

RENAN JOSÉ DOS SANTOS VIANA

**ABORDAGENS HEURÍSTICAS PARA
OTIMIZAÇÃO DE UM SERVIÇO DE
TRANSPORTE REATIVO A DEMANDA**

Dissertação apresentada a Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

VIÇOSA
MINAS GERAIS-BRASIL
2016

RENAN JOSÉ DOS SANTOS VIANA

**ABORDAGENS HEURÍSTICAS PARA OTIMIZAÇÃO DE UM
SERVIÇO DE TRANSPORTE REATIVO A DEMANDA**

Dissertação apresentada à Universidade Federal de Viçosa, como parte das exigências do Programa de Pós-Graduação em Ciência da Computação, para obtenção do título de *Magister Scientiae*.

APROVADA: 10 de junho de 2016.



Elizabeth Fialho Wanner



Gustavo Peixoto Silva



André Gustavo dos Santos
(Orientador)

**Ficha catalográfica preparada pela Biblioteca Central da Universidade
Federal de Viçosa - Câmpus Viçosa**

T

V614a Viana, Renan José dos Santos, 1991-
2016 Abordagens heurísticas para otimização de um serviço de
transporte reativo a demanda / Renan José dos Santos Viana. –
Viçosa, MG, 2016.
xii,116f. : il. (algumas color.) ; 29 cm.

Inclui apêndice.

Orientador: André Gustavo dos Santos.

Dissertação (mestrado) - Universidade Federal de Viçosa.

Referências bibliográficas: f.110-116.

1. Pesquisa operacional. 2. Programação heurística.
3. Algoritmos. 4. Otimização matemática. 5. Otimização
combinatória. 6. Logística. I. Universidade Federal de Viçosa.
Departamento de Informática. Programa de Pós-graduação em
Ciência da Computação. II. Título.

CDD 22. ed. 003

Dedico este trabalho a Deus e a toda minha família.

AGRADECIMENTOS

Agradeço primeiramente a Deus pela minha vida e família. O agradeço também por me conceder saúde para correr atrás de todos os meus sonhos. Sou eternamente grato aos meus pais, Paulo e Sueli, pela criação e grande dedicação na minha formação. À minha irmã Juliana e meu cunhado Nagai pelo apoio e concessão de importantes conselhos ao longo desta trajetória.

Agradeço também ao meu orientador André, pelo companheirismo, por todo conhecimento compartilhado, por acreditar na minha capacidade, pela compreensão e grande incentivo ao longo destes anos. Agradeço ao professor José Ivo pelas orientações e dicas.

Gostaria de agradecer à Paôla pelo apoio mais que especial e carinho durante estes anos. Aos meus amigos de mestrado agradeço por todos momentos de diversões e pelo companheirismo. Em especial aos meus amigos Jonatas, Matheus, Maurício, Felipe, Julian, Vivian, Thales e Dâmaris.

Enfim, agradeço à CAPES e à FAPEMIG, pelo apoio financeiro, que contribuiu para realização desta pesquisa.

Sumário

Lista de Figuras	vii
Lista de Tabelas	ix
Resumo	xi
Abstract	xii
1 Introdução	1
1.1 Considerações Iniciais	1
1.2 Problema e sua Importância	2
1.3 Hipótese	4
1.4 Objetivos	4
1.4.1 Objetivos Específicos	4
2 Revisão Bibliográfica	6
2.1 Otimização Multiobjetivo	6
2.1.1 Relação de Dominância e Otimalidade Pareto	7
2.2 Algoritmos Evolutivos Multiobjetivo	8
2.2.1 Non-Dominated Sorting Genetic Algorithm II	9
2.2.2 Indicator-based Multiobjective Evolutionary Algorithm	13
2.2.3 Strenght Pareto Evolutionary Algorithm 2	16
2.3 Métodos de Avaliação de Algoritmos Multiobjetivo	19
2.3.1 Cobertura (C)	20
2.3.2 Binary Additive ϵ -Indicator ($I_{\epsilon+}$)	21
3 Transporte Reativo a Demanda	22
3.1 Problemas de Roteamento de Veículos com Coletas e Entregas	22
3.2 Dial-a-Ride Problem	23

3.2.1	Modelo Matemático DARP	25
3.3	Trabalhos Relacionados	27
3.3.1	Trabalhos que tratam o caso estático	28
3.3.2	Trabalhos que tratam o caso dinâmico	32
3.3.3	Delimitação do Escopo do Trabalho	34
3.4	Serviço de Transporte Reativo a Demanda	35
3.4.1	Definição do Problema	36
4	Materiais e Métodos	42
4.1	Representação de uma solução	42
4.2	Geração da População Inicial	43
4.3	Operadores de Seleção e Variação	44
4.3.1	Operador de Seleção	44
4.3.2	Operador de Recombinação (Crossover)	45
4.3.3	Operador de Mutação	48
4.4	Abordagens para Resolução de um Serviço DRT Dinâmico	51
4.4.1	Simulação do serviço DRT Dinâmico	51
4.4.2	Abordagem Baseada em Métodos Exatos	54
4.4.3	Abordagem Heurística Proposta	55
5	Resultados	58
5.1	Ambiente Computacional	58
5.2	Conjunto de Instâncias de Teste	58
5.3	Conjuntos de Testes Estatísticos	59
5.4	Dados de Entrada	60
5.5	Abordagens Aplicadas ao Serviço DRT Estático	60
5.5.1	Análise dos Resultados	62
5.6	Abordagens Aplicadas ao Serviço DRT Dinâmico	72
5.6.1	Considerações Iniciais	72
5.6.2	Método de Avaliação de Desempenho	72
5.6.3	Resultados dos Experimentos Computacionais	73
6	Conclusões	83
6.1	Publicações	84
A	Calibração das Abordagens Heurísticas	85
A.1	Calibração das Abordagens Propostas	85
A.1.1	Calibração NSGA2	87

A.1.2	Calibração IBEA	96
A.1.3	Calibração SPEA2	103
	Referências Bibliográficas	110

Lista de Figuras

2.1	Ordenação por grau de dominância [Deb, 2001]	10
2.2	Distância de Agrupamento [Deb, 2001]	10
2.3	Procedimento do NSGA-II [Deb, 2001]	12
4.1	Estrutura da Abordagem Heurística	56
5.1	Gráfico de Médias - Comparação das Abordagens Desenvolvidas para Resolução do Caso Estático - Conjuntos "Gravit100" e "Rnd100"	64
5.2	Gráfico de Médias - Comparação das Abordagens Desenvolvidas para Resolução do Caso Estático - Conjunto "Gravit100"	67
5.3	Gráfico de Médias - Comparação das Abordagens Desenvolvidas para Resolução do Caso Estático - Conjunto "Rnd100"	70
5.4	Comparação do Aumento do Número de Veículos com a Chegada de Novas Requisições para a Instância Gravit100_0	75
5.5	Comparação do Aumento do Número de Veículos com a Chegada de Novas Requisições para a Instância Rnd100_0	75
A.1	Gráfico <i>boxplot</i> - Recombinação 2 e Mutação 1	88
A.2	Gráfico <i>boxplot</i> - Recombinação 3 e Mutação 1	89
A.3	Gráfico <i>boxplot</i> - Recombinação 4 e Mutação 1	91
A.4	Gráfico <i>boxplot</i> - Recombinação 2 e Mutação 2	92
A.5	Gráfico <i>boxplot</i> - Recombinação 3 e Mutação 2	93
A.6	Gráfico <i>boxplot</i> - Recombinação 4 e Mutação 2	95
A.7	Gráfico <i>boxplot</i> - Recombinação 2 e Mutação 1	97
A.8	Gráfico <i>boxplot</i> - Recombinação 3 e Mutação 1	98
A.9	Gráfico <i>boxplot</i> - Recombinação 4 e Mutação 1	99
A.10	Gráfico <i>boxplot</i> - Recombinação 2 e Mutação 2	100
A.11	Gráfico <i>boxplot</i> - Recombinação 3 e Mutação 2	101
A.12	Gráfico <i>boxplot</i> - Recombinação 4 e Mutação 2	102

A.13 Gráfico <i>boxplot</i> - Recombinação 2 e Mutação 1	104
A.14 Gráfico <i>boxplot</i> - Recombinação 3 e Mutação 1	105
A.15 Gráfico <i>boxplot</i> - Recombinação 4 e Mutação 1	106
A.16 Gráfico <i>boxplot</i> - Recombinação 2 e Mutação 2	107
A.17 Gráfico <i>boxplot</i> - Recombinação 3 e Mutação 2	108
A.18 Gráfico <i>boxplot</i> - Recombinação 4 e Mutação 2	109

Lista de Tabelas

1.1	Irregularidades Cometidas	3
2.1	Relações de Dominância	8
4.1	Representação da Solução	43
4.2	Exemplo de um Conjunto de Requisições, Ordenadas por Horário de Coleta; em Destaque o Horário de Chegada das 10 Requisições Seguintes.	53
4.3	Horários de Coleta e Entrega Definidos pela 1 ^o Iteração do Algoritmo; Horários Hachurados são Fixos (Inalteráveis) na 2 ^o Iteração.	54
5.1	Conjunto de Abordagens Avaliadas	61
5.2	Resultados dos Testes Estatísticos - Conjuntos “ <i>Gravit100</i> ” e “ <i>Rnd100</i> ”, com Nível de Confiança de 99%	62
5.3	Teste de <i>Scott-Knott</i> , com Nível de Confiança de 99% - Conjuntos “ <i>Gravit100</i> ” e “ <i>Rnd100</i> ”	63
5.4	Resultados dos Testes Estatísticos - Conjunto “ <i>Gravit100</i> ”, com Nível de Confiança de 99%	65
5.5	Teste de <i>Scott-Knott</i> , com Nível de Confiança de 99% - Conjunto “ <i>Gravit100</i> ”	65
5.6	Resultados dos Testes Estatísticos - Conjunto “ <i>Rnd100</i> ”, com Nível de Confiança de 99%	68
5.7	Teste de <i>Scott-Knott</i> , com Nível de Confiança de 99% - Conjunto “ <i>Rnd100</i> ”	69
5.8	Tabela Comparativa Composta por Todas as Análises	71
5.9	Resultados Obtidos pela Abordagem EcA para Cada Instância do Conjunto “ <i>Gravit100</i> ”	77
5.10	Resultados Obtidos pela Abordagem EcA para Cada Instância do Conjunto “ <i>Rnd100</i> ”	78
5.11	Comparação do Número de Veículos Obtidos em cada Abordagem - Conjunto “ <i>Gravit100</i> ”	79

5.12	Comparação da Duração Total das Rotas Obtidas em cada Abordagem - Conjunto “Gravit100”	80
5.13	Comparação do Número de Veículos Obtidos em cada Abordagem - Conjunto “Rnd100”	81
5.14	Comparação da Duração Total das Rotas Obtidas em cada Abordagem - Conjunto “Rnd100”	82
A.1	Configurações de Parâmetros Analisadas	86
A.2	Experimentos Estatísticos - Algoritmo NSGA-II	87
A.3	Teste de <i>Scott-Knott</i> - NSGA2 - Recombinação 2 e Mutação 1	88
A.4	Teste de <i>Scott-Knott</i> - NSGA2 - Recombinação 3 e Mutação 1	89
A.5	Teste de <i>Scott-Knott</i> - NSGA2 - Recombinação 4 e Mutação 1	90
A.6	Teste de <i>Scott-Knott</i> - NSGA2 - Recombinação 2 e Mutação 2	92
A.7	Teste de <i>Scott-Knott</i> - NSGA2 - Recombinação 3 e Mutação 2	93
A.8	Teste de <i>Scott-Knott</i> - NSGA2 - Recombinação 4 e Mutação 2	94
A.9	Experimentos Estatísticos - Algoritmo IBEA	96
A.10	Teste de <i>Scott-Knott</i> - IBEA - Recombinação 2 e Mutação 1	97
A.11	Teste de <i>Scott-Knott</i> - IBEA - Recombinação 3 e Mutação 1	98
A.12	Teste de <i>Scott-Knott</i> - IBEA - Recombinação 4 e Mutação 1	99
A.13	Teste de <i>Scott-Knott</i> - IBEA - Recombinação 2 e Mutação 2	100
A.14	Teste de <i>Scott-Knott</i> - IBEA - Recombinação 3 e Mutação 2	101
A.15	Teste de <i>Scott-Knott</i> - IBEA - Recombinação 4 e Mutação 2	102
A.16	Experimentos Estatísticos - Algoritmo SPEA2	103
A.17	Teste de <i>Scott-Knott</i> - SPEA2 - Recombinação 2 e Mutação 1	104
A.18	Teste de <i>Scott-Knott</i> - SPEA2 - Recombinação 3 e Mutação 1	105
A.19	Teste de <i>Scott-Knott</i> - SPEA2 - Recombinação 4 e Mutação 1	106
A.20	Teste de <i>Scott-Knott</i> - SPEA2 - Recombinação 2 e Mutação 2	107
A.21	Teste de <i>Scott-Knott</i> - SPEA2 - Recombinação 3 e Mutação 2	108
A.22	Teste de <i>Scott-Knott</i> - SPEA2 - Recombinação 4 e Mutação 2	109

Resumo

VIANA, Renan José dos Santos, M.Sc., Universidade Federal de Viçosa, junho de 2016. **Abordagens Heurísticas para Otimização de um Serviço de Transporte Reativo a Demanda** Orientador: André Gustavo dos Santos.

Transporte reativo a demanda, na língua inglesa Demand Responsive Transport (DRT) é uma forma de prover transporte, seja para passageiros ou mercadorias, na qual o serviço é ativado sob demanda. Ao contrário dos serviços tradicionais de transporte público, os quais operam por meio de rotas, horários e pontos de atendimento fixos, os serviços DRT operam de formas flexíveis ou semi-flexíveis. Para utilização do serviço, passageiros devem enviar requisições, nas quais informam locais e horários desejados de embarque e desembarque. A partir das requisições, ocorre o processo de roteamento dos veículos e agendamento dos atendimentos. Usuários provenientes de requisições diferentes, mas com características em comum, seja área e/ou momento de atuação do serviço podem ser atendidos simultaneamente pelo mesmo veículo. Devido a esta forma de prover transporte, para alguns pesquisadores do tema, serviços DRT são considerados uma forma intermediária de transporte, situada entre os serviços de transporte público (caráter geral e compartilhado) e os táxis (personalizado e individual) e contribuem direta e indiretamente na redução de alguns dos principais problemas comuns em centros urbanos, tais como: excesso de veículos nas vias trafegando com baixa ocupação, poluição, congestionamentos, exclusão social relacionada ao acesso a meios de transporte público e etc. Neste trabalho, foram propostos modelos de programação linear mista, abordagens multiobjetivo e abordagens heurísticas para otimização de um serviço DRT introduzido na literatura, o qual foi explorado para os casos estático e dinâmico. As abordagens apresentadas foram avaliadas por meio de experimentos computacionais e testes estatísticos sobre conjuntos de instâncias com diferentes características, que indicaram as melhores abordagens para cada situação.

Abstract

VIANA, Renan José dos Santos, M.Sc., Universidade Federal de Viçosa, June of 2016. **Heuristic Approaches to Optimizing a Demand Responsive Transport.** Adviser: André Gustavo dos Santos.

Demand responsive transport is a way to provide transportation for passengers or goods, in which the service is activated on demand. Unlike traditional public transport services, which operate through fixed routes, schedules and service points, DRT services operate in flexible or semi-flexible way. In order to use the service, passengers must submit requests, in which they inform the desired local and times of departure and arrival. The routing of vehicles and the scheduling of calls are performed based on those requests. Users from different requests, but with common features like area and/or moment of the service can be served simultaneously by the same vehicle. Due to this way of providing transport, some researchers consider the DRT services an intermediate form of transport, situated between public transport services (general purpose and shared) and taxis (custom and individual) and contribute directly and indirectly in reducing some of the major common problems in urban centers, such as: too many vehicles traveling on the roads with low occupancy, pollution, congestion, social exclusion related to access to public transportation, etc. In this work, we proposed mixed linear programming models, multi-objective approaches and heuristics approaches for optimization of a DRT service from the literature, which was exploited for the static and dynamic case. The approaches presented were evaluated through computational experiments and statistical tests using sets of instances with different characteristics, showing the best approaches for each situation.

Capítulo 1

Introdução

1.1 Considerações Iniciais

Introduzido por Dantzig & Ramser [1959], o Problema de Roteamento de Veículos (PRV), na língua inglesa *Vehicle Routing Problem* (VRP) é um dos problemas mais importantes e estudados de otimização combinatória. Intensamente explorado pela comunidade científica desde sua criação, o PRV em sua forma mais simples consiste em determinar um conjunto ótimo de rotas, as quais são realizadas por uma frota de veículos com a finalidade de servir um dado conjunto de clientes. Precusores em problemas de roteirização, os autores propuseram o primeiro modelo matemático assim como a primeira abordagem para resolução do problema, que foi aplicado a uma situação real de distribuição de gasolina para estações de serviço. Posteriormente, Clarke & Wright [1964] propuseram uma outra abordagem de melhor desempenho para mesma aplicação. O clássico PRV generaliza o Problema do Caixeiro Viajante (TSP, do inglês *Traveling Salesman Problem*), portanto também é classificado como um problema NP-difícil.

Após quase seis décadas desde sua primeira aparição, uma busca com o título *Vehicle Routing Problem* na ferramenta de pesquisa de trabalhos científicos Google Acadêmico resulta em mais de 420 mil trabalhos que abordam, além da versão clássica do problema, diversas outras variantes, tais como: imposição de janelas de tempo [Bräysy & Gendreau, 2005a,b], multi-objetivos a serem otimizados [Jozefowicz et al., 2008], aplicação em contextos dinâmicos [Pillac et al., 2013; Psaraftis et al., 2016], entre outras. Além da crescente aparição de novas variações do PRV, vários métodos têm sido propostos para resolução das mesmas, como por exemplo abordagens baseadas em métodos exatos e aproximados. De acordo com Toth & Vigo [2014], o alto interesse em diferentes variantes do PRV por parte da comunidade científica não se deve apenas

por serem problemas de otimização combinatória de difícil resolução, mas também pela relevância prática apresentada.

Seja qual for a aplicação e objetivo a ser otimizado, as pesquisas que abordam o PRV se assemelham pela busca por um planejamento otimizado e de qualidade. Dentro deste contexto, se encontram os serviços de transporte de passageiros operacionalizados sob demanda, também conhecidos como Serviços de Transporte Reativos a Demanda (DRT, do inglês *Demand Responsive Transport*). Inicialmente viabilizados para o transporte de pessoas com necessidades especiais, os serviços baseados no DRT transportam passageiros de acordo com suas necessidades, sendo flexíveis em relação à locais e horários. Para serviços de transporte de passageiros, um planejamento otimizado significa definir rotas (roteamento) e horários de atendimento (agendamento) que demandam baixo custo de operação e oferecem um serviço de alta qualidade.

