instância em que existam vértices coin-

cidentes. Nesse caso, dois veículos se deslocando sincronizados a partir de dois vértices coincidentes e com destino a outros dois vértices coincidentes terão dispersão entre eles igual a zero, independente do tamanho do deslocamento. Porém, se o deslocamento pelas mesmas arestas não for sincronizado, a proximidade espacial entre os vértices coincidentes torna-se irrelevante frente a ausência de proximidade temporal. A partir dessas considerações, foram elaboradas vizinhanças que favoreçam a descoberta de boas configurações em termos de proximidade temporal na visita a vértices espacialmente próximos.

Como a dispersão total é calculada de forma segmentada, a partir de fatias de tempo, os movimentos focam na redução do distanciamento dos veículos em cada fatia. Por isso, inicialmente é necessário definir o conceito de par de vértices mais custoso, que estabelece os vértices que mais contribuem para o custo local de uma fatia de tempo.

Dada uma fatia de tempo e encontradas as rotas mais dispersas durante a sua duração, é verificado se as rotas se aproximam ou distanciam durante a fatia. Desse modo, é possível determinar se o par de vértices que mais influencia no afastamento é o que determina as posições iniciais ou finais dos veículos. A Figura 4.12 exemplifica a descoberta de um par de vértices mais custoso de uma fatia.

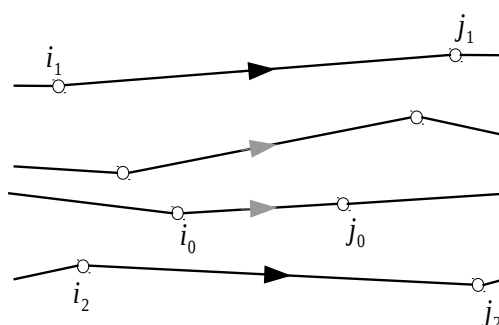


Figura 4.12 – Par de vértices mais custoso para uma fatia de tempo. As setas indicam o posicionamento em determinado momento e a direção de deslocamento dos veículos durante a fatia de tempo limitada pelo instante de partida de i_0 e o instante de chegada em j_0 . Os veículos mais dispersos durante a fatia partiram de i_1 e i_2 e se dirigem a j_1 e j_2 , respectivamente. A sua trajetória implica no aumento da distância relativa durante a fatia em questão. Portanto, nesse caso, o par mais custoso para a fatia de tempo é o par de destino, formado pelos vértices j_1 e j_2 .

Se os veículos mantêm distância constante, é adotado por definição o par de vértices determinante da posição final. Em nada interfere o fato de que as unidades estejam em deslocamento com duração maior que a duração da fatia, e a visita ao par de vértices selecionado ocorra de fato em momento anterior ou posterior. Mas essa situação torna possível que o mesmo par de vértices seja determinante da dispersão em diversas fatias subsequentes. Na figura, por exemplo, os mesmos vértices j_1 e j_2 continuam sendo responsáveis pela dispersão de outras fatias além da delimitada por j_0 . A ocorrência de tais repetições é observada, para que um par de vértices seja considerado apenas uma vez,

evitando movimentos redundantes. Os pares de vértices responsáveis pela dispersão de qualquer fatia, descartando as repetições, formam o conjunto que orienta as vizinhanças de aproximação das rotas.

A noção heurística que norteia os movimentos da vizinhança é a necessidade de quebrar a relação temporal existente entre os vértice mais custosos de cada fatia. Para isso, foram implementados movimentos referidos como *inserção anterior*, *inserção posterior* e *troca de vértices*. Os três movimentos partem da seleção de dois vértices i e j identificados como o par mais custoso de uma fatia. Enquanto o vértice i é mantido em sua rota, os seus vizinhos mais próximos presentes em $Nn(i)$ são envolvidos, um de cada vez, em alterações na rota de j . A Figura 4.13 ilustra os movimentos considerando o vértice $i' \in Nn(i)$, retirado de sua rota para ser inserido em posições anterior e posterior a j e também trocado de posição com j . Após aplicar os movimentos com todos os vértices em $Nn(i)$, são testados os movimentos em que os vértices em $Nn(j)$ modificam a rota de i .

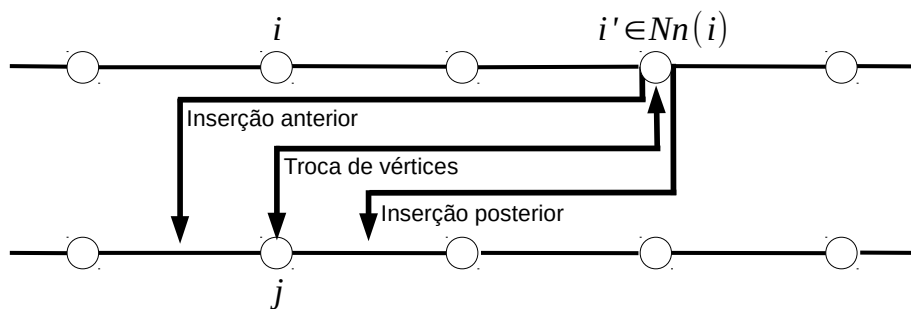


Figura 4.13 – Movimentos de inserção e troca.

É importante notar que a remoção de um terceiro vértice de sua posição original, como i' na Figura 4.13, também representa uma modificação nas rotas e tem impacto na função objetivo.

Outra vizinhança é gerada pelo movimento referido como *ciclo*. Esse movimento tem objetivo semelhante ao movimento de troca apresentado na figura, ou seja, de remover um dos vértices da sua posição (j) e substituí-lo por um vizinho do outro (i'). Mas no caso da troca, o vértice expulso (j na figura) é colocado na posição do seu substituto, sem que haja qualquer indicação de que esta seja uma boa posição. No movimento de ciclo, definidos os vértices mais custosos de uma fatia, i e j , e um terceiro vértice $i' \in Nn(i)$, j também será substituído por i' , a exemplo do movimento de trocas. Mas a posição ocupada por i' originalmente será ocupada por um vértice pertencente a $N(i')$, e assim sucessivamente, cada novo vértice substituído por outro selecionado dentre seus vizinhos, até que j esteja presente em uma dessas vizinhanças. O vértice j é então reinsertado no roteamento, e o ciclo é fechado. A Figura 4.14 ilustra o movimento de ciclo.

O movimento representado na figura exige a existência de um caminho de i' até j no

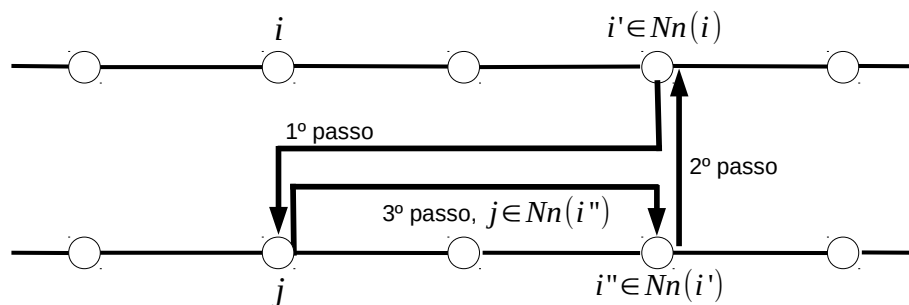


Figura 4.14 – Movimento de ciclo com um ciclo de tamanho 3. O vértice $i' \in Nn(i)$ é colocado no lugar do vértice j . A posição de i' é ocupada por um de seus vizinhos, i'' . Como j é vizinho de i'' , é reinserido na posição desse vértice e o ciclo se encerra.

qual um vértice sempre é sucedido por outro selecionado dentre seus vizinhos. Caminhos mínimos com essa configuração são identificados com o método de Floyd, descrito em (TAHA, 2006), aplicado sobre um grafo dirigido no qual arcos ponderados com peso unitário conectam cada vértice $i \in V$ aos seus vizinhos mais próximos em $Nn(i)$. Originalmente, o resultado do método é um conjunto de caminhos mais curtos, um para cada par de vértices no grafo se tal caminho existir. Para sua aplicação na vizinhança de ciclos, o método foi adaptado de modo a permitir que sejam descobertos todos os caminhos caso existam múltiplas alternativas com o mesmo comprimento mínimo entre dois vértices.

Uma vez que o número de vértices em cada conjunto Nn é restrito, o grafo dirigido gerado a partir desses conjuntos não será necessariamente fortemente conexo, e podem não existir caminhos ligando certos pares de vértices. Não havendo caminho entre um par de vértices i' e j , o movimento não é executado. Havendo múltiplos caminhos possíveis com o mesmo comprimento mínimo, são adotados no máximo dez, selecionados aleatoriamente. Ciclos de tamanho dois (i' substitui j e j substitui i') são ignorados, por representarem um movimento já contemplado na vizinhança de troca.

Os movimentos são aplicados em sequência, até a geração da primeira solução nova que melhore a função objetivo (*first improvement*), de modo que nem todos os movimentos são testados. Além disso, se a sequência de movimentos for sempre a mesma, sempre o mesmo vizinho será gerado. Por isso, todas as decisões que impactam na ordem de avaliação dos movimentos são tomadas de forma aleatória, como a ordem em que os pares i e j são retirados do conjunto de pares mais custosos a serem tratados, qual dos vértices do par i e j terá seus vértices mais próximos envolvidos nos movimentos primeiro, a ordem em que os vértices do conjunto $Nn(i)$ são aplicados, e a própria ordem em que os movimentos de troca e inserção são aplicados.

Finalmente, uma última vizinhança com objetivo de aproximar rotas é obtida com movimentos de trocas entre as rotas dentro de cada fatia. Essa vizinhança é referida como

trocas na fatia. Para cada instante determinado pela chegada de um veículo a um vértice, forma-se um conjunto composto por esse vértice e por um vértice de cada uma das outras rotas, a saber: os vértices onde estão parados os veículos em visita no instante referido e os vértices de onde partiram por último os veículos em deslocamento naquele momento. Todos os possíveis pares formados por vértices desse conjunto são trocado de rotas entre si.

4.2.4.2 Vizinhaça de redução de rotas

Entre dois veículos posicionados na garagem devido a rotas já encerradas, a distância relativa é igual a zero. Por consequência, a partir do momento em que o último veículo conclui sua rota, a máxima distância entre todos os veículos é nula e não há mais dispersão a ser acrescida à função objetivo. Por isso, é interessante a construção de roteamentos curtos, que reduzam o tempo de conclusão da última rota.

A Figura 4.15(a) ilustra um roteamento a ser percorrido por dois veículos. Elipses escurecidas destacam os pares de vértices que exibem proximidade em relação ao instante em que são visitados, e é possível perceber que, em relação ao sincronismo entre vértices vizinhos, as rotas estão bem organizadas. Mas a Figura 4.15(b) apresenta um roteamento alternativo, em que esses pares se mantém os mesmos, mas as rotas são mais curtas por evitar um cruzamento presente no roteamento anterior.

Os movimentos de vizinhaça que visam encurtar as rotas são elaborados sobre a seguinte ideia heurística: o conjunto de múltiplas rotas é considerado como uma rota única a ser cumprida pelos m veículos, e movimentos são efetuados sobre essa rota abstrata de modo a evitar cruzamentos. Essa ideia é ilustrada nas Figuras 4.15(c) e 4.15(d), em que os pares de vértices são considerados agrupamentos atômicos e o roteamento inteiro é reduzido a uma rota ligando esses agrupamentos. O mesmo cruzamento tratado nas Figuras 4.15(a) e 4.15(b) é eliminado na alteração efetuada sobre a rota na Figura 4.15(c) para gerar a rota na Figura 4.15(d). As alterações efetuadas nas rotas são inspiradas nos movimentos das vizinhaças clássicas 2-opt e 3-opt.

Para que a abstração tenha sentido, é necessário que uma eventual alteração seja aplicada de forma idêntica em todas as rotas da solução. Isso é fundamental para evitar que alterações visando a redução do roteamento causem prejuízo severo à outra característica abordada, de sincronismo entre veículos. A Figura 4.16 ilustra um caso em que o cruzamento entre arestas presente na Figura 4.15(a) só foi corrigido em uma rota. Percebe-se no destaque a formação de dois pares de vértices próximos do ponto de vista temporal, porém mais dispersos do que os pares presentes na solução original.

Obviamente, a construção de agrupamentos atômicos de vértices e a abstração de considerar diversos arcos como um apenas não é evidente na maioria dos casos práticos.

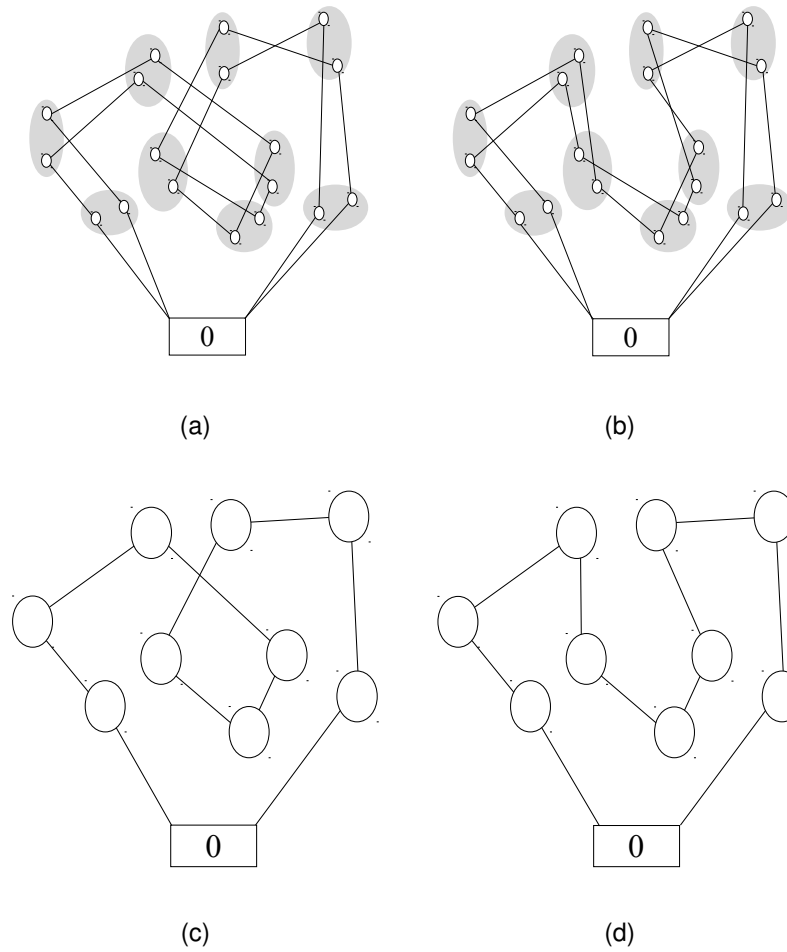


Figura 4.15 – mTSP-D e remoção de cruzamento, na solução com duas rota e na abstração que considera a rota única.

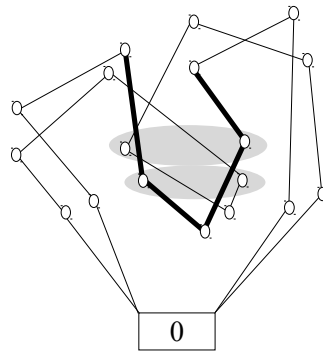


Figura 4.16 – Movimento de eliminação de cruzamento aplicado a apenas uma das rotas.

Existem, por exemplo, casos em que um arco (longo) é concomitante a vários de outra rota, ou que um vértice é visitado durante um intervalo em que todos os demais veículos estão em deslocamento. Por isso, a aplicação dos movimentos exige uma estratégia para a definição dos pontos em que arcos podem ser removidos para gerar soluções alternativas.

A aplicação de um movimento da vizinhança 2-opt a uma rota implica na remoção de dois arcos não adjacentes, sendo a rota reconectada com a adição de arcos diferentes, em procedimento repetido para todos os pares de arcos possíveis. Na aplicação de movimentos semelhantes a uma solução do mTSP-D, a definição de pontos de remoção de arcos é mais complexa. Ao invés de duas arestas, são removidos dois conjuntos de arestas. Esses conjuntos são construídos de modo que contenham uma aresta de cada rota e que essas arestas sejam pelo menos parcialmente concomitantes, ou seja, exista pelo menos um instante de tempo em que todas as arestas estejam sendo percorridas pelos respectivos veículos. Para fins de construção dos conjuntos, o tempo em que o veículo está parado em um vértice é computado como parte da aresta que parte do vértice. A Figura 4.17 ilustra os possíveis conjuntos de arestas, formados conforme essa descrição. É possível perceber que não são considerados conjuntos após o momento em que o primeiro veículo retornou à garagem.

Estabelecidos os conjuntos de arestas conforme a figura, é possível explicar as vizinhanças de redução de rotas. A primeira vizinhança será referida como m2-opt, versão do 2-opt clássico voltada para as múltiplas rotas do mTSP-D. Nela, os movimentos de remoção de arestas e reconstrução das rotas são feitos para todas as combinações de tais conjuntos em pares. Mas dois conjuntos formam um par válido apenas se não apresentarem interseção (o mesmo arco nos dois conjuntos) e se pelo menos para uma rota apresentarem arcos não adjacentes. Essa última restrição é necessária para que o movimento resulte em alteração em pelo menos uma rota.

Após a remoção de arestas, as rotas podem ser reconstruídas pela reinserção das arestas retiradas, o que sempre resulta na mesma solução original e por isso esse movimento é ignorado. O movimento m2-opt restante é exemplificado na Figura 4.18, através

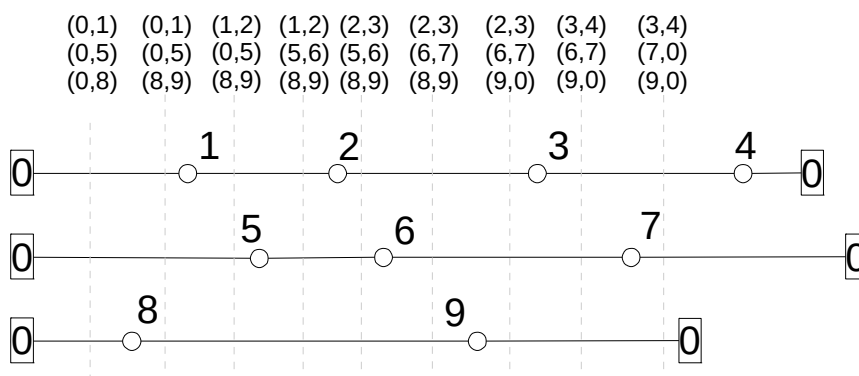


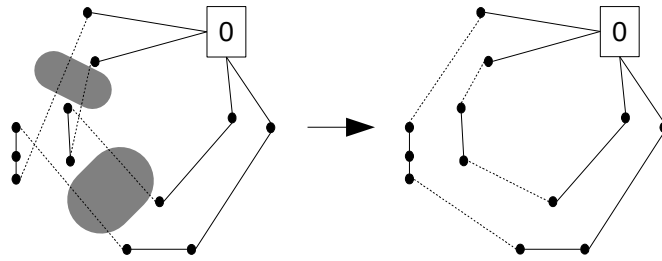
Figura 4.17 – Conjuntos de arestas que podem ser removidas para aplicação de um movimento de redução de rotas. As arestas cruzadas por uma mesma linha vertical formam um conjunto, e são listadas na parte superior. Os vértices são representados sem consideração do tempo de serviço, pois para fins de geração dos conjuntos esse tempo é considerado como parte do tempo das arestas que sucedem a visita.

de um caso em que não há arestas adjacentes no par de conjuntos, e outro em que tal situação ocorre. No segundo caso, é interessante destacar que embora a rota interna não sofra alteração, em função das arestas adjacentes, a rota externa muda e portanto a solução gerada é diferente.

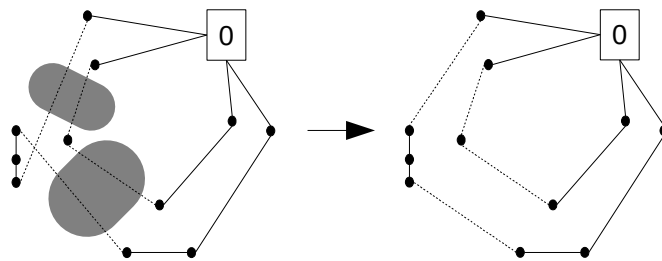
De forma similar, é implementada uma vizinhança referida como m3-opt, inspirada na clássica vizinhança 3-opt. Nesse caso, os movimentos devem ser aplicados a partir da composição de trios de conjunto de arestas, de modo que em cada rota sejam removidas três arestas. É importante que exista pelo menos uma rota em que as arestas indicadas nos conjuntos não sejam adjacentes. A recomposição das rotas se dá pela inserção de arestas de modo a reconectar as rotas, e existem oito formas possíveis. Uma resulta nas rotas originais, e outras três passam pela reinserção de uma aresta e na alteração apenas das outras duas, em movimentos equivalentes a outros já implementados na vizinhança m2-opt. Os quatro movimentos resultantes são exemplificados na Figura 4.19, e se diferenciam pelas formas de recomposição do roteamento.

No exemplo, as arestas a serem removidas de cada rota da solução inicial são identificadas por (i_k, i_k') , (j_k, j_k') , (l_k, l_k') , sendo k o veículo da rota. A escolha dessas arestas para remoção nos movimentos ilustrados permite exemplificar três situações distintas: quando não há arestas adjacentes entre as arestas removidas, no caso da rota mais externa ($k = 1$); quando todas as arestas retiradas são adjacentes, no caso da rota mais interna ($k = 3$); e quando duas das arestas são adjacentes, no caso da rota colocada entre as outras ($k = 2$). Cada movimento se caracteriza pelas arestas inseridas, conforme a Tabela 4.1.

Todo movimento efetuado sobre as rotas tem um impacto na dispersão. Porém,



(a) Caso em que os conjuntos não apresentam arestas adjacentes

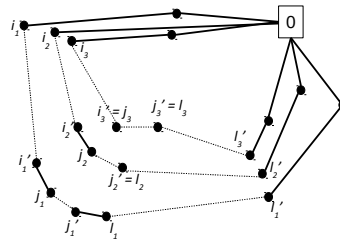


(b) Caso em que os conjuntos tem arestas adjacentes na rota interna

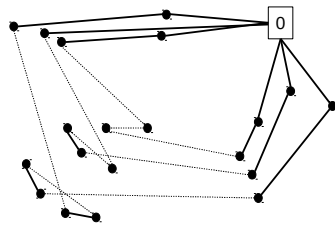
Figura 4.18 – O movimento m2-opt, aplicado a roteamentos com duas rotas. Cada figura representa o movimento aplicado à solução apresentada à esquerda, gerando a solução apresentada à direita. As arestas pontilhadas representam as que são retiradas no roteamento da esquerda e as que são inseridas no roteamento da direita. As áreas sombreadas tranpassam as arestas que formam um conjunto.