1.2 Problema e sua Importância

Serviços de transporte tradicionais, como por exemplo o transporte público provido por meio de ônibus e micro-ônibus, funcionam de forma eficiente e satisfatória em períodos do dia em que a demanda é alta e constante, pois desta forma ocorre um alto compartilhamento do serviço. Em contrapartida, em momentos que a demanda pelo serviço é baixa ou escassa, ocorre drasticamente a redução deste potencial de compartilhamento. Tal situação, decorre principalmente da estrutura pouco flexível em relação a locais e horários adotada [Errico et al., 2013].

Como alternativa, serviços DRT proveem transporte flexível e compartilhado à passageiros e mercadorias. Usuários formulam requisições definindo locais e horários de embarque e desembarque específicos, e a partir destes dados as rotas e os horários de atendimentos são definidos [Cordeau et al., 2007]. Esta forma de operação sob demanda altera um antigo paradigma, no qual rotas e horários de atendimentos são definidos de acordo com os interesses dos provedores.

Viabilizados para atenderem setores da população que lidam com a ineficiência dos serviços de transporte tradicionais e também com os altos custos dos serviços altamente personalizados oferecidos pelos táxis, os DRTs preenchem esta lacuna provendo serviços de transporte flexíveis ou semi-flexíveis que combinam os benefícios políticos, econômicos e sociais dos ônibus com os serviços de alta qualidade dos táxis [Executive et al., 2004]. Ainda, contribuem indiretamente na solução de outros problemas graves que são visíveis principalmente nos grandes centros, tais como: superlotação das vias, engarrafamentos, ruídos, poluição, entre outros.

Segundo dados obtidos na página do Departamento Nacional de Trânsito (DENATRAN), a frota de veículos no Brasil em 2000 era composta por quase 30 milhões de veículos. De acordo com o último senso realizado em março de 2016, este número cresceu para quase 92 milhões, mais de 3 vezes o valor anterior. Estas informações apontam outro ponto a favor da popularização de novas formas de transporte coletivo.

Em uma pesquisa realizada pelo Instituto Brasileiro de defesa do Consumidor (IDEC) e publicada em [IDEC, 2013], alguns dos principais problemas enfrentados pelos usuários dos serviços de transporte público foram provados por meio de números. A pesquisa analisou os serviços de transporte público providos por meio de ônibus em 10 trajetos das cidades: Belo Horizonte e São Paulo. Constatou-se que, em São Paulo, um passageiro pode chegar a caminhar até 1,5 Km para chegar no ponto de ônibus mais próximo e a distância percorrida média é de 275 metros. Já em Belo Horizonte a situação é ainda mais crítica, um passageiro pode chegar a caminhar até 2,4 Km para chegar em um local onde consiga obter o serviço e a distância média chega a 744 metros. Segundo a pesquisa, em ambas cidades há casos de excesso de tempo de espera pelo serviço, superlotação (que ocorre em 14% das viagens em São Paulo) e excesso de velocidade por parte dos motoristas (além de manobras perigosas e paradas longe da calçada). Outra situação importante apontada pela pesquisa é a falta de acessibilidade nos ônibus para idosos e deficientes (em ambas cidades, 20% dos veículos não eram equipados com rampas de acesso). Esta reportagem retrata as deficiências do transporte público apenas em duas cidades (Belo Horizonte e São Paulo). Entretanto, esta é a realidade na grande maioria dos grandes centros urbanos. A seguir, é apresentada a Tabela 1.1, a qual foi construída por meio de dados fornecidos pela pesquisa e que aponta o número de irregularidades encontradas nos serviços de transporte público nas duas cidades analisadas.

Tabela 1.1. Irregularidades Cometidas

Critérios	São Paulo	Belo Horizonte
Qualidade da Viagem	10	11
Estrutura do Ponto	13	8
Estrutura do Meio de Transporte	51	4
Atendimento ao Usuário	11	7
Total	85	30

1.3 Hipótese

Segundo Ambrosino et al. [2004], a evolução de novas formas de serviços de transporte público flexíveis, os quais estão situados em algum ponto entre os serviços de transporte público operacionalizados por meio de ônibus com rotas fixas, destinados para uso geral e os serviços de táxis altamente flexíveis e personalizados, tem ocorrido principalmente em decorrência dos desenvolvimentos das tecnologias de comunicação e sistemas de otimização.

A utilização de abordagens heurísticas no processo de planejamento das rotas realizadas pelos serviços de transporte reativo a demanda permite minimizar custos e maximizar o número de atendimento de solicitações, assim como a qualidade do serviço. Isto é importante tanto para as empresas que fornecem o serviço quanto para os clientes que o utilizam.

1.4 Objetivos

O objetivo geral desta dissertação é propor e desenvolver abordagens heurísticas para otimizar o processo de construção de rotas efetuado em serviços de Transporte Reativo a Demanda, com o propósito de encontrar boas soluções tanto no caso estático (todas as solicitações são conhecidas antes do processo de construção das rotas) quanto no dinâmico (solicitações de transporte são gradualmente recebidas ao longo do tempo). As heurísticas propostas serão avaliadas por meio de testes computacionais com instâncias de teste presentes na literatura e os resultados obtidos e os tempos de execuções gastos serão comparados com os já encontrados, a fim de validar a eficiência das mesmas.

1.4.1 Objetivos Específicos

Os objetivos específicos desse trabalho são:

- Implementar heurísticas propostas na literatura para a resolução do problema no caso estático;
- Propor e desenvolver novas abordagens heurísticas para tratar o problema no caso estático;
- Propor e desenvolver abordagens heurísticas para tratar o problema no caso dinâmico;
- Propor e implementar estratégias que possibilitem melhorar a qualidade das soluções;

- Identificar os parâmetros que permitem as abordagens encontrarem os melhores resultados;
- Realizar experimentos computacionais para avaliar o desempenho das abordagens;
- Realizar comparações entre os resultados encontrados e os resultados obtidos pelas abordagens da literatura;
- Apresentar os resultados à comunidade acadêmica.

Capítulo 2

Revisão Bibliográfica

2.1 Otimização Multiobjetivo

Problemas de otimização multiobjetivo (MOOP, do inglês Multi-Objective Optimization problems), também conhecidos por problemas de otimização vetorial (POV) são problemas que lidam com mais de uma função objetivo (maximização ou minimização) usualmente conflitantes entre si. Funções objetivo são conflitantes quando a melhora em uma somente é possível por meio da piora de pelo menos alguma das demais. Na prática, na maioria dos problemas de tomada de decisão é clara a existência de múltiplos objetivos [Deb, 2001].

O conceito de ótimo utilizado dentro do contexto da otimização multiobjetivo foi originalmente proposto e abordado em trabalhos na área da economia (Edgeworth [1881] e Pareto [1896]). Devido a existência de n objetivos a serem otimizados, a busca pela solução ótima, comum em problemas mono-objetivo é substituída pela busca por um conjunto de soluções que possuem um bom equilíbrio entre os critérios desejados, conhecidas como soluções Pareto ótimo [Talbi, 2009].

Segundo Talbi [2009] e Coello et al. [2007], um MOOP pode ser definido da seguinte maneira:

$$MOOP = \begin{cases} \min F(x) = ((f_1(x), f_2(x), \dots, f_m(x))) \\ x \in S. \end{cases}$$

em que m ($m > 2$) é o número de objetivos a serem otimizados ($F = (f_1, f_2, \dots, f_m)$), $x = (x_1, x_2, \dots, x_n)$ representa o vetor de variáveis de decisão e X o conjunto de soluções viáveis (região factível), as quais devem respeitar um conjunto de restrições impostas pelo problema em questão, definidas por equações, inequações e limites. Para cada solução $x \in X$ é associado um vetor de objetivo (ponto) $z \in Z$, denotado por $F : X \rightarrow$

Z com $z = F(x) = (f_1(x), f_2(x), \dots, f_m(x))$. O espaço de busca X representa o espaço de decisão ou espaço de parâmetros, enquanto o espaço ao qual o vetor de objetivo z pertence é chamado de espaço de objetivo ($Z = F(X)$).

Em MOOP não é comum existir uma solução x^* , que possui o valor ótimo para todos os n objetivos. Diante deste fato, outros conceitos foram criados e estabelecidos para considerar a otimalidade, sendo a relação de dominância o principal deles.

2.1.1 Relação de Dominância e Otimalidade Pareto

A maioria dos algoritmos de otimização multiobjetivo utiliza a relação de dominância para guiar a pesquisa no espaço de busca. Duas soluções são comparadas com base se uma domina a outra ou não. Segundo Deb [2001], o conceito de dominância pode ser definido da seguinte forma:

Definição 1 (Dominância de Pareto) Dado um MOOP com m funções objetivo a serem otimizadas, sendo minimizadas ou maximizadas, uma solução x^1 domina uma outra solução x^2 (ou matematicamente $x^1 \prec x^2$) se:

1. A solução x^1 não é pior que a solução x^2 em nenhum dos m objetivos.
2. A solução x^1 é estritamente melhor que a solução x^2 em pelo menos um dos m objetivos.

Se qualquer uma destas duas condições acima for violada, a solução x^1 não domina a solução x^2 . Além disso, se x^1 não domina x^2 , não implica que x^2 domina x^1 .

Quando o problema possui objetivos conflitantes entre si, raramente existe uma solução que domina todas as demais. A seguir, algumas outras definições presentes dentro do contexto da otimização de problemas multi-objetivo.

Definição 2 (Conjunto de Soluções Não-Dominadas) Dado um conjunto de soluções P , o conjunto de soluções não dominadas P' é formado por todas aquelas que não são dominadas por nenhum outro membro de P .

Definição 3 (Conjunto Pareto-Ótimo) Quando o conjunto de soluções P é todo o espaço de busca (região factível), ou seja $P = X$, o conjunto de soluções não dominadas P' é chamado de Conjunto Pareto-Ótimo.

De acordo com Eiben & Smith [2003], todas as soluções que pertencem ao conjunto de soluções não-dominadas são incapazes de melhorar a qualidade de algum dos

objetivos otimizados sem piorar a qualidade de algum outro dos demais, geralmente localizadas na fronteira das regiões viáveis do espaço de busca.

Definição 4 (*Fronteira Pareto-Ótima*) Conjunto de vetores de objetivo, o qual é a imagem do conjunto Conjunto Pareto-Ótimo no espaço de objetivos.

Existem outras relações que podem ser utilizadas para comparar dois vetores de objetivo $z^1, z^2 \in Z$. Algumas das principais são apresentadas na Tabela 2.1.

Tabela 2.1. Relações de Dominância

Relação	Notação	Interpretação
domina estritamente	$z^1 \prec\prec z^2$	z^1 é melhor que z^2 em todos os objetivos
domina	$z^1 \prec z^2$	z^1 não é pior que z^2 em todos os objetivos e melhor em pelo menos um objetivo
domina fracamente	$z^1 \preceq z^2$	z^1 não é pior que z^2 em todos os objetivos
não comparáveis	$z^1 \parallel z^2$	z^1 não domina fracamente z^2 e a recíproca também é verdadeira

Uma vez com o conjunto de soluções Pareto-ótimo nas mãos, o tomador de decisão se depara com a tarefa complexa de selecionar uma solução em detrimento das demais, principalmente em situações nas quais não se sabe qual objetivo priorizar. Nesta situação todas as soluções são igualmente importantes. Diante de tal fato, na otimização multi-objetivo há duas metas [Deb, 2001]:

- Encontrar um conjunto de soluções o mais próximo possível do conjunto de soluções Pareto-Ótimo.
- Encontrar um conjunto de soluções com alta diversidade.

2.2 Algoritmos Evolutivos Multiobjetivo

Algoritmos Evolutivos (EA, do inglês Evolutionary Algorithms) são métodos de otimização que baseiam-se nos princípios da evolução natural para compor procedimentos de busca e otimização [Deb, 2001]. Iterativamente, simulam a evolução de um conjunto de soluções utilizando o conceito de competição, que garante a sobrevivência dos indivíduos mais aptos. Baseados na biologia, os EA são compostos por mecanismos de seleção, mutação e combinação. O sucesso destes métodos de otimização é motivado pela habilidade de resolver problemas complexos de otimização de vários domínios e pelo fato de serem aplicados com sucesso em diversos problemas reais [Talbi, 2009].

Segundo Yu & Gen [2010], EAs possuem três principais características:

- **Baseado em população:** EAs mantêm um conjunto de soluções intitulado população. A população é o princípio básico do processo evolutivo.

- **Orientado por aptidão:** Para cada solução (indivíduo) da população é atribuído um valor de aptidão. Indivíduos mais aptos têm maiores chances de sobrevivência para as próximas gerações.
- **Dirigido por variação:** os indivíduos são submetidos à diversas operações de variação com o intuito de se obter variações genéticas, que são importantes para exploração do espaço de busca.

Quando projetados para tratarem problemas multiobjetivo, EAs são chamados de algoritmos evolutivos multiobjetivo (MOEA, do inglês *Multiobjective Evolutionary Algorithms*) e o tipo de otimização é intitulada otimização multiobjetivo evolutiva (EMO, do inglês *Evolutionary Multiobjective Optimization*). Segundo Abraham & Jain [2005], EAs se destacam na resolução de problemas com múltiplas funções objetivo pelo fato de manterem um conjunto de soluções (população) e durante o processo de reprodução combiná-las para a obtenção de novas soluções. Sendo assim, capazes de obterem vários membros do conjunto Pareto-Ótimo em apenas uma execução, ao contrário de métodos convencionais que necessitam de várias execuções independentes para obterem o conjunto de soluções Pareto-Ótimo.

2.2.1 Non-Dominated Sorting Genetic Algorithm II

O algoritmo *Non-Dominated Sorting Genetic Algorithm II* (NSGA-II) proposto por Deb et al. [2002], pertence a classe de MOEA elitistas. Sucessor do algoritmo NSGA proposto por Srinivas & Deb [1994], o NSGA-II é um dos algoritmos mais utilizados para solucionar problemas de otimização que possuem mais de um objetivo. Este fato é justificado principalmente por encontrar soluções com boa qualidade para vários problemas presentes na literatura. O NSGA-II foi proposto com base em vários aspectos que antes eram criticados em seu antecessor. Basicamente, foram realizadas três aprimoramentos: houve o aperfeiçoamento do método de ordenação das soluções não-dominadas, obtendo uma versão com menor complexidade de tempo, a inclusão do elitismo e a eliminação do parâmetro de compartilhamento σ_{share} .

O processo de classificação chamado de *Fast Nondominated Sorting* atribui a cada solução da população uma das diversas classes de não-dominância chamadas fronteiras. O conjunto de soluções é distribuído nas fronteiras F_1, F_2, \dots, F_k de acordo com o grau de dominância de cada uma, de modo que em uma mesma fronteira nenhuma solução domine alguma das demais. Após o processo de classificação, a primeira fronteira contém os melhores indivíduos da população (não dominadas por nenhuma outra solução) da geração corrente, enquanto que na última fronteira estão as piores soluções da

população. A Figura 2.1 apresenta uma ilustração de um problema com duas funções objetivo, sendo que f_1 é maximizada e f_2 minimizada. Os cinco pontos são distribuídos em três fronteiras.

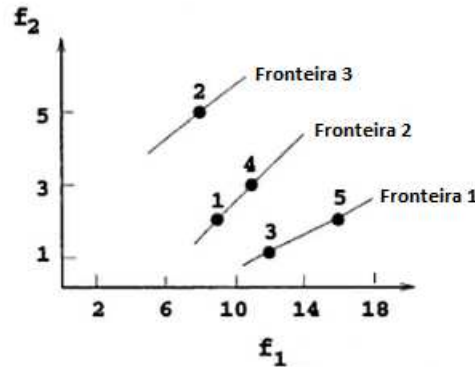


Figura 2.1. Ordenação por grau de dominância [Deb, 2001]

Para garantir a diversidade, o NSGA-II utiliza a abordagem chamada distância de agrupamento (do inglês *Crowding Distance*) que estima a densidade de soluções em volta de cada solução da população. Este método busca uma melhor distribuição da população nas fronteiras, onde soluções distantes das demais são melhores avaliadas. Na Figura 2.2, a distância de agrupamento da solução i é calculada com o auxílio das soluções vizinhas mais próximas ($i - 1$ e $i + 1$).

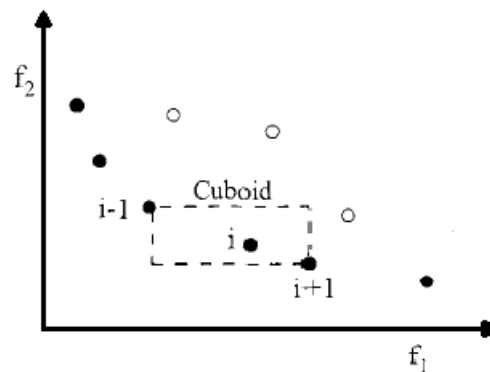


Figura 2.2. Distância de Agrupamento [Deb, 2001]

As abordagens apresentadas permitem ao NSGA-II avaliar a qualidade das soluções em termos da convergência em relação ao conjunto Pareto-Ótimo e em termos de diversidade, respectivamente. Dados estes valores, uma solução x é selecionada quando comparada a uma solução y , se a mesma possui uma melhor classificação, ou seja, está em uma fronteira de menor índice. Se ambas possuírem a mesma classificação, a solu-

ção x é selecionada se apresentar um valor superior de distância de agrupamento em relação a y .

O Algoritmo 1 apresenta todo o procedimento do NSGA-II. Os módulos nos quais as soluções são classificadas e avaliadas segundo o fator diversidade são apresentados a seguir.

Algoritmo 1: NSGA-II

Entrada: N , tempo de máximo de execução

Saída: P_{final}

1 **início**

2 $n \leftarrow 0$

3 $P_n \leftarrow$ Gerar População Inicial();

4 Aplica Ordenação por não-dominância(P_n);

5 Aplica Distância de multidão(P_n);

6 **enquanto** *tempo de máximo de execução não for atingido* **faça**

7 $Q_n \leftarrow$ Aplica operadores de Seleção, Crossover e Mutação (P_n);

8 $R_n \leftarrow P_n \cup Q_n$

9 Aplica Ordenação por não-dominância(R_n);

10 $P_{n+1} \leftarrow \emptyset$

11 $k \leftarrow 1$

12 **enquanto** $|P_{n+1}| + F_k \leq N$ **faça**

13 $P_{n+1} \leftarrow P_{n+1} \cup F_k$

14 $k \leftarrow k + 1$

15 Aplica Distância de multidão(F_k);

16 Copiar as $N - |P_{n+1}|$ soluções de F_k para P_{n+1}

17 $n \leftarrow n + 1$

18 $P_{final} \leftarrow P_n$

19 **retorna** P_{final}

Inicialmente, é criada de forma aleatória uma população inicial P_0 de tamanho N (linhas 3 a 5) que é um dos parâmetros de entrada. A cada geração do algoritmo NSGA-II, todas as soluções de P_n são atribuídas a diferentes classes de não-dominância. Uma população filha Q_n de tamanho N é criada por meio dos operadores de seleção (torneio binário), recombinação e mutação aplicados à P_n (linha 7). Em seguida, as duas populações são combinadas, resultando no conjunto de soluções R_n de tamanho $2N$, o qual é classificado utilizando a ordenação por não-dominância (linhas 8 e 9). A construção da próxima população P_{n+1} de tamanho N inicia com as soluções da melhor fronteira de R_n , passa para a segunda melhor fronteira, e continua o processo seguindo a ordem de classificação das fronteiras (linhas 12 a 14). Quando não for possível armazenar completamente a fronteira corrente em conjunto com as fronteiras anteriores nas N vagas da próxima população, a fronteira será ordenada em ordem decrescente

de distância de multidão (linha 15). Posteriormente, as $N - |P_{n+1}|$ primeiras soluções serão utilizadas para completar P_{n+1} (linha 16). Em outras palavras, quando houver mais soluções na fronteira corrente que vagas livres na próxima população, as soluções com maior valor de diversidade serão selecionadas.

As gerações prosseguem até que um determinado critério de parada seja satisfeito (linha 6). Ao final, o algoritmo retorna as soluções não dominadas presentes na população obtida (linha 19). A Figura 2.3 ilustra o processo evolutivo em uma geração do NSGA-II.

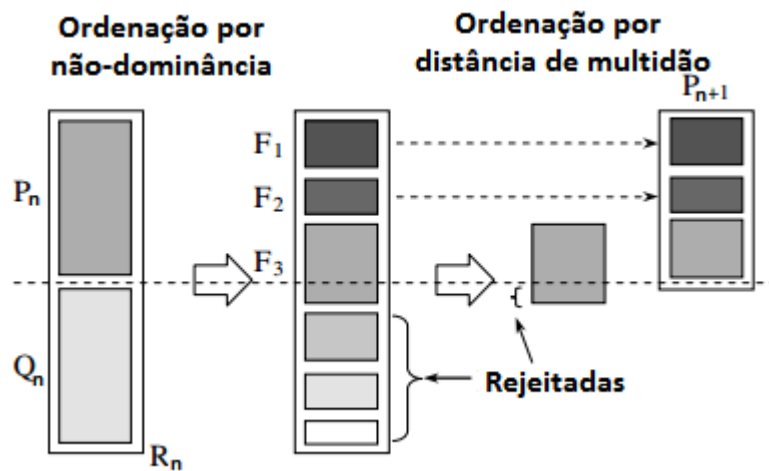


Figura 2.3. Procedimento do NSGA-II [Deb, 2001]

2.2.1.1 Fast Non-Dominated Sorting

O método de ordenação por não-dominância classifica todas as soluções da população P de acordo com o respectivo grau de dominância. Possui complexidade de tempo $O(MN^2)$, sendo M o número de objetivos e N o tamanho da população. Dada uma população como parâmetro de entrada, este método retorna um conjunto de fronteiras ($F = F_1, F_2, \dots, F_k$). O Algoritmo 2 apresenta como é realizado o procedimento de classificação das soluções. Inicialmente, todas as soluções são comparadas por pares, de modo que o conjunto S_i (que armazena todas as soluções dominadas pelo elemento i) e o valor n_i (quantidade de soluções que dominam o elemento i) são definidos para cada indivíduo da população (linha 2 a 12). Posteriormente, é realizada a definição do ranqueamento das soluções, assim como a construção das fronteiras (linha 13 a 23).

Algoritmo 2: ALGORITMO DE ORDENAÇÃO POR NÃO-DOMINÂNCIA

Entrada: P
Saída: F

```

1 início
2   para cada solução  $i \in P$  faça
3      $S_i \leftarrow \emptyset$ 
4      $n_i \leftarrow 0$ 
5     para cada solução  $j \in P$  faça
6       se  $i \prec j$  então
7          $S_i \leftarrow S_i \cup \{j\}$ 
8       se  $j \prec i$  então
9          $n_i \leftarrow n_i + 1$ 
10    se  $n_i = 0$  então
11       $i_{rank} \leftarrow 1$ 
12       $F_1 \leftarrow F_1 \cup \{i\}$ 
13     $k \leftarrow 1$ 
14    enquanto  $F_k \neq \emptyset$  faça
15       $Q \leftarrow \emptyset$ 
16      para cada solução  $i \in F_k$  faça
17        para cada solução  $j \in S_i$  faça
18           $n_j \leftarrow n_j - 1$ 
19          se  $n_j = 0$  então
20             $j_{rank} \leftarrow k + 1$ 
21             $Q \leftarrow Q \cup \{j\}$ 
22       $k \leftarrow k + 1$ 
23       $F_k = Q$ 
24    retorna  $F$ 

```

2.2.1.2 Crowding Distance

O método que calcula a distância de multidão estima a densidade de vetores de objetivo em volta de um dado ponto i . Para isso, utiliza como referência dois pontos adjacentes para o cálculo da distância. Basicamente, o valor $P[i]_{distancia}$ é o tamanho do maior cuboide (espaço sem nenhum outro ponto da população) que envolve o ponto i . Soluções que possuem suas respectivas imagens no espaço de objetivos afastadas dos demais pontos recebem maiores valores de distância de multidão. O Algoritmo 3 explicita como é realizado este procedimento.

2.2.2 Indicator-based Multiobjective Evolutionary Algorithm

O algoritmo *Indicator-based Multiobjective Evolutionary Algorithm* (IBEA) proposto por Zitzler & Künzli [2004] é um dos primeiros MOEAs da classe de algoritmos ba-

Algoritmo 3: ALGORITMO DE DISTÂNCIA DE MULTIDÃO

Entrada: P

```

1 início
2    $l \leftarrow |P|$ 
3   para cada solução  $i \in P$  faça
4      $P[i]_{distancia} \leftarrow 0$ 
5   para cada objetivo  $m$  faça
6      $P \leftarrow \text{Ordena}(P, m)$ 
7      $P[1]_{distancia} \leftarrow \infty$ 
8      $P[l]_{distancia} \leftarrow \infty$ 
9     para  $i = 2$  até  $(l - 1)$  faça
10     $P[i]_{distancia} \leftarrow P[i]_{distancia} + (P[i + 1]_m - P[i - 1]_m) / (f_m^{max} - f_m^{min})$ 

```

seados em indicadores, nos quais a busca pelo espaço de soluções é guiada por um indicador de qualidade de performance. O processo de atribuição da aptidão (do inglês *fitness*) consiste em comparações par-a-par realizadas por meio de um indicador de qualidade, o qual determina um valor que representa a diferença de qualidade entre duas soluções. Há diversos indicadores de qualidade na literatura que podem ser incorporados ao IBEA, por exemplo o *binary ϵ -indicator* [Zitzler et al., 2003], *binary additive ϵ -indicator* [Zitzler et al., 2003], *hypervolume indicator* entre outros. A possibilidade de escolha do indicador evidencia a flexibilidade que o IBEA oferece dando ao tomador de decisões a capacidade de incorporar informações de preferência.

Em geral, um indicador de qualidade é uma função que mapeia k conjuntos aproximados à um número real, sendo os mais comuns, os indicadores de qualidade unários ($k = 1$) [Zitzler & Künzli, 2004]. Dentre os três indicadores citados como exemplo, o *binary additive ϵ -indicator* ($I_{\epsilon+}$) foi escolhido para incorporar o IBEA proposto, sendo assim denotado $IBEA_{\epsilon+}$.

Os indicadores de qualidade binários permitem comparar a qualidade de cada um dos dois conjuntos aproximados analisados em relação ao outro. O $I_{\epsilon+}(A, B)$ determina o quanto o conjunto aproximado A é pior que o B em relação a todos os objetivos, ou seja, quantifica o fator ϵ mínimo necessário para que qualquer vetor de objetivo do conjunto aproximado B seja ϵ -dominado por pelo menos um vetor de objetivo do conjunto aproximado A . Mais precisamente, este indicador retorna o valor mínimo que precisa ser adicionado em B para que o mesmo seja dominado fracamente por A . Formalmente esta função pode ser definida da seguinte maneira na Equação (2.1):

$$I_{\epsilon+}(A, B) = \min_{\epsilon} \{ \forall x^2 \in B \exists x^1 \in A : f_i(x^1) - \epsilon \leq f_i(x^2) \text{ para } i \in \{1, \dots, n\} \} \quad (2.1)$$

Segundo Zitzler & Künzli [2004], a razão de se utilizar indicadores de qualidade binários é devido aos mesmos representarem uma natural extensão da relação de dominância Pareto. Portanto podem ser utilizados para o cálculo da aptidão das soluções de forma similar aos esquemas comuns baseados em dominância Pareto. O Algoritmo 4 apresenta todo o procedimento do IBEA.

Algoritmo 4: IBEA

Entrada: N , k , tempo máximo de execução

Saída: P_{final}

```

1 início
2    $n \leftarrow 0$ 
3    $P_n \leftarrow$  Gerar População Inicial();
4   repita
5     Calcula o valor de aptidão ( $P_n$ ,  $k$ );
6     enquanto  $|P_n| > N$  faça
7       Selecciona a solução  $i \in P_n$  com menor valor de aptidão;
8       Remova  $i$  de  $P_n$ ;
9       Atualiza o valor de aptidão de todas soluções ( $P_n$ ,  $k$ );
10    se tempo máximo de execução não foi atingido então
11       $Q_n \leftarrow$  Aplica operadores de Seleção, Crossover e Mutação ( $P_n$ );
12       $P_{n+1} \leftarrow P_n \cup Q_n$ 
13       $n \leftarrow n + 1$ 
14    até tempo máximo de execução ser atingido;
15   $P_{final} \leftarrow P_n$ 
16 retorna  $P_{final}$ 

```

Inicialmente, é criada de forma aleatória uma população inicial P_0 de tamanho N (linha 3) que é um dos parâmetros de entrada. A cada geração do algoritmo IBEA, para cada solução de P_n é associado um valor de aptidão, que representa a qualidade da mesma e será levado em consideração no processo de redução do tamanho da população ao seu tamanho definido (N). Em seguida, é realizado o procedimento de redução da população, o qual só será executado a partir da segunda geração. Neste passo (linhas 6 a 10), inicialmente a população se encontra com tamanho $2N$ e a cada iteração a solução com menor valor de aptidão é removida da mesma, seguida da atualização do valor de aptidão das demais soluções. Este procedimento é encerrado, quando o tamanho da população retorna ao valor estabelecido (N). Posteriormente, um conjunto de soluções filhas Q_n é criado por meio dos operadores de seleção, crossover e mutação e em seguida armazenado em conjunto com a população corrente P_n com o intuito de gerar a população da próxima geração P_{n+1} (linhas 12 e 13). A execução do IBEA é encerrada quando o critério de parada é satisfeito.

2.2.2.1 Atribuição de Aptidão

Dada uma população P , o processo de atribuição de aptidão do IBEA busca classificar cada solução de acordo com sua utilidade segundo os objetivos tratados. Para isso, a ideia proposta pelos autores para avaliar uma solução se baseia na soma dos valores retornados pelo indicador de qualidade I em relação ao restante da população P , como é demonstrado na Equação (2.2):

$$F'(x^1) = \sum_{x^2 \in P \setminus \{x^1\}} I(\{x^2\}, \{x^1\}) \quad (2.2)$$

Definindo I como o $I_{\epsilon+}$, para um problema com n objetivos, a Equação (2.2) ficaria da seguinte forma:

$$F'(x^1) = \sum_{x^2 \in P \setminus \{x^1\}} \max_{i \in \{1, \dots, n\}} (f_i(x^2) - f_i(x^1)) \quad (2.3)$$

Para avaliar uma solução x^1 , é realizado o somatório da maior distância entre a solução corrente e cada uma das outras soluções de P segundo os objetivos a serem otimizados. Sendo assim, o valor de aptidão $F'(x^1)$ representa a perda de qualidade caso a solução x^1 seja removida de P . Dado este fato, busca-se a maximização de F' , ou seja, maximiza-se a chance de x^1 permanecer na população. Adotando o $I_{\epsilon+}$ como indicador de qualidade, a razão de $F'(x^1)$ pelo tamanho da população N resulta no fator ϵ médio necessário para cobrir a solução x^1 com as demais soluções de P .

De acordo com Zitzler & Künzli [2004], em vez de utilizar o esquema de atribuição de aptidão mencionado acima, um outro esquema semelhante, o qual amplifica a influência das soluções dominantes de P sobre as dominadas é definido para realizar tal tarefa no IBEA. A seguir, na Equação (2.4) é apresentado este novo esquema:

$$F'(x^1) = \sum_{x^2 \in P \setminus \{x^1\}} -e^{\frac{-I(\{x^2\}, \{x^1\})}{K}} \quad (2.4)$$

O parâmetro K é um fator de escala que depende do indicador de qualidade utilizado e do problema tratado. O valor e é o número de Euler ou também chamado constante de Euler.

2.2.3 Strength Pareto Evolutionary Algorithm 2

O *Strength Pareto Evolutionary Algorithm 2* (SPEA2) é um MOEA elitista assim como o NSGA-II. Proposto por Zitzler et al. [2001], o SPEA2 foi introduzido com o intuito de tentar corrigir as fragilidades encontradas em seu predecessor SPEA proposto em

Zitzler & Thiele [1999]. Tais aperfeiçoamentos são: aprimoramento do esquema de atribuição de aptidão utilizado, adição de uma técnica de estimativa de densidade da vizinhança mais próxima e utilização de um novo método de compactação do arquivo. O procedimento do SPEA2 é apresentado no Algoritmo 5.

Algoritmo 5: SPEA2

Entrada: N , N' , tempo máximo de execução
Saída: P_{final}

```

1 início
2    $n \leftarrow 0$ 
3    $P_n \leftarrow$  Gerar População Inicial();
4    $A_n \leftarrow \emptyset$ 
5   enquanto tempo máximo de execução não for atingido faça
6     Calcula o valor de aptidão ( $P_n$ );
7      $A_{n+1} \leftarrow$  Seleção Populacional ( $P_n \cup A_n$ ,  $N'$ );
8     se tempo máximo de execução não foi atingido então
9        $P_{n+1} \leftarrow$  Aplica operadores de Seleção, Crossover e Mutação
10      ( $A_{n+1}$ );
10      $n \leftarrow n + 1$ 
11  $P_{final} \leftarrow A_n$ 
12 retorna  $P_{final}$ 

```

Inicialmente, cria-se de forma aleatória uma população inicial P_0 com N indivíduos e um conjunto de soluções A_0 vazio chamado arquivo (linhas 3 e 4), o qual será utilizado para armazenar os melhores indivíduos encontrados durante as gerações. A cada iteração do laço principal (linha 5 a 10), calcula-se o valor de aptidão de todas as soluções de P_n (procedimento explicado na seção 2.2.3.1) e em seguida realiza-se o procedimento de atualização do arquivo de tamanho N' (linha 7), que será explicado na seção 2.2.3.2. Caso o critério de parada (tempo de execução) não seja satisfeito, são gerados novos indivíduos para a próxima população P_{n+1} utilizando A_{n+1} que foi atualizado no passo anterior (linha 8). Uma vez satisfeito o critério de parada, o SPEA2 retorna a última atualização do arquivo (linhas 11 e 12), a qual contém as melhores soluções encontradas durante as gerações.

2.2.3.1 Atribuição de Aptidão

O método de atribuição de aptidão do SPEA2, um dos módulos que foram aperfeiçoados em relação ao seu predecessor, realiza o cálculo para cada indivíduo da população levando em consideração tanto as soluções que o dominam quanto as soluções que são dominadas pelo mesmo.

Para cada solução i presente na população P e no arquivo A é atribuído um valor de força S_i que representa o número de soluções de $P \cup A$ que a mesma domina. Matematicamente, o cálculo da força de uma solução i é representada da seguinte forma:

$$S_i = |\{j | j \in P_n \cup A_n \wedge i \prec j\}| \quad (2.5)$$

onde $|\cdot|$ corresponde a cardinalidade do conjunto. De acordo com a Equação (2.5) soluções que não dominam nenhuma outra possuem $S_i = 0$. Com base nos valores S calculados, a primeira parte da aptidão das soluções é calculada da seguinte forma:

$$R_i = \sum_{j \in P \cup A, j \prec i} s(j) \quad (2.6)$$

Basicamente, o valor de aptidão inicial R_i calculado na Equação (2.6) representa o somatório dos valores das forças de todas soluções que dominam a solução i . Caso uma solução não seja dominada por nenhuma outra, sua aptidão será $R_i = 0$. Por outro lado, se a solução i for dominada por diversas outras soluções, seu valor de R_i será elevado.

A segunda parte do cálculo da aptidão é a medida de densidade, que foi incorporada para discriminar indivíduos com valores R_i iguais. O cálculo desta densidade basicamente utiliza o inverso da distância da solução i em relação ao k -ésimo vizinho mais próximo, sendo realizado da seguinte forma:

$$D_i = \frac{1}{(\alpha_i^k + 2)} \quad (2.7)$$

de modo que, α_i^k representa a distância no espaço de objetivos do indivíduo i em relação ao seu k -ésimo vizinho mais próximo, sendo $k = \sqrt{N + N'}$. O valor final da aptidão F_i referente a solução i , é obtido por meio da soma dos valores R_i e D_i como apresentado na Equação (2.8).

$$F_i = R_i + D_i \quad (2.8)$$

2.2.3.2 Atualização do Arquivo

O processo de atualização do arquivo mantém para próxima geração ($n + 1$) as N' melhores soluções que estão presentes na população P_n e no arquivo A_n . Para isso, todas soluções não-dominadas contidas em $P_n \cup A_n$ são transferidas para o novo arquivo

A_{n+1} . Em outras palavras, são selecionadas todas as soluções que possuem o valor de aptidão menor que 1. Este procedimento é representado matematicamente da seguinte forma:

$$A_{n+1} = \{i | i \in P_n \cup A_n \wedge F_i < 1\} \quad (2.9)$$

Após a cópia de todas soluções não-dominadas, podem ocorrer três situações, que são: o número de soluções contidas no novo arquivo ser igual ao número máximo de soluções estabelecido ($|A_{n+1}| = N'$), ou ser inferior ($|A_{n+1}| < N'$), ou ser superior ($|A_{n+1}| > N'$). Para o primeiro caso, não há necessidade de ajuste na cardinalidade de A_{n+1} e o procedimento de atualização é finalizado. Caso ocorra a segunda situação, o conjunto de soluções $P_n \cup A_n$ é ordenado em ordem crescente de aptidão e as $N' - |A_{n+1}|$ primeiras soluções com $F_i > 1$ são utilizadas para completar o novo arquivo. Por fim, se verificar-se o terceiro caso, é necessário realizar o procedimento de compactação do arquivo, o qual iterativamente retira indivíduos de A_{n+1} até a condição $|A_{n+1}| = N'$ ser satisfeita.

A cada iteração, o procedimento de compactação do arquivo remove o indivíduo $i \in A_{n+1}$ que possui a menor distância em relação a outro indivíduo do conjunto. Caso houver vários indivíduos com a mesma menor distância, a segunda menor distância é considerada, e assim por diante enquanto for necessário para viabilizar o desempate. Este processo é finalizado quando $|A_{n+1}| = N'$.