Tabela 4.1 – Arestas inseridas em cada movimento da vizinhança m3-opt.

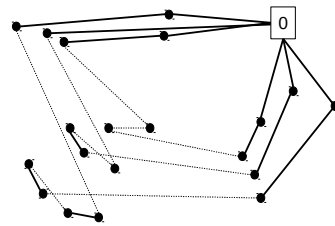
i	N_i
Movimento 1	$(i_k, j_k'), (l_k, i_k'), (j_k, l_k')$
Movimento 2	$(i_k, l_k), (j_k', i_k'), (j_k, l_k')$
Movimento 3	$(i_k, j_k), (i_k', l_k), (j_k', l_k')$
Movimento 4	$(i_k, j_k'), (l_k, j_k), (i_k', l_k')$



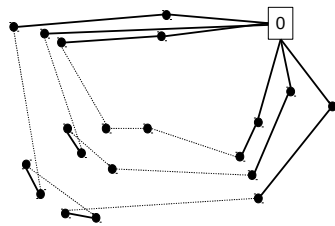
(a) Rotas iniciais



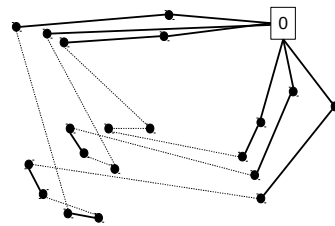
(b) Movimento 1



(c) Movimento 2



(d) Movimento 3



(e) Movimento 4

Figura 4.19 – Os quatro movimentos que definem a vizinhança m3-opt, aplicados a um roteamento com três rotas.

o grande número de movimentos possíveis levou à adoção de uma estratégia heurística para reduzir o número de movimentos que serão efetivamente realizados e que terão sua dispersão calculada. Essa estratégia, usada tanto no m2-opt quanto no m3-opt, consiste em verificar a variação na duração total do roteamento causada por cada movimento, já que reduzir esse tempo é a razão da implementação da vizinhança. A duração é definida pelo momento de retorno do último veículo à garagem, e esse valor pode ser obtido apenas pela avaliação da duração das arestas removidas e inseridas. Se o tempo total de roteamento aumentou, o movimento é descartado. Caso tenha diminuído, o movimento é marcado como interessante, e as informações que o definem são guardadas para futura aplicação.

O ganho no tempo total é apenas um indicador, pois não contempla a avaliação das alterações em relação ao sincronismo das rotas. Por isso, os movimentos considerados interessantes são ordenados de acordo com a magnitude da redução da duração do roteamento, do maior para o menor ganho de tempo, e os movimentos passam a ser efetuados, gerando novas soluções para as quais a função objetivo definida na Equação (4.2) é aplicada. A busca termina quando a primeira melhoria na dispersão é encontrada.

O mTSP-D foi descrito a partir de um grafo não dirigido, havendo simetria no conjunto de arestas. Isso implica que uma rota terá a mesma duração se percorrida em qualquer dos dois sentidos possíveis. Mas é importante perceber que uma solução completa, considerando todas as rotas, não é simétrica. Os tempos de chegada de cada veículo na garagem são provavelmente diferentes, de modo que o percursos no sentido inverso modifica a sincronização dos veículos e modifica o valor de avaliação da dispersão. Em função dessa assimetria, e como os movimentos do m2-opt e do m3-opt provocam a inversão de trechos de rotas, após cada movimento o roteamento é avaliado considerando os seus dois sentidos possíveis.

4.2.4.3 Procedimento de Busca Local com as Vizinhanças Descritas

O procedimento de Busca Local é realizado sobre as duas primeiras populações, a gerada pelo método construtivo e a primeira gerada por operadores genéticos, e, a partir de então, a cada β gerações. A cada execução da Busca Local, a parcela da população composta pelos $lsSize$ melhores indivíduos é submetida a um processo de melhoria pela exploração sistemática de suas vizinhanças, conforme descrito no Algoritmo 4.

Algoritmo 4 Pseudocódigo do método de Busca Local.

Input:

curPop - população corrente
nIter - número de iterações
nNeigh - número de esquemas de vizinhança
lsSize - tamanho da parcela da população para explorar
NI - conjunto de todas as soluções cuja vizinhança já foi explorada

sort(*curPop*) comentário: ordem decrescente de aptidão calculada conforme Equação (4.37)

```

for j from 0 to lsSize - 1 do
  x* ← curPop[j]
  it ← 0
  i ← 0
  while it < nIter and i < nNeigh do
    if x* ∈ NIi then
      i ← i + 1
    else
      Gerar x' ∈ Ni(x*)
      if ObjectiveFunction(x') < ObjectiveFunction(x*) then
        x* ← x'
        i ← 0
      else
        NIi ← NIi ∪ {x*}
        i ← i + 1
      end if
    end if
    it ← it + 1
  end while
  curPop[j] ← x*
end for
  
```

O algoritmo começa com a ordenação dos indivíduos da população de acordo com seus valores de avaliação, de modo que a Busca Local seja aplicada aos melhores. O uso de apenas um subconjunto da população foi adotada para equilibrar o *trade-off* entre o número de gerações possíveis de obter com o tempo disponibilizado e a diversidade presente em uma população. Por ter grande custo computacional, a Busca Local sobre toda a população exige populações pequenas, sob pena de que o LSGA encerre com poucas gerações. Por outro lado, a população pequena reduz a diversidade representada e favorece a convergência prematura.

Para cada indivíduo selecionado, a busca inicia com a adoção da solução representada como a solução incumbente x^* inicial. A busca é limitada para cada solução por um número máximo de iterações permitidas $nIter$. Esse valor estabelece um critério que pode causar a parada no procedimento mesmo antes de chegar a um indivíduo sem vizinhos de melhor avaliação. Na prática, o valor é definido de forma a restringir a Busca Local quando essa se configura como uma série de movimentos de pequena melhora, resultando em muitos passos onerosos em termos de tempo computacional consumido.

O outro parâmetro que limita o número de iterações do laço aplicado a uma solução

é o número $nNeigh$ de vizinhanças que podem ser exploradas. Esse último parâmetro corresponde a cardinalidade de um conjunto de esquemas de vizinhanças, descrito na Tabela 4.2. Esses esquemas são ordenados de modo a estabelecer uma hierarquia que sistematiza a execução de todos os movimentos de aproximação e redução de rotas descritos. O índice que indica a vizinhança corrente é inicializado em 0 sempre que uma nova solução é selecionada para exploração, o que indica que a Busca Local começa pelos movimentos de inserção e trocas.

Tabela 4.2 – Tipos de vizinhanças exploradas durante a Busca Local.

i	N_i
0	inserções anterior e posterior e trocas
1	m2-opt
2	ciclos
3	m3-opt
4	trocas na fatia

Antes de explorar a solução incumbente x^* com os movimentos da vizinhança N_i , é necessário verificar se essa solução já está contida no conjunto NI_i das soluções previamente exploradas com a vizinhança N_i e que não puderam ser melhoradas. Se $x^* \in NI_i$, é sabido que sua exploração com N_i é inócua, e o índice que indica o esquema de vizinhança corrente é incrementado. Caso contrário, é feita a busca na vizinhança N_i pelo primeiro vizinho com avaliação melhor que x^* . Se um vizinho com tal característica for encontrado, é adotado como incumbente e a vizinhança ativa volta a ser a de maior prioridade (N_0). Caso contrário, x^* é adicionado à lista NI_i e o índice da vizinhança ativa é incrementado.

Quando a exploração de um indivíduo é concluída, a solução incumbente é armazenada nesse indivíduo, substituindo a solução nele contida no início da Busca Local.

4.3 RESULTADOS E DISCUSSÕES

Nessa seção, são detalhados os experimentos realizados e os resultados obtidos na solução do mTSP-D. Os experimentos computacionais foram realizados em uma máquina *dual chip* com dois processadores Intel E5-2680, 64 GB de memória RAM e *clock* de 2,7 GHz. Os algoritmos foram implementados em Java e executados na implementação do Oracle Hotspot, versão 1.8.0 e revisão 72, sobre Linux O.S.. Nenhum tipo de paralelismo foi implementado, então cada execução usou apenas um processador.

4.3.1 Instâncias e Parâmetros

O mTSP-D é um problema novo e requer um conjunto específico de instâncias para avaliação. As instâncias utilizadas foram construídas a partir de dezesseis instâncias propostas em (NECULA; BREABAN; RASCHIP, 2015) para o mTSP. Essas dezesseis instâncias, por sua vez, foram criadas a partir da adaptação para o mTSP de quatro instâncias originalmente elaboradas para o TSP e que fazem parte do repositório TSPLIB, a saber: *eil51*, *berlin52*, *eil76* e *rat99*. O número representado no nome da instância indica o seu número de vértices. Para a geração das instâncias do mTSP, a cada instância original é adicionada informação referentes ao número de veículos/rotas, que pode ser 2, 3, 5 ou 7. Dessa forma, cada uma das instâncias escolhidas do TSPLIB gerou quatro instâncias para o mTSP.

O mTSP-D exige ainda a inclusão de informação referente ao tempo de serviço em cada vértice. Cada uma das dezesseis instâncias de (NECULA; BREABAN; RASCHIP, 2015) resultou então em três novas instâncias, uma com tempo de serviço uniforme de uma unidade de tempo, outra com tempo de serviço uniforme de dez unidades de tempo, e uma última com tempo aleatório gerado entre uma e dez unidades de tempo, com probabilidade uniforme. Essas três alternativas de tempo de serviço são identificadas nas instâncias como “s1”, “s10” e “sR”, respectivamente, e o nome de cada uma das 48 instâncias para o mTSP-D indica a instância original do TSPLIB (que contempla o número de vértices), seguido do número de veículos e da especificação do tempo de serviço. A instância “rat99-m5s10”, por exemplo, possui 99 vértices, cinco veículos e tempos de serviço uniformes iguais a dez. Todas as instâncias criadas estão disponíveis em <http://github.com/gdhein/mTSP-D>.

Os procedimentos descritos para o LSGA são definidos com diversos parâmetros, e os valores utilizados são descritos na Tabela 5.2. O ajuste dos parâmetros do Algoritmo Genético foi feito de forma empírica, a partir de um subconjunto de seis instâncias.

Tabela 4.3 – Parâmetros do LSGA.

parâmetro	descrição	valor
β	número de gerações entre execuções de Busca Local	1000
cp	probabilidade de cruzamento	95%
es	tamanho da elite	50
$lsSize$	número de soluções exploradas em cada Busca Local	500
$maxTime$	tempo máximo de execução	$\lceil n \cdot m / 15 \rceil$
$nIter$	número de iterações na Busca Local	50
$ Nn(i) $	tamanho do conjunto de vizinhos mais próximos de cada vértice	$\max(2m, n/10)$
ps	tamanho da população	1000

4.3.2 Resultados Computacionais

A Tabela 4.4 apresenta os resultados dos experimentos computacionais, resumindo os resultados de vinte execuções para cada instância através do valor da função objetivo da melhor solução, o valor médio entre as vinte soluções para cada instância, o desvio padrão e o coeficiente de variação (CV). O coeficiente de variação representa a razão entre o desvio padrão e a média e é interessante para a avaliação da convergência das soluções.

O coeficiente de variação máximo encontrado foi de 3,12%, única ocorrência acima de 3%, referente à instância *rat99-m3s1*. O valor médio para o coeficiente de variação, considerando o valor obtido para cada uma das 48 instâncias, resultou em 1,29%, o que atesta a consistência das soluções encontradas entre as vinte execuções. Esse valor aumenta para 1,91% se forem considerados apenas as instâncias com 99 vértices, e diminui para 0,97% para instâncias com 51 vértices. Se as 48 instâncias forem agrupadas por tempo de serviço, o CV médio é de 1,52% para uma unidade de tempo, 1,10% para dez unidades de tempo, e 1,25% para as instâncias com tempo de serviço gerado entre uma e dez unidades de tempo. Além disso, se as instâncias forem divididas em 16 conjuntos, agrupando as três que compartilham o número de vértices e veículos e divergem no tempo de serviço, tem-se que em 10 dos 16 conjuntos o maior coeficiente de variação está atrelado à instância de tempo de serviço de uma unidade de tempo. Tais resultados parecem apontar que instâncias com tempos de serviço menores em relação aos tempos de deslocamento apresentam maior dificuldade para resolver.

De modo a avaliar a eficácia do LSGA na melhoria da população inicial, foi calculado, para cada execução, o *gap* entre a solução final do LSGA e a melhor solução da população inicial gerada pelo algoritmo construtivo. Considerando as 960 execuções, o *gap* médio foi de 44,72%. A instância com menor melhoria apresentou *gap* de 27,55%, e a com maior melhoria teve *gap* de 69,82%.

O *gap* entre duas soluções sol_a e sol_b é calculado como $gap = \left(1 - \frac{sol_a}{sol_b}\right) \cdot 100$. No caso do parágrafo anterior, sol_a corresponde a solução retornada pela LSGA e sol_b corresponde a melhor solução do algoritmo construtivo. Um *gap* positivo, portanto, indica que a avaliação da primeira solução é melhor do que a avaliação da segunda.

O mesmo *gap* foi usado na avaliação da sensibilidade do LSGA em relação a algumas variações no procedimento. Primeiramente, o LSGA foi executado sem o procedimento de Busca Local, sendo reduzido a um Algoritmo Genético. O *gap* entre o LSGA e o GA é de 6,2%, em média, se forem consideradas apenas as melhores soluções de cada método com cada instância, e aumenta para 10,2% se forem consideradas todas as execuções de todas as instâncias.

Outras configurações alternativas avaliadas contemplam a execução da Busca Local com apenas uma das duas concepções de vizinhança, descritas nas secções 4.2.4.1

Tabela 4.4 – Resultados Computacionais.

Instância	Tempo ^a	Melhor Solução ^b	Média ^c	σ ^d	CV ^e
eil51-m2s1	7	1353,34	1416,44	42,26	1,49
eil51-m3s1	11	2075,95	2152,37	45,47	1,06
eil51-m5s1	17	2749,71	2885,67	81,17	1,41
eil51-m7s1	24	3329,20	3364,19	28,66	0,43
eil51-m2s10	7	2956,53	3017,85	54,92	0,91
eil51-m3s10	11	3520,16	3676,89	69,09	0,94
eil51-m5s10	17	4229,19	4349,97	77,55	0,89
eil51-m7s10	24	4507,23	4673,05	91,54	0,98
eil51-m2sR	7	2151,14	2201,63	27,45	0,62
eil51-m3sR	11	2824,16	2947,68	67,35	1,14
eil51-m5sR	17	3348,95	3456,83	94,15	1,36
eil51-m7sR	24	3998,28	4062,61	35,73	0,44
berlin52-m2s1	7	520237,12	548690,43	22679,85	2,07
berlin52-m3s1	11	737613,41	746808,25	9579,86	0,64
berlin52-m5s1	18	1036774,61	1054453,90	15149,00	0,72
berlin52-m7s1	25	1300384,19	1330210,69	19778,24	0,74
berlin52-m2s10	7	542477,40	573862,82	15876,56	1,38
berlin52-m3s10	11	763611,17	772030,46	9644,64	0,62
berlin52-m5s10	18	1064817,80	1079343,82	10804,12	0,50
berlin52-m7s10	25	1323599,86	1359972,94	23853,50	0,88
berlin52-m2sR	7	524510,93	566487,05	17857,83	1,58
berlin52-m3sR	11	749047,62	761899,92	15449,40	1,01
berlin52-m5sR	18	1049964,48	1071314,52	11341,11	0,53
berlin52-m7sR	25	1316034,72	1335764,85	10325,07	0,39
eil76-m2s1	11	1651,83	1751,53	74,85	2,14
eil76-m3s1	16	2179,53	2483,74	102,91	2,07
eil76-m5s1	26	2895,14	3085,44	96,00	1,56
eil76-m7s1	36	3218,26	3417,25	91,83	1,34
eil76-m2s10	11	3815,96	3915,11	43,96	0,56
eil76-m3s10	16	4438,33	4663,70	121,94	1,31
eil76-m5s10	26	4938,73	5208,22	127,68	1,23
eil76-m7s10	36	5017,31	5219,94	128,71	1,23
eil76-m2sR	11	2838,75	2958,35	55,52	0,94
eil76-m3sR	16	3446,35	3597,59	69,15	0,96
eil76-m5sR	26	3984,39	4194,43	115,20	1,37
eil76-m7sR	36	4129,37	4426,37	136,33	1,54
rat99-m2s1	14	5327,53	5924,58	340,80	2,88
rat99-m3s1	20	8602,45	9583,69	598,84	3,12
rat99-m5s1	33	12927,75	13486,67	498,68	1,85
rat99-m7s1	47	13589,17	14014,74	228,70	0,82
rat99-m2s10	14	10174,17	10553,88	266,61	1,26
rat99-m3s10	20	13691,69	14708,89	636,17	2,16
rat99-m5s10	33	17164,85	18748,14	580,24	1,55
rat99-m7s10	47	18528,39	19103,09	474,27	1,24
rat99-m2sR	14	7849,15	8432,41	374,96	2,22
rat99-m3sR	20	11473,44	12464,34	714,34	2,87
rat99-m5sR	33	15603,62	16471,01	568,20	1,72
rat99-m7sR	47	16273,46	16942,80	430,73	1,27

considerando 20 execuções por instância: ^a tempo de cada execução da instância, em minutos, ^b valor da função objetivo da melhor solução, ^c média dos valores de função objetivo nas vinte soluções para a instância, ^d desvio padrão, ^e coeficiente de variação.

e 4.2.4.2. Foi definido como LSGA-rr a configuração em que a Busca Local foi executada apenas com as vizinhanças de redução de rotas (m2-opt e m3-opt), e como LSGA-ar a configuração em que a Busca Local foi executada apenas com vizinhanças de aproximação das rotas (inserções e trocas, ciclos e trocas na fatia). O *gap* entre a LSGA e LSGA-ar é de 6,2%, em média, para os melhores resultados de cada instância, e 8,5%, em média, para todas as soluções, enquanto entre a LSGA e LSGA-rr é de 2,6% para os melhores resultados e 5,4% para todas as soluções.

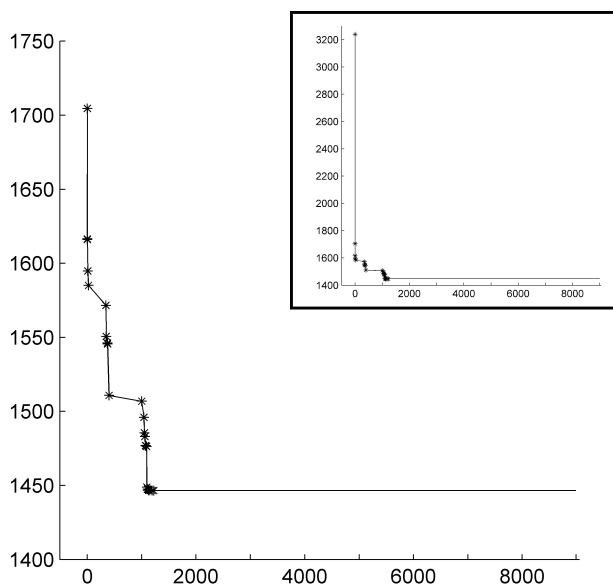
A ordem em que as vizinhanças são aplicadas também foi modificada para avaliação: as vizinhanças de redução de rotas primeiro, ou as vizinhanças de aproximação de rotas primeiro. Quando comparados com o LSGA padrão, cuja ordem das vizinhanças está descrita na Tabela 4.2, estas apresentam piora tanto na avaliação da melhor quanto na média das execuções, mas a diferença média é sempre inferior a 1%.

Sempre que foi feita uma comparação do LSGA com alguma configuração alternativa, a mesma população inicial foi usada para ambos os métodos, de modo que não há possibilidade de interferência da qualidade dessa população nos resultados obtidos.

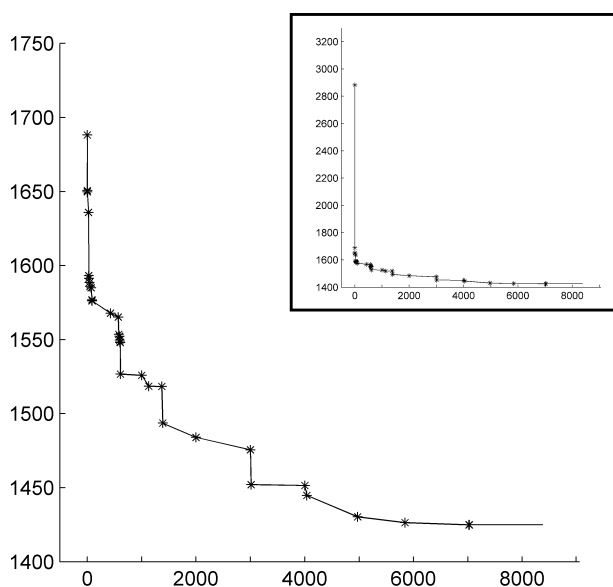
Foi realizada ainda uma avaliação do andamento do processo evolutivo durante o tempo de execução, definido *a priori* para todas as execuções de uma instância em função de m e n . A Figura 4.20 apresenta um gráfico com a evolução da incumbente em duas execuções com a mesma instância, uma em que a solução final é gerada com relativamente poucas gerações, e outra com a solução final gerada mais tardiamente. Destaque-se que essa instância é a que apresenta, considerando a média de suas vinte execuções, a geração da solução final mais cedo dentro do processo evolutivo, conforme apontado adiante na Tabela 4.5. Na Figura 4.20(a), existem 24 trocas de incumbente, considerando a substituição da incumbente inicial, gerada na população inicial pelo algoritmo construtivo. Já na Figura 4.20(b) esse número sobe para 31. Entre ambas, o resultado final em termos de valor de função objetivo apresenta pequena variação, sendo que a dispersão da solução da execução ilustrada na Figura 4.20(a) apresenta valor 1,5% superior (pior, portanto) em relação à melhor solução da outra execução.

A Figura 4.21 apresenta evolução semelhante, mas relativa a execução de melhor solução da instância rat99-m5sA. Neste caso, a execução termina após uma geração em que foi realizada Busca Local, e nessa busca foi obtida a solução final. Como as gerações com Busca Local demandam mais tempo que as demais, é relativamente comum entre as instâncias maiores (99 vértices ou 76 vértices e cinco ou sete veículos) que a execução termine em gerações desse tipo, embora não seja regra que a execução de Busca Local melhore a incumbente.