2.3 Métodos de Avaliação de Algoritmos Multiobjetivo

Na otimização de problemas mono-objetivo, para avaliar a qualidade de uma solução obtida pela abordagem proposta, basta apenas calcular a diferença relativa entre o valor encontrado e o melhor valor conhecido (possivelmente a solução ótima) para a instância de teste em questão. Por sua vez, para problemas que otimizam múltiplos objetivos não existe uma forma de medida natural, sendo assim, a avaliação dos resultados se torna uma tarefa não trivial.

Gerar o conjunto de soluções Pareto-ótimo geralmente é computacionalmente dispendioso e muitas vezes inviável, em razão da complexidade de submeter MOOPs à métodos exatos [Zitzler et al., 2003]. Dessa forma, o principal objetivo da otimização multi-objetivo torna-se encontrar um conjunto de soluções não-dominadas próximas do conjunto Pareto-ótimo e com boa distribuição pela fronteira.

De acordo com Zitzler et al. [2003], a saída dos MOEAs ou outras heurísticas populacionais é composta por um conjunto de soluções incomparáveis, denominado conjunto de aproximação [Hansen & Jaszkiewicz, 1998]. Formalizado da seguinte forma:

Definição 5 (*Conjunto de Aproximação*) *Seja $A \subseteq Z$ um conjunto de vetores de objetivo, o qual é intitulado um conjunto de aproximação se nenhum de seus elementos não domine fracamente qualquer outro vetor de objetivo do conjunto, sendo Z o espaço de objetivos.*

Frequentemente, a abordagem mais utilizada para avaliar o desempenho de um MOEA é baseada em indicadores de qualidade, a qual reduz cada conjunto de aproximação obtido a um valor numérico que representa a qualidade do mesmo. Na literatura há métricas para medir a proximidade do conjunto aproximado analisado em relação ao conjunto Pareto-ótimo (*Error ratio, Set Coverage Metric, Generational Distance*, etc.), outras para avaliar a diversidade das soluções obtidas (*Spacing Spread*, etc.) e as métricas que realizam as duas avaliações (*Hipervolume, binary ϵ -indicator, binary Additive ϵ -indicator*, etc.). De acordo com Zitzler et al. [2003], para realizar comparações entre um grupo de conjuntos de aproximação provenientes dos algoritmos analisados, os indicadores de qualidade mais aptos são: *binary ϵ -indicator* e o *set coverage metric*. A justificativa para tal fato é que ambos provêm compatibilidade e robustez para a maioria das relações de dominância, ou seja, ambos são capazes de representar de forma aceitável tais relações. A seguir estas duas técnicas de medida de desempenho são apresentadas.

2.3.1 Cobertura (C)

O indicador de qualidade cobertura (em inglês *Set Coverage Metric*) foi proposto por Zitzler & Thiele [1998]. Esta métrica mapeia um par ordenado (X, Y) para um valor dentro do intervalo $[0, 1]$. O cálculo $C(A, B)$ determina a proporção de soluções do conjunto B que são dominadas fracamente (cobertas) por soluções contidas no conjunto A . Tanto $C(A, B)$ quanto $C(B, A)$ devem ser considerados, pelo de não serem necessariamente complementares. Matematicamente pode ser representada da seguinte forma:

$$C(A, B) = \frac{|\{b \in B | \exists a \in A : a \preceq b\}|}{|B|} \quad (2.10)$$

Logo, o intuito é obter valores baixos para C . Caso $C(A, B)$ é igual a 1, todas as soluções em B são dominadas fracamente por soluções da fronteira A . Em contrapartida,

tida, se o valor retornado é igual a 0, significa a situação oposta, na qual nenhuma das soluções de B são dominadas fracamente por soluções de A .

2.3.2 Binary Additive ϵ -Indicator ($I_{\epsilon+}$)

A métrica Binary Additive ϵ -Indicator foi proposta por Zitzler et al. [2003]. Baseada no conceito de ϵ -dominância, $I_{\epsilon+}(A, B)$ calcula o fator mínimo ϵ necessário para qualquer vetor de objetivo do conjunto de aproximação B seja ϵ -dominado por pelo menos um vetor de objetivo do conjunto de aproximação A . Considerando que o problema seja de minimização com n objetivos, matematicamente este cálculo é realizado da seguinte forma:

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \{ \forall z^2 \in B \exists z^1 \in A : z^1 \preceq_{\epsilon+} z^2 \} \quad (2.11)$$

em que um vetor de objetivos $z^1 \in Z$ ϵ -domina um outro vetor $z^2 \in Z$ ($z^1 \preceq_{\epsilon+} z^2$) se e somente se:

$$\forall 1 \leq i \leq n : z_i^1 \leq \epsilon + z_i^2 \quad (2.12)$$

Devido a complexidade de obter o conjunto Pareto-ótimo da maioria dos MOPs, Fonseca et al. [2005] propuseram uma versão unária desta métrica, na qual o par de conjuntos é formado pelo conjunto de aproximação A obtido pelo algoritmo avaliado e pelo conjunto R composto pelas melhores soluções não-dominadas conhecidas. Note, que quanto mais próximo de 0 está o valor obtido por $I_{\epsilon+}(A, R)$, melhor é a qualidade do conjunto de aproximação avaliado.

Capítulo 3

Transporte Reativo a Demanda

Neste capítulo, são apresentados o serviço de transporte abordado, assim como os problemas selecionados dentro campo da Pesquisa Operacional (PO) para serem a base da modelagem do mesmo. Inicialmente, nas Seções 3.1 e 3.2 são apresentados os problemas relacionados. Posteriormente, na Seção 3.4 é feita a introdução do serviço DRT abordado neste trabalho, seguida de uma revisão da literatura na Seção 3.3.

3.1 Problemas de Roteamento de Veículos com Coletas e Entregas

A classe de problemas de roteamento de veículos com coletas e entregas (VRPPD, do inglês *Vehicle Routing Problem with Pickups and Deliveries*) compreende todos os problemas caracterizados pelo transporte de bens/passageiros entre pontos (locais) de coleta e de entrega. Este transporte é operacionalizado por meio de uma frota de veículos inicialmente localizada em um depósito, a qual deve satisfazer um conjunto de requisições de transporte. Cada requisição é composta por um ponto de coleta, um ponto de entrega e uma demanda que deverá ser transportada entre estes dois pontos (Toth & Vigo [2002]; Parragh et al. [2008]). Geralmente, o objetivo tratado nesta classe de problemas consiste em planejar um conjunto de rotas minimizando os custos de operação a fim de satisfazer as demandas de um conjunto de consumidores.

Segundo Parragh et al. [2008], a classe VRPPD pode ser dividida em duas subclasses de problemas: Desemparelhados e Emparelhados. Na subclasse que compreende os problemas classificados como desemparelhados, cada carga coletada pode ser utilizada para suprir a demanda de qualquer consumidor, ou seja, não há vínculo entre um ponto de coleta e um ponto de entrega que restrinja a distribuição das cargas trans-

portadas. Na literatura há o Coleta e Entrega - Problema de Roteamento de veículo (PDVRP, do inglês *Pickup and Delivery Vehicle Routing Problem*) e o Coleta e Entrega - Problema do Caixeiro Viajante (PDTSP, do inglês *Pickup and Delivery Traveling Salesman Problem*) nos casos em que há múltiplos veículos e somente um único veículo, respectivamente.

A segunda subclasse engloba problemas nos quais as requisições são associadas à pares ordenados de pontos, contendo um ponto onde ocorrerá a coleta da carga e um segundo ponto onde ocorrerá a entrega da mesma. Referente a esta subclasse há na literatura o clássico Problema de Coleta e Entrega (PDP, do inglês *Pickup and Delivery Problem*) e o *Dial-a-Ride Problem* (DARP). Ambos problemas são baseados em requisições de transporte compostas por um par de pontos. Entretanto, há uma diferença importante entre os mesmos. O primeiro problema lida com o transporte de bens/produtos, enquanto o segundo foca o transporte de passageiros. Geralmente, esta diferença é expressa por meio de restrições adicionais, ou por objetivos extras ou até mesmo por objetivos substituindo os clássicos objetivos relacionados ao custo, a fim de manter a qualidade do serviço prestado aos clientes. Estas restrições e objetivos adicionais o tornam ainda mais complexo e difícil de ser resolvido. Os casos nos quais há somente um veículo disponível no PDP e também no DARP são nomeados SPDP e SDARP, respectivamente.

Dentre todos estes problemas, o foco deste trabalho é direcionado para o DARP. Esta escolha se deve ao fato de que o modelo proposto para o serviço DRT abordado foi construído com base neste problema.

3.2 Dial-a-Ride Problem

O *Dial-a-Ride Problem* (DARP) é um problema de roteirização e programação de veículos no qual a carga transportada são passageiros e o par de pontos (localizações) é formado por um ponto de embarque e um segundo ponto onde deverá ser realizado o desembarque. Consiste em projetar rotas e definir horários de atendimento para um conjunto de passageiros, os quais formulam requisições de coleta e entrega entre origens e destinos desejados. O transporte é realizado por uma frota de veículos que fornece um serviço compartilhado, ou seja, vários passageiros provenientes de requisições diferentes podem estar no mesmo veículo ao mesmo tempo. O processo de planejamento das rotas busca atender um conjunto de requisições minimizando os custos de operação e respeitando um conjunto de restrições, como por exemplo: número de veículos disponíveis, capacidade dos veículos, janelas de tempo, entre outras [Cordeau & Laporte,

2007]. Encontrar uma solução viável para o DARP é NP-difícil [Cordeau, 2006]. Neste tipo de problema, uma requisição de transporte enviada por um cliente é composta por um ponto de embarque (coleta), um ponto de desembarque (entrega) e suas respectivas janelas de tempo. As janelas de tempo impõem intervalos, nos quais os atendimentos (embarque e desembarque) podem ocorrer ao longo do dia.

Segundo Toth & Vigo [2014], várias rotas devem ser definidas e cada uma deve satisfazer as seguintes restrições:

- Atendimento: cada usuário deve ser servido somente uma vez.
- Depósito: todo veículo deve iniciar e finalizar a rota no depósito.
- Acoplamento: a coleta e a entrega do usuário deve ser realizada pelo mesmo veículo.
- Precedência: cada usuário deve ser coletado e depois entregue pelo veículo.
- Capacidade: a capacidade máxima do veículo nunca deve ser violada.
- Janelas de tempo: cada usuário deve embarcar e desembarcar em horários dentro dos intervalos definidos pelas janelas de tempo.
- Duração máxima da rota: a duração total das rotas não devem exceder um valor máximo estabelecido.
- Tempo máximo de viagem: o tempo total de viagem de cada passageiro não deve exceder um valor máximo estabelecido.

A aplicação mais comum do DARP são os serviços de transporte porta-a-porta destinados a idosos e deficientes. Neste tipo de serviço, os clientes frequentemente formulam duas requisições durante o dia. Uma requisição solicitando o transporte da casa do usuário para algum outro local (do inglês *outbound request*) como por exemplo hospitais ou clínicas e outra solicitando o transporte da viagem de volta (do inglês *inbound request*) [Cordeau et al., 2007]. Embora geralmente direcionado para o transporte de passageiros com pouca mobilidade, serviços DARP podem ser viabilizados para vários outros segmentos da população, como por exemplo: transporte escolar, ambulâncias, turismo, etc. [Paquette et al., 2009].

Aprofundando dentre problemas presentes na literatura relacionados ao DARP, é possível afirmar que o mesmo generaliza vários problemas de roteamento como VRPPD e o Problema de Roteamento de Veículos com Janelas de Tempo (VRPTW, do inglês *Vehicle Routing Problem with Time Windows*). O que torna o DARP diferente da

maioria destes problemas é a perspectiva humana que está presente em transporte de passageiros, ou seja, a qualidade do serviço deve estar em equilíbrio com a minimização dos custos operacionais [Cordeau & Laporte, 2007]. Na literatura, o DARP também é caracterizado por um Problema de Roteamento de Veículos com Coleta e Entrega e com Janelas de Tempo (VRPPDTW, do inglês *Vehicle Routing Problem with Pickup and Delivery and Time Windows*).

Serviços *Dial-a-Ride* podem operar tanto em um contexto estático quanto em um contexto dinâmico. No primeiro caso, todas as requisições são conhecidas antes do início do processo de planejamento das rotas, as quais os veículos irão percorrer ao longo do dia. Já no segundo caso, além das requisições enviadas antes da elaboração das rotas, novas requisições podem surgir ao longo do dia e as rotas já existentes são ajustadas em tempo real para suprir as novas demandas. Caso não seja possível acomodar as novas requisições nestas rotas, outras são criadas para efetuar o serviço [Cordeau, 2006]. Na prática, problemas dinâmicos são frequentemente tratados como sequências de subproblemas estáticos. A solução atual pode ser adaptada sempre que uma nova requisição é realizada, ou um conjunto composto por novas requisições pode ser armazenado e periodicamente incorporado em lotes às rotas existentes [Cordeau et al., 2007].

3.2.1 Modelo Matemático DARP

Um dos principais modelos matemáticos propostos para o DARP foi apresentado em Cordeau [2006]. Neste modelo de programação mista é tratado o caso estático do problema, o qual foi definido da seguinte forma: dado um conjunto de n usuários (ou requisições) a serem servidos e um grafo completo direcionado $G = (N, A)$, onde $N = P \cup D \cup \{0, 2n + 1\}$ e A são os conjuntos de todos vértices e de todas as arestas, respectivamente. Os subconjuntos $P = \{1, \dots, n\}$ e $D = \{n + 1, \dots, 2n\}$ contêm os vértices de embarque e de desembarque dos passageiros, respectivamente. Para cada usuário i é associado um vértice de embarque i e um vértice de desembarque $n + i$. Os vértices 0 e $2n + 1$ representam os depósitos de origem e de destino dos veículos.

Seja K o conjunto de veículos disponíveis. Para cada $k \in K$ são conhecidos sua capacidade máxima Q_k e também a duração total máxima T_k da rota realizada pelo mesmo. Em cada vértice $i \in N$ são definidos uma carga q_i e uma duração de serviço não negativa d_i dedicada para o embarque ou desembarque do(s) passageiro(s). Para os vértices de coleta e entrega define-se $q_i = -q_{n+i}$ ($i = 1, \dots, n$) e $d_i \geq 0$, e para os depósitos $q_0 = q_{2n+1} = 0$ e $d_0 = d_{2n+1} = 0$. Para cada aresta $(i, j) \in A$ é associado um custo de roteamento c_{ij} e um tempo de viagem t_{ij} . É definido um tempo de viagem

máximo L para os usuários. Além disso, para cada nó é associada uma janela de tempo $[e_i, l_i]$, onde e_i e l_i representam o horário mais cedo e o mais tarde para início do serviço no nó i , respectivamente.

As variáveis de decisão definidas para este modelo do DARP são:

- x_{ij}^k : variável binária definida como 1 se o veículo k percorre a aresta (i, j) , 0 caso contrário.
- B_i^k : horário no qual o veículo k inicia o serviço no nó i , 0 se o nó não é servido pelo veículo.
- Q_i^k : ocupação do veículo k após visitar o nó i , 0 se o nó não é servido pelo veículo.
- L_i^k : duração da viagem do usuário i no veículo k , 0 se o usuário não é servido pelo veículo.

Função Objetivo

$$F = \min \sum_{k \in K} \sum_{i \in N} \sum_{j \in N} x_{ij}^k c_{ij}^k \quad (3.1)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j \in N} x_{ij}^k = 1, \forall i \in P \quad (3.2)$$

$$\sum_{j \in N} x_{ij}^k - \sum_{j \in N} x_{n+i,j}^k = 0, \forall i \in P, k \in K \quad (3.3)$$

$$\sum_{j \in N} x_{0j}^k = 1, \forall k \in K \quad (3.4)$$

$$\sum_{j \in N} x_{ji}^k - \sum_{j \in N} x_{ij}^k = 0, \forall i \in P \cup D, k \in K \quad (3.5)$$

$$\sum_{i \in N} x_{i,2n+1}^k = 1, \forall k \in K \quad (3.6)$$

$$B_j^k \geq B_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k), \quad (3.7)$$

$$\forall i \in N, j \in N, k \in K$$

$$\text{onde } M_{ij}^k \geq \max\{0, l_i + d_i + t_{ij} - e_j\}$$

$$Q_j^k \geq Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k), \quad (3.8)$$

$$\forall i \in N, j \in N, k \in K$$

$$\text{onde } W_{ij}^k \geq \min\{Q_k, Q_k + q_i\}$$

$$L_i^k = B_{n+1}^k - (B_i^k + d_i), \forall i \in P, \forall k \in K \quad (3.9)$$

$$B_{2n+1}^k - B_0^k \leq T_k, \forall k \in K \quad (3.10)$$

$$e_i \leq B_i^k \leq l_i, \forall i \in N, k \in K \quad (3.11)$$

$$t_{i,n+i} \leq L_i^k \leq L, \forall i \in P, k \in K \quad (3.12)$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{Q_k, Q_k + q_i\}, \forall i \in N, k \in K \quad (3.13)$$

$$x_{ij}^k \in \{0, 1\}, \forall i \in N, j \in N, k \in K \quad (3.14)$$

A função objetivo (3.1) minimiza o custo total de roteamento. As restrições (3.2) e (3.3) garantem que cada requisição seja servida exatamente uma vez e que o mesmo veículo visite os nós de origem e entrega de cada usuário. As restrições (3.4), (3.5) e (3.6) asseguram que o início da rota seja no depósito de origem e o término no depósito de destino e que se o veículo visita um nó, ele deve deixar este nó. As restrições (3.7) e (3.8) garantem a consistência dos horários de visita e da ocupação dos veículos. A restrição (3.9) calcula a duração de cada rota. A restrição (3.10) assegura que as durações das rotas sejam inferiores ou iguais ao limite imposto. As restrições (3.11) e (3.12) limitam o início do serviço em cada nó e o tempo de viagem de cada usuário, respectivamente. A restrição (3.13) garante que a capacidade dos veículos não seja violada. E por fim, a restrição (3.14) define que as variáveis x_{ij}^k sejam binárias.

3.3 Trabalhos Relacionados

As primeiras publicações no campo de problemas relacionados ao transporte de passageiros surgiram na década de 70 (Wilson et al. [1971]; Wilson & Wayne [1976]; Wilson & Colvin [1977]). Desde então, problemas que abordam o tema tem recebido considerável atenção dentro da comunidade científica. Extensas revisões da literatura sobre o DARP e algumas de suas principais variantes foram realizadas por Cordeau & Laporte [2007] e Parragh et al. [2010a]. Nesta seção, alguns dos principais trabalhos são explicitados seguindo uma estrutura baseada em qual caso está sendo tratado (estático ou dinâmico) e também qual abordagem foi utilizada para resolução (exatas, heurísticas e metaheurísticas).

3.3.1 Trabalhos que tratam o caso estático

Como já supracitado, serviços DARP que operam de forma estática planejam as rotas e os horários de atendimento quando já possuem o conjunto completo de requisições de transporte. A seguir alguns dos principais trabalhos presentes na literatura que tratam esta variante.