Os dados relativos a comparação da geração de surgimento da melhor solução de uma execução em relação ao total de gerações realizadas dentro do tempo disponibilizado são apresentados de forma resumida na Tabela 4.5. Observa-se que uma geração mais tardia nas instâncias maiores, especialmente nas instâncias com 99 vértices e três ou mais



(a) execução com surgimento mais adiantado da solução final



(b) execução com surgimento mais tardio da solução final

Figura 4.20 – Evolução da incumbente durante duas execuções relativas à instância *eil51-m2s1*. O eixo vertical representa o valor de função objetivo, e o horizontal o número da geração dentro do processo evolutivo. Mesmo não havendo nova incumbente, a linha segue para indicar as gerações posteriores ao surgimento da solução final. Para facilitar a visualização, a primeira solução, gerada pelo algoritmo construtivo, é ignorada no gráfico maior. A configuração considerando tal solução podem ser observada no canto superior direito de cada figura.

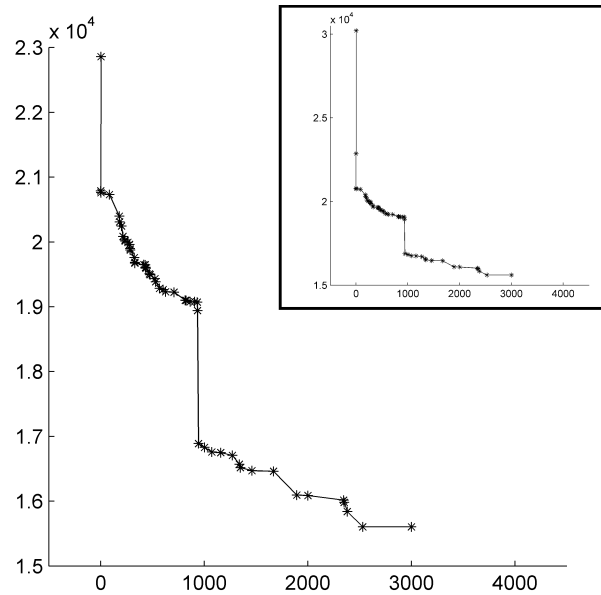


Figura 4.21 – Evolução da incumbente durante a execução de melhor solução da instância rat99-m5sA.

veículos, e nas instâncias com 76 vértices e cinco ou sete veículos. Destaca-se ainda o conjunto de instâncias com 51 vértices, onde em três casos a geração média de surgimento da melhor solução também se aproxima da geração final.

Como o mTSP-D é um problema novo, com uma função objetivo baseada em uma métrica também original, é importante avaliar as soluções de um ponto de vista mais subjetivo, verificando a configuração das rotas frente à característica desejada de proximidade entre veículos.

A Figura 4.22 permite a visualização da configuração apresentada pela melhor solução encontrada pelo LSGA para duas instâncias do mTSP-D. Nas primeiras figuras, 4.22(a) e 4.22(b), é possível observar que as rotas correspondentes são altamente correlacionadas no espaço. Na Figura 4.22(c), a mesma solução da Figura 4.22(b) é ilustrada, mas agora apresentando o progresso no tempo, mostrando claramente que a proximidade espacial também se manifesta como proximidade temporal, ou seja, a dispersão dos veículos é minimizada.

Os maiores distanciamentos entre rotas são causados por vértices remotos, onde não há vértice próximo para que outro veículo se aproxime. A Figura 4.23 permite observar como as soluções são construídas de modo a minimizar o impacto dos vértices distantes dos demais, ao combinarem deslocamentos que colocam os veículos novamente agrupados após o atendimento às porções do espaço onde os vértices se apresentam esparsos. Na figura, é possível observar os veículos agrupados logo antes e logo após os momentos em que apresentam considerável dispersão para atenderem tanto os vértices do canto inferior direito quanto os vértices do topo da figura. A alternativa de deixar apenas um veículo responsável pelos vértices distantes é pior para a avaliação da solução, por resultar

Tabela 4.5 – Momento do surgimento da melhor solução e sua relação com o total de gerações.

Instância	Geração ^a	Total de gerações ^b	% posterior ^c
eil51p2eq10u.tsp	4779,80	8316,30	42,52
eil51p2eq1u.tsp	3732,35	8667,85	56,94
eil51p2eqA.tsp	4836,40	8858,15	45,40
eil51p3eq10u.tsp	3607,10	7860,65	54,11
eil51p3eq1u.tsp	5107,95	7487,70	31,78
eil51p3eqA.tsp	3765,65	8339,15	54,84
eil51p5eq10u.tsp	3655,55	3966,70	7,84
eil51p5eq1u.tsp	3523,35	4117,60	14,43
eil51p5eqA.tsp	3454,15	4038,80	14,48
eil51p7eq10u.tsp	2751,00	3001,35	8,34
eil51p7eq1u.tsp	3651,00	4001,75	8,76
eil51p7eqA.tsp	2601,00	3252,50	20,03
berlin52p2eq10u.tsp	4403,90	7349,35	40,08
berlin52p2eq1u.tsp	5492,75	7536,20	27,12
berlin52p2eqA.tsp	5596,50	7490,95	25,29
berlin52p3eq10u.tsp	5313,50	7845,40	32,27
berlin52p3eq1u.tsp	4999,85	8005,45	37,54
berlin52p3eqA.tsp	6037,75	7959,70	24,15
berlin52p5eq10u.tsp	4726,85	5924,55	20,22
berlin52p5eq1u.tsp	5458,95	6088,20	10,34
berlin52p5eqA.tsp	4849,90	6001,90	19,19
berlin52p7eq10u.tsp	4101,00	5681,90	27,82
berlin52p7eq1u.tsp	5201,00	5963,05	12,78
berlin52p7eqA.tsp	5101,00	5879,80	13,25
eil76p2eq10u.tsp	3935,50	4858,30	18,99
eil76p2eq1u.tsp	3603,80	4950,60	27,20
eil76p2eqA.tsp	4111,90	5333,65	22,91
eil76p3eq10u.tsp	3963,35	4771,35	16,93
eil76p3eq1u.tsp	3788,15	4589,05	17,45
eil76p3eqA.tsp	3246,00	4540,95	28,52
eil76p5eq10u.tsp	3165,70	3369,30	6,04
eil76p5eq1u.tsp	3742,10	3861,85	3,10
eil76p5eqA.tsp	2950,00	3127,95	5,69
eil76p7eq10u.tsp	3001,00	3115,40	3,67
eil76p7eq1u.tsp	4101,00	4256,55	3,65
eil76p7eqA.tsp	3218,70	3494,75	7,90
rat99p2eq10u.tsp	2730,60	3315,40	17,64
rat99p2eq1u.tsp	2917,60	3747,05	22,14
rat99p2eqA.tsp	3013,80	3552,75	15,17
rat99p3eq10u.tsp	2782,95	2859,05	2,66
rat99p3eq1u.tsp	3161,10	3350,60	5,66
rat99p3eqA.tsp	2729,70	2967,80	8,02
rat99p5eq10u.tsp	2561,40	2694,80	4,95
rat99p5eq1u.tsp	3530,25	3663,45	3,64
rat99p5eqA.tsp	2901,00	2915,30	0,49
rat99p7eq10u.tsp	3101,00	3402,15	8,85
rat99p7eq1u.tsp	4751,00	4960,90	4,23
rat99p7eqA.tsp	3801,00	3962,05	4,06

considerando 20 execuções por instância: ^a geração média de surgimento da melhor solução, ^b média do total de gerações nas execuções com a instância, ^c percentual do número de gerações total posterior ao surgimento da melhor solução. Os percentuais abaixo de 10% aparecem em destaque.

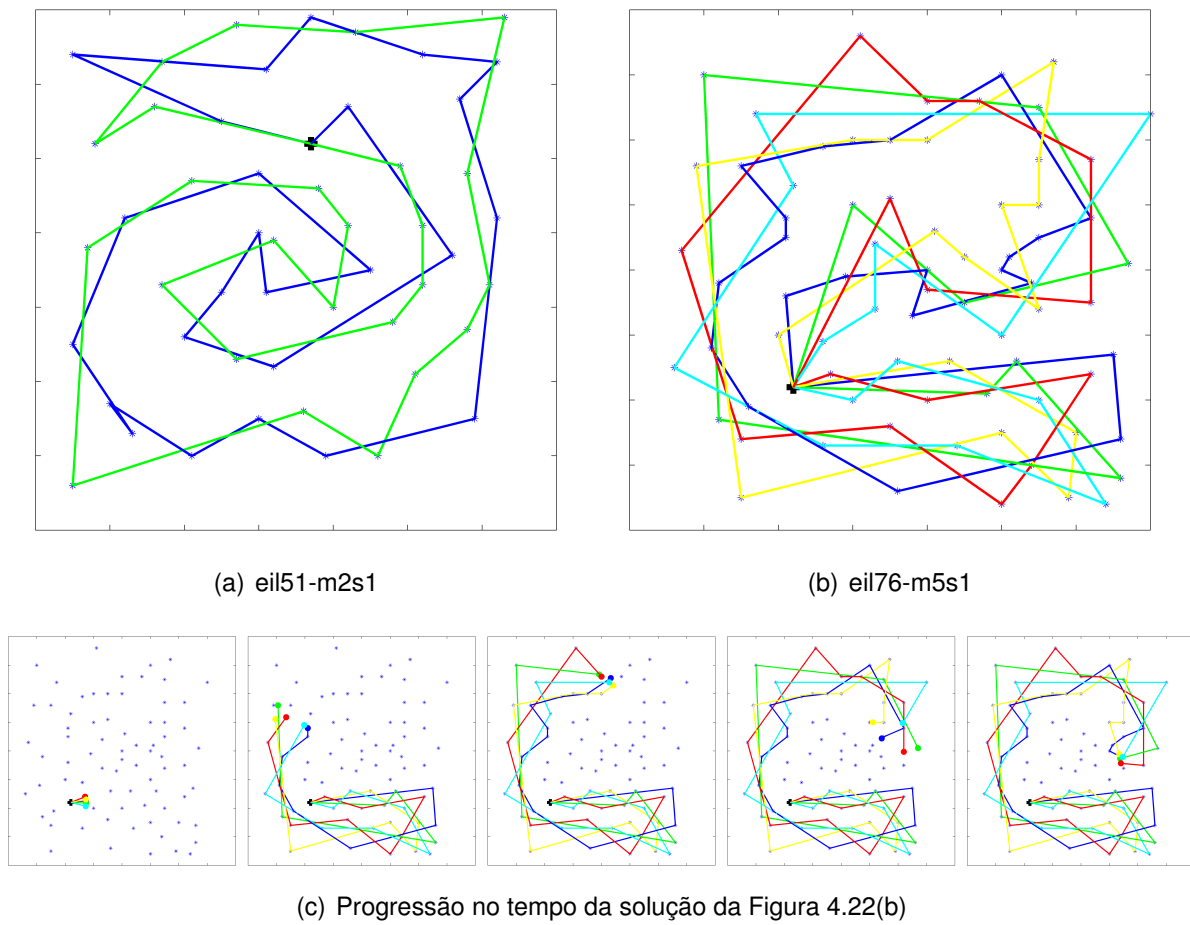


Figura 4.22 – Melhores soluções encontradas para as instâncias eil51-m2s1 e eil76-m5s1. Nas figuras, estão representadas as rotas percorridas por cada veículo. Uma solução mTSP-D produz rotas correlacionadas no tempo e no espaço, minimizando a dispersão.

atraso e, por consequência, distanciamento dos demais para a sequência do roteamento.

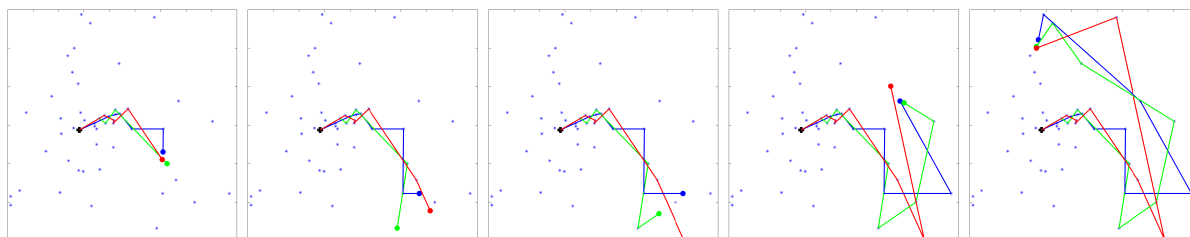


Figura 4.23 – Solução com alta dispersão momentânea para atender vértices remotos. Trata-se de uma solução para a instância berlin52-m3s1. Na primeira imagem, os veículos apresentam-se agrupados, antes de começarem a atender os vértices do canto inferior direito. O atendimento aos vértices dessa região se dá pela atribuição de um vértice distante para cada veículo, de modo que a ocorre dispersão momentânea. Como todos estiveram em deslocamento para posições distantes, não há atraso considerável de um em relação aos demais para a sequência, e os vértices aparecem próximos novamente na quarta imagem. Há apenas um pequeno adiantamento de um dos veículos, que se destina a outro vértice remoto. Por fim, após novo afastamento em função de um vértice, os veículos voltam a se agrupar.

É importante perceber como uma solução mTSP-D difere de uma solução mTSP que tem como função objetivo a redução de custos de deslocamento. Como exemplo, é destacada a situação que pode ser observada na visita aos dois vértices posicionados mais próximos ao canto superior direito na Figura 4.22(a). Contrariamente ao que se esperaria em um problema com objetivo de redução de custos, os dois vértices são visitados por veículos distintos, mesmo que o primeiro veículo cruze praticamente sobre o vértice visitado pelo segundo. No caso do mTSP-D, a vantagem da solução dada pelo LSGA quando considerada a proximidade espacial pode ser facilmente inferida a partir da Figura 4.22(a), mas é interessante perceber também a vantagem para a proximidade temporal.

Para ilustrar essa vantagem, a Figura 4.24 apresenta duas soluções parcialmente executadas, com o referido vértice destacado e designado para diferentes veículos. A solução da Figura 4.24(a) mostra a mesma solução encontrada pelo LSGA e apresentado na Figura 4.22(a), porém em uma configuração com o roteamento apenas parcialmente percorrido. A solução da Figura 4.24(b) mostra a situação parcial de uma solução alternativa, em que o vértice em foco foi designado para o outro veículo, que passava próximo a ele na configuração anterior. É possível observar que na solução alternativa um dos veículos resulta atrasado em relação ao outro, enquanto na solução original os veículos prosseguem próximos para o restante da execução das rotas planejadas.

A mesma solução exibe uma rota que apresenta um cruzamento, localizado no canto inferior esquerdo e ilustrado de forma ampliada na Figura 4.25. Esta também é uma situação indesejável em problemas que visam minimização de custo de deslocamento. Mas para o mTSP-D este é um movimento interessante para a manutenção da proximidade temporal. Essa ordem de visita aos vértices atrasa o respectivo veículo, permitindo que o

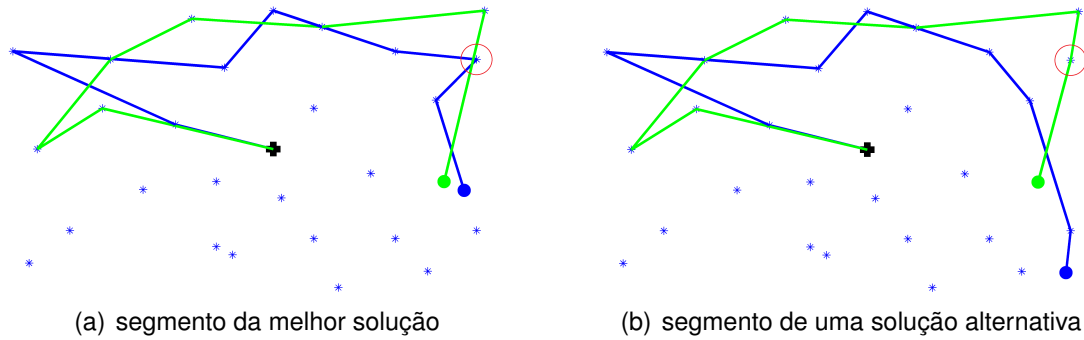


Figura 4.24 – Comparação de duas soluções parciais para a instância eil51-m2s1. O círculo vermelho indica um vértice de interesse. (a) Melhor solução, em que um veículo passa muito próximo a um vértice visitado por outro. (b) Solução alternativa, construída manualmente, em que o mesmo veículo visita os dois vértices próximos

outro visite um vértice distante e volte para que ambos sigam próximos para o restante das respectivas rotas. Se o sequenciamento fosse alterado, evitando o cruzamento, a continuação do percurso teria um veículo adiantado em relação ao outro.

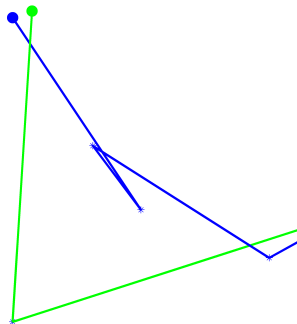


Figura 4.25 – Ampliação do canto inferior esquerdo da Figura 4.22(a).

Por fim, é interessante destacar ainda dois outros comportamentos verificados em soluções do mTSP-D, relativo a alternâncias de direção que provocam cruzamentos entre rotas. Os dois comportamentos referidos são visíveis considerando a Figura 4.26, que apresenta dois momentos distintos de uma mesma solução. A Figura 4.26(a) demonstra um segmento da solução em que um veículo apresenta arcos mais longos, enquanto o outro circula a sua volta, repetidas vezes cruzando os arcos longos percorridos pelo primeiro. Isso fica evidente se colocado em números: até o ponto apresentado, um dos veículos atendeu 16 vértices, contra 26 do outro.

O segundo comportamento é visível no segmento a partir do instante ilustrado na Figura 4.26(a) e terminado no instante ilustrado na Figura 4.26(b). Nesse caso, os dois veículos se deslocam de modo a forçar o cruzamento entre suas rotas. Esse comportamento atrasa as rotas em relação ao seu tempo total, o que é compensado pelos pontos de grande aproximação. Vale destacar que tal comportamento é recorrente no percurso

desse conjunto de vértices, acontecendo em diversas soluções da instância, embora em alguns casos no sentido inverso.

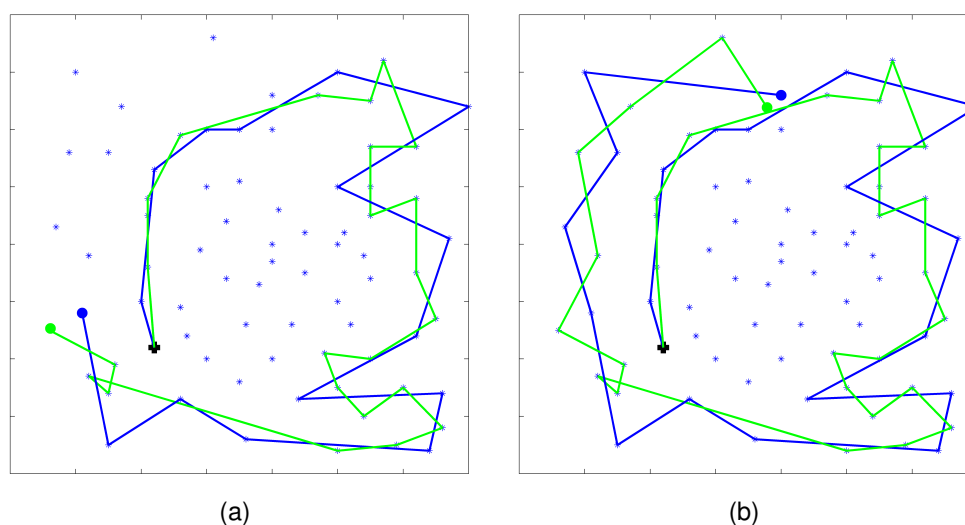


Figura 4.26 – Alternâncias de direção provocando cruzamentos entre rotas. São ilustradas as rotas parciais em dois instantes de uma solução obtida para a instância eil76-m2s1. A solução apresentada não é a melhor encontrada para a instância, e possui dispersão 2,32% maior que a melhor.

4.4 RESUMO DO CAPÍTULO

Nesse capítulo, foi proposto um novo problema de roteamento: o Problema do Múltiplo Caixeiro Viajante com Objetivo de Minimização da Dispersão. O objetivo é encontrar rotas, definidas por sequências de vértices, que minimizem uma métrica de dispersão entre veículos. Isso obriga os veículos a viajarem próximos uns dos outros, o que lhes permite atuar como reforço ou cobertura uns para os outros em caso de ocorrência de situação de emergência.

A métrica de dispersão induz uma espécie de sincronização através da imposição de proximidade espacial e temporal entre veículos. A métrica requer a divisão do tempo de roteamento em fatias de tempo, que são definidas em termos de alterações nos vetores de velocidade dos veículos. Para cada fatia de tempo, calcula-se a distância instantânea entre qualquer par de veículos.

Para resolver o problema, é proposto um Algoritmo Genético com Busca Local, com operadores genéticos e vizinhanças especializados. Essas vizinhanças são baseadas em métodos bem conhecidos da literatura, incluindo troca e inserção, bem como uma generalização de 2-opt e 3-opt para várias rotas.

Finalmente, é apresentado um conjunto de 48 instâncias para testes, construídas a partir de instâncias da literatura, chegando a 99 vértices e sete veículos. São apresentados resultados computacionais que mostram que a abordagem resolve o problema de forma eficaz, produzindo rotas que de fato apresentam as características desejadas de dispersão reduzida entre os veículos.

5 O PROBLEMA DO ROTEAMENTO DE ARCOS BI OBJETIVO COM COLETA DE PRÊMIOS E MAXIMIZAÇÃO DA DISPERSÃO

Este capítulo aborda Problema do Roteamento de Arcos Biobjetivo com Coleta de Prêmios e Maximização da Dispersão ou BARP-PD. Nesse problema, as soluções são construídas a fim de maximizar tanto a totalidade dos prêmios coletados ao cruzar determinados arcos quanto um objetivo de dispersão dos veículos. Em geral, esses dois objetivos são conflitantes, estabelecendo-se portanto o *trade-off* que caracteriza os problemas biobjetivo.

O primeiro objetivo, de coleta de prêmios, permite que as soluções sejam construídas de modo a direcionar os veículos para que percorram certos arcos ou transitem por regiões de interesse. Na concepção do grafo sobre o qual o roteamento será feito, uma hierarquia é estabelecida entre os arcos a partir da diversidade dos valores a eles atribuídos, e a busca pela maximização do prêmio coletado tende a gerar rotas em que arcos melhor colocados na hierarquia sejam atendidos.

O segundo objetivo é concebido de forma a manter os veículos afastados uns dos outros. A métrica proposta para calcular a dispersão é baseada no cálculo das distâncias instantâneas entre veículos, conforme descrito na Seção 3, e a função objetivo de dispersão é formalizada na Seção 5.1.