3.3.1.1 Abordagens Exatas

O trabalho mais antigo que trata a variante DARP que aborda a utilização de um único veículo foi conduzido por Psaraftis [1980]. O autor baseia-se no princípio intitulado "requisição-imediata", o qual pressupõe que todo cliente que envia uma requisição de transporte deseja ser atendido o mais cedo possível. O problema foi formulado e resolvido utilizando um algoritmo de programação dinâmica, o qual otimiza dois objetivos em uma única função objetivo: minimiza a soma ponderada do tempo de conclusão da rota e a insatisfação do cliente. Ao invés dos clientes associarem janelas de tempo às requisições, o motorista impõe restrições fundamentadas no conceito de "máxima posição de mudança", que limita a diferença entre a posição de cada requisição na lista de reservas e a sua posição na ordem de atendimento na rota, ou seja, a ordem dos atendimentos deve estar coerente com a ordem em que as requisições foram recebidas. Somente instâncias pequenas puderam ser resolvidas pelo algoritmo proposto. Posteriormente, Psaraftis [1983] propôs modificações tanto no problema, no qual introduz janelas de tempo na partida e na chegada, quanto no algoritmo proposto, onde alterou o sentido da busca.

Cordeau [2006] introduziu uma formulação de programação inteira-mista para o DARP básico e um algoritmo *branch-and-cut*. O algoritmo adota inequações válidas já utilizadas para tratar conhecidos problemas da literatura, como: *Traveling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP) e o *Pickup and Delivery Problem* (PDP), assim como propõem novas inequações válidas para o DARP.

Em Ropke et al. [2007], foram introduzidos novos algoritmos *branch-and-cut* para o clássico Problema de Coleta e Entrega com Janelas de Tempo (PDPTW, do inglês *Pickup and Delivery Problem with Time Windows*) e o DARP. Propõem também duas novas formulações para o PDPTW, as quais, ao contrário da formulação apresentada por Cordeau [2006], possuem um número exponencial de restrições, mas conduzem para algoritmos mais eficientes, porque contêm menos variáveis e fornecem limites mais apertados. Várias famílias de inequações válidas são introduzidas para fortalecer as duas formulações propostas. Os algoritmos foram testados em vários conjuntos de instâncias de teste do PDPTW e DARP. Ambos algoritmos obtiveram desempenho

superior ao algoritmo *branch-and-cut* proposto por Cordeau [2006], o qual encontrou soluções ótimas para instâncias com no máximo 36 requisições. Os algoritmos propostos por Ropke et al. [2007] resolveram de forma ótima instâncias com até 96 requisições.

Parragh [2011] introduziu duas formulações (com dois e três índices) e um algoritmo *branch-and-cut*, todos baseados nos trabalhos desenvolvidos por Cordeau [2006] e Ropke et al. [2007] e adaptados para tratar a variante DARP com veículos heterogêneos (HDARP, do inglês *Heterogeneous Dial-a-Ride Problem*). Esta variante do problema surge em decorrência do transporte de idosos e deficientes. Todas inequações válidas propostas para o DARP padrão foram adequadas para a variante abordada. Instâncias com até 40 requisições foram resolvidas de forma ótima. Parragh et al. [2012] desenvolveram duas novas formulações matemáticas para tratar o DARP com usuários e veículos heterogêneos. Ambas formulações são utilizadas para compor algoritmos *branch-and-cut*. Ao contrário do trabalho anterior, os tempos de viagem máximo dos usuários não são considerados explicitamente, os mesmos são considerados em função de janelas de tempo associadas aos pontos de coleta e entrega. Esta modificação na característica do problema possibilitou a aplicação de um algoritmo de geração de colunas, e conseqüentemente permitiu resolver instâncias bem maiores que o algoritmo anterior.

Uma variante do HDARP, a qual aborda múltiplos depósitos (MD-H-DARP, do inglês *Multi-Depot Heterogeneous Dial-a-Ride Problem*) foi introduzida e resolvida por meio de um algoritmo *branch-and-cut* proposto por Braekers et al. [2014]. O problema abordado considera que usuários podem ter necessidades especiais (por exemplo deficientes), a frota de veículos é heterogênea e disponibilizada em múltiplos depósitos distribuídos ao longo da área de serviço. O algoritmo proposto foi baseado no algoritmo *branch-and-cut* proposto por Ropke et al. [2007] para tratar o DARP padrão, entretanto algumas adaptações realizadas e elementos são provenientes de Cordeau [2006].

Outra variante do DARP é o DARP realista (RDARP, do inglês *Realistic Dial-a-Ride Problem*), a qual foi estudada em Liu et al. [2015]. Nesta variante são considerados simultaneamente múltiplas viagens, veículos e passageiros heterogêneos entre outras características e restrições que surgiram ao longo dos anos. Foram propostos dois modelos de programação linear inteira-mista, assim como, apresentadas oito famílias de inequações válidas. Os autores também elaboraram um algoritmo *branch-and-cut* para aperfeiçoar e resolver os dois modelos propostos, respectivamente.

3.3.1.2 Abordagens Heurísticas

Dada a complexidade do DARP e conseqüentemente a capacidade de resolver de forma ótima apenas instâncias de teste de pequeno porte por meio de métodos exatos, como alternativa, ao longo das últimas décadas várias abordagens heurísticas têm sido propostas para tratar o problema.

Alguns dos primeiros trabalhos que utilizam heurísticas para resolução do SDARP foram Sexton & Bodin [1985a] e Sexton & Bodin [1985b]. Os autores definem que cada requisição de transporte é composta por um ponto de coleta, um ponto de entrega e somente uma janela de tempo, a qual é associada à entrega do passageiro. No conjunto, foram propostos uma heurística de roteamento e um algoritmo de agendamento que utilizam decomposição de Benders.

Jaw et al. [1986] propuseram uma abordagem de inserção sequencial para resolver o DARP. O algoritmo seleciona os passageiros de acordo com o horário de coleta mais cedo e os aloca de forma viável à alguma rota de modo a acarretar o menor aumento possível na função objetivo.

3.3.1.3 Metaheurísticas

Toth & Vigo [1996] estudaram o DARP aplicado a um serviço de transporte real para deficientes em uma área urbana. Neste trabalho foi desenvolvido uma heurística de inserção paralela, a qual atribui cada requisição à uma rota. Também é descrito um algoritmo de busca local que aplica movimentos e trocas a fim de refinar as rotas construídas. Este trabalho foi aperfeiçoado em Toth & Vigo [1997] por meio da aplicação de uma abordagem baseada na metaheurística Busca Tabu após o procedimento de inserção paralela com a finalidade de otimizar as rotas.

Em Cordeau & Laporte [2003], uma heurística baseada na metaheurística Busca Tabu é proposta para tratar o DARP estático, a qual minimiza o custo total de roteamento considerando restrições de janelas de tempo, tempo máximo de viagem do usuário e duração máxima das rotas.

Uma heurística de duas-fases foi desenvolvida por Parragh et al. [2009] para tratar a variante DARP com dois objetivos. A primeira fase é constituída por uma heurística baseada na resolução por soma ponderada dos objetivos na qual é utilizado o conceito da metaheurística *Variable Neighborhood Search* (VNS). A segunda fase é uma abordagem baseada no path relinking (PL), a qual busca todas soluções não encontradas durante a primeira fase. Em Parragh et al. [2010b] foi apresentada uma heurística baseada na metaheurística VNS que utiliza três classes de vizinhança. A abordagem proposta foi comparada com o algoritmo proposto por Cordeau & Laporte [2003].

Em Chevrier et al. [2010] um serviço DRT é abordado como um caso especial do DARP no qual são adicionadas algumas especificidades do serviço e uma formulação multi-objetivo. Uma abordagem evolutiva foi proposta, assim como uma nova representação para as soluções e novos operadores de variação. Tais módulos foram integrados a três algoritmos estado da arte: NSGA-II, SPEA-2, e IBEA. Para intensificar a busca no espaço de soluções, Chevrier *et al.* Chevrier et al. [2012] trataram o DARP aplicado ao contexto do DRT por meio de uma abordagem evolutiva multi-objetivo híbrida baseada nos algoritmos utilizados em Chevrier et al. [2010]. As rotas são otimizadas por meio de um algoritmo de busca local dentro do operador de mutação. Esta estratégia é baseada na meta-heurística *Iterated Local Search* (ILS) e contém o algoritmo 2-opt integrado.

Parragh & Schmid [2013] propuseram uma abordagem híbrida para tratar o DARP. A metaheurística *Large Neighborhood Search* (LNS) e um algoritmo de geração de colunas são combinados para buscar soluções que minimizem os custos de roteamento enquanto respeitam as janelas de tempo, o tempo máximo de viagem dos passageiros, duração máxima das rotas e capacidade dos veículos.

Miranda et al. [2012]; Gomes et al. [2014]; Mendes et al. [2016] abordam o problema roteamento de veículos para o transporte reativo a demanda (PRVTRD). No primeiro trabalho os autores apresentaram uma heurística reativa baseada na metaheurística GRASP para tratar a versão estática do problema, assim como Algoritmos de replanejamento para resolução do caso dinâmico. Já no segundo, o problema foi tratado considerando múltiplos objetivos. Foi proposto uma abordagem heurística, na qual rotas viáveis são construídas por meio de um procedimento guloso e também aleatório, e posteriormente submetidas a uma busca local, que minimiza os custos operacionais e maximiza a qualidade do serviço. Por fim, o terceiro trabalho também aborda o problema considerando múltiplos objetivos. Inicialmente, cinco funções objetivo foram consideradas e por meio da utilização da Árvore de Agregação, os objetivos foram agregados formando dois novos objetivos. O problema com os dois objetivos foi resolvido usando os algoritmos NSGA-II e SPEA 2. Além disso, duas novas funções objetivo foram inseridas no problema e uma nova agregação dos objetivos foi feita resultando em um novo problema bi-objetivo, o qual também foi resolvido utilizando as duas abordagens utilizadas anteriormente.

Em Viana et al. [2015] foi proposta uma abordagem heurística baseada no NSGA-II para otimizar o serviço DRT introduzido em Chevrier et al. [2010, 2012]. Os operadores de variação apresentados realizam somente movimentos que mantenham a viabilidade das soluções, concentrando a busca somente no espaço viável de soluções. A abordagem proposta foi comparada à também baseada no NSGA-II [Chevrier et al.,

2012], mas que permite movimentos que inviabilizam a solução e obteve resultados superiores.

3.3.2 Trabalhos que tratam o caso dinâmico

O termo dinâmico, entre outras definições, significa mudança, ou seja, algo que se altera de alguma forma. No contexto dos serviços DARP, este termo indica que as rotas são planejadas em tempo real a medida que chegam novas requisições de transporte. Existem dois possíveis cenários: as rotas que já foram criadas são adaptadas para servir a nova demanda ou novas rotas são criadas para atender as novas requisições. A seguir são apresentados os trabalhos mais relevantes na literatura que abordam a variante dinâmica do DARP. A estrutura utilizada é a mesma empregada na seção de trabalhos que abordam o caso estático.

3.3.2.1 Abordagens Exatas

Abordagens baseadas em métodos exatos não têm sido utilizadas para tratar a forma dinâmica do DARP. Segundo Parragh et al. [2008], uma possível razão para tal fato, é que no contexto de roteamento dinâmico, o conceito de soluções ótimas torna-se discutível. Entretanto, mesmo que a forma na qual o serviço opera seja dinâmica, é possível aceitar como solução ótima o melhor planejamento para o momento atual, segundo os objetivos tratados.

Em Viana & dos Santos [2015] foi proposta uma abordagem que simula um serviço DRT dinâmico, utilizando dois modelos de programação linear inteira-mista baseados no modelo DARP proposto por Cordeau [2006] e em especificações do serviço DRT apresentadas em Chevrier et al. [2012]. O algoritmo executa sequencialmente os modelos buscando encontrar soluções que necessitam do menor número de veículos e da menor duração total das rotas com esse número de veículos, suficientes para atender todas as requisições que são recebidas em tempo real ao longo da operação do serviço. Ambos modelos foram apresentados na seção anterior e o algoritmo proposto será apresentado posteriormente.

3.3.2.2 Abordagens Heurísticas

A primeira abordagem proposta para tratar o DARP na forma dinâmica foi apresentada por Daganzo [1978]. Três diferentes algoritmos de roteamento foram estudados, sendo o modo de inserção a principal diferença entre os mesmos. No primeiro, após cada ponto de parada, o veículo é direcionado para o ponto com inserção viável mais próximo.

No segundo, visita o ponto de coleta ou o ponto de entrega com inserção viável mais próximo, alternando as escolhas realizadas. Por último, o terceiro insere um número fixo de coletas na rota e depois realiza as entregas.

Madsen et al. [1995] propuseram um algoritmo baseado no trabalho de Jaw et al. [1986] para otimizar um serviço DARP real aplicado ao transporte de idosos e deficientes em Copenhagen. O planejamento leva em consideração a otimização de múltiplos objetivos, requisições novas que podem chegar ao longo do dia e a disposição de uma frota heterogênea de veículos, na qual um modelo qualquer pode não estar disponível a todo momento. Novas requisições são atribuídas à uma rota levando em consideração a dificuldade de inserção. O algoritmo de inserção proposto foi batizado com o nome REBUS.

Colorni & Righini [2001] abordam o DARP dinâmico dividindo-o em dois subproblemas: o primeiro agrupa os passageiros em subconjuntos e o segundo roteia cada subconjunto. Os dois subproblemas são solucionados alternadamente por meio de um algoritmo de duas-fases. Considerando somente as requisições mais urgentes, os subproblemas de roteamento são solucionados de forma ótima por meio de um algoritmo *branch-and-bound*.

Horn [2002b] propôs um software que gerencia um serviço de transporte reativo a demanda, no qual os passageiros podem solicitar o serviço com antecedência ou em um curto prazo em relação ao horário desejado. O procedimento de inserção das novas requisições busca o menor custo de inserção possível. Periodicamente, é aplicado um método de busca local com o intuito de otimizar as rotas. Em Horn [2002a] por sua vez, o software foi testado em um framework de modelagem intitulado LITRES-2.

Coslovich et al. [2006] desenvolveram um algoritmo de duas-fases baseado em perturbações em rotas. A primeira fase é realizada *off-line* e tem como objetivo a criação de uma vizinhança viável a partir da rota corrente. A segunda fase é executada em tempo real toda vez que uma nova requisição é recebida, sendo capaz de determinar de forma rápida a inclusão ou rejeição de uma requisição por meio de um simples processo de inserção. As perturbações são realizadas utilizando o algoritmo de troca de arestas *2-opt*.

Xiang et al. [2008] abordaram o DARP na forma dinâmica que considera com alterações nos tempos de viagem, novas requisições, ausência de passageiros, inoperância de veículos, cancelamento de requisições, engarrafamentos e etc. É proposto um esquema flexível de agendamento e uma heurística para re-otimizar o planejamento quando novos eventos ocorrerem.

3.3.2.3 Metaheurísticas

Em Attanasio et al. [2004], a heurística Busca Tabu proposta por Cordeau & Laporte [2003] foi adaptada por meio de paralelização para tratar o DARP dinâmico. Diferentes implementações paralelas foram testadas no trabalho desenvolvido.

Hanne et al. [2009] abordam o DARP aplicado ao contexto de hospitais na Alemanha. O problema é tratado por meio de um Algoritmo Evolucionário.

Beaudry et al. [2010] tratam o DARP com restrições adicionais que surgem em decorrência da aplicação do problema em um específico contexto hospitalar. Neste trabalho é proposta uma heurística de duas-fases, sendo a primeira fase um algoritmo que gera uma solução viável por meio de um simples esquema de inserção. E a segunda fase, uma heurística baseada na metaheurística Busca Tabu que procura melhorar a solução obtida na primeira fase.

Schilde et al. [2011] abordam o DARP dinâmico e estocástico com transporte de retorno esperado, o qual foi aplicado ao transporte de pacientes de uma empresa não governamental localizada na Áustria. O serviço lida com requisições enviadas com antecedência e também com outras que chegam ao longo do dia, podendo ser requisições que solicitam o transporte da casa do paciente até o hospital ou a viagem de volta. Além disso, algumas requisições de ida para o hospital podem influenciar com uma certa probabilidade a viagem de volta, adquirindo assim, além da forma dinâmica de operar, a característica de estocasticidade. Foi utilizada uma versão dinâmica do *Variable Neighborhood Search* (VNS), um VNS estocástico e uma abordagem de múltiplo planejamento e uma abordagem de múltiplos cenários aplicadas às versões do VNS anteriores.

Berbeglia et al. [2012] introduziram uma abordagem híbrida que combina um algoritmo de programação por restrições exato e uma heurística baseada na metaheurística Busca Tabu para otimizar o DARP no contexto dinâmico.

3.3.3 Delimitação do Escopo do Trabalho

Dentre as diversas pesquisas apresentadas na seção anterior, o serviço DRT introduzido em Chevrier et al. [2010, 2012] foi selecionado para estudo. As abordagens propostas em ambos trabalhos exploram todo o espaço de busca, incluindo soluções inviáveis, as quais são penalizadas quando avaliadas. Uma pesquisa preliminar mostrou que manter a busca somente no espaço de busca viável possibilita obter resultados ainda melhores, considerando os mesmos objetivos [Viana et al., 2015]. Para a continuação do estudo, novos operadores foram propostos e aplicados nos três algoritmos estado da

arte (NSGA-II, SPEA-2, e IBEA) já utilizados pela literatura. Além disso, também é abordado e tratado o caso dinâmico do serviço.

3.4 Serviço de Transporte Reativo a Demanda

Transporte reativo a demanda (DRT, do inglês *Demand Responsive Transport*) é a definição utilizada para serviços de transporte que oferecem transporte sob demanda por meio de uma frota de veículos (ônibus, vans, carros, etc.), que são agendados para coletar e entregar passageiros de acordo com suas necessidades [Mageean & Nelson, 2003]. Para solicitar o serviço, usuários formulam requisições nas quais determinam locais e horários de embarque e desembarque desejados [Ambrosino et al., 2004]. Usualmente este tipo de serviço de transporte é compartilhado, ou seja, usuários provenientes de solicitações diferentes, mas com características em comum, seja área e/ou momento de atuação do serviço podem ser atendidos simultaneamente pelo mesmo veículo [Cordeau et al., 2007].

Há diversas razões que justificam o aumento significativo da popularidade dos serviços DRT nos últimos anos. Mageean & Nelson [2003] citam que as principais são: deficiências dos serviços de transporte público convencionais e serviços de táxi; deficiências dos serviços de transporte especiais; novos desenvolvimentos na comunidade do transporte; e um interesse em seu potencial no combate à exclusão social.

Em geral, serviços de transporte público operam por meio de rotas e horários fixos. São caracterizados pela falta de flexibilidade em ajustar o planejamento de acordo com a variação da demanda ao longo do dia e com as necessidades dos passageiros. Este tipo de serviço de transporte é apropriado para situações onde a demanda é alta e constante, sendo complexa a tarefa de adaptar-se economicamente e adequadamente à frequente variações da demanda [Crainic et al., 2012].