A concepção do BARP-PD lhe confere características que são interessantes para duas importantes aplicações de roteamento de arcos: vigilância e distribuição de ajuda humanitária. A distribuição de ajuda humanitária em geral é realizada em locais de difícil tráfego, como áreas atingidas por catástrofes naturais ou áreas de conflito, e a dispersão está diretamente relacionada com a eficácia da distribuição. Conforme apontado por (VICTORIANO; ORTUÑO; TIRADO, 2009), dispor os veículos em comboios aumenta a segurança da carga, mas a dispersão dos veículos aumenta a robustez do sistema de entrega. Como algumas rotas podem ser bloqueadas por causa de graves danos à infraestrutura de transportes (em áreas de desastre) ou ações hostis (em áreas de conflito), a possibilidade de que boa parte dos veículos finalizem suas rotas é maior quando eles viajam separados. Além disso, é mais fácil que os veículos se desloquem despercebidos, característica importante em regiões hostis, quando viajam separados. Nessa aplicação, o objetivo de coleta de prêmios está relacionado com a eficiência, uma vez que pode ser utilizado para dirigir as rotas para locais em que a ajuda é mais necessária.

A aplicação em vigilância está voltada principalmente para o problema do planejamento de rotas para a polícia ou patrulhas de segurança (SHAFABI; HAGHANI, 2015; WILLEMSE; JOUBERT, 2012). Nesse caso, a coleta de prêmio permite que a vigilância seja priorizada em algumas ruas ou áreas, de acordo com a sua incidência de crimes ou outras características (arredores de bancos, lojas de jóias e estações de trem, por exem-

plo). Além disso, se for considerado que uma patrulha cuidando de uma determinada área é capaz de servir ocorrências próximas à sua localização, a dispersão é importante para aumentar a área total coberta por múltiplas patrulhas.

Outra interessante aplicação ligada à vigilância é a busca por possíveis ameaças nucleares, conforme descrita em (HOCHBAUM; LYU; ORDÓÑEZ, 2014). Neste problema, um conjunto de caminhões equipados com equipamento altamente sensível é utilizado para patrulhar uma área e detectar radiação incomum. Em tal aplicação, coleta de prêmios pode ser usada para favorecer a busca em locais mais suspeitos ou mais povoados. Já a dispersão reduz a chance de intersecção entre as áreas cobertas por dois caminhões, aumentando assim a área total coberta. Além disso, os veículos viajando separados reduz a chance de serem percebidos como algo incomum.

Para gerar soluções para o problema novo, foram implementados dois métodos diferentes. Como os objetivos são conflitantes, estes métodos respeitam o *trade-off* presente em problemas biobjetivo e geram conjuntos de soluções não dominadas. O primeiro método é um Algoritmo Genético Multiobjetivo (MOGA), e é essencialmente o NSGA II com procedimentos especializados para construir a população inicial e para criar as gerações seguintes. O segundo método é um Algoritmo Genético Multiobjetivo com Busca Local (MOGLS) que apresenta diversas similaridades com o primeiro, mas incorpora um procedimento de Busca Local e usa um conjunto de arquivo separado da população para armazenar as soluções não dominadas encontradas.

O capítulo está organizado da seguinte forma. Na Seção 5.1, o BARP-PD é descrito em detalhes, especialmente a função objetivo de dispersão. Os métodos implementados são descritos na Seção 5.2. Na Seção 5.3, são descritos os testes realizados e apresentados os resultados obtidos.

5.1 DEFINIÇÃO DO PROBLEMA

As rotas são construídas sobre um grafo conexo $G = (N, A)$, onde $N = \{0, 1, \dots, n\}$ é o conjunto de $n + 1$ nós e A é um conjunto de arcos que conectam nós em N . O nó 0 é a garagem, e a localização de cada nó $i \in V$ é dada por um par ordenado de coordenadas (x_i, y_i) . Os m veículos idênticos em $K = \{1, 2, \dots, m\}$ partem do nó 0 para realizar m rotas abertas, definidas por sequências de arcos. Um custo c_{ij} e um prêmio p_{ij} estão associados com cada arco $a_{ij} \in A$. O custo representa o tempo de viagem necessário para atravessar completamente o respectivo arco e o prêmio representa um ganho obtido se o arco é incluído em alguma rota. O prêmio pode ser zero em alguns arcos, mas é sempre não negativo.

Existe um comprimento máximo s para as rotas, estabelecido para representar o turno de trabalho do pessoal que conduz os veículos. Todos os veículos partem da gara-

gem no momento 0 e coletam prêmios durante todo o turno. É usual que o último arco em cada rota excede o limite definido por s . Em tais casos, o prêmio recolhido é proporcional ao comprimento percorrido. Finalmente, um arco pode ser percorrido várias vezes em uma solução, mas seu prêmio será recolhido apenas uma vez.

A Figura 5.1(a) apresenta uma representação gráfica das rotas abertas de uma solução do problema descrito. A definição das fatias de tempo correspondentes pode ser percebida na Figura 5.1(b). Como a finalização do percurso de um arco implica no imediato início do percurso do arco seguinte, tanto a chegada quanto a partida de um vértice são identificados por apenas um marcador de tempo. É possível identificar ainda que não existem marcadores de tempo (que definem as fatias) associados aos instantes de chegada nos nós 4, 7 e 10. Isso porque o tempo s que determina o final do turno está identificado pelo marcador t_8 , e encerra o período sobre o qual a dispersão deve ser considerada.

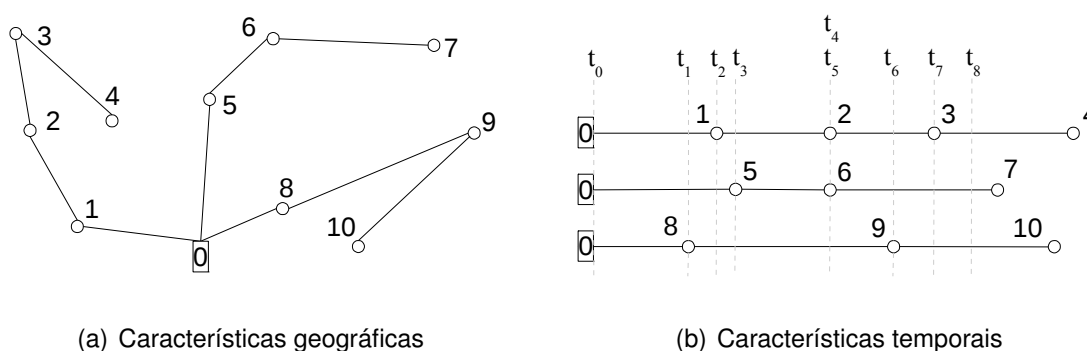


Figura 5.1 – Rotas e as respectivas fatias de tempo.

Observe que em uma solução com g arcos e m veículos existem $g - m$ marcadores de tempo correspondentes aos instantes em que os veículos mudam de direção, já que há uma mudança de direção quando cada travessia de arco é concluída, exceto para as m concluídas após o turno estar finalizado. Existem também dois marcadores de tempo representando a partida da garagem no tempo zero e o final do turno no tempo s . O total de $g - m + 2$ marcadores indica a existência de $g - m + 1$ fatias de tempo em uma solução.

Nesse problema, a métrica de dispersão descrita na Seção 3 é utilizada em uma função objetivo concebida para favorecer soluções em que os veículos se deslocam afastados uns dos outros. O valor de dispersão atribuído a cada fatia individualmente, identificado por c_τ na Equação (5.1), é determinado pelo par de veículos mais próximos naquele intervalo de tempo.

$$c_\tau = \min_{\substack{k=1, \dots, m \\ k'=k+1, \dots, m}} \int_0^{T_\tau} d_{kk'}(t) dt \quad (5.1)$$

A dispersão total de uma solução, identificada como C , é dada pelo somatório dos

valores individuais de dispersão das $g - m + 1$ fatias.

$$C = \sum_{\tau=1}^{g-m+1} c_{\tau} \quad (5.2)$$

Para que se obtenha soluções com os veículos distribuídos pelo espaço dos nós, essa função objetivo deve ser maximizada, considerando as relações de dominância existentes em conjuntos de soluções de problemas biobjetivo.

5.2 MÉTODOS DE SOLUÇÃO

Dois métodos foram implementados para gerar conjuntos de aproximação. Os métodos têm diferenças importantes, mas também compartilham algumas características e procedimentos. O restante dessa seção é dedicada aos detalhes desses métodos. As Seções 5.2.1 e 5.2.2 descrevem o laço principal de cada um. As Seções 5.2.3 a 5.2.5 descrevem as questões relacionadas à geração e representação de soluções, comuns a ambos. Finalmente, o procedimento de Busca Local, específico do MOGLS, é descrito na Seção 5.2.6

5.2.1 Algoritmo Genético Multiobjetivo

O Algoritmo 5 apresenta o pseudocódigo do MOGA proposto. Apesar de algumas questões de implementação, este pseudocódigo representa o laço principal do método NSGA II, conforme descrito na Seção 2.4.3.

Algoritmo 5 Pseudocódigo do MOGA

```

curPop ← InitialPopulation(nRCL, ps1)
FastNondominatedSort(curPop)
CrowdingDistanceComputation(curPop)
while currentTime < maxTime do
  newPop ← MakeNewPopulation(curPop, ps1)
  newPop ← newPop ∪ curPop
  FastNondominatedSort (newPop)   comentário: F0 a Ff fronteiras não-dominadas
  f' ← f
  while |newPop| − |Ff'| > ps1 do
    newPop ← newPop \ Ff'
    f' ← f' − 1
  end while
  CrowdingDistanceComputation(newPop)
  Remover de newPop os |newPop| − ps1 indivíduos de menor distancia de povoamento em Ff'
  curPop ← newPop
end while
return o conjunto de aproximação na fronteira F0 de curPop

```

O algoritmo construtivo descrito na Seção 5.2.3 é utilizado para gerar uma população inicial *curPop* com *ps*₁ indivíduos.

Esta população é, então, organizada e avaliada como no algoritmo de NSGA II. Inicialmente ela é classificada em níveis de dominância e em seguida é feita a avaliação da distância de povoamento (*crowding distance*) de cada indivíduo. Da mesma forma que no método NSGA II, o valor de classificação e a distância são usados para definir o operador de comparação que orienta o procedimento de seleção.

O procedimento entra em um laço de repetição no qual a população corrente é melhorada. Uma nova população é criada com a aplicação de operadores genéticos conforme procedimento descrito na Seção 5.2.4. A nova população é combinada com *curPop*, resultando em um conjunto *newPop* com 2*ps*₁ indivíduos.

O conjunto *newPop* é reduzido para *ps*₁ indivíduos com a estratégia definida no NSGA-II. Inicialmente, após uma classificação por níveis de dominância, as fronteiras de pior avaliação (maior índice) que tiverem todos os seus indivíduos computados como excesso em relação a *ps*₁ são completamente descartadas. Caso o número de indivíduos permaneça maior que *ps*₁, são eliminados os indivíduos de pior valor para distância de povoamento na fronteira remanescente de mais alto índice. A população reduzida entra em uma nova iteração como corrente, e o procedimento é repetido até que o limite de tempo de execução seja alcançado.

5.2.2 Algoritmo Genético Multiobjetivo com Busca Local

Um MOGLS também foi implementado para gerar um conjunto de soluções que aproximam a fronteira de Pareto do BARP-PD. Um pseudocódigo que descreve o método é apresentado no Algoritmo 6.

Algoritmo 6 Pseudocódigo do MOGLS

```

Archive ← ∅
curPop ← InitialPopulation(nRCL, ps2)
g ← 1
while currentTime < maxTime do
  FastNondominatedSort (curPop)  comentário: F0 a Ff fronteiras não-dominadas
  CrowdingDistanceComputation(curPop)
  Archive ← soluções não-dominadas de (Archive ∪ F0)
  curPop ← MakeNewPopulation(curPop, ps2)
  Elite ← SelectElite(es, Archive)
  if (g mod β = 1) then
    LocalSearch(curPop, Elite)
  else
    curPop ← curPop ∪ Elite
  end if
  g ← g + 1
end while
return o conjunto de aproximação em Archive

```

O conjunto de aproximação é mantido separado da população em um conjunto de arquivo *Archive*, que contém todas as soluções não-dominadas encontradas durante todo o processo. O procedimento começa com um conjunto *Archive* vazio e uma população de *ps*₂ indivíduos gerada com o algoritmo construtivo descrito na Seção 5.2.3. O contador de gerações *g* é inicializado para 1. Populações sucessivas são geradas iterativamente no laço principal, limitado por um tempo máximo de execução.

Iniciando cada iteração, a população é organizada em fronteiras não dominadas e a distância de povoamento é calculada para todos os indivíduos logo após a ordenação. O arquivo *Archive* é então atualizado com as soluções presentes na fronteira *F*₀, que contém os indivíduos não dominados. Uma nova população é gerada com o procedimento descrito na Seção 5.2.4, e substitui a atual. O conjunto *Elite* é criado como um subconjunto do arquivo, com soluções escolhidas por torneio binário. Entre duas soluções selecionadas aleatoriamente, aquela com maior distância de povoamento na fronteira de aproximação atual, mantida em *Archive*, é favorecida. Como *Archive* contém apenas soluções não dominadas, entre elas não existe diferença de níveis de dominância.

A cada β gerações, um procedimento de Busca Local é executado, a fim de melhorar as fronteiras não dominadas representadas por *Elite* e por *F*₀. O método de Busca Local e as formas como as novas soluções são incluídas na população estão descritos na Seção 5.2.6. Nas iterações sem Busca Local as soluções de *Elite* são incluídas diretamente na população. Finalmente, o contador de gerações *g* é incrementado e nova

iteração do laço externo é executada se o tempo disponível não foi excedido.

5.2.3 Geração da População Inicial

O Algoritmo 7 apresenta um algoritmo construtivo, guloso e aleatorizado, utilizado para criar a população inicial dos métodos de solução propostos. Cada iteração do laço externo fornece uma solução ou indivíduo I , gerado conforme descrição que segue.

Algoritmo 7 Pseudocódigo do método de construção da população inicial

Input:

$nRCL$: número de candidatos a considerar na seleção aleatória
 ps : tamanho da população

$Pop \leftarrow \emptyset$

while $|Pop| < ps$ **do**

 Cria um novo indivíduo I

$Out \leftarrow A$

$LN \leftarrow \emptyset$

for k **from** 1 **to** m **do**

$Candidates \leftarrow \{a_{0i} \in Out : \forall i \in N\}$

if $Candidates = \emptyset$ **then**

$Candidates \leftarrow \{a_{0i} \in A : \forall i \in N\}$

end if

 Seleciona aleatoriamente um arco $a_{0i_k} \in Candidates$

$Out \leftarrow Out \setminus \{a_{0i_k}\}$

 Inicia uma nova rota no indivíduo I pela inserção do arco a_{0i_k}

$LN \leftarrow LN \cup \{i_k\}$

$t_i \leftarrow c_{0i_k}$

end for

$j' \leftarrow \underset{i \in LN}{\operatorname{argmin}} \{t_i\}$

while $t_{j'} < s$ **do**

$Candidates \leftarrow \{a_{jh} \in Out : \forall h \in N \wedge p_{jh} > 0\}$

if $Candidates = \emptyset$ **then**

$Candidates \leftarrow \{a_{jh} \in A : \forall h \in N\}$

end if

$eval_{jh} \leftarrow \min_{j \in LN \setminus \{j'\}} \{c_{hj}\}, \forall a_{jh} \in Candidates$

 Cria RCL com os $nRCL$ arcos em $Candidates$ com maior $eval$

 Seleciona aleatoriamente um arco $a_{j'r} \in RCL$

 Insero o arco $a_{j'r}$ no indivíduo I

$Out \leftarrow Out \setminus \{a_{j'r}\}$

$LN \leftarrow LN \setminus \{j'\}$

$LN \leftarrow LN \cup \{l^*\}$

$t_l \leftarrow t_{j'} + c_{j'r}$

$j' \leftarrow \underset{i \in LN}{\operatorname{argmin}} \{t_i\}$

end while

$Pop \leftarrow Pop \cup \{I\}$

end while

return Pop

De modo a evitar a inclusão do mesmo arco duas vezes, o conjunto *Out* é criado para manter registro dos arcos não incluídos na solução em construção. Inicialmente, ele contém todos os arcos de *A*.

Uma rota começa com um arco ligando a garagem a outro nó. Um arco a_{0i_k} é selecionados aleatoriamente entre aqueles em *Out* se houver pelo menos um arco que começa no nó 0. Caso contrário, o arco será selecionado entre aqueles em *A*, o que significa que estará duplicado na solução. Neste ponto, não há qualquer favorecimento para arcos em função do prêmio relacionado, a fim de permitir diferentes arcos de partida e promover a diversidade da população inicial. O arco selecionado é removido de *Out* e incluído na solução. O nó i_k é incluído no conjunto *LN*, cujos elementos são os últimos nós de cada rota parcial, e o tempo estimado de chegada neste nó (t_{i_k}) é calculado. Este procedimento é repetido *m* vezes, definindo um arco de partida para cada rota.

A seguir, é encontrada entre as *m* rotas parciais aquela que está atrasada em relação às outras, e o último nó desta rota é identificado como $j' \in LN$. Este rota é escolhida para ser a próxima a incorporar um novo arco, selecionado a partir de uma lista restrita de arcos candidatos *RCL* construída como segue.

Inicialmente, são considerados como candidatos aqueles arcos que começam em j' , que estejam em *Out* e que tenham prêmio positivo, de modo a gerar soluções iniciais com um bom objetivo de prêmio coletado. Se não houver um arco com tais características, passam a ser considerados candidatos todos os arcos que partem de j' e são elementos de *A*. Os arcos candidatos são colocados em *Candidates* A *RCL* é construída com os $nRCL$ arcos $a_{j'h} \in Candidates$ com maior avaliação de acordo com a distância mínima a partir do nó *h* para todos os nós em $LN \setminus \{j'\}$. O objetivo é incluir na solução um arco entre aqueles com uma alta distância mínima para outras rotas, o que é coerente com a função objetivo de dispersão. Um arco $a_{j'l} \in RCL$ é selecionado aleatoriamente para ser incluído na rota atrasada. O arco é removido de *Out*, *l* substitui j' no conjunto *LN* como o novo nó final em sua rota parcial e t_l é calculado.

Um indivíduo ou solução é considerado completo e incluído na população se todas as rotas têm comprimento igual a ou maior do que a duração do turno *s*.

5.2.4 Geração de Novas Populações

A geração das novas populações segue o procedimento descrito no algoritmo 8.

Algoritmo 8 Pseudocódigo da geração de novas populações

Input:*curPop* : população corrente*ps* : tamanho da população*newPop* $\leftarrow \emptyset$ **while** $|newPop| < ps$ **do** $I_a \leftarrow \text{SelectIndividual}(curPop)$ **repeat** $I_b \leftarrow \text{SelectIndividual}(curPop)$ **until** $I_b \neq I_a$ **if** $\text{GenerateRandomNumber}(\in [0, 1]) < cp$ **then** $Offspring \leftarrow \text{Crossover}(I_a, I_b)$ comentário: $Offspring = \{O_a, O_b\}$ **if** $\text{GenerateRandomNumber}(\in [0, 1]) < mpC$ **then** $\text{Mutate}(O_a)$ **end if** **if** $\text{GenerateRandomNumber}(\in [0, 1]) < mpC$ **then** $\text{Mutate}(O_b)$ **end if** **else** $Offspring \leftarrow \text{Duplicate}(I_a, I_b)$ comentário: $Offspring = \{O_a, O_b\}$ **if** $\text{GenerateRandomNumber}(\in [0, 1]) < mpD$ **then** $\text{Mutate}(O_a)$ **end if** **if** $\text{GenerateRandomNumber}(\in [0, 1]) < mpD$ **then** $\text{Mutate}(O_b)$ **end if** **end if** $newPop \leftarrow newPop \cup \{O_a, O_b\}$ **end while****return** *newPop*

Dois indivíduos pais, I_a e I_b , diferentes entre si, são selecionados por torneio binário, usando o operador de comparação do NSGA II. Dois novos indivíduos são gerados através de cruzamento ou reprodução assexuada, de acordo com uma probabilidade de cruzamento cp , e formam *Offspring*. O operador de cruzamento é descrito na Seção 5.2.5 e a reprodução assexuada simplesmente replica cada um dos indivíduos pais em um indivíduo de *Offspring*. Um operador de uma mutação é aplicado a cada novo indivíduo de acordo com uma probabilidade de mpC se o indivíduo é o resultado de um cruzamento, ou com uma probabilidade mpD se o indivíduo é o resultado de uma reprodução assexuada. A sequência de seleção-reprodução-mutação é repetida para completar a nova população com ps indivíduos, e a nova população é retornada.

5.2.5 Cromossomo e Operadores Genéticos

O objetivo de dispersão do BARP-PD apresenta as mesmas características de sincronismo descritas para a função objetivo do mTSP-D. Por isso, o cromossomo implementado tem a mesma estrutura descrita na Seção 4.2.3. Enquanto favorece as relações importantes para o objetivo de dispersão, tal estrutura não traz prejuízo para a propagação de informações relativas à coleta de prêmios.

Foi implementado um operador de mutação simples, em que uma rota é escolhida aleatoriamente e um par de arcos adjacentes a_{il} e a_{lj} é substituído nesta rota por outro par a_{ik} e a_{kj} . A escolha do nó k é feita aleatoriamente entre todos os nós $N \setminus \{l\}$.

O operador de cruzamento implementado combina as soluções de dois indivíduos pais, p_1 e p_2 , para criar soluções o_1 e o_2 para uma prole de dois novos indivíduos. A principal diferença em relação ao operador implementado para o mTSP-D está no número de pontos de cruzamento, que para o BARP-PD ficou limitado a um. Essa diferença se explica pela dispersão entre as aresta ou arcos novos inseridos em cada ponto de cruzamento.