Ainda, segundo Executive et al. [2004], cada vez menos os serviços de transporte público convencionais, por exemplo os que operam por meio de ônibus, são capazes de atender as necessidades de grande parte da população. Esta situação é ocasionada por diversos fatores, tais como: aumento dos níveis de renda da população, o qual consequentemente impulsiona a compra de veículos próprios; dispersão dos centros de atividades (negócios); e alteração dos padrões de rotas decorrentes desta nova distribuição dos usuários. Tal incapacidade contribui diretamente em um serviço de transporte menos eficaz e menos eficiente como transporte de pessoas. Consequentemente, esta concepção impulsiona ainda mais a crescente aquisição de carros privados, a qual determina o declínio na utilização deste tipo de transporte, claramente perceptível nos

últimos anos. Este cenário é conhecido como ciclo vicioso do declínio do transporte público.

Devido à característica de operação sob-demanda, um serviço DRT é útil em áreas esparsas e de baixa densidade demográfica, as quais lidam com a falta de serviços de transporte devido ao fato dos provedores não aceitarem o custo de um serviço de transporte subutilizado [Chevrier, 2008]. Um serviço DRT é considerado uma forma intermediária de prover transporte, situado em algum lugar entre os ônibus (transporte compartilhado e geral) e os táxis (transporte individual e personalizado) [Mageean & Nelson, 2003]. Ele procura combinar as vantagens do transporte coletivo e dos táxis e ao mesmo tempo evitar suas desvantagens.

Historicamente, os serviços DRT evoluíram a partir dos serviços *dial-a-ride*, os quais eram providos pelas autoridades ou por grupos comunitários para atender um público específico formado por pessoas com pouca mobilidade (por exemplo idosos e deficientes). Este serviço é operacionalizado de porta a porta e os usuários precisam solicitar o transporte alguns dias antes do dia da realização do serviço para que os operadores efetuem o planejamento das rotas e dos horários de atendimento. Destinados também para outros públicos, como por exemplo passageiros que habitam regiões distantes dos centros urbanos com baixa demanda, os serviços *dial-a-ride* frequentemente são criticados pelo fato de possuírem alto custo de operação, pela falta de flexibilidade no planejamento das rotas e pela incapacidade de gerir altas demandas [Mageean & Nelson, 2003; Brake et al., 2004]. Entretanto, grandes avanços nas áreas da comunicação e computação, que se tornaram mais poderosas, acessíveis e portáteis, permitiram a evolução e popularização de novas formas de serviços de transporte flexíveis.

3.4.1 Definição do Problema

Em sua forma geral, um serviço DRT pode ser formulado com base em modelos do DARP. Os modelos matemáticos propostos são adaptações do modelo DARP apresentado em Cordeau [2006], aos quais são incorporadas especificações do serviço DRT abordado em Chevrier et al. [2010] e Chevrier et al. [2012]. O serviço de transporte sob-demanda tratado busca otimizar três objetivos. Deseja-se minimizar o número de veículos utilizados, a duração total das rotas e o atraso nos atendimentos. Dado que o problema abordado é uma generalização do DARP, o mesmo é considerado NP-Difícil dado a sua complexidade.

O serviço DRT abordado pode ser definido por meio de um grafo completo direcionado $G = (V, A)$, onde $V = V^+ \cup V^- \cup \{0, 2n+1\}$ é o conjunto de todos vértices e A é o conjunto de todas as arestas. Os subconjuntos $V^+ = \{1, \dots, n\}$ e $V^- = \{n+1, \dots, 2n\}$

contêm todos vértices de coleta e entrega, respectivamente. Um conjunto formado por n usuários (ou requisições) devem ser servidos. Cada requisição consiste de um vértice de coleta i e um vértice de entrega $n+i$. Os vértices 0 e $2n+1$ representam os depósitos de origem e destino.

Assim como no DARP, seja K uma frota homogênea de veículos com k veículos idênticos, todos com capacidade Q . Para cada usuário $i = 1 \dots n$ é associado um número de passageiros q_i e uma duração de serviço d_i para entrada e saída de passageiros. Para os vértices de coleta e entrega defina-se $q_i = -q_{n+i}$ ($i = 1, \dots, n$) e $d_i > 0$, e para os depósitos $q_0 = q_{2n+1} = 0$ e $d_0 = d_{2n+1} = 0$. Para cada aresta $(i, j) \in A$ é associado um tempo de viagem t_{ij} não nulo.

Cada usuário $i = 1, \dots, n$ define um horário de coleta desejado h_{i+} . A duração da janela de tempo w_{i+} em um ponto de coleta é proporcional à duração $t_{i,n+i}$ do percurso de i para $n+i$, definida como: $w_{i+} = k_w \cdot t_{i,n+i}$, sendo k_w um coeficiente não negativo que determina a duração alocada para a janela de coleta. O horário máximo de coleta h'_{i+} é então definido como $h'_{i+} = h_{i+} + w_{i+}$. O horário teórico de chegada no ponto de entrega h_{i-} é a soma do horário de coleta desejado e a duração do percurso de i para $n+i$ ($h_{i+} + t_{i,n+i}$). O horário máximo de entrega h'_{i-} é definido como: $h'_{i-} = h_{i+} + (k_r \cdot t_{i,n+i})$ sendo k_r um coeficiente de relaxação ($k_r > k_w$ e $k_r \geq 1$).

Devido a existência de três objetivos, o serviço DRT abordado é modelado utilizando um modelo matemático principal que otimiza o número de veículos utilizados e também outros dois modelos derivados do primeiro, os quais otimizam os demais objetivos, mas com adaptações que basicamente transformam o objetivo do modelo principal em restrição. Abaixo as principais variáveis de decisão presentes em todos os três modelos de programação linear mista:

- x_{ij}^k : variável binária definida como 1 se o veículo k percorre a aresta (i, j) , 0 caso contrário.
- v^k : variável binária definida como 1 se o veículo k é utilizado, 0 caso contrário.

Outras variáveis de decisão são utilizadas para modelar as restrições e os objetivos:

- t^k : tempo total de viagem do veículo k se o mesmo é utilizado, 0 caso contrário.
- D_i : atraso na entrega do usuário i
- H_i^k : horário de chegada do veículo k no nó i , 0 se não é servido pelo veículo.
- Q_i^k : número de passageiros a bordo do veículo k após visitar o nó i , 0 se não é servido pelo veículo.

3.4.1.1 Modelo 1

O primeiro modelo matemático proposto tem como objetivo buscar o menor número possível de veículos necessários para servir um conjunto de requisições e que satisfaça um grupo de restrições impostas pelo problema. A quantidade de veículos disponíveis k é igual ao número de requisições a serem atendidas e a quantidade mínima de veículos necessários k_{min} é a solução ótima objetivada. No pior caso será utilizado um veículo para servir cada requisição ($k_{min} = n$). O serviço DRT que busca minimizar o número de veículos utilizados pode ser formulado por meio do seguinte modelo de programação linear mista:

Função Objetivo

$$F_1 = \min \sum_{k \in K} v^k \quad (3.15)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \forall i \in V^+ \quad (3.16)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0, \forall i \in V^+, k \in K \quad (3.17)$$

$$\sum_{j \in V} x_{0j}^k = v^k, \forall k \in K \quad (3.18)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0, \forall i \in V^+ \cup V^-, k \in K \quad (3.19)$$

$$\sum_{i \in V} x_{i,2n+1}^k = v^k, \forall k \in K \quad (3.20)$$

$$\sum_{j \in V} x_{ij}^k \leq v^k, \forall i \in V^+, k \in K \quad (3.21)$$

$$H_j^k \geq H_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k), \forall i \in V, j \in V, k \in K \quad (3.22)$$

$$\text{onde } M_{ij}^k \geq \max\{0, h_{i-} + d_i + t_{ij} - h_{j+}\}$$

$$Q_j^k \geq Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k), \forall i \in V, j \in V, k \in K \quad (3.23)$$

$$\text{onde } W_{ij}^k \geq \min\{Q, Q + q_i\}$$

$$t^k = H_{2n+1}^k - H_0^k, \forall k \in K \quad (3.24)$$

$$D_i \geq \max\{0, H_i^k - h_{i-}\}, \forall i \in V^-, k \in K \quad (3.25)$$

$$v^{k-1} \geq v^k, \forall k \in K \setminus 1 \quad (3.26)$$

$$\begin{aligned}
v^k &\in \{0, 1\}, \forall k \in K & (3.27) \\
x_{ij}^k &\in \{0, 1\}, \forall i \in V, j \in V, k \in K \\
t^k &\geq 0, \forall k \in K \\
h_{i^+} &\leq H_i^k \leq h_{i^+} + w_{i^+}, \forall i \in V^+, k \in K \\
h_{i^-} &\leq H_i^k \leq h'_{i^-}, \forall i \in V^-, k \in K \\
\max\{0, q_i\} &\leq Q_i^k \leq \min\{Q, Q + q_i\}, \forall i \in V, k \in K
\end{aligned}$$

A função objetivo (3.15) minimiza o número de veículos utilizados. As restrições (3.16) e (3.17) garantem que cada requisição seja servida exatamente uma vez e que o mesmo veículo visite os nós de origem e entrega. As restrições (3.18), (3.19) e (3.20) asseguram que se o veículo k é utilizado, o início da rota seja no depósito de origem e o término no depósito de destino e que se o veículo visita um nó, ele deve deixar este nó. A restrição (3.21) certifica que o veículo será definido como utilizado se o mesmo serve alguma requisição. As restrições (3.22) e (3.23) garantem a consistência dos horários de visita e ocupação dos veículos. As restrições (3.24) e (3.25) definem a duração da rota de cada veículo e o atraso da chegada em cada nó de entrega. A restrição (3.26) remove uma simetria na utilização dos veículos, forçando o uso dos veículos de menores índices. Finalmente, o conjunto de restrições (3.27) define as variáveis binárias e os limites de cada uma das outras variáveis do modelo, assegurando que cada usuário seja servido dentro de sua respectiva janela de tempo e que a capacidade dos veículos seja respeitada a cada visita.

3.4.1.2 Modelo 2

O segundo modelo matemático tem como objetivo a minimização da duração total das rotas. Para isso, o tamanho da frota de veículos disponíveis é fixado e assume o valor K_{min} obtido por meio da resolução do primeiro modelo, sendo assim um parâmetro de entrada. Além de uma função objetivo diferente, o segundo modelo carece de algumas modificações em relação as restrições do modelo anterior que utilizam a variável de decisão v^k . As restrições (3.18) e (3.20) são alteradas (variável v^k é substituída pela constante 1), pois neste modelo todos os veículos disponíveis (K_{min}) são utilizados, sendo assim não é mais necessário identificar quais foram alocados. Da mesma forma, como a variável v^k não é considerada no modelo 2, às restrições (3.21) e (3.26) foram removidas. Com o intuito de facilitar a compreensão e permitir uma visão mais clara do modelo 2, mesmo as restrições que não foram alteradas serão repetidas. O segundo modelo de programação linear mista pode ser definido como:

Função Objetivo

$$F_2 = \min \sum_{k \in K_{min}} t^k \quad (3.28)$$

Sujeito a:

$$\sum_{k \in K_{min}} \sum_{j \in V} x_{ij}^k = 1, \forall i \in V^+ \quad (3.29)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0, \forall i \in V^+, k \in K_{min} \quad (3.30)$$

$$\sum_{j \in V} x_{0j}^k = 1, \forall k \in K_{min} \quad (3.31)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0, \forall i \in V^+ \cup V^-, k \in K_{min} \quad (3.32)$$

$$\sum_{i \in V} x_{i,2n+1}^k = 1, \forall k \in K_{min} \quad (3.33)$$

$$H_j^k \geq H_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k), \quad (3.34)$$

$$\forall i \in V, j \in V, k \in K_{min}$$

$$\text{onde } M_{ij}^k \geq \max\{0, h_{i-} + d_i + t_{ij} - h_{j+}\}$$

$$Q_j^k \geq Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k), \quad (3.35)$$

$$\forall i \in V, j \in V, k \in K_{min}$$

$$\text{onde } W_{ij}^k \geq \min\{Q, Q + q_i\}$$

$$t^k = H_{2n+1}^k - H_0^k, \forall k \in K_{min} \quad (3.36)$$

$$d_i \geq \max\{0, H_i^k - h_{i-}\}, \forall i \in V^-, k \in K_{min} \quad (3.37)$$

$$x_{ij}^k \in \{0, 1\}, \forall i \in V, j \in V, k \in K_{min} \quad (3.38)$$

$$t^k \geq 0, \forall k \in K_{min}$$

$$D_i \geq 0, \forall i \in V^-$$

$$h_{i+} \leq H_i^k \leq h_{i+} + w_{i+}, \forall i \in V^+, k \in K_{min}$$

$$h_{i-} \leq H_i^k \leq h'_{i-}, \forall i \in V^-, k \in K_{min}$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{Q, Q + q_i\}, \forall i \in V, k \in K_{min}$$

A função objetivo (3.28) minimiza a duração total das rotas. As restrições (3.29) e (3.30) garantem que cada requisição seja servida exatamente uma vez e que o mesmo veículo visite os nós de origem e entrega. As restrições (3.31), (3.32) e (3.33) asseguram que o início da rota seja no depósito de origem e o término no depósito de destino e que se o veículo visita um nó, ele deve deixar este nó. As restrições (3.34) e (3.35) garantem a consistência dos horários de visita e ocupação dos veículos. As restrições (3.36) e (3.37) definem a duração da rota de cada veículo e o atraso da chegada em cada nó de entrega. O conjunto de restrições (3.38) define as variáveis binárias e os limites de cada uma das outras variáveis do modelo, assegurando que cada usuário seja servido dentro de sua respectiva janela de tempo e que a capacidade dos veículos seja respeitada a cada visita.

3.4.1.3 Modelo 3

O terceiro modelo matemático tem como objetivo a minimização do atraso total nos atendimentos. Para isso, de forma similar ao modelo 2, o tamanho da frota de veículos disponíveis é fixado e assume o valor K_{min} obtido por meio da resolução do primeiro modelo, sendo assim um parâmetro de entrada. A única diferença para o modelo 2 é a função objetivo, a qual pode ser definida da seguinte forma:

Função Objetivo

$$F_3 = \min \sum_{i \in V^-} D_i \quad (3.39)$$

A função objetivo (3.39) minimiza o atraso no atendimento das requisições. As restrições deste modelo assim como as descrições de cada uma foram apresentadas na seção anterior.

Capítulo 4

Materiais e Métodos

Neste capítulo, inicialmente será apresentada a forma em que uma solução foi representada computacionalmente. Em seguida são descritos os módulos que compuseram os três algoritmos evolutivos (NSGA-II, IBEA e SPEA2) estado da arte analisados, tais como: procedimento de geração da população inicial e operadores de seleção, recombinação e mutação propostos. Após apresentação das abordagens viabilizadas para otimização do serviço DRT estático, serão apresentadas as abordagens exata e heurística criadas para resolução do caso dinâmico.

4.1 Representação de uma solução

Geralmente, a forma utilizada para representar uma solução em problemas de roteamento de veículos é basicamente por meio de um conjunto de rotas, as quais são formadas por uma sequência de pontos (vértices). Explicando por meio de uma forma mais técnica, uma solução consiste de vetores de vetores. Cada vetor rota representa o trajeto que será percorrido por um veículo, sendo que os pontos estão dispostos na ordem em que serão visitados. O primeiro e o último ponto de cada rota representam o depósito. Entretanto, é comum encontrar na literatura trabalhos que abordam o DARP ou problemas relacionados, nos quais uma rota é composta por uma sequência de identificadores de requisições e não por uma sequência de pontos. Nesta pesquisa, foi utilizada esta segunda forma de representação.

Como cada requisição de transporte é associada a um par de pontos, os quais representam os locais de embarque e de desembarque desejados, a primeira ocorrência do identificador da requisição na rota representa o embarque do passageiro e a segunda o desembarque do mesmo. Para facilitar a identificação da ação (embarque ou desembarque) que irá ocorrer em cada parada do veículo, a ocorrência do identificador

que representa o embarque será precedida pelo sinal positivo (+) e a que representa o desembarque será precedida pelo sinal negativo (-). Note que os identificadores em conjunto com os devidos sinais são armazenados no vetor na ordem em que deverão ser atendidos pelo veículo. Abaixo, a Tabela 4.1 apresenta um exemplo de solução para uma instância de teste fictícia composta por 6 requisições de transporte e por uma frota composta por 2 veículos.

Tabela 4.1. Representação da Solução

Veículo 1										
Requisição		1	2	1	2					
Ponto associado		+1	+2	-1	-2					
Rota final	D	+1	+2	-1	-2	D				
Veículo 2										
Requisição		3	4	4	3	5	5	6	6	
Ponto associado		+3	+4	-4	-3	+5	-5	+6	-6	
Rota final	D	+3	+4	-4	-3	+5	-5	+6	-6	D

4.2 Geração da População Inicial

Para gerar a população inicial, que irá se desenvolver ao longo das gerações efetuadas pelos MOEAs foi proposta uma abordagem baseada no procedimento de geração introduzido por Chevrier et al. [2012]. A principal diferença entre as abordagens é o critério adotado para priorizar o atendimento às requisições mais urgentes. A abordagem proposta considera o horário de coleta definido pelas requisições e o tempo necessário para transitar do depósito até o ponto de coleta na escolha das requisições mais urgentes. Desta forma, terão prioridade as requisições que demandam atendimentos que exigem maior antecedência da partida dos veículos que se encontram inicialmente no depósito. Por outro lado, o procedimento de geração da população inicial da literatura, a princípio, aloca um veículo diferente para toda requisição definida como "*minimal requests*", que são requisições nas quais a janela de tempo associada ao embarque seja violada sempre que o veículo atenda outra requisição primeiro.

Levando em consideração que toda a frota de veículos se encontra no depósito, o procedimento proposto inicialmente ordena a lista com todas as requisições disponíveis de acordo com o horário de coleta e o tempo gasto para transitar entre o depósito e o ponto em que deverá ser efetuado o embarque do passageiro ($h_{i+} - t_{D,i}$). Assim,

as requisições mais próximas ao depósito e mais urgentes terão prioridade em relação as demais. Após o processo de ordenação, as m primeiras requisições da lista são atribuídas à m veículos distintos, sendo m um parâmetro de entrada. Posteriormente, as demais requisições são ordenadas de acordo com os horários de coleta (h_{i+}) e em sequência são adicionadas de forma aleatória e viável a uma rota já existente. A aleatoriedade é utilizada para obter diversidade na população inicial. Caso não seja possível efetuar a operação anterior de forma viável, um novo veículo é alocado para servir a requisição. Em cada solução, o número de veículos utilizados pode variar entre m e o número de requisições disponíveis (pior caso, considerando a função objetivo que minimiza o número de veículos utilizados).

4.3 Operadores de Seleção e Variação

Os operadores de seleção e variação são responsáveis respectivamente por selecionar os indivíduos mais aptos da população e criar novas soluções com boa diversidade. A seguir, são detalhados e explicados os operadores de seleção, mutação e recombinação aqui propostos.

4.3.1 Operador de Seleção

O operador de seleção simula a seleção natural dando aos indivíduos mais aptos maior chance de reprodução e, conseqüentemente, uma maior oportunidade de manterem suas características ao longo das próximas gerações.

Geralmente, nos EAs, apenas alguns indivíduos da população corrente participam da reprodução. É necessário escolher de alguma forma as soluções, nas quais os operadores de variação serão aplicados. Todas as soluções selecionadas são armazenadas em um espaço chamado *mating pool* e posteriormente são utilizadas para geração da população composta por soluções filhas.

Em todas as abordagens propostas, a estratégia utilizada para seleção foi o torneio binário, no qual dois pares de indivíduos são selecionados de forma aleatória e apenas o melhor de cada par é selecionado para reprodução. No caso do algoritmo NSGA2, o indivíduo do par com melhor classificação, ou se houver empate neste critério, o que possui maior valor de distância de multidão é selecionado. No IBEA vence a disputa o indivíduo com maior valor de aptidão. E por fim, no SPEA2 é selecionado indivíduo com menor valor de aptidão, que ao contrário do MOEA anterior significa uma solução de melhor qualidade.

4.3.2 Operador de Recombinação (Crossover)

O operador de recombinação, também conhecido por crossover, produz novas soluções por meio de trocas de informações entre dois indivíduos da população corrente intitulados soluções pais. Após a criação da *mating pool* onde estão armazenadas, as soluções pais são utilizadas par-a-par para criação de novos indivíduos denominados soluções filhas. Este par de soluções é combinado de forma que os filhos herdem as características de ambos pais. A execução do operador de recombinação é controlada por uma probabilidade de ocorrência P_c chamada taxa de recombinação. No presente trabalho, foram testados quatro operadores de recombinação, os quais serão apresentados a seguir.

4.3.2.1 Operador de Recombinação 1

Proposto em [Viana et al., 2015], dadas duas soluções pais ($P1$ e $P2$), cria-se uma solução filha $F1$ idêntica à $P1$. Posteriormente, é selecionada aleatoriamente em $P2$ uma rota rt . Para que não existam duplicatas, são removidas de $F1$ todas as requisições que são atendidas em rt . Caso este procedimento resulte em rotas vazias, as mesmas são removidas de $F1$. Por fim, rt é adicionada em $F1$. O processo de criação de $F1$ é explicitado no Algoritmo 6. O processo de criação da segunda solução filha $F2$ é realizado da mesma forma, mas invertendo os papéis das soluções pais.

Algoritmo 6: CROSSOVER 1

Entrada: $P1, P2$

Saída: $F1$

```

1 início
2    $F1 \leftarrow P1$ 
3    $rt \leftarrow$  Seleciona Rota Randomicamente ( $P2$ );
4   para cada requisição  $req \in rt$  faça
5     para cada rota  $r \in F1$  faça
6       se Atende ( $r, req$ ) então
7         Remove Requisição ( $r, req$ );
8     pare
9   para cada rota  $r \in F1$  faça
10    se Rota Vazia ( $r$ ) então
11      Remove Rota ( $F1, r$ );
12  senão
13    Calcula Custo e Define Horários de Atendimento ( $F1, r$ );
14  Insere Rota ( $F1, rt$ );
15 retorna  $F1$ 

```

4.3.2.2 Operador de Recombinação 2

Dadas duas soluções pais ($P1$ e $P2$), cria-se uma solução filha $F1$ a partir de $P1$. Posteriormente, é selecionada em $P2$ a rota rt com menor valor da razão entre a duração total e o número de requisições atendidas. Desta forma, é selecionada a rota com melhor custo benefício de $P2$. Em seguida, todas as rotas de $F1$ que atendem pelo menos uma das requisições que são atendidas em rt são inseridas em uma lista R e removidas de $F1$. Por fim, rt é adicionada em $F1$ e todas as requisições atendidas pelas rotas presentes em R são re-inseridas em $F1$ com o menor custo de inserção possível (menor acréscimo na duração total), respeitando a ordem em que as rotas foram adicionadas em R e também a ordem em que as requisições são atendidas nas mesmas. O propósito de excluir as rotas afetadas e replanejar o atendimento de suas requisições é motivado pelo surgimento de longos períodos de ociosidade dos veículos, os quais precisam esperar pelo início das janelas de tempo dos atendimentos posteriores aos removidos da rota. Caso não exista possibilidade de inserção viável, uma nova rota será criada para atender a requisição. O processo de criação de $F1$ é explicitado no Algoritmo 7. A segunda solução filha $F2$ é criada por meio do mesmo procedimento mas invertendo os papéis das soluções pais.

Algoritmo 7: CROSSOVER 2

Entrada: $P1, P2$
Saída: $F1$

- 1 **início**
- 2 $F1 \leftarrow P1$
- 3 $rt \leftarrow$ Seleciona Rota Melhor Custo Benefício ($P2$);
- 4 $R \leftarrow \emptyset$
- 5 **para** cada requisição $req \in rt$ **faça**
- 6 **para** cada rota $r \in F1$ **faça**
- 7 **se** *Atende* (r, req) **então**
- 8 $R \leftarrow R \cup \{r\}$
- 9 Remove Rota ($F1, r$);
- 10 **pare**
- 11 Insere Rota ($F1, rt$);
- 12 **para** cada rota $r \in R$ **faça**
- 13 **para** cada requisição $req \in r$ **faça**
- 14 Insere com Menor Custo ($F1, req$);
- 15 **retorna** $F1$

4.3.2.3 Operador de Recombinação 3

O operador de recombinação 3 possui as etapas iniciais (linhas 1 a 4) semelhantes ao último descrito. Em seguida, são removidas de $F1$ todas as requisições que são atendidas em rt para que não existam duplicatas. Caso este processo resulte em rotas vazias, as mesmas deverão ser removidas. Após isso, rt é adicionada em $F1$. Baseado no operador de recombinação proposto em [Chiang & Hsu, 2014], todas as rotas de $F1$ que atendem somente uma ou duas requisições e a rota com maior valor da razão entre a duração total e o número de requisições atendidas são adicionadas em uma lista R e então removidas de $F1$. Por fim, todas as requisições atendidas pelas rotas de R são re-inseridas em $F1$ com o menor custo de inserção possível (menor acréscimo na duração total), respeitando a ordem em que as rotas foram adicionadas em R e também a ordem em que as requisições são atendidas nas mesmas. Este procedimento de replanejamento das requisições tenta eliminar rotas que subutilizam a capacidade de atendimento dos veículos, ou seja, procura maximizar o compartilhamento dos mesmos. Da mesma forma que no último operador, novas rotas poderão ser criadas para atender estas requisições. O processo de criação de $F1$ é explicitado no Algoritmo 8. A segunda solução filha $F2$ é criada por meio do mesmo procedimento mas invertendo os papéis das soluções pais.

4.3.2.4 Operador de Recombinação 4

Baseado no operador de recombinação clássico, a partir de duas soluções pais ($P1$ e $P2$), o procedimento de recombinação 4 inicia com a criação de uma solução filha $F1$ que a princípio é uma cópia de $P1$. Posteriormente, é selecionado um ponto de corte entre 0 e $\min(|P1| \text{ e } |P2|)$. Todas as rotas em $F1$ após o ponto de corte são removidas e para cada rota em $P2$ após o ponto de corte é realizada uma cópia, a qual é concatenada no vetor de rotas de $F1$. Em seguida, todas as requisições que são atendidas pelas rotas originadas de $P2$ são removidas das rotas vindas de $P1$ para que não ocorra duplo atendimento. Caso este processo resulte em rotas vazias, as mesmas são removidas. Por fim, todas as requisições que ainda não foram alocadas à alguma rota de $F1$ são re-inseridas com o menor custo de inserção possível (menor acréscimo na duração total), respeitando a ordem em que estão dispostas no conjunto de requisições. Este último procedimento é realizado com o auxílio de um conjunto S que armazena as requisições atendidas por alguma rota de $F1$ e com o conjunto total de requisições disponíveis Req . O processo de criação de $F1$ é explicitado no Algoritmo 9. A segunda solução filha $F2$ é criada por meio do mesmo procedimento mas invertendo os papéis das soluções pais.

Algoritmo 8: CROSSOVER 3

Entrada: $P1, P2$
Saída: $F1$

- 1 **início**
- 2 $F1 \leftarrow P1$
- 3 $rt \leftarrow$ Seleciona Rota Melhor Custo Benefício ($P2$);
- 4 $R \leftarrow \emptyset$
- 5 **para** cada requisição $req \in rt$ **faça**
- 6 **para** cada rota $r \in F1$ **faça**
- 7 **se** *Atende* (r, req) **então**
- 8 Remove Requisição (r, req);
- 9 **pare**
- 10 **para** cada rota $r \in F1$ **faça**
- 11 **se** *Rota Vazia* (r) **então**
- 12 Remove Rota ($F1, r$);
- 13 **senão**
- 14 Calcula Custo e Define Horários de Atendimento ($F1, r$);
- 15 **para** cada rota $r \in F1$ **faça**
- 16 **se** *Atende uma ou duas requisições* (r) ou $r ==$ *Rota Pior Custo Benefício* ($F1$) **então**
- 17 $R \leftarrow R \cup \{r\}$
- 18 Remove Rota ($F1, r$);
- 19 Insere Rota ($F1, rt$);
- 20 **para** cada rota $r \in R$ **faça**
- 21 **para** cada requisição $req \in r$ **faça**
- 22 Insere com Menor Custo ($F1, req$);
- 23 **retorna** $F1$

4.3.3 Operador de Mutação

O principal objetivo do operador de mutação é introduzir diversidade na população corrente. Nesta dissertação foi analisado o desempenho de dois operadores de mutação, os quais realizam a troca de apenas uma requisição entre diferentes rotas da solução. De forma semelhante ao operador de recombinação, a execução deste operador é controlada por uma probabilidade Pm chamada taxa de mutação.

4.3.3.1 Operador de Mutação 1

Utilizado em Viana et al. [2015], este operador realiza a troca de uma única requisição (cliente) entre duas diferentes rotas da solução, de forma que mantenha a viabilidade da mesma. Sendo assim, o objetivo deste operador é inserir diversidade, mantendo a viabilidade das soluções. O procedimento inicia selecionando de forma randômica duas

Algoritmo 9: CROSSOVER 4

Entrada: $P1, P2, Req$ **Saída:** $F1$

```

1 início
2    $F1 \leftarrow P1$ 
3    $S \leftarrow \emptyset$ 
4    $corte \leftarrow$  Selecciona Ponto de Corte ( $P1, P2$ );
5   Apaga Rotas após Ponto de Corte ( $F1, corte$ );
6   Copia Rotas após Ponto de Corte ( $F1, P2, corte$ );
7    $i = corte$ 
8   enquanto  $i < |P2|$  faça
9     para cada requisição  $r \in P2_i$  faça
10       $S \leftarrow S \cup \{r\}$ 
11       $i \leftarrow i + 1$ 
12   para cada requisição  $r \in S$  faça
13      $k = 0$ 
14     enquanto  $k < corte$  faça
15       se Atende ( $F1_k, r$ ) então
16         Remove Requisição ( $F1_k, r$ );
17         pare
18          $k \leftarrow k + 1$ 
19   para cada rota  $k \in F1$  faça
20     se Rota Vazia ( $F1_k$ ) então
21       Remove Rota ( $F1, k$ );
22     senão
23       Calcula Custo e Define Horários de Atendimento ( $F1, k$ );
24    $i = 0$ 
25   enquanto  $i < corte$  faça
26     para cada requisição  $r \in F1_i$  faça
27        $S \leftarrow S \cup \{r\}$ 
28        $i \leftarrow i + 1$ 
29   para cada requisição  $r \in Req$  faça
30     se  $r \notin S$  então
31       Insere com Menor Custo ( $F1, r$ );
32 retorna  $F1$ 

```

diferentes rotas da solução corrente ($r1$ e $r2$). Em seguida é selecionada randomicamente e removida de $r1$ uma requisição req . A tentativa de reinserção de req é iniciada em $r2$. Caso não seja possível inserir a requisição de forma que mantenha a viabilidade (por exemplo, respeitar as janelas de tempo e limite de ocupação dos veículos), é realizada uma nova tentativa na rota subsequente. Este processo continua enquanto não encontrar uma inserção viável e houver rotas. Se além de $r1$ nenhuma outra rota

da solução corrente for capaz de atender a requisição de forma viável, uma nova rota (veículo) é alocada para efetuar o atendimento. O procedimento que analisa a viabilidade da inserção inicia as tentativas sempre ao final da rota, de modo que afete poucos atendimentos e conseqüentemente resulte em um menor acréscimo no atraso total. Os Algoritmos 10 e 11 apresentam o operador de mutação e o procedimento de inserção viável, respectivamente.

Algoritmo 10: MUTAÇÃO 1

Entrada: S
Saída: S

- 1 **início**
- 2 $r1 \leftarrow$ Seleciona Randomicamente uma Rota (S);
- 3 **repita**
- 4 $r2 \leftarrow$ Seleciona Randomicamente uma Rota (S);
- 5 **até** $r1 \neq r2$;
- 6 $req \leftarrow$ Seleciona Randomicamente e Remove Requisição ($r1, S$);
- 7 $i \leftarrow 0$
- 8 $j \leftarrow r2$
- 9 **enquanto** $i < |S_{rotas}|$ **faça**
- 10 **se** $j \neq r1$ **então**
- 11 **se** *Inserção Viável* (S, j, req) **então**
- 12 **pare**
- 13 $i \leftarrow i + 1$
- 14 $j \leftarrow (j + 1) \% |S_{rotas}|$
- 15 **se** $i == |S_{rotas}|$ **então**
- 16 Criar Nova Rota (S, req);
- 17 **retorna** S

Algoritmo 11: INSERÇÃO VIÁVEL

Entrada: S, k, req

- 1 **início**
- 2 $i \leftarrow |S_k| - 1$
- 3 **enquanto** $i \geq 0$ **faça**
- 4 **se** *Inserção Viável do Ponto de Coleta* (S_k, i, req) **então**
- 5 $j \leftarrow |S_k| - 1$
- 6 **enquanto** $j \neq i$ **faça**
- 7 **se** *Inserção Viável do Ponto de Entrega* (S_k, j, req) **então**
- 8 **retorna** *True*
- 9 $j \leftarrow j - 1$
- 10 $i \leftarrow i - 1$
- 11 **retorna** *False*

4.3.3.2 Operador de Mutação 2

A utilização do operador de mutação 2 se dá pelo fato do mesmo ser simples e demandar pouco tempo de execução. O procedimento inicia selecionando de forma randômica uma rota $r1$ e uma requisição req atendida pela mesma. Posteriormente, é criada uma nova rota para atender req . Este procedimento introduz diversidade, entretanto ao contrário do outro operador de mutação sempre aumenta em uma unidade o número de veículos utilizados. O Algoritmo 12 explicita o operador de mutação.

Algoritmo 12: MUTAÇÃO 2

Entrada: S

Saída: S

1 **início**

2 $r1 \leftarrow$ Seleciona Randomicamente uma Rota (S);

3 $req \leftarrow$ Seleciona Randomicamente e Remove Requisição ($S, r1$);

4 Criar Nova Rota (S, req);

5 **retorna** S

4.4 Abordagens para Resolução de um Serviço DRT Dinâmico

Esta seção descreve as abordagens propostas para otimização de um serviço DRT que opera de forma dinâmica. Inicialmente é descrita a forma como o serviço DRT abordado foi adaptado, para que, dado um conjunto de requisições, seja simulado um serviço DRT dinâmico. Em seguida, as abordagens propostas são apresentadas.

4.4.1 Simulação do serviço DRT Dinâmico

Como já mencionado, um serviço DRT que opera de forma dinâmica deve ser capaz de lidar com novas requisições que chegam gradualmente ao longo da operação do serviço. As rotas que ainda não foram iniciadas ou que já estão em processo de atendimento devem ser adaptadas em tempo real para atender as novas solicitações de transporte. Caso não seja possível alocar de forma viável as novas requisições nas rotas já existentes, outros veículos são utilizados para atendê-las.

A partir destas informações, a natureza dinâmica é introduzida no serviço DRT abordado da seguinte forma: dado um conjunto com n requisições, é construída uma lista com todas, a qual é ordenada em ordem crescente de acordo com o horário desejado

de coleta h_{i+} informado por cada usuário ao solicitar o transporte. Para simular um serviço dinâmico a execução da abordagem é dividida em várias iterações, que lidam com novas requisições e com todas as outras tratadas nas iterações anteriores de forma acumulativa.

O processo de otimização inicia com as m_1 primeiras requisições da lista ordenada, sendo m (parâmetro de entrada) um vetor que armazena a quantidade de requisições que estarão disponíveis em cada iteração. Na segunda iteração, a otimização é realizada para as $m_1 + m_2$ primeiras requisições. Na terceira para $m_1 + m_2 + m_3$ primeiras requisições e assim por diante até atribuir todas as n requisições à uma rota. A inclusão das novas requisições é sempre realizada de forma acumulativa, onde a iteração corrente lida com todas as requisições disponíveis nas iterações anteriores somadas às novas requisições recebidas. A cada iteração, o algoritmo retorna a melhor solução encontrada para as requisições disponibilizadas até o momento. Durante a simulação do ambiente dinâmico, as requisições já disponíveis são divididas em 3 status, apresentados abaixo:

1. Requisição não atendida.
2. Requisição atendida mas não finalizada (passageiro a bordo do veículo).
3. Requisição atendida e finalizada (passageiro entregue).

Na primeira iteração do algoritmo todas as requisições disponíveis são classificadas como não atendidas. De forma a simular um serviço DRT real, este conjunto de requisições são solicitações de transporte realizadas antes do início do processo de planejamento das rotas. Já as demais requisições, que se tornam disponíveis nas iterações posteriores, simulam a chegada de novas demandas e a necessidade de replanejamento em tempo real.

4.4.1.1 Exemplo da Simulação do Contexto Dinâmico

Para exemplificar como é realizado a simulação de um serviço DRT dinâmico é utilizado como exemplo um pequeno conjunto de requisições. A Tabela 4.2 apresenta um conjunto R com $n = 20$ requisições ordenadas de acordo com os respectivos h_{i+} . Das vinte requisições, dez estarão disponíveis antes do início do processo de planejamento das rotas e as outras dez se tornarão disponíveis posteriormente, sendo assim, $m_1 = m_2 = 10$. O h_{i+} da m -ésima requisição (hachurado na Tabela 4.2) é definido como o horário da chegada das m novas requisições. Os IDs estão fora de ordem devido à ordenação.

Tabela 4.2. Exemplo de um Conjunto de Requisições, Ordenadas por Horário de Coleta; em Destaque o Horário de Chegada das 10 Requisições Seguintes.