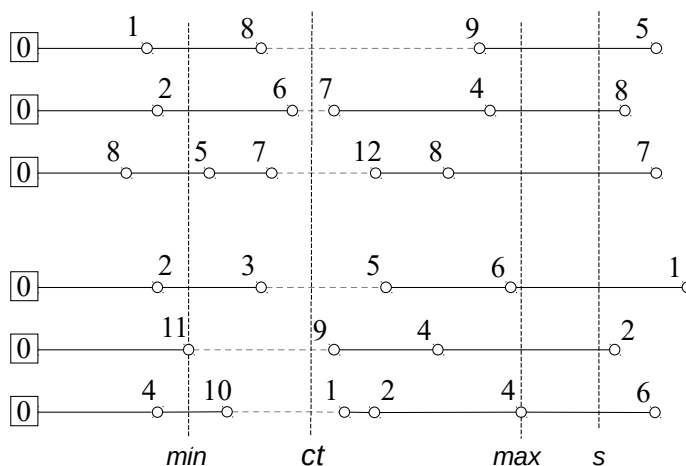
No caso do mTSP-D, em cada ponto de cruzamento, segmentos de rotas próximas entre si são ligados por novas arestas a outros segmentos de rotas também próximas. Portanto, as novas arestas ligam vértices próximos a outros também próximos, e devem apresentar uma proximidade relativa, o que está de acordo com o objetivo de minimização da dispersão do mTSP-D. Já no caso de um objetivo de maximização da dispersão, as novas rotas são construídas a partir de segmentos de rotas que se apresentam dispersas nos indivíduos pais. Em tal situação, mesmo partindo de nós distantes para outros nós também distantes, não é possível afirmar que os arcos novos não terão pontos de aproximação, ou mesmo cruzamentos. Por isso, optou-se por um operador de cruzamento de um ponto, que mantém a geração de novos indivíduos, importante para a diversidade na população, porém reduzindo a apenas um os pontos com possível redução da dispersão pela inclusão de novos arcos.

O primeiro passo do procedimento de cruzamento é a definição de um marcador de tempo de cruzamento ct , para ser utilizado como um ponto de seccionamento das rotas. Em seguida, são removidos de p_1 e p_2 todos os arcos planejados para estarem sendo atravessados por algum veículo quando o tempo de cruzamento ct ocorre. Assim, cada rota em ambas soluções é dividida em dois segmentos, o primeiro com arcos a serem percorridos antes de ct e o segundo com arcos a serem percorridos após ct .

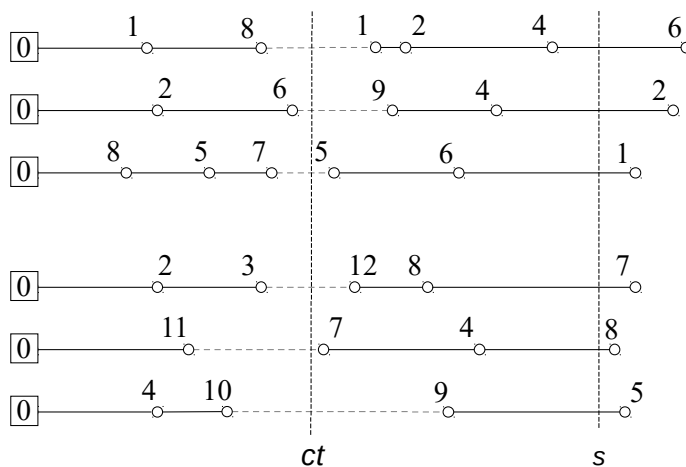
Cada rota da solução o_1 combina o primeiro segmento de uma rota de p_1 com o segundo segmento de uma rota de p_2 . Na construção da solução o_2 , a contribuição de cada indivíduo pai é invertida. A definição sobre as rotas dos pais que serão combinadas nas rotas dos filhos é feita de forma aleatória a cada cruzamento.

Limites mínimo e máximo são fixados para determinar um intervalo válido para a escolha de ct . O limite mínimo min é o instante em que o último veículo termina seu

primeiro arco, considerando todas as rotas de $p1$ e $p2$. Esse limite garante que todas as rotas terão ao menos um arco completo em seu primeiro segmento. O limite máximo max é dado pelo instante em que o último arco que extrapola o turno é começado, de modo a evitar que todas as rotas tenham seus arcos finais (que em geral terminam fora do turno) removidos. O procedimento de cruzamento é ilustrado na Figura 5.2.



(a) Soluções originais



(b) Soluções resultantes

Figura 5.2 – Cruzamento aplicado a dois indivíduos com três rotas cada. As linhas verticais em (a) representam ct entre os valores mínimo e máximo possíveis, além do final do turno. Os arcos representados por linhas tracejadas em (a) são aqueles removidos das rotas, e em (b) são os incluídos para ligar os dois segmentos em cada rota nas soluções dos novos indivíduos.

Finalmente, é importante notar que algumas das novas rotas criadas podem resultar mais longas do que as rotas dos indivíduos originais, e alguns arcos podem ser deslocados no tempo de modo a serem iniciados após o final do turno. Esses arcos são descartados, removidos de suas rotas. Por outro lado, algumas rotas podem se tornar mais curtas do que o limite. Nesses casos, arcos são escolhidos aleatoriamente, sempre dentre os que

partem do último vértice da rota parcial, para serem incluídos no final da rota, até que o limite s seja alcançado. Normalmente, isso implica nas inserções de apenas um ou dois arcos aleatórios. Este problema de factibilidade ocorre também no operador de mutação ou em movimentos de vizinhança, e o tratamento é o mesmo em todas as operações.

5.2.6 Busca Local

Em métodos de Busca Local projetados para problemas com um objetivo, a vizinhança de uma solução incumbente é explorada para encontrar uma solução que a melhore. Em problemas biobjetivo, não existe uma solução incumbente, mas um conjunto que forma uma fronteira não dominada corrente. A forma como se dá no MOGLS a busca por novas soluções que representem melhora nesse conjunto é descrita a seguir.

5.2.6.1 Procedimento de Melhoria de Fronteira

O método de Busca Local faz uso do procedimento de melhoria de fronteira descrito no algoritmo 9, que é inspirado no procedimento de *Multi-Objective Local Search* descrito em (ARROYO; ARMENTANO, 2005).

Algoritmo 9 Pseudocódigo do procedimento de melhoria de uma fronteira

Input:

ND : conjunto de soluções não dominadas
 $nIter$ - número de iterações
 $nMax$ - número de soluções a avaliar em cada iteração
 L - conjunto de todas as soluções cuja vizinhança já foi explorada

```

 $S \leftarrow ND$ 
while  $nIter > 0 \wedge |S| > 0$  do
  if  $|S| > nMax$  then
     $Snot \leftarrow$  conjunto de  $|S| - nMax$  soluções selecionadas aleatoriamente de  $S$ 
     $S \leftarrow S \setminus \{Snot\}$ 
  end if
   $S' \leftarrow \emptyset$ 
  for all  $x \in S$  do
    if  $x \notin L$  then
       $N(x) \leftarrow$  GenerateNeighborhood( $x$ )
       $L \leftarrow L \cup \{x\}$ 
       $ND \leftarrow$  soluções não-dominadas de  $(ND \cup N(x))$ 
       $S' \leftarrow$  soluções não-dominadas de  $(S' \cup \{N(x) \cap ND\})$ 
    end if
  end for
   $S \leftarrow$  soluções não-dominadas de  $(S' \cup Snot)$ 
   $nIter \leftarrow nIter - 1$ 
end while
return  $ND$ 
  
```

O Algoritmo 9 recebe uma fronteira de soluções não dominadas ND para ser melhorada, e esta é duplicada em S . A partir de então, o conjunto ND permanece como o conjunto de todas soluções não dominadas encontradas, enquanto S passa a ser utilizado como o conjunto de soluções por explorar. O laço principal é executado até que o conjunto S torne-se vazio ou o número máximo de iterações $nIter$ seja atingido. Dentro do laço, se o conjunto S tem mais do que $nMax$ soluções, é reduzido para a cardinalidade definida por esse parâmetro. Os parâmetros $nIter$ e $nMax$ estabelecem limites devido ao alto custo computacional de exploração da vizinhança.

Soluções removidas de S são armazenadas no conjunto $Snot$. A vizinhança $N(x)$ é gerada para cada solução x em S e que não esteja em L . O conjunto L contém todas as soluções já explorados em execuções de Busca Local, e é usado para evitar que as mesmas soluções sejam exploradas repetidas vezes, em diferentes iterações.

Se $N(x)$ é gerado, a solução x é incluída no conjunto de L . Atualizar L neste ponto do algoritmo é uma diferença importante do procedimento do Algoritmo 9 quando comparado com o seminal. No método descrito em (ARROYO; ARMENTANO, 2005), todas soluções que fazem parte do conjunto não dominado inicial (o ND inicial) são incluídas em L . No entanto, como a Busca Local é limitada na amplitude por $nMax$ e na profundidade por $nIter$, não é garantido que todas as soluções no ND inicial realmente sejam exploradas no procedimento de melhoria de fronteira.

Um novo conjunto de soluções não dominadas é construído após a união de $N(x)$ e ND e a remoção das soluções dominadas do conjunto resultante. Além disso, as novas soluções em $N(x)$ que não foram dominadas por qualquer solução em ND são incluídas no conjunto S' . No final do laço *for*, as soluções de S' são as soluções novas e não dominadas. Essas soluções, combinadas com as soluções deixadas de lado e armazenadas em S_{not} , constituem o novo conjunto S para a próxima iteração da geração de vizinhos.

5.2.6.2 Fronteiras Submetidas à Busca Local e Inserção de Soluções Resultantes na População

O método de melhoria de fronteira descrita no Algoritmo 9 é executada duas vezes sempre que a Busca Local é realizada. A primeira execução melhora o conjunto *Elite*, subconjunto do arquivo *Archive*, que em gerações sem Busca Local é diretamente incluído na população em processo de elitismo. O conjunto não dominado resultante é referido na sequência como $ND1$.

A segunda execução melhora o conjunto de soluções não dominadas da população atual, identificado como F_0 após o *nondominated sorting process*. O conjunto melhorado resultante é referido como $ND2$. De modo a evitar o desperdício de tempo computacional, o método de melhoria é aplicado sobre os conjuntos *Elite* e F_0 apenas após a remoção de soluções já em L .

Os conjuntos $ND1$ and $ND2$ representam o resultado real do processo de Busca Local. De modo a participarem nos futuros processos evolutivos de seleção-reprodução, suas soluções devem ser incluídas na população atual. Isso ocorre de acordo com as cardinalidades dos conjuntos originais *Elite* e F_0 e conjuntos resultantes $ND1$ e $ND2$.

Se $|Elite| + |F_0| = |ND1| + |ND2|$, todos os indivíduos nos conjuntos *Elite* e F_0 são descartados e aqueles em $ND1$ e $ND2$ são incluídos diretamente na população atual.

Outra situação ocorre se $|Elite| + |F_0| > |ND1| + |ND2|$, isto é, o número de soluções em $ND1$ e $ND2$ não é suficiente para preencher o tamanho da população. Neste caso, os conjuntos *Elite* e F_0 não são totalmente descartados, mas reduzidos para se ajustarem à diferença $\Delta = (|Elite| + |F_0|) - (|ND1| + |ND2|)$. Em primeiro lugar, são eliminados do conjunto *Elite* todas as soluções que permaneceram no $ND1$ ou têm soluções descendentes em $ND1$, e de F_0 todas as soluções que permaneceram no $ND2$ ou têm soluções descendentes em $ND2$. A ideia é manter a diversidade da população, primeiramente eliminando indivíduos que tem seu patrimônio genético preservado de alguma forma nos indivíduos resultantes. Uma solução x' é dita descendente de x se ela foi gerada como um vizinho de x , ou assim recursivamente.

Vale salientar que o número de soluções descartadas de *Elite* e F_0 nunca excederá $|ND1| + |ND2|$, já que cada solução nos conjuntos resultantes é igual a ou descendente de

apenas uma solução dos conjuntos originais. Mas é possível que duas ou mais soluções resultantes descendam da mesma solução original, de modo que a simples remoção dos descendentes não represente uma redução suficiente em $Elite$ e F_0 para que $ND1$ e $ND2$ sejam incluídos na população sem exceder seu tamanho. Nesse caso, outras soluções são removidas aleatoriamente de ambos os conjuntos originais, até que $|Elite| + |F_0| = \Delta$. Em seguida, as soluções em $ND1$ e $ND2$ e as soluções restantes em $Elite$ e F_0 são incluídas na população corrente. Este procedimento está ilustrado na Figura 5.3, para uma população de quinze indivíduos.

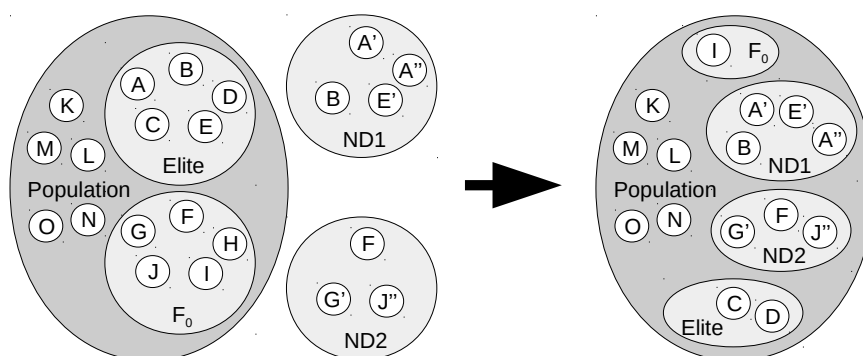


Figura 5.3 – $ND1$ e $ND2$ não preenchem a população. Nem todas as soluções em $Elite$ e F_0 são descartadas, apenas aquelas em $Elite \cap ND1$ e em $F_0 \cap ND2$, B e F , e soluções com descendentes, A , E , G e J . Outra solução deve ser descartada, e H é selecionado aleatoriamente. As quinze soluções restantes formam a população.

Uma terceira situação é apresentada na Figura 5.4 e ocorre se $|Elite| + |F_0| < |ND1| + |ND2|$, ou seja, o simples descarte de $Elite$ e F_0 não gera espaço suficiente para a inclusão de $ND1$ e $ND2$ na população. De modo a favorecer a diversidade, inicialmente $ND1$ é reduzido pelo descarte de indivíduos em $Elite \cap ND1$, uma vez que estes já estão armazenados em *Archive*. Se esta redução não é suficiente, outros indivíduos são selecionados aleatoriamente e eliminados da população, até que a inserção de $ND1$ e $ND2$ não exceda o número de indivíduos.

Há ainda uma situação que ocorre se Busca Local é extremamente bem sucedida e gera soluções em número maior do que o tamanho da população. Nesse caso, os conjuntos $ND1$ e $ND2$ são mesclados, gerando um novo conjunto $ND1$ de soluções não-dominadas e um conjunto $ND2$ de soluções dominadas por $ND1$. Se $ND1$ sozinho tem muitas soluções, um número adequado é selecionado aleatoriamente para ser retirado de $ND1$ e inserido diretamente em *Archive*, enquanto $ND2$ é completamente descartado. Caso contrário, soluções são selecionadas de forma aleatória a partir de $ND2$ para preencher a população juntamente com $ND1$.

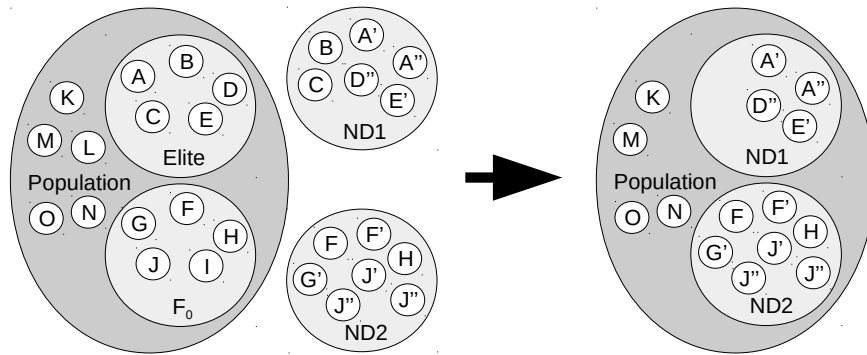


Figura 5.4 – $ND1$ e $ND2$ juntos têm mais indivíduos do que $Elite$ e $F0$. $Elite$ e $F0$ são totalmente descartados, juntamente com os indivíduos em $ND1 \cap Elite$, B and C . Outro indivíduo deve ser descartado da população, e L é selecionado aleatoriamente.

5.2.6.3 Vizinhanças

A Busca Local explora quatro vizinhanças definidas pelos seguintes movimentos: (1) um arco a_{ij} é substituído por dois arcos a_{ik} e a_{kj} ; (2) dois arcos a_{il} e a_{lj} são substituídos por dois arcos a_{ik} e a_{kj} ; (3) dois arcos a_{il} e a_{lj} são substituídos por um arco a_{ij} ; (4) dois arcos a_{ij} e a_{kl} , colocados em rotas diferentes e pelo menos parcialmente concomitantes, são substituídos por arcos a_{il} e a_{kj} . O quarto movimento representa a troca de segmentos entre rotas e é importante para evitar cruzamentos de rotas, muito ruins para o objetivo de dispersão.

Existe um alto custo computacional associado à exploração das vizinhanças, uma vez que esta exploração implica em um grande número de movimentos possíveis. Para reduzir esse custo, foram estabelecidas algumas condições para decidir se um movimento deve ser aplicado ou não. O primeiro e o segundo movimentos são executados se a troca de arcos representa um ganho para o objetivo de coleta de prêmios, ou se o movimento envolve pelo menos um arco que é determinante para a dispersão de pelo menos uma fatia de tempo. O quarto movimento é executado se a troca de arcos representa um ganho de prêmio coletado ou se $c_{ij} + c_{kl} > c_{il} + c_{kj}$. O terceiro movimento é sempre executado, uma vez que o número de vizinhos gerados nessa vizinhança é pequeno.

Finalmente, é necessário definir quando um arco é determinante para a dispersão de uma fatia de tempo, situação referida como um critério no parágrafo anterior. O conceito de arco mais custoso é semelhante ao conceito de par de vértices mais custoso, utilizado nas vizinhanças de aproximação das rotas do mTSP-D e descrito na Seção 4.2.4.1. Considere a Figura 5.5, na qual uma fatia de tempo e segmentos de quatro rotas estão representados. Os arcos mais próximos são a_{kl} e a_{mn} , então esses arcos são determinantes para o custo local associada à fatia de tempo. É interessante notar que o arco a_{ij} , que determina os limites da fatia, não contribui para sua dispersão, nem de qualquer outra fatia.

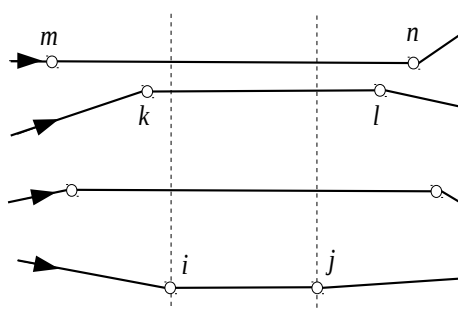


Figura 5.5 – Arcos determinantes para a dispersão de uma fatia de tempo. Linhas verticais tracejadas representam os limites da fatia.

5.3 RESULTADOS E DISCUSSÕES

Os testes computacionais foram realizados na mesma máquina e com as mesmas características de implementação descritas na Seção 4.3. Na sequência dessa seção, serão descritas as instâncias usadas para os teste, os valores atribuídos aos parâmetros dos métodos de solução, e serão apresentados e discutidos os resultados obtidos para o BARP-PD.

5.3.1 Instâncias e Parâmetros

Normalmente, instâncias para problemas de roteamento de arcos com coleta de prêmios apresentam apenas informações sobre os conjuntos de arcos. Cada arco é representado por um par de nós e seus valores associados, como prêmio, custo ou a demanda. No entanto, a métrica de dispersão introduzida neste trabalho é calculada com base nas posições instantâneas dos veículos, e essas posições dependem da localização dos arcos.

De modo a suprir este requisito, foram geradas novas instâncias que listam as coordenadas de todos os nós de N . O primeiro conjunto de instâncias foi gerado de acordo com (FEILLET; DEJAX; GENDREAU, 2005a): o grafo é completo; todos os pontos foram gerados aleatoriamente com coordenadas dentro de um quadrado com largura de 500 unidades de distância; cada custo c_{ij} corresponde a distância euclidiana entre os nós i e j ; cada arco tem uma probabilidade de 50% de ter um prêmio associado $p_{ij} \neq 0$, o qual é gerado aleatoriamente no intervalo $[c_{ij}, 1.5c_{ij}]$; e o limite para o comprimento das rotas é definido em 1500.

Há também diferenças significativas entre as novas instâncias e aquelas criadas em (FEILLET; DEJAX; GENDREAU, 2005a): as novas instâncias geradas não tem capacidade associada a arcos, uma vez que o prêmio de cada arco é coletado apenas uma vez; e os valores de c_{ij} não são arredondados para uma precisão de 10^{-1} . Além disso, o problema abordado em (FEILLET; DEJAX; GENDREAU, 2005a) não impõe limite para o número de

veículos. Foram definidas então instâncias com 2, 3 e 4 veículos, como em (ARCHETTI et al., 2010), e ainda instâncias com 5 veículos. As instâncias geradas têm 900, 1600, 2500 ou 3600 arcos (ou 30, 40, 50 e 60 nós). Da combinação entre diferentes valores de veículos e nós foi obtido um conjunto inicial com dezesseis diferentes instâncias.

Foram gerados ainda três outros conjuntos, cada um com mais dezesseis instâncias. Eles diferem na duração do turno, aumentando o tempo para 2000, ou no intervalo em que os valores dos prêmios são gerados, permitindo valores de p_{ij} até menores do que c_{ij} . A Tabela 5.1 resume as características dos conjuntos de instâncias. São, portanto, 64 instâncias no total.

Tabela 5.1 – Características que variam entre os conjuntos de instâncias

Tipo do conjunto	Intervalo do valor do prêmio	Turno
0	$[c_{ij}, 1.5c_{ij}]$	1500
1	$[c_{ij}, 1.5c_{ij}]$	2000
2	$[0.1c_{ij}, 2.0c_{ij}]$	1500
3	$[0.1c_{ij}, 2.0c_{ij}]$	2000

Todas as instâncias estão disponíveis em <https://github.com/gdhein/arcRouting>.

Os valores dos parâmetros para MOGA e MOGLS foram empiricamente ajustados através de vários experimentos executados sobre um subconjunto das instâncias, e estão resumidos na Tabela 2. É importante notar que os tamanhos de população ps_1 e ps_2 foram obtidos de forma independente para cada método. A diferença pode ser atribuída à existência do conjunto *Archive* no MOGLS, uma vez que o MOGA mantém inseridos na população todos os indivíduos não dominados encontrados.

Tabela 5.2 – Parâmetros e valores usados nos métodos MOGA e MOGLS

parameter	description	value
ps_1	tamanho da população (MOGA)	1300
ps_2	tamanho da população (MOGLS)	800
cp	probabilidade de cruzamento	90%
mpC	probabilidade de mutação após cruzamento	5%
mpD	probabilidade de mutação após duplicação dos pais	30%
es	tamanho da elite	40
β	número de gerações entre execuções de Busca Local	50
$nIter$	número de iterações na Busca Local	5
$nMax$	número de soluções exploradas em cada iteração de Busca Local	20
$maxTime$	tempo máximo de execução	$\lceil n \cdot m / 10 \rceil$

5.3.2 Resultados Computacionais

Os métodos MOGA e MOGLS foram executados dez vezes cada um para cada instância. A primeira avaliação realizada considera o impacto do método de Busca Local. Considerando todas as 640 execuções de cada método, o método MOGLS apresentou menos de 26% do número total de gerações apresentado pelo MOGA, o que evidencia o custo computacional relacionado com o processo de Busca Local.

Uma segunda avaliação considera o número absoluto de soluções no conjunto de aproximação final gerado por cada um dos métodos. MOGLS apresentou conjuntos de aproximação com 719,7 soluções em média, enquanto o MOGA apresentou conjuntos de aproximação com apenas 382,2 soluções. Além disso, não há nenhuma instância em que o MOGA tenha mais soluções do que o MOGLS, considerando-se os resultados médios das dez execuções de cada método.

A diferença é mais expressiva se forem consideradas fronteiras de referência obtidas para cada instância pela união de todos os 20 conjuntos de aproximação resultantes seguida da remoção de todas as soluções dominadas. Os conjuntos de aproximação gerados por MOGLS apresentaram, em média, 132,8 soluções que aparecem na fronteira de referência, enquanto os conjuntos gerados por MOGA apresentaram apenas 15,2 soluções de referência, em média.

A avaliação foi aprofundada pelo uso das métricas de desempenho $D1_R$, com normalização, e indicador de hipervolume. No caso da métrica $D1_R$, cada conjunto de soluções obtido em uma execução foi avaliado em relação ao conjunto de referência Ref constituído pelas soluções não dominadas encontradas nas vinte execuções da instância, conforme descrito anteriormente. No caso do cálculo do indicador de hipervolume, o ponto de referência utilizado no cálculo é dado pelos valores mínimos de ambos os objetivos na fronteira de referência da instância.

A Tabela 5.3 resume os resultados para essas métricas, apresentando os valores médios e os melhores valores encontrados considerando as dez execuções de cada método para cada instância. A quarta coluna apresenta o tempo, em minutos, de cada execução da respectiva instância.

Todas as soluções estão disponíveis em <https://github.com/gdhein/arcRouting>.

Tabela 5.3 – Resultados Computacionais

A	m	Tipo	Tempo	Melhores soluções				Média			
				Hipervolume($\times 10^6$)		$D1_R$		Hipervolume($\times 10^6$)		$D1_R$	
				MOGA	MOGLS	MOGA	MOGLS	MOGA	MOGLS	MOGA	MOGLS
900	2	0	6	1025,79	1037,60	0,51	0,11	1020,30	1036,31	1,08	0,22
900	2	1	6	1821,76	1836,53	0,67	0,15	1810,11	1830,80	1,01	0,34
900	2	2	6	1904,48	1929,48	0,67	0,15	1891,39	1926,53	0,93	0,29
900	2	3	6	3442,30	3526,71	1,08	0,17	3418,13	3506,32	1,47	0,41
900	3	0	9	1011,51	1060,75	3,90	0,60	973,72	1049,78	5,48	1,19
900	3	1	9	2061,30	2139,27	4,73	1,05	2002,76	2114,08	5,76	1,77

Continua na próxima página

Tabela 5.3 – Continuação da página anterior

A	m	Tipo	Tempo	Melhores soluções				Média			
				Hipervolume(*10 ⁶)		D1 _R		Hipervolume(*10 ⁶)		D1 _R	
				MOGA	MOGLS	MOGA	MOGLS	MOGA	MOGLS	MOGA	MOGLS
900	3	2	9	1729,18	1837,10	3,98	0,48	1695,14	1796,41	4,90	1,91
900	3	3	9	2621,24	2827,30	4,20	0,38	2591,20	2742,27	4,91	2,21
900	4	0	12	1071,97	1139,26	3,21	0,38	1021,45	1088,80	6,19	2,74
900	4	1	12	2565,94	2805,18	5,80	0,36	2492,34	2655,15	7,46	4,07
900	4	2	12	1691,24	1800,53	4,13	1,28	1611,87	1748,92	6,76	2,86
900	4	3	12	3464,81	3659,62	5,99	1,38	3345,24	3476,68	7,61	5,87
900	5	0	15	1126,88	1209,35	4,00	0,61	1029,14	1123,78	10,27	4,98
900	5	1	15	1003,56	1064,79	4,99	0,90	907,49	962,25	8,16	4,14
900	5	2	15	1296,46	1449,83	6,31	1,08	1180,73	1355,32	10,38	4,68
900	5	3	15	2202,86	2403,01	6,87	2,67	2132,98	2294,34	9,12	4,30
1600	2	0	8	1508,48	1541,04	4,21	0,06	1505,09	1540,14	5,27	0,21
1600	2	1	8	3137,16	3197,56	0,61	0,19	3084,18	3168,98	4,20	1,19
1600	2	2	8	1856,19	1898,53	0,93	0,04	1770,29	1890,80	4,82	0,09
1600	2	3	8	3499,64	3684,67	4,02	0,05	3465,68	3667,18	4,56	0,14
1600	3	0	12	1808,16	1857,79	2,20	0,72	1770,76	1829,20	4,27	1,62
1600	3	1	12	3239,75	3313,44	1,45	0,52	3186,68	3284,60	3,99	1,42
1600	3	2	12	1592,40	1665,73	2,44	0,66	1566,52	1650,29	3,72	1,22
1600	3	3	12	2475,56	2563,65	3,00	1,32	2435,77	2529,71	3,83	2,30
1600	4	0	16	1419,98	1497,40	4,04	0,64	1378,24	1423,21	6,44	3,69
1600	4	1	16	1605,93	1653,97	3,02	0,74	1499,68	1541,47	5,32	3,11
1600	4	2	16	1786,28	1910,61	3,79	0,64	1729,60	1850,61	5,71	2,67
1600	4	3	16	3316,82	3408,57	2,28	1,57	3225,74	3360,96	4,11	2,30
1600	5	0	20	925,72	971,71	3,33	1,01	860,20	941,49	8,16	2,78
1600	5	1	20	1714,19	1859,25	4,34	0,28	1640,35	1691,07	8,38	5,72
1600	5	2	20	1105,58	1205,27	6,66	1,07	1038,72	1151,63	9,39	4,20
1600	5	3	20	1807,72	1946,78	4,02	1,05	1724,79	1864,93	6,96	3,89
2500	2	0	10	1540,81	1546,47	0,26	0,11	1533,26	1540,68	0,64	0,37
2500	2	1	10	3072,21	3093,08	0,48	0,17	3041,73	3075,68	1,35	0,62
2500	2	2	10	2593,78	2653,86	1,29	0,21	2574,45	2645,69	1,61	0,39
2500	2	3	10	3253,25	3345,09	1,08	0,21	3223,94	3324,09	2,42	0,50
2500	3	0	15	1466,81	1519,55	3,90	0,75	1428,99	1501,55	5,86	2,06
2500	3	1	15	2745,67	2798,20	2,66	0,68	2666,40	2758,90	5,35	1,93
2500	3	2	15	2291,92	2375,81	2,63	0,60	2254,77	2332,61	3,35	2,11
2500	3	3	15	3798,12	3958,78	2,24	0,62	3708,60	3900,97	3,48	1,55
2500	4	0	20	1084,83	1119,27	2,13	0,47	1064,63	1085,24	3,39	2,44
2500	4	1	20	1903,11	1960,30	2,49	1,01	1824,68	1858,24	4,62	3,70
2500	4	2	20	1383,88	1394,57	2,01	1,97	1326,16	1356,58	4,79	3,78
2500	4	3	20	3249,46	3377,10	3,41	0,96	3169,83	3215,89	4,96	4,03
2500	5	0	25	1252,38	1319,76	4,66	1,43	1188,07	1235,14	8,51	5,99
2500	5	1	25	907,29	924,23	4,48	1,91	856,34	880,06	6,31	4,22
2500	5	2	25	2203,36	2390,35	7,34	0,45	2046,37	2204,68	10,93	6,13
2500	5	3	25	2192,23	2478,62	6,17	0,91	2098,84	2255,43	8,31	5,74
3600	2	0	12	1776,32	1817,96	1,35	0,20	1762,41	1803,13	1,77	0,66
3600	2	1	12	2884,59	2957,20	1,87	0,21	2859,81	2930,41	3,58	0,81
3600	2	2	12	1615,36	1637,46	0,30	0,01	1593,92	1621,20	1,53	0,63
3600	2	3	12	3710,54	3763,74	0,92	0,20	3653,68	3744,32	1,54	0,38
3600	3	0	18	1015,88	1033,69	1,22	0,36	992,05	1017,87	2,87	1,41
3600	3	1	18	1582,84	1619,71	1,49	0,70	1548,79	1571,94	3,11	2,15
3600	3	2	18	1785,62	1850,93	2,35	0,64	1715,60	1823,29	5,20	1,88
3600	3	3	18	2526,46	2620,71	2,78	0,84	2437,27	2555,11	4,51	2,34

Continua na próxima página

Tabela 5.3 – Continuação da página anterior

A	m	Tipo	Tempo	Melhores soluções				Média			
				Hiper-volume(*10 ⁶)		D1 _R		Hiper-volume(*10 ⁶)		D1 _R	
				MOGA	MOGLS	MOGA	MOGLS	MOGA	MOGLS	MOGA	MOGLS
3600	4	0	24	1366,99	1405,38	2,11	0,70	1326,54	1379,11	4,51	1,86
3600	4	1	24	2404,26	2475,43	2,67	1,15	2358,48	2401,01	4,19	2,89
3600	4	2	24	2324,70	2484,61	3,69	0,58	2259,51	2418,74	6,25	2,19
3600	4	3	24	2551,06	2674,13	2,36	0,30	2479,19	2540,29	3,98	3,17
3600	5	0	30	692,21	758,69	6,02	0,46	664,86	707,76	8,33	4,98
3600	5	1	30	1054,42	1072,54	2,20	1,96	981,13	1005,24	6,65	4,97
3600	5	2	30	1671,98	1732,74	3,29	1,25	1555,44	1630,79	6,43	3,80
3600	5	3	30	2199,77	2236,13	3,40	2,77	2033,00	2153,28	7,19	4,76

É importante lembrar que o método MOGA implementado utiliza as estratégias evolutivas muito utilizadas e com reconhecida qualidade do método NSGA II. Mas os valores apresentados na tabela demonstram a superioridade do método MOGLS sobre o MOGA, já que o primeiro apresenta melhores resultados em ambas as métricas para todas as instâncias. Ressalta-se, assim, que a Busca Local, apesar de reduzir a diversidade por limitar o número de gerações, é importante para superar as dificuldades impostas pelas características de sincronização.

As Figuras 5.6(a)-5.6(d) apresentam quatro conjuntos de aproximações obtidos com diferentes execuções na mesma instância, e permitem comparar os melhores e piores conjuntos de aproximações gerados por cada método considerando a métrica $D1_R$. É possível identificar que as soluções geradas por MOGLS são consideravelmente melhores se observadas a extensão da fronteira e a proximidade com o conjunto de referência.

Observando essas representações de fronteiras de referência, foi possível identificar uma característica que está relacionada com os tipos de instância, mais especificamente com o intervalo em que os valores de prêmios dos arcos foram gerados. Fronteiras relacionadas com instâncias do tipo 0 ou 1 apresentam um Joelho proeminente, ilustrado na Figura 5.7(a), enquanto que as fronteiras relativas às instâncias de tipo 2 e 3 são mais arredondadas, como na Figura 5.7(b).

Conforme indicado em (BRANKE et al., 2004), tais Joelhos estão relacionadas com regiões no espaço de soluções em que pequenas melhorias em um objetivo resultam em grande deterioração (relativa) no outro. O trabalho faz referência ainda ao fato de que tal Joelho facilita a decisão de um eventual decisor humano responsável por escolher uma solução dentre todas da fronteira, uma vez que o Joelho demarca a região com melhor equilíbrio entre os objetivos. Nas demais regiões, pequenas melhorias em um objetivo levam a degradação relativamente severa no outro.

No que diz respeito ao conjunto de instâncias propostas no presente trabalho, essa característica está associada com a variação na razão p_{ij}/c_{ij} . Quando a variação desta razão é pequena, a troca de um arco por outro na solução, ambos com prêmio, terá impacto relativamente pequeno no objetivo de coleta de prêmios, mas pode ter impacto significativo

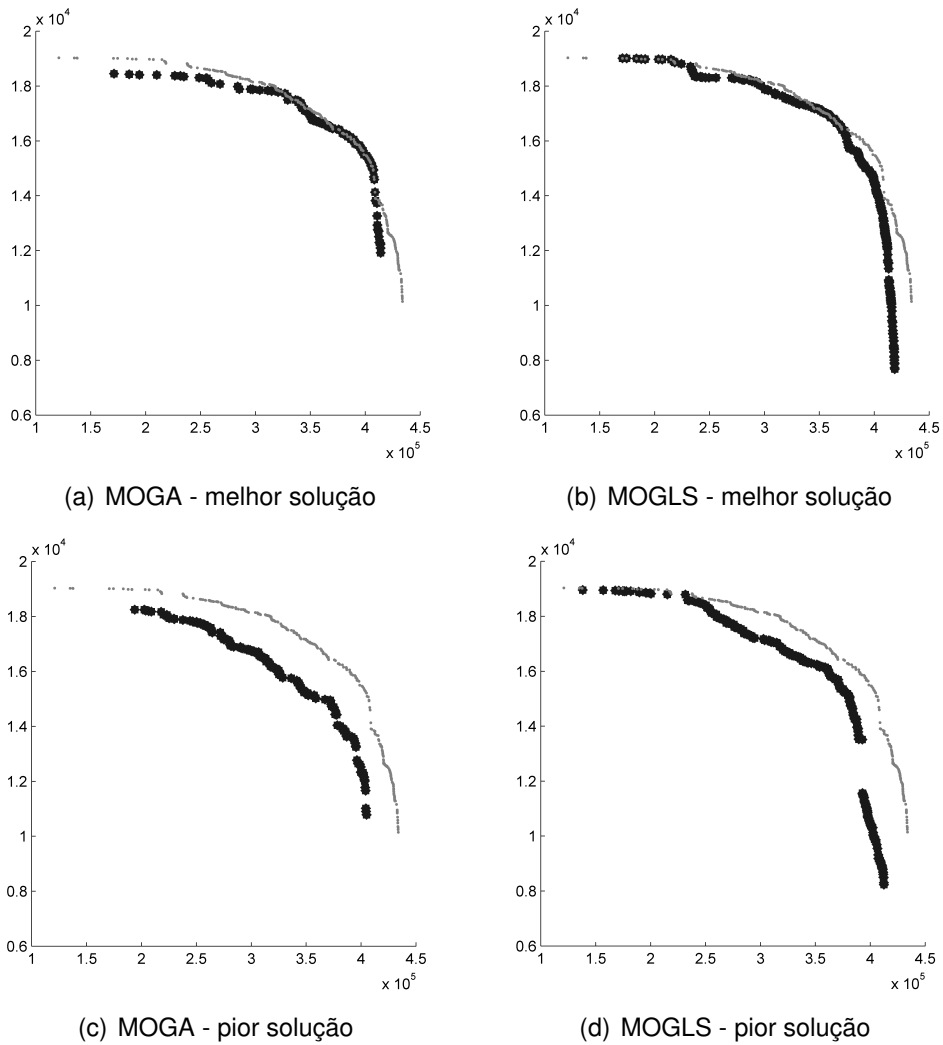


Figura 5.6 – Melhores e piores fronteiras de aproximação para MOGA e MOGLS de acordo com a métrica $D1_R$. Todas as fronteiras estão relacionadas a uma instância do tipo três, com $|A| = 3600$ e $m = 5$. Pequeno pontos cinza representam soluções na fronteira de referência, e grandes pontos pretos representam a soluções da fronteira de aproximação da respectiva execução. O eixo vertical representa o objetivo de coleta de prêmios, enquanto o horizontal representa o objetivo de dispersão.

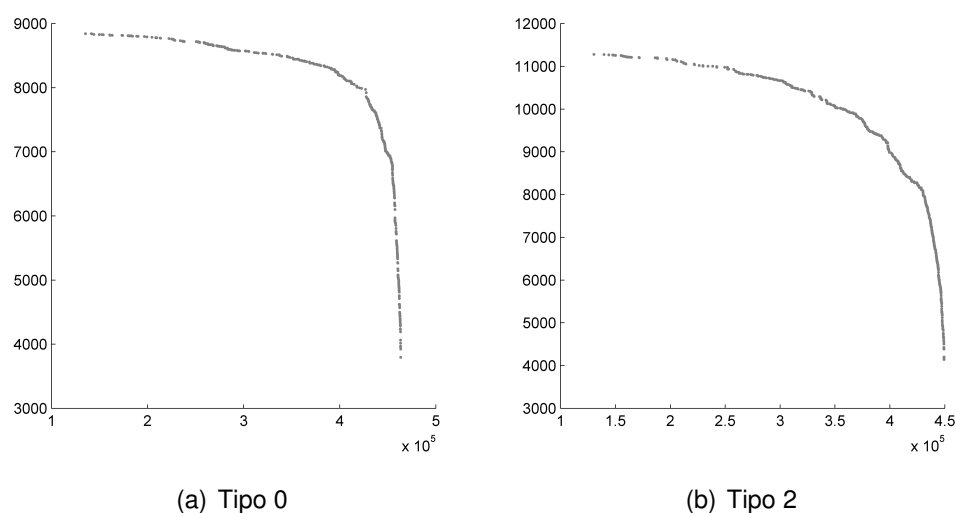


Figura 5.7 – Comparação de duas fronteiras de referência para instâncias com diferentes variações nos valores de prêmio associado aos arcos. As duas fronteiras estão relacionadas com instâncias com $|A| = 1600$ and $m = 4$.

no objetivo de dispersão. Entre soluções com grande parte das rotas planejadas compostas por arcos que agregam prêmios, a variação nos valores de prêmio coletado tendem a ser menos severas do que nos valores de dispersão. Uma vez que soluções com alto valor de coleta de prêmios estejam saturadas, as melhorias na dispersão são alcançados com trocas de arcos que agregam prêmio por outros com $p_{ij} = 0$. A situação então se inverte e mesmo pequenas melhorias no objetivo de dispersão resultam em significativa degradação do objetivo de coleta de prêmios.

Para avaliar esta interpretação, foi executado um teste com uma instância com $p_{ij} = c_{ij}$, gerada a parte dos conjuntos dos outros testes. A fronteira de referência resultante é apresentada na Figura 5.8. É possível identificar um grande segmento na fronteira onde apenas pequenas mudanças no objetivo de coleta de prêmios são verificadas, e também um joelho ainda mais proeminente.

Por outro lado, quando a razão p_{ij}/c_{ij} apresenta uma variação mais alta, tal como nas instâncias de tipo 2 ou 3, alterações em arcos tendem a representar sempre um impacto considerável para os dois objetivos. Segmentos da fronteira que aparecem como linhas praticamente horizontais em uma representação gráfica se tornam improváveis.

Por fim, como a métrica e o objetivo de dispersão são novos na literatura, é importante analisar graficamente as rotas planejadas e a disposição temporal dos veículos. A Figura 5.9 apresenta uma sequência de quadros de uma solução com cinco veículos. Esta solução foi escolhida dentre aquelas localizados no centro da fronteira encontrada em uma execução, isto é, apresenta um equilíbrio entre os dois objetivos. É possível observar a coleta de prêmios enquanto os veículos são mantidos separados. Uma animação com o progresso contínuo desta solução e das demais apresentadas na sequência dessa seção

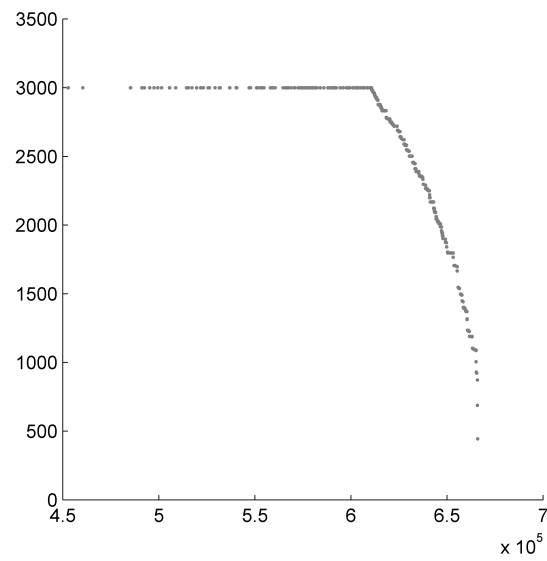


Figura 5.8 – Fronteira de referência para uma instância com $p_{ij} = c_{ij}$.

estão disponíveis em <https://github.com/gdhein/arcRouting>.

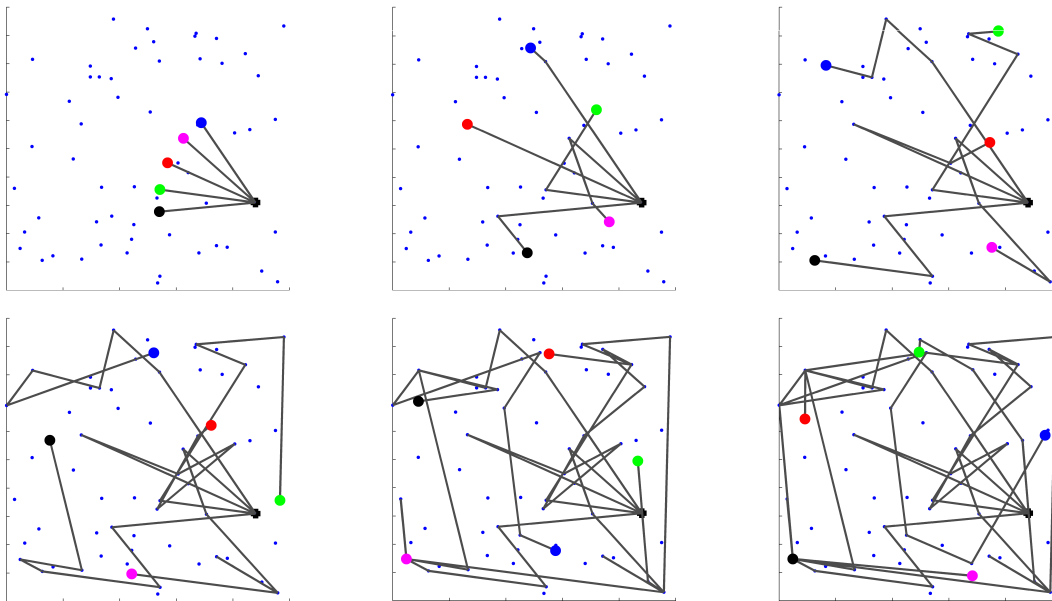


Figura 5.9 – Progressão no tempo de uma solução. A solução apresentada é de uma execução com a instância do tipo 1, de 3600 arcos e cinco veículos, e apresenta um equilíbrio entre os objetivos, não estando próxima de nenhuma das extremidades da fronteira. Os veículos são representados como grandes pontos, e apenas arcos com prêmio diferente de zero são pintados quando atravessados.

Considerando-se soluções extremas, em que um dos objetivos é amplamente favorecido em detrimento do outro, as configurações observadas deixaram clara a natureza biobjetivo do BARP-PD. Em soluções com dispersão muito elevada, foi possível observar configurações com os veículos cruzando muitas vezes os mesmos poucos arcos, concen-

trados em diferentes extremos do ambiente quadrado. A sequência na Figura 5.10 ilustra essa situação, ao representar graficamente uma solução em que os dois veículos envolvidos ficam em um ciclo, cada um se deslocando por arcos simétricos entre um mesmo par de nós colocados em cantos opostos do espaço do grafo, conforme representado entre a terceira e a quinta imagens. Apenas no último instante um dos veículos muda de trajetória e parte para um novo arco. É interessante notar que este novo arco é vantajoso do ponto de vista da dispersão, mas apenas porque uma pequena porção será percorrida no tempo restante de turno. Houvesse mais tempo, o veículo se dirigiria para a parte superior do espaço, e haveria diminuição na distância relativa entre os veículos.

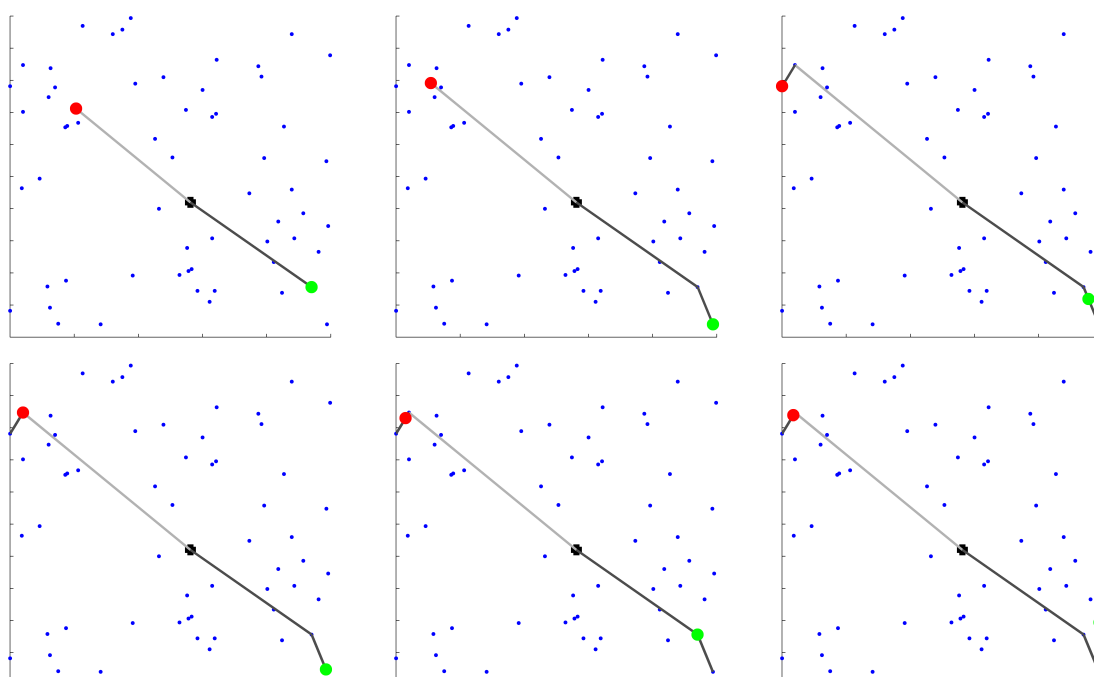


Figura 5.10 – Progressão no tempo de uma solução com máxima dispersão. A solução apresentada é de uma execução com a instância do tipo 2, de 3600 arcos e dois veículos, e se encontra no extremo da fronteira que apresenta valor máximo para a dispersão. A sequência ilustra os veículos andando em ciclos em extremos opostos do espaço dos nós. O último dos seis quadros apresenta a posição dos veículos ao final do turno.

A máxima dispersão entre os veículos se dá exatamente nos extremos do espaço. A sequência ilustrada sugere que num contexto apenas de maximização da dispersão, os veículos se dirigem para esses extremos e lá permanecem. Por isso, o objetivo relacionado à dispersão de veículos em um contexto de roteamento deve estar associado a problemas multiobjetivo, em que existe a imposição de outro(s) objetivo(s) na construção de soluções. Caso contrário, as rotas se resumem a deslocamentos para posições vantajosas, que serão mantidas durante o roteamento.

Por fim, em soluções altamente rentáveis em termos de prêmio total coletado, foi possível observar dois ou mais veículos se deslocando próximos. Situações como essas podem ser observadas na Figura 5.11. Após um começo com um bom afastamento entre

os veículos, estes se aproximam em diversas oportunidades. Vale lembrar que a não dominância entre soluções evita em parte a ocorrência de soluções com dispersão muito ruim. Entre duas soluções com igual valor de prêmio coletado, a de maior dispersão dominará a outra.

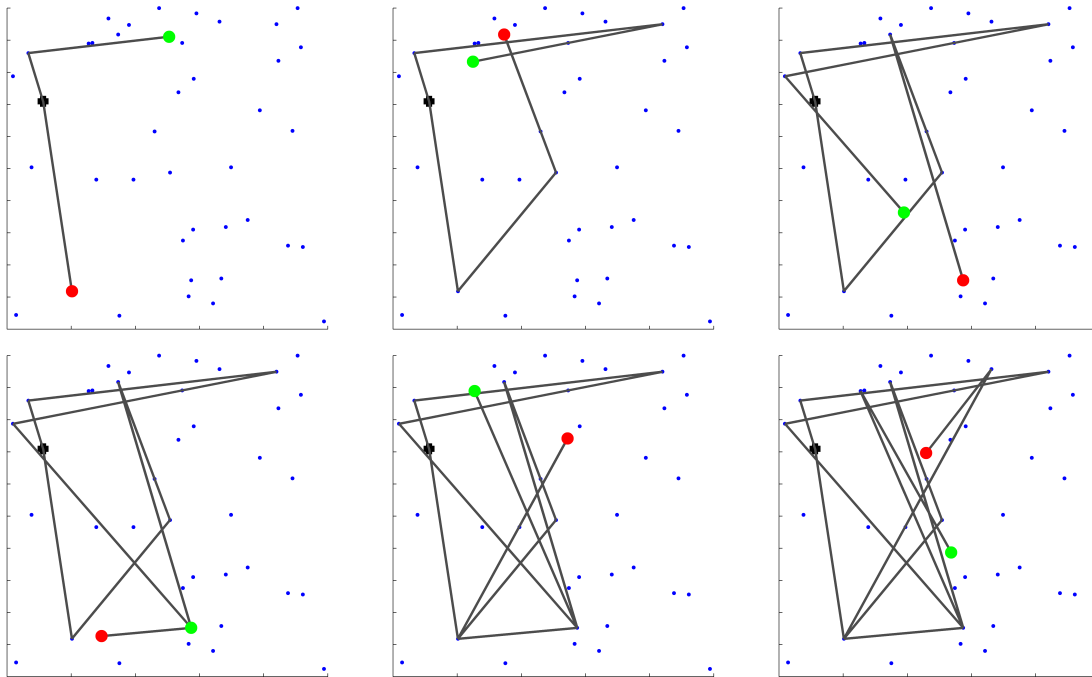


Figura 5.11 – Progressão no tempo de uma solução com máximo prêmio coletado. A solução apresentada é de execução com a instância do tipo 3, de 1600 arcos e dois veículos, e se encontra no extremo da fronteira que apresenta valor máximo para a coleta de prêmio. A sequência ilustra momentos de aproximação entre os veículos. O último dos seis quadros apresenta a posição dos veículos ao final do turno.

5.4 RESUMO DO CAPÍTULO

Nesse capítulo, foi proposto um novo problema de roteamento de arcos biobjetivo, o BARP-PD, com objetivos de maximizar tanto um valor de prêmio coletado quanto a dispersão dos veículos baseada na métrica descrita na Seção 3. A dispersão é relevante para a vigilância e aplicações logísticas de ajuda humanitária.

Foram apresentados dois métodos implementados para a obtenção de soluções para o problema. O primeiro é basicamente a estratégia evolutiva do método NSGA II, e o segundo incorpora um procedimento de Busca Local para melhorar conjuntos não dominados. Ambos os métodos utilizam um algoritmo construtivo para gerar a população inicial e operadores genéticos especializados no laço evolutivo.

Finalmente, foi descrito um conjunto de 64 instâncias originais de *benchmarking*

e apresentados resultados computacionais. Mostra-se que a Busca Local é importante para melhorar a qualidade das soluções, pois o Algoritmo Genético que faz uso de Busca Local apresenta melhores conjuntos de aproximação como resultado. Além disso, foi possível observar a natureza biobjetivo do problema e bons conjuntos de aproximação foram encontrados em tempo razoável.

6 CONCLUSÃO

O presente trabalho apresentou uma nova métrica de dispersão não linear usada na definição de dois novos problemas de roteamento. A métrica está baseada na divisão do tempo de roteamento em fatias de tempo, cada fatia determina um intervalo em que todos os vetores velocidade dos veículos se mantêm inalterados. Dentro da fatia, os veículos são tomados aos pares e a sua distância relativa ao longo de todo o intervalo determina o que é chamado de dispersão entre os veículos na fatia.

A partir dessa métrica, foram propostos dois problemas de roteamento. No primeiro problema, a solução deve prover rotas que mantenham os veículos relativamente próximos, minimizando a dispersão total. Para o cálculo da função objetivo, a dispersão de cada fatia assume o valor do seu par de veículos mais dispersos, e a dispersão total é dada pelo somatório dos valores de dispersão das fatias. Tem-se, portanto, a minimização do somatório das dispersões máximas entre veículos nas fatias. O segundo problema estabelece um objetivo oposto, em que as distâncias relativas dos veículos devem ser maximizadas, mantendo-os dispersos no ambiente. Nesse caso, tem-se a maximização do somatório das dispersões mínimas entre veículos dentro das fatias de tempo.

Tanto a maximização quanto a minimização da dispersão resultam em dependência entre as rotas. Essa dependência tem características particulares em relação a outros problemas de roteamento com sincronização por ser relativa aos veículos e avaliada constantemente, e estabelece uma interdependência espacial e temporal entre os veículos durante todo o roteamento. Por essa característica, o problema apresenta alguma semelhança com problemas de dissimilaridade de caminhos ou rotas, embora muitos desses problemas façam uso de métricas que desconsideram a relação temporal, focando apenas na espacial. Em outros problemas normalmente associados à sincronização existem restrições de sequenciamento ou concomitância na visita a vértices ou no deslocamento por arcos. A semelhança dos problemas propostos na tese é menor com esse tipo de problema, pois nesses casos a sincronização é necessária em instantes ou intervalos definidos, e entre subconjuntos de veículos.

Por se tratarem de problemas novos, com funções objetivo baseadas em métrica também nova e que implicam em característica de sincronização bastante particular, torna-se inviável a comparação dos resultados obtidos com outros encontrados na literatura. Por essa razão, os métodos de solução propostos tinham objetivo primordial de apresentarem resultados consistentes, robustos, e que permitissem avaliar se o comportamento desejado das equipes era obtido nas soluções encontradas. Verificou-se a necessidade de especialização nos métodos para contemplar a dependência espacial e temporal, e percebeu-se a importância da Busca Local para melhorar a robustez dos métodos e melhorar a qualidade das soluções obtidas.

É possível perceber que a Busca Local implementada é mais complexa para o mTSP-D do que para o BARP-PD. Isso pode ser atribuído a duas características do primeiro problema que favoreceram a elaboração de movimentos específicos para ele. A primeira característica refere-se ao tempo de duração das rotas, já que não existe no mTSP-D um tempo de turno a ser cumprido, como ocorre no problema biobjetivo. A percepção de que a redução na duração do roteamento contribui para a redução do custo total abriu a possibilidade de explorar esse fator. A segunda característica refere-se à transitividade da relação de proximidade, que favorece a elaboração dos movimentos de trocas de vértices. A troca de um vértice por outro próximo tende a manter a proximidade espacial em relação às outras rotas nos pontos em que já existe proximidade temporal.

Os distanciamentos máximo e mínimo entre os veículos, se tirados do contexto de roteamento, caracterizam-se como questão de posicionamento, já que a solução ideal seria aquela que destinasse os veículos para os pontos mais ou menos distantes entre si e os mantivesse ali, parados. Mesmo não sendo esse o objetivo no caso de roteamento, a métrica proposta contempla essa situação, o que foi evidenciado nas soluções do BARP-PD em que o objetivo de dispersão é máximo. Os veículos não ficaram parados nos vértices mais distantes pela necessidade de constante movimento imposta pelo problema, mas cada um ficou alternando sua posição entre dois vértices apenas. No contexto de roteamento, em que se destaca a necessidade de visitar vértices, os objetivos implementados se mostraram adequados para obtenção de soluções nas quais é possível equilibrar a dispersão e o cumprimento de tarefas em campo.

Vale lembrar que a união das características dos problemas propostos cobrem um conjunto de variações encontradas em problemas de roteamento, já listadas anteriormente. Embora isso não sirva para atestar com certeza que a aplicação da métrica em outros problemas de roteamento resultará em sucesso, pode servir como um indicativo. Abre-se assim a possibilidade de apresentação de sugestões de trabalhos futuros relacionadas a novos problemas, como exemplo pode ser citado o problema de caminhos dissimilares.

Os testes permitiram a observação de relações entre as características das instâncias e a qualidade das soluções obtidas. Instâncias do mTSP-D em que o tempo de serviço é menos significativo em relação ao tempo de deslocamento apresentaram maior coeficiente de variação médio entre as soluções, o que é um indicativo de maior dificuldade de convergência. Em instâncias do BARP-PD, a relação entre custo e prêmio de um arco aparece como determinante do formato da fronteira de Pareto. Um estudo específico sobre a relação das características de instâncias com a dificuldade para objetivos de posicionamento relativo e contínuo, contemplando uma avaliação do relevo do espaço de soluções definido por tais objetivos, também aparece como um trabalho futuro promissor.

A métrica de dispersão proposta na tese assume todas as distâncias calculadas de acordo com a distância euclidiana. Uma terceira sugestão para trabalho futuro é a criação e aplicação de uma métrica de dispersão derivada da aqui proposta, mas considerando

a distância instantânea calculada por outra métrica. A métrica L_1 (ou métrica do táxi, ou métrica de Manhatthan) aparece como alternativa.

Finalmente, existe também a possibilidade do estudo de outros métodos de solução para os problemas descritos. Para que a comparação de resultados seja possível, tanto as instâncias criadas quanto as soluções obtidas estão disponíveis na Internet.

REFERÊNCIAS BIBLIOGRÁFICAS

- AFIFI, S.; DANG, D.-C.; MOUKRIM, A. A simulated annealing algorithm for the vehicle routing problem with time windows and synchronization constraints. In: **7th International Conference, Learning and Intelligent Optimization (LION 7)**. [S.l.: s.n.], 2013. p. 259–265.
- AKGÜN, V.; ERKUT, E.; BATTÀ, R. On finding dissimilar paths. **European Journal of Operational Research**, v. 121, n. 2, p. 232 – 246, 2000.
- ALTWAIJRY, N.; MENAI, M. E. B. Data structures in multi-objective evolutionary algorithms. **Journal of Computer Science and Technology**, v. 27, n. 6, p. 1197–1210, 2012.
- AMAYA, A.; LANGEVIN, A.; TRÉPANIÉ, M. The capacitated arc routing problem with refill points. **Operations Research Letters**, v. 35, n. 1, p. 45 – 53, 2007.
- AMAYA, C.; LANGEVIN, A.; TRÉPANIÉ, M. A heuristic method for the capacitated arc routing problem with refill points and multiple loads. **JORS**, v. 61, n. 7, p. 1095–1103, 2010.
- ARCHETTI, C. et al. The undirected capacitated arc routing problem with profits. **Comput. Oper. Res.**, Elsevier Science Ltd., Oxford, UK, UK, v. 37, n. 11, p. 1860–1869, nov. 2010.
- ARROYO, J. E. C.; ARMENTANO, V. A. Genetic local search for multi-objective flowshop scheduling problems. **European Journal of Operational Research**, v. 167, n. 3, p. 717–738, 2005.
- ASSAD, A. A.; GOLDEN, B. L. Chapter 5 arc routing methods and applications. In: **Network Routing**. [S.l.]: Elsevier, 1995, (Handbooks in Operations Research and Management Science, v. 8). p. 375 – 483.
- BALAS, E. The prize collecting traveling salesman problem. **Networks**, Wiley Subscription Services, Inc., A Wiley Company, v. 19, n. 6, p. 621–636, 1989.
- BALDACCI, R.; MINGOZZI, A.; ROBERTI, R. Recent exact algorithms for solving the vehicle routing problem under capacity and time window constraints. **European Journal of Operational Research**, v. 218, n. 1, p. 1 – 6, 2012.
- BANDYOPADHYAY, S. et al. A simulated annealing-based multiobjective optimization algorithm: Amosa. **IEEE Transactions on Evolutionary Computation**, v. 12, n. 3, p. 269–283, June 2008.
- BEKTAS, T. The multiple traveling salesman problem: an overview of formulations and solution procedures. **Omega**, v. 34, n. 3, p. 209 – 219, 2006.
- BERRY, A.; VAMPLEW, P. An efficient approach to unbounded bi-objective archives -: Introducing the mak_tree algorithm. In: **Proceedings of the 8th Annual Conference on Genetic and Evolutionary Computation**. New York, NY, USA: ACM, 2006. (GECCO '06), p. 619–626. ISBN 1-59593-186-4.
- BLACK, D.; EGGLESE, R.; WØHLK, S. The time-dependent prize-collecting arc routing problem. **Computers & Operations Research**, v. 40, n. 2, p. 526 – 535, 2013. ISSN 0305-0548.

BOWERMAN, R.; HALL, B.; CALAMAI, P. A multi-objective optimization approach to urban school bus routing: Formulation and solution method. **Transportation Research Part A: Policy and Practice**, Elsevier, v. 29, n. 2, p. 107–123, 1995.

BRANDÃO, J. A tabu search algorithm for the heterogeneous fixed fleet vehicle routing problem. **Computers & Operations Research**, v. 38, n. 1, p. 140 – 151, 2011.

BRANKE, J. et al. Finding knees in multi-objective optimization. In: _____. **Parallel Problem Solving from Nature - PPSN VIII: 8th International Conference, Birmingham, UK, September 18-22, 2004. Proceedings**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 722–731.

BREDSTRÖM, D.; RÖNNQVIST, M. Combined vehicle routing and scheduling with temporal precedence and synchronization constraints. **European Journal of Operational Research**, v. 191, n. 1, p. 19 – 31, 2008.

CÁCERES-CRUZ, J. et al. Applying a savings algorithm for solving a rich vehicle routing problem in a real urban context. **Lecture Notes in Management Science - 5th International Conference on Applied Operational Research, Proceedings**, v. 5, p. 84–92, 2013.

CAMACHO-COLLADOS, M.; LIBERATORE, F.; ANGULO, J. A multi-criteria police districting problem for the efficient and effective design of patrol sector. **European Journal of Operational Research**, v. 246, n. 2, p. 674 – 684, 2015.

CARRABS, F.; CORDEAU, J.-F.; LAPORTE, G. Variable neighborhood search for the pickup and delivery traveling salesman problem with lifo loading. **INFORMS Journal on Computing**, v. 19, n. 4, p. 618–632, 2007.

CATTARUZZA, D. et al. A memetic algorithm for the multi trip vehicle routing problem. **European Journal of Operational Research**, v. 236, n. 3, p. 833 – 848, 2014.

CHEUNG, B. K.-S. et al. Dynamic routing model and solution methods for fleet management with mobile technologies. **International Journal of Production Economics**, v. 113, n. 2, p. 694 – 705, 2008.

CORDEAU, J.-F. et al. The vehicle routing problem. In: TOTH, P.; VIGO, D. (Ed.). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. cap. VRP with Time Windows, p. 157–193. Disponível em: <<http://dl.acm.org/citation.cfm?id=505847.505854>>.

CORMEN, T. H. et al. **Introduction to Algorithms, Third Edition**. 3rd. ed. [S.l.]: The MIT Press, 2009. ISBN 0262033844, 9780262033848.

CURTIN, K. M.; HAYSLETT-MCCALL, K.; QIU, F. Determining optimal police patrol areas with maximal covering and backup covering location models. **Networks and Spatial Economics**, v. 10, n. 1, p. 125–145, 2007.

CZUMAJ, A. Euclidean traveling salesman problem. In: _____. **Encyclopedia of Algorithms**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014. p. 1–6. ISBN 978-3-642-27848-8. Disponível em: <http://dx.doi.org/10.1007/978-3-642-27848-8_131-3>.

CZYZAK, P.; JASZKIEWICZ, A. Pareto simulated annealing – a metaheuristic technique for multiple-objective combinatorial optimization. **Journal of Multi-Criteria Decision Analysis**, John Wiley & Sons, Ltd., v. 7, n. 1, p. 34–47, 1998.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. **Trans. Evol. Comp**, IEEE Press, Piscataway, NJ, USA, v. 6, n. 2, p. 182–197, abr. 2002.

_____. An interactive evolutionary multiobjective optimization method based on progressively approximated value functions. **IEEE Transactions on Evolutionary Computation**, v. 14, n. 5, p. 723–739, Oct 2010.

DELL'OLMO, P.; GENTILI, M.; SCOZZARI, A. On finding dissimilar pareto-optimal paths. **European Journal of Operational Research**, v. 162, n. 1, p. 70 – 82, 2005.

DERIGS, U.; PULLMANN, M.; VOGEL, U. Truck and trailer routing—problems, heuristics and computational experience. **Computers & Operations Research**, v. 40, n. 2, p. 536 – 546, 2013.

DESAULNIERS, G. et al. The vehicle routing problem. In: TOTH, P.; VIGO, D. (Ed.). Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. cap. VRP with Pickup and Delivery, p. 225–242.

DEWILDE, T. et al. Heuristics for the traveling repairman problem with profits. **Computers & Operations Research**, v. 40, n. 7, p. 1700 – 1707, 2013.

DOHN, A.; KOLIND, E.; CLAUSEN, J. The manpower allocation problem with time windows and job-teaming constraints: A branch-and-price approach. **Computers & Operations Research**, v. 36, n. 4, p. 1145 – 1157, 2009.

DREXL, M. Rich vehicle routing in theory and practice. **Logistics Research**, v. 5, n. 1, p. 47–63, 2012a.

_____. Synchronization in vehicle routing - A survey of vrps with multiple synchronization constraints. **Transportation Science**, v. 46, n. 3, p. 297–316, 2012b.

_____. Applications of the vehicle routing problem with trailers and transshipments. **European Journal of Operational Research**, v. 227, n. 2, p. 275 – 283, 2013.

EISELT, H. A.; GENDREAU, M.; LAPORTE, G. Arc routing problems, part I: the chinese postman problem. **Operations Research**, v. 43, n. 2, p. 231–242, 1995a.

_____. Arc routing problems, part II: the rural postman problem. **Operations Research**, v. 43, n. 3, p. 399–414, 1995b.

EKSIOGLU, B.; VURAL, A. V.; REISMAN, A. The vehicle routing problem: A taxonomic review. **Computers & Industrial Engineering**, v. 57, n. 4, p. 1472 – 1483, 2009.

FEILLET, D.; DEJAX, P.; GENDREAU, M. The profitable arc tour problem: Solution with a branch-and-price algorithm. **Transportation Science**, v. 39, n. 4, p. 539–552, 2005a.

_____. Traveling salesman problems with profits. **Transportation Science**, v. 39, n. 2, p. 188–205, 2005b.

FIELDSEND, J. E.; EVERSON, R. M.; SINGH, S. Using unconstrained elite archives for multiobjective optimization. **IEEE Transactions on Evolutionary Computation**, v. 7, n. 3, p. 305–323, June 2003.

FIKAR, C.; HIRSCH, P. A matheuristic for routing real-world home service transport systems facilitating walking. **Journal of Cleaner Production**, v. 105, p. 300 – 310, 2015.

FLEMING, C. L.; GRIFFIS, S. E.; BELL, J. E. The effects of triangle inequality on the vehicle routing problem. **European Journal of Operational Research**, v. 224, n. 1, p. 1 – 7, 2013.

FRAMINAN, J. M.; LEISTEN, R. A multi-objective iterated greedy search for flowshop scheduling with makespan and flowtime criteria. **OR Spectrum**, v. 30, n. 4, p. 787–804, 2008.

GABBAY, H. **An overview of vehicular scheduling problems**. [S.I.], 1974.

GHAFURIAN, S.; JAVADIAN, N. An ant colony algorithm for solving fixed destination multi-depot multiple traveling salesmen problems. **Applied Soft Computing**, v. 11, n. 1, p. 1256 – 1262, 2011.

GRANDINETTI, L. et al. An optimization-based heuristic for the multi-objective undirected capacitated arc routing problem. **Computers & Operations Research**, v. 39, n. 10, p. 2300 – 2309, 2012.

HÀ, M. H. et al. Solving the close-enough arc routing problem. **Networks**, v. 63, n. 1, p. 107–118, 2014.

HACHICHA, M. et al. Heuristics for the multi-vehicle covering tour problem. **Computers & Operations Research**, v. 27, n. 1, p. 29 – 42, 2000.

HAUPT, R. L.; HAUPT, S. E. **Practical genetic algorithms**. Hoboken, N.J.: J. Wiley, 2004.

HELBIG, M.; ENGELBRECHT, A. P. Performance measures for dynamic multi-objective optimisation algorithms. **Information Sciences**, v. 250, p. 61 – 81, 2013.

HENKE, T.; SPERANZA, M. G.; WÄSCHER, G. The multi-compartment vehicle routing problem with flexible compartment sizes. **European Journal of Operational Research**, v. 246, n. 3, p. 730 – 743, 2015.

HOCHBAUM, D. S.; LYU, C.; ORDÓÑEZ, F. Security routing games with multivehicle chinese postman problem. **Netw.**, Wiley-Interscience, New York, NY, USA, v. 64, n. 3, p. 181–191, out. 2014.

HOLLAND, J. **Adaptation in natural and artificial systems**. The University of Michigan Press, Ann Arbor, 1975.

ISHIBUCHI, H.; YOSHIDA, T.; MURATA, T. Balance between genetic search and local search in memetic algorithms for multiobjective permutation flowshop scheduling. **IEEE Transactions on Evolutionary Computation**, v. 7, n. 2, p. 204–223, April 2003.

JAFFRÈS-RUNSER, K.; GORCE, J.; COMANICIU, C. A multiobjective tabu framework for the optimization and evaluation of wireless systems. In: _____. **Local Search Techniques: Focus on Tabu Search**. Vienna, AUT: I-Tech, 2008. p. 29–54.

JENSEN, M. T. Reducing the run-time complexity of multiobjective eas: The nsga-ii and other algorithms. **Trans. Evol. Comp**, IEEE Press, Piscataway, NJ, USA, v. 7, n. 5, p. 503–515, out. 2003.

JIANG, S. et al. Consistencies and contradictions of performance metrics in multiobjective optimization. **IEEE Transactions on Cybernetics**, v. 44, n. 12, p. 2391–2404, Dec 2014.

JOHNSON, D. S.; MCGEOCH, L. A. The Traveling Salesman Problem: A Case Study in Local Optimization. In: _____. **Local Search in Combinatorial Optimization**. London: John Wiley and Sons, 1997. p. 215–310.

JOZEFOWIEZ, N.; SEMET, F.; TALBI, E.-G. Multi-objective vehicle routing problems. **European Journal of Operational Research**, v. 189, n. 2, p. 293 – 309, 2008.

KARA, I.; BEKTAS, T. Integer linear programming formulations of multiple salesman problems and its variations. **European Journal of Operational Research**, v. 174, n. 3, p. 1449–1458, November 2006.

KARAKATIC, S.; PODGORELEC, V. A survey of genetic algorithms for solving multi depot vehicle routing problem. **Applied Soft Computing**, v. 27, p. 519 – 532, 2015.

KNOWLES, J.; CORNE, D. On metrics for comparing nondominated sets. In: **Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on**. [S.l.: s.n.], 2002. v. 1, p. 711–716.

KNOWLES, J. D.; CORNE, D. W. Approximating the nondominated front using the pareto archived evolution strategy. **Evol. Comput.**, MIT Press, Cambridge, MA, USA, v. 8, n. 2, p. 149–172, jun. 2000.

KRITIKOS, M. N.; IOANNOU, G. The balanced cargo vehicle routing problem with time windows. **International Journal of Production Economics**, v. 123, n. 1, p. 42 – 51, 2010.

KULKARNI, A. J.; TAI, K. Probability collectives: A multi-agent approach for solving combinatorial optimization problems. **Applied Soft Computing**, v. 10, n. 3, p. 759 – 771, 2010.

LABADI, N.; PRINS, C.; REGHIOUI, M. Grasp with path relinking for the capacitated arc routing problem with time windows. In: _____. **Advances in Computational Intelligence in Transport, Logistics, and Supply Chain Management**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. p. 111–135.

LACOMME, P.; PRINS, C.; SEVAUX, M. A genetic algorithm for a bi-objective capacitated arc routing problem. **Computers & Operations Research**, v. 33, n. 12, p. 3473 – 3493, 2006. Part Special Issue: Recent Algorithmic Advances for Arc Routing Problems.

LAWLER, E. L. et al. **The Traveling salesman problem : a guided tour of combinatorial optimization**. Chichester, New York, Brisbane: J. Wiley and sons, 1987. (Wiley interscience series in discrete mathematics and optimization).

LEISS, E. et al. Multi-objective variable neighborhood search algorithms for a single machine scheduling problem with distinct due windows. **Electronic Notes in Theoretical Computer Science - Proceedings of the 2011 Latin American Conference in Informatics (CLEI)**, v. 281, p. 5 – 19, 2011.

LI, J.-Q.; MIRCHANDANI, P. B.; BORENSTEIN, D. Real-time vehicle rerouting problems with time windows. **European Journal of Operational Research**, v. 194, n. 3, p. 711 – 727, 2009.

LIN, S.-W.; YU, V. F.; LU, C.-C. A simulated annealing heuristic for the truck and trailer routing problem with time windows. **Expert Systems with Applications**, v. 38, n. 12, p. 15244 – 15252, 2011.

LIU, L.; MU, H.; YANG, J. Simulated annealing based grasp for pareto-optimal dissimilar paths problem. **Soft Computing**, p. 1–17, 2016.

LIU, L. et al. A simulated annealing for multi-criteria optimization problem: {DBMOSA}. **Swarm and Evolutionary Computation**, v. 14, p. 48 – 65, 2014.

LOMBARD, K.; CHURCH, R. The gateway shortest path problem: generation of alternative routes for a corridor location problem. **Geographical Systems**, v. 1, p. 25–45, 1993.

LÓPEZ-SÁNCHEZ, A. et al. A multi-start algorithm for a balanced real-world open vehicle routing problem. **European Journal of Operational Research**, v. 238, n. 1, p. 104 – 113, 2014.

MALANDRAKI, C.; DASKIN, M. S. Time dependent vehicle routing problems - formulations, properties and heuristic algorithms. **Transportation Science**, v. 26, n. 3, p. 185–200, 1992.

_____. The maximum benefit Chinese postman problem and the maximum benefit traveling salesman problem. **European Journal of Operational Research**, v. 65, n. 2, p. 218–234, 1993.

MANKOWSKA, D. S.; MEISEL, F.; BIERWIRTH, C. The home health care routing and scheduling problem with interdependent services. **Health Care Management Science**, v. 17, n. 1, p. 15–30, 2014.

MARTÍ, R. et al. Multiobjective grasp with path relinking. **European Journal of Operational Research**, v. 240, n. 1, p. 54 – 71, 2015.

MARTÍ, R.; VELARDE, J. L. G.; DUARTE, A. Heuristics for the bi-objective path dissimilarity problem. **Computers & Operations Research**, v. 36, n. 11, p. 2905 – 2912, 2009.

MARTÍNEZ-VARGAS, A. et al. Application of nsga-ii algorithm to the spectrum assignment problem in spectrum sharing networks. **Applied Soft Computing**, v. 39, p. 188 – 198, 2016.

MASSON, R.; LEHUÉDÉ, F.; PÉTON, O. The dial-a-ride problem with transfers. **Computers & Operations Research**, v. 41, p. 12 – 23, 2014.

MITROVIĆ-MINIĆ, S.; LAPORTE, G. The pickup and delivery problem with time windows and transshipment. **INFOR: Information Systems and Operational Research**, v. 44, n. 3, p. 217–227, 2006.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Comput. Oper. Res.**, Elsevier Science Ltd., Oxford, UK, v. 24, n. 11, p. 1097–1100, nov. 1997.

MOSTAGHIM, S.; TEICH, J.; TYAGI, A. Comparison of data structures for storing pareto-sets in moeas. In: **Evolutionary Computation, 2002. CEC '02. Proceedings of the 2002 Congress on**. [S.l.: s.n.], 2002. v. 1, p. 843–848.

MOUSAVI, S. M. et al. A bi-objective inventory optimization model under inflation and discount using tuned pareto-based algorithms: Nsga-ii, nrga, and {MOPSO}. **Applied Soft Computing**, v. 43, p. 57 – 72, 2016.

ÑAHUIS, F. V. C.; FLORES, M. J. R. Automação do Despacho Dinâmico de Viaturas para o Atendimento das Ordens de Serviços nas Redes de Distribuição de Energia Elétrica. In: **Anais do XLV Simpósio Brasileiro de Pesquisa Operacional**. Natal/RN: [s.n.], 2013.

NARASIMHA, K. V. et al. An ant colony optimization technique for solving min–max multi-depot vehicle routing problem. **Swarm and Evolutionary Computation**, v. 13, p. 63 – 73, 2013.

NAZIF, H.; LEE, L. S. Optimised crossover genetic algorithm for capacitated vehicle routing problem. **Applied Mathematical Modelling**, v. 36, n. 5, p. 2110 – 2117, 2012.

NEBRO, A. J. et al. Abyss: Adapting scatter search to multiobjective optimization. **IEEE Transactions on Evolutionary Computation**, v. 12, n. 4, p. 439–457, Aug 2008.

NECULA, R.; BREABAN, M.; RASCHIP, M. Performance evaluation of ant colony systems for the single-depot multiple traveling salesman problem. In: **Hybrid Artificial Intelligent Systems**. [S.l.]: Springer, 2015. p. 257 – 268.

NGUEVEU, S. U.; PRINS, C.; CALVO, R. W. A hybrid tabu search for the m-peripatetic vehicle routing problem. In: _____. **Matheuristics: Hybridizing Metaheuristics and Mathematical Programming**. Boston, MA: Springer US, 2010. p. 253–266.

PARRAGH, S. N.; DOERNER, K. F.; HARTL, R. F. A survey on pickup and delivery problems - part I: Transportation between customers and depot. **Journal für Betriebswirtschaft**, v. 58, n. 1, p. 21–51, 2008a.

_____. A survey on pickup and delivery problems - part II: Transportation between pickup and delivery locations. **Journal für Betriebswirtschaft**, v. 58, n. 2, p. 81–117, 2008b.

PIA, A. D.; FILIPPI, C. A variable neighborhood descent algorithm for a real waste collection problem with mobile depots. **International Transactions in Operational Research**, Blackwell Publishing Ltd, v. 13, n. 2, p. 125–141, 2006.

PILLAC, V. et al. A review of dynamic vehicle routing problems. **European Journal of Operational Research**, v. 225, n. 1, p. 1 – 11, 2013.

QU, Y.; BARD, J. F. A grasp with adaptive large neighborhood search for pickup and delivery problems with transshipment. **Computers & Operations Research**, v. 39, n. 10, p. 2439 – 2456, 2012.

RASHIDI, H.; FARAHANI, R. Z. A hybrid ant colony system for partially dynamic vehicle routing problem. **American Journal of Operational Research**, v. 2, n. 4, p. 31 – 44, 2012.

RIBEIRO, G. M.; LAPORTE, G. An adaptive large neighborhood search heuristic for the cumulative capacitated vehicle routing problem. **Computers & Operations Research**, v. 39, n. 3, p. 728 – 735, 2012.

RODRÍGUEZ, A.; RUIZ, R. A study on the effect of the asymmetry on real capacitated vehicle routing problems. **Computers & Operations Research**, v. 39, n. 9, p. 2142 – 2151, 2012.

ROSA, B. D. et al. The arc routing and scheduling problem with transshipment. **Transportation Science**, v. 36, n. 3, p. 301–313, 2002.

SALAZAR-AGUILAR, M. A.; LANGEVIN, A.; LAPORTE, G. Synchronized arc routing for snow plowing operations. **Computers & Operations Research**, v. 39, n. 7, p. 1432 – 1440, 2012.

_____. The synchronized arc and node routing problem: Application to road marking. **Computers & Operations Research**, v. 40, n. 7, p. 1708 – 1715, 2013.

SHAFABI, A.; HAGHANI, A. Generalized maximum benefit multiple chinese postman problem. **Transportation Research Part C: Emerging Technologies**, v. 55, p. 261 – 272, 2015.

SHAHRIARI, M. Multi-objective optimization of discrete time–cost tradeoff problem in project networks using non-dominated sorting genetic algorithm. **Journal of Industrial Engineering International**, v. 12, n. 2, p. 159–169, 2016.

SHANG, R. et al. An improved decomposition-based memetic algorithm for multi-objective capacitated arc routing problem. **Applied Soft Computing**, v. 19, p. 343 – 361, 2014.

SILVA, M. M. et al. A simple and effective metaheuristic for the minimum latency problem. **European Journal of Operational Research**, Elsevier, v. 221, n. 3, p. 513–520, 2012.

SIMON, D. **Evolutionary Optimization Algorithms**. [S.l.]: John Wiley & Sons, 2013.

SINGH, A.; BAGHEL, A. S. A new grouping genetic algorithm approach to the multiple traveling salesperson problem. **Soft Computing**, v. 13, n. 1, p. 95–101, 2008.

SPLIET, R.; GABOR, A. F.; DEKKER, R. The vehicle rescheduling problem. **Computers & Operations Research**, Elsevier Science Ltd., Oxford, UK, UK, v. 43, p. 129–136, mar. 2014.

TAHA, H. A. **Operations Research: An Introduction (8th Edition)**. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006. ISBN 0131889230.

TALARICO, L.; SÖRENSEN, K.; SPRINGAEL, J. The k-dissimilar vehicle routing problem. **European Journal of Operational Research**, v. 244, n. 1, p. 129 – 140, 2015.

TAVARES, G. et al. Optimisation of MSW collection routes for minimum fuel consumption using 3D GIS modelling. **Waste Management**, v. 29, n. 3, p. 1176–1185+, 2009.

THIELE, L. et al. A preference-based evolutionary algorithm for multi-objective optimization. **Evol. Comput.**, MIT Press, Cambridge, MA, USA, v. 17, n. 3, p. 411–436, set. 2009.

THYAGARAJAN, K. et al. Planning dissimilar paths for military units. **Military Operations Research**, v. 40, n. 1, p. 25 – 42, 2005.

TLILI, T. et al. On solving the multi-depot vehicle routing problem. In: _____. **Proceedings of 3rd International Conference on Advanced Computing, Networking and Informatics: ICACNI 2015, Volume 2**. New Delhi: Springer India, 2016. p. 103–108.

TOTH, P.; VIGO, D. An exact algorithm for the vehicle routing problem with backhauls. **Transportation Science**, v. 31, n. 4, p. 372–385, 1997.

TOTH, P.; VIGO, D. (Ed.). **The Vehicle Routing Problem**. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2001. ISBN 0-89871-498-2.

URK, R. van; MES, M. R.; HANS, E. W. Anticipatory routing of police helicopters. **Expert systems with applications**, Elsevier, v. 40, n. 17, p. 6938 – 6947, 2013.

VANSTEENWEGEN, P.; SOUFFRIAU, W.; OUDHEUSDEN, D. V. The orienteering problem: A survey. **European Journal of Operational Research**, v. 209, n. 1, p. 1 – 10, 2011.

VENKATESH, P.; SINGH, A. Two metaheuristic approaches for the multiple traveling salesperson problem. **Applied Soft Computing**, v. 26, p. 74 – 89, 2015.

VIDAL, T. et al. Heuristics for multi-attribute vehicle routing problems: A survey and synthesis. **European Journal of Operational Research**, v. 231, n. 1, p. 1 – 21, 2013.

VITORIANO, B.; ORTUÑO, T.; TIRADO, G. Hads, a goal programming-based humanitarian aid distribution system. **Journal of Multi-Criteria Decision Analysis**, John Wiley & Sons, Ltd., v. 16, n. 1-2, p. 55–64, 2009.

WEX, F.; SCHRYEN, G.; NEUMANN, D. Assignments of collaborative rescue units during emergency response. **IJISCRAM**, v. 5, n. 4, p. 63–80, 2013.

WILLEMSE, E. J.; JOUBERT, J. W. Applying min-max k postmen problems to the routing of security guards. **The Journal of the Operational Research Society**, v. 63, n. 2, p. 245–260, 2012.

WØHLK, S. A decade of capacitated arc routing. In: _____. **The Vehicle Routing Problem: Latest Advances and New Challenges**. Boston, MA: Springer US, 2008. p. 29–48.

ZACHARIADIS, E.; KIRANOUDIS, C. Local search for the undirected capacitated arc routing problem with profits. **European Journal of Operational Research**, v. 210, n. 2, p. 358 – 367, 2011.

ZITZLER, E.; LAUMANN, M.; THIELE, L. Spea2: Improving the strength pareto evolutionary algorithm for multiobjective optimization. In: **Evolutionary Methods for Design, Optimisation, and Control**. [S.l.]: CIMNE, Barcelona, Spain, 2002. p. 95–100.

ZITZLER, E.; THIELE, L. Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. **IEEE Transactions on Evolutionary Computation**, v. 3, n. 4, p. 257–271, Nov 1999.

APÊNDICE A – LISTA DE TRABALHOS ORIGINAIS

Ao longo do desenvolvimento da tese, os seguintes artigos foram submetidos para publicação:

1) Artigos publicados

GARCIA, V. J.; BERNARDON, D. P. ; ABAIDE, A. R. ; ARAÚJO, O. C. B. ; DHEIN, G. . Reliability assessment by coordinating maintenance vehicles in electric power distribution systems. In: **Procedia: Social and Behavioral Sciences**, v. 111, p. 1045-1053, 2014.

DHEIN, G. ; ZANETTI, M. S. ; MÜLLER, F. M. ; ARAÚJO, O. C. B. ; CARDOSO JUNIOR, G. . Minimização da Dispersão em um Problema de Roteamento de Veículos. In: XLVIII SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2016.

GARCIA, V. J.; ARAÚJO, O. C. B. ; DHEIN, G. ; BERNARDON, D. P. ; CARDOSO JUNIOR, G. . Proposta de modelo matemático para o problema do despacho de ordens de serviço emergenciais em concessionária de distribuição de energia elétrica. In: XLV Simpósio Brasileiro de Pesquisa Operacional, 2013, Natal. **Anais... XLV SBPO 2013**, 2013.

GARCIA, V. J.; BERNARDON, D. P. ; DHEIN, G. ; ARAÚJO, O. C. B. ; ABAIDE, A. ; KUMMER NETO, A. F. ; FONINI, J. ; DAZA, E. F. B. . Multitasked maintenance crews to serve emergency scenarios in electric distribution utilities. In: 48th International Universities' Power Engineering Conference, 2013, Dublin. **Anais... UPEC 2013**, 2013.

FONINI, J. ; DAZA, E. F. B. ; GARCIA, V. J. ; BERNARDON, D. P. ; ABAIDE, A. R. ; ARAÚJO, O. C. B. ; DHEIN, G. . Sistema de Apoio à Decisão para Despacho Automático e Integrado de Ordens de Serviços Emergenciais. In: VII Congresso de Inovação Tecnológica em Energia Elétrica, 2013, Rio de Janeiro. **Anais... CITENEL 2013**, 2013.

GARCIA, V. J.; ARAÚJO, O. C. B. ; DHEIN, G. ; BERNARDON, D. P. ; CARDOSO JUNIOR, G. . Problema de Roteamento de Veículos para Atendimento de Ordens Emergenciais em Concessionária de Distribuição de Energia Elétrica. In: XLIV SBPO - Simpósio Brasileiro de Pesquisa Operacional, 2012, Rio de Janeiro. **Anais... SBPO 2012**, 2012.

2) Artigos submetidos, em processo de revisão

DHEIN, G. ; ZANETTI, M. S. ; ARAÚJO, O. C. B. ; CARDOSO JUNIOR, G. . The Multiple Traveling Salesman Problem with Minimizing Dispersion Objective. **Applied Soft Computing**.

DHEIN, G. ; ARAÚJO, O. C. B. ; CARDOSO JUNIOR, G. . Genetic Algorithms for a New Bi-Objective Arc Routing Problem with Profit Collection and Dispersion of Vehicles. **Engineering Applications of Artificial Intelligence**.