ID	Origem	Destino	q_i	h_{i+}	h'_{i+}	h_{i-}	h'_{i-}
2	132	47	1	29160	29309,6	29908	30282
19	1109918	1109789	1	29160	29221,2	29466	29619
17	1110044	1109944	1	29280	29380,0	29780	30030
15	1109885	141	1	30000	30102,0	30510	30765
11	1110116	1110039	1	30180	30284,8	30704	30966
1	131	46	1	30240	30370,0	30890	31215
10	1110116	1110023	1	30300	30364,4	30622	30783
0	129	1110532	1	30360	30482,8	30974	31281
9	1109807	1109789	1	30480	30589,6	31028	31302
14	1110646	1109789	1	30540	30635,2	31016	31254
13	153	1110033	1	30660	30766,8	31194	31461
8	1110057	1109962	1	30660	30741,6	31068	31272
5	137	1110033	1	30960	31085,6	31588	31902
4	1109799	1110040	1	31020	31052,0	31180	31260
3	132	1110600	1	31080	31145,6	31408	31572
6	1109793	1109993	1	31260	31355,6	31738	31977
7	1109807	1109944	1	31440	31525,2	31866	32079
18	2344054	1109814	1	31680	31736,0	31960	32100
16	155	1109960	1	32160	32266,4	32692	32958
12	2263832	1109944	1	32280	32369,6	32728	32952

Na primeira iteração, após a execução em sequência dos dois modelos, é obtida uma solução na qual os horários de visita de cada uma das m_1 primeiras requisições estão definidos. Utilizando o horário de chegada da nova demanda como base, é possível atualizar o status destas requisições. A Tabela 4.3 apresenta os horários de visita obtidos na primeira iteração, sendo que os horários hachurados são aqueles que aconteceram antes ou no momento da chegada da nova demanda, consequentemente fixados e mantidos inalterados nas próximas iterações. Os horários não hachurados representam as visitas (embarque ou desembarque de passageiros) que ainda não foram realizadas até o momento da chegada das novas requisições, sendo assim suscetíveis a alterações durante os planejamentos posteriores. A possibilidade de alterar estes horários de atendimento introduz flexibilidade no processo de otimização dos atendimentos das novas requisições. Por exemplo: em diversas situações, um veículo pode alterar sua rota original para possibilitar o atendimento de outros passageiros que não estavam previamente alocados. Embora seja natural a realização de desvios, os mesmos são permitidos, apenas quando os horários de embarque e desembarque dos passageiros já atribuídos à rota não excedem os limites impostos.

Tabela 4.3. Horários de Coleta e Entrega Definidos pela 1^o Iteração do Algoritmo; Horários Hachurados são Fixos (Inalteráveis) na 2^o Iteração.

ID	Coleta	Entrega
2	29160,0	29911,0
19	29221,2	29530,2
17	29353,0	29856,0
15	30000,0	30513,0
11	30284,8	30946,0
1	30301,0	31007,0
10	30300,0	30625,0
0	30360,0	31130,0
9	30480,0	31031,0
14	30540,0	31019,0

4.4.2 Abordagem Baseada em Métodos Exatos

A abordagem proposta para tratar o serviço DRT operacionalizado na forma dinâmico é composta por duas etapas. Na primeira etapa é resolvido o modelo de programação linear mista 1 (minimiza número de veículos) e na segunda etapa o modelo 2 (minimiza duração total das rotas), os quais já foram apresentados. Em ambas etapas, a resolução busca a solução ótima. Entretanto, em algumas situações, esta tarefa é impossibilitada devido ao limite máximo de tempo de execução imposto. Por meio desta abordagem, a otimização do serviço é realizada de forma mono-objetiva, ou seja, em cada etapa é otimizado somente um objetivo.

O modelo 1 é executado com o intuito de encontrar o número mínimo de veículos necessários para atender o conjunto de requisições disponíveis no momento. Em seguida, dado o número de veículos obtido na primeira etapa, o modelo 2 é executado minimizando a duração total das rotas. Como parâmetro, são informados o conjunto de requisições R , a cardinalidade n do conjunto $|R|$ e o vetor m que armazena a quantidade de novas requisições que surgirão em cada iteração do planejamento das rotas. O Algoritmo 13 descreve a estrutura da abordagem proposta.

Inicialmente, os dois modelos são executados para determinar o número de veículos necessários para atender as m_1 requisições disponíveis e para este número mínimo de veículos definir a duração total mínima das rotas para atender à todas requisições. Na segunda iteração, o número de requisições disponíveis é incrementado em m_2 unidades. Para atualizar o status das requisições disponíveis na iteração anterior ($i - 1$), o horário desejado de coleta mais tarde dentre todas as requisições disponíveis no momento é definido como o horário de chegada das m_i novas requisições e a solução obtida por meio da execução do segundo modelo na iteração anterior é utilizada para verificação

Algoritmo 13: ABORDAGEM BASEADA EM MÉTODOS EXATOS

Entrada: R, n, m **Saída:** sol

```

1 início
2    $nReq \leftarrow m$ 
3    $sol \leftarrow null$ 
4    $i \leftarrow 1$ 
5   enquanto  $nReq \neq n$  faça
6      $v \leftarrow \text{modelo1}(R, nReq, sol);$ 
7      $sol \leftarrow \text{modelo2}(R, nReq, sol, v);$ 
8      $i \leftarrow i + 1$ 
9      $nReq \leftarrow nReq + m_i$ 
10 retorna  $sol$ 

```

dos horários de visita definidos. Assim, todas as requisições que a coleta já ocorreu mas que ainda não foram entregues são definidas com o status (2) e todas em que a coleta e entrega já foram efetuadas são definidas com o status (3). As demais permanecem com o status (1). A partir da segunda iteração, as requisições que passarem do status (1) para o status (2) terão o horário do atendimento no qual foi realizado o embarque do passageiro fixado. Nas que passaram para o status (3), tanto o horário do atendimento do embarque quando do desembarque serão fixados, não podendo mais serem alterados nas próximas iterações.

4.4.3 Abordagem Heurística Proposta

A abordagem heurística proposta para otimizar o serviço DRT abordado, quando operacionalizado na forma dinâmica é composta por duas fases. A primeira fase realiza o planejamento das rotas e dos horários de atendimentos, para o conjunto de requisições que são enviadas antes do início da operação do serviço. O procedimento de otimização nesta fase é conduzido por um MOEA. A segunda fase viabiliza o replanejamento das rotas sempre que novas requisições são recebidas ao longo da operação do serviço. Note, que a primeira fase da abordagem heurística é equivalente à primeira iteração da abordagem baseada em métodos exatos e a segunda fase é constituída por todas iterações realizadas a partir da segunda iteração. A estrutura da abordagem heurística proposta é detalhada na Figura 4.1.

Diferentemente da abordagem baseada em métodos exatos, a abordagem heurística foi projetada para otimizar simultaneamente todos os três objetivos apresentados. Para isso, um conjunto de soluções são analisadas durante o procedimento. A seguir, as duas fases que compõem a abordagem heurística serão explicitadas.

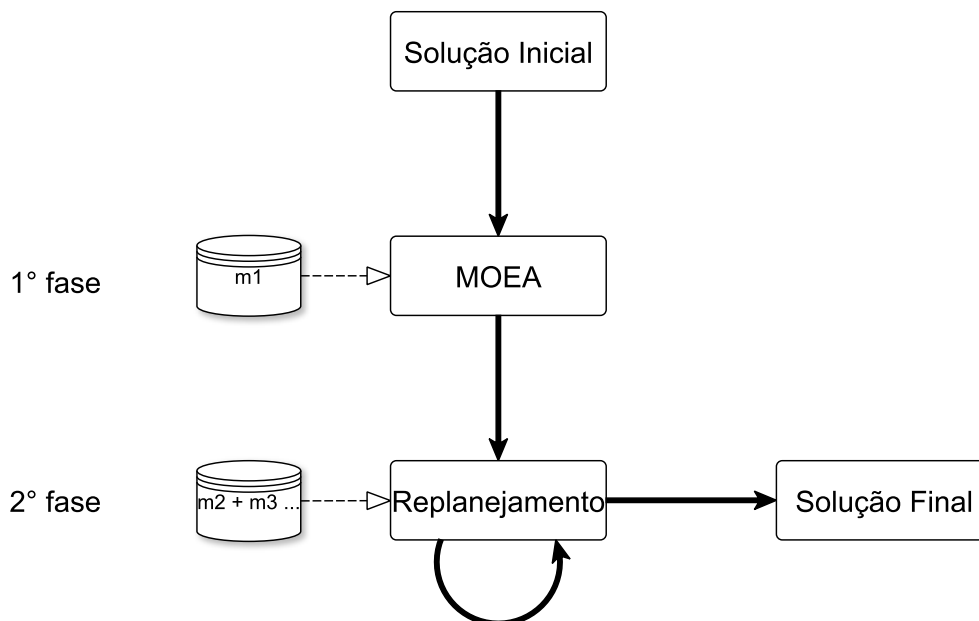


Figura 4.1. Estrutura da Abordagem Heurística

4.4.3.1 Fase 1

Dado um conjunto de requisições de transporte recebidas com certa antecedência (m_1 primeiras requisições), para otimizar o planejamento das rotas, foi utilizado um MOEA, o qual considera os três objetivos (descritos na descrição do problema) durante o processo de otimização. O algoritmo retorna um conjunto de soluções não-dominadas, das quais apenas uma será selecionada para a segunda fase da abordagem proposta.

O procedimento de seleção da solução que será utilizada na segunda fase baseia-se na forma que o algoritmo IBEA avalia a aptidão das soluções e depois constrói a nova população (apresentado na seção 2.2.2). Inicialmente é calculado o valor de aptidão de todas as soluções, o qual representa por meio de um valor numérico o quanto cada solução é importante para o conjunto. Em seguida, de forma iterativa até que a quantidade de soluções definida seja satisfeita, a solução com menor valor de aptidão é removida e o valor de aptidão das demais é atualizado. Para esta abordagem, apenas uma solução é necessária para a continuidade do processo de otimização, pois na prática apenas uma é efetuada. A primeira fase é finalizada quando restar apenas uma solução, a qual será ponto de partida da segunda fase.

4.4.3.2 Fase 2

A segunda fase da abordagem heurística proposta é responsável por realizar o replanejamento das rotas sempre que o serviço DRT receba novas requisições. Exceto pela primeira iteração, esta fase é constituída por todas as demais iterações realizadas pela abordagem. Em cada uma dessas iterações, as m_i requisições, as quais representam todas aquelas que se tornaram disponíveis na iteração i , são alocadas em alguma rota já existente ou caso não seja possível, em uma nova rota. Este procedimento realiza inserções que resultam no menor acréscimo na duração total da solução. Para que seja explorado um maior número de possibilidades de inserções, um conjunto de soluções é construído por meio de sequências diferentes de inserções das m_i requisições. Devido ao fato de que na prática apenas uma solução deve ser efetuada, para selecionar uma dentre o conjunto de soluções construídas é realizado o mesmo procedimento de seleção que foi utilizado na primeira fase. Desta forma é definida uma única solução, a qual será utilizada na próxima iteração, caso seja necessário (surja uma nova demanda).

Capítulo 5

Resultados

Neste capítulo são apresentados o planejamento dos experimentos computacionais e os resultados obtidos pelas abordagens propostas. Inicialmente são explicitados o ambiente computacional, os conjuntos de instâncias de testes utilizados, os quais são comuns para todas as abordagens analisadas e o conjunto de testes estatísticos utilizados para validação dos resultados. As apresentações e análises dos resultados são divididas em duas partes, que são: resultados obtidos pelas abordagens propostas para resolução do serviço DRT estático e resultados provenientes das abordagens direcionadas para o serviço DRT dinâmico.

5.1 Ambiente Computacional

Para a realização dos testes computacionais foi utilizado um computador Intel Core I5 3.40 GHz, com 16GB RAM, executando o sistema operacional *Windows 7*.

Os métodos propostos foram codificados utilizando a linguagem C++. As abordagens baseadas em métodos exatos foram testadas utilizando o software CPLEX12.6 e implementadas utilizando a biblioteca *Concert C++*.

5.2 Conjunto de Instâncias de Teste

Para validação das abordagens propostas foram utilizados dois conjuntos de instâncias de teste introduzidos por [Chevrier et al., 2010] e também utilizados em [Chevrier et al., 2012]. O primeiro conjunto, chamado “*Rnd100*” é composto por 10 instâncias com uma distribuição quase homogênea de clientes (janelas de tempo conflitantes), as quais contêm 100 pedidos gerados aleatoriamente. O segundo conjunto, intitulado

“*Gravit100*” contém 10 instâncias com uma distribuição heterogênea de clientes, que têm 100 pedidos gerados usando um modelo geográfico de pessoas ou de fluxos de mercadorias.

5.3 Conjuntos de Testes Estatísticos

Para validar os resultados obtidos, experimentos estatísticos foram realizados por meio da Análise de Variância (ANOVA) paramétrica ou do teste não paramétrico *Kruskal-Wallis*, utilizado quando os dados não atingem pelo menos um dos três principais pressupostos da análise ANOVA, os quais são: amostras independentes, distribuição normal dos resíduos e homogeneidade das variâncias. Estes testes são acompanhados do teste de *Scott-Knott*. Para realização da análise ANOVA, antes foram verificadas a normalidade dos resíduos e a homogeneidade das variâncias por meio dos teste de *Shapiro-Wilk* e de *Levene*, respectivamente. O primeiro pressuposto citado é garantido no planejamento dos experimentos. A ferramenta estatística utilizada para realização dos experimentos estatísticos foi o R.

O teste de *Shapiro-Wilk* verifica, com certo grau de confiança (definido em 99%), se o conjunto de dados analisado segue a distribuição normal. A verificação da normalidade realizada sobre os resíduos retorna um valor intitulado P-valor, o qual indica quando acima de 0.01, que não se rejeita a hipótese de que os resíduos seguem uma distribuição normal.

A igualdade ou também homogeneidade das variâncias é verificada por meio do teste de *Levene*, que também retorna o P-valor. Quando acima de 0.01, o P-valor indica que não há diferença estatisticamente significativa entre os desvios padrões e portanto não se rejeita a hipótese de igualdade das variâncias. O nível de confiança definido para este teste foi de 99%.

Quando os três pressupostos da ANOVA são atingidos e a mesma leva a rejeição da hipótese nula (P-valor < 0.01), a qual afirma que não há diferença significativa entre as médias, pode-se concluir que as médias analisadas diferem significativamente.

O teste de *Kruskal-Wallis* funciona de forma análoga à análise ANOVA, entretanto não possui restrições para ser considerado. Após a rejeição da hipótese nula, seja qual método utilizado, o teste de *Scott-Knott* é aplicado para apontar qual ou quais tratamentos são superiores aos demais. Este teste foi realizado com nível de significância de 99%.

5.4 Dados de Entrada

Para a execução das abordagens, foram definidos alguns parâmetros de entrada:

- Para simular uma frota composta por veículos comuns, todos terão capacidade máxima para quatro pessoas ($Q = 4$);
- Para as abordagens que otimizam o serviço DRT para o caso estático não foi considerado duração de serviço ($d_i = 0, \forall i \in V^+ \cup V^-$). Baseado em [Chevrier et al., 2012];
- Para as abordagens que otimizam o serviço DRT para o caso dinâmico, a duração do serviço em todos os pontos de visita foi definido como sendo 3 ($d_i = 3, \forall i \in V^+ \cup V^-$);
- As constantes de janela de tempo e de relaxação são estabelecidas em 0.2 e 1.5, respectivamente ($k_w = 0.2$ e $K_r = 1.5$). Baseado em [Chevrier et al., 2012].

5.5 Abordagens Aplicadas ao Serviço DRT Estático

Esta seção apresenta os experimentos computacionais que envolvem as abordagens propostas para resolução do caso estático do serviço DRT abordado. O desempenho dos algoritmos NSGA-II, IBEA e SPEA2 é avaliado para 6 combinações de operadores de variação (operadores de recombinação 2, 3 e 4 e operadores de mutação 1 e 2). Para comparação, os resultados obtidos pela abordagem proposta em [Viana et al., 2015] composta pelo algoritmo NSGA-II e operadores de recombinação 1 e de mutação 1 são incluídos na análise. No total 19 abordagens foram avaliadas.

Com o intuito de realizar uma comparação justa das abordagens avaliadas, primeiro foi realizado um estudo, no qual são encontradas as configurações de parâmetros que permitem cada uma obter seu melhor desempenho. Os experimentos de calibração, assim como as análises estatísticas utilizadas para validações dos resultados são apresentados no Apêndice A. A Tabela 5.1 aponta as abordagens avaliadas e também os tratamentos definidos para cada uma após os experimentos de calibração. As 19 abordagens são identificadas da seguinte forma: "algoritmo-C(operador de recombinação)M(operador de mutação)".

Após a fase de calibração, para analisar o comportamento das abordagens, cada uma foi executada 10 vezes utilizando a configuração de parâmetros ideal. Note, que a avaliação das 19 abordagens totaliza 190 conjuntos de pontos não-dominados para

cada instância de teste utilizada. O critério de parada estabelecido foi 1 minuto de execução e o tamanho da população foi fixado em 100 soluções.

Tabela 5.1. Conjunto de Abordagens Avaliadas

Abordagem	Tratamento Utilizado
IBEA-C2M1	$P_c = 1.0$ e $P_m = 0.8$
IBEA-C2M2	$P_c = 1.0$ e $P_m = 0.2$
IBEA-C3M1	$P_c = 0.2$ e $P_m = 1.0$
IBEA-C3M2	$P_c = 0.8$ e $P_m = 1.0$
IBEA-C4M1	$P_c = 1.0$ e $P_m = 1.0$
IBEA-C4M2	$P_c = 1.0$ e $P_m = 0.2$
NSGA2-C1M1	$P_c = 0.2$ e $P_m = 1.0$
NSGA2-C2M1	$P_c = 0.8$ e $P_m = 1.0$
NSGA2-C2M2	$P_c = 1.0$ e $P_m = 0.2$
NSGA2-C3M1	$P_c = 0.2$ e $P_m = 1.0$
NSGA2-C3M2	$P_c = 0.8$ e $P_m = 1.0$
NSGA2-C4M1	$P_c = 1.0$ e $P_m = 1.0$
NSGA2-C4M2	$P_c = 1.0$ e $P_m = 0.2$
SPEA2-C2M1	$P_c = 0.8$ e $P_m = 1.0$
SPEA2-C2M2	$P_c = 0.8$ e $P_m = 1.0$
SPEA2-C3M1	$P_c = 0.2$ e $P_m = 1.0$
SPEA2-C3M2	$P_c = 0.8$ e $P_m = 1.0$
SPEA2-C4M1	$P_c = 1.0$ e $P_m = 0.8$
SPEA2-C4M2	$P_c = 1.0$ e $P_m = 0.2$

Como o conjunto Pareto-ótimo não é conhecido para nenhuma das instâncias de teste utilizadas, a avaliação das abordagens foi realizada por meio de um conjunto referência composto pelos pontos não-dominados encontrados nas 190 execuções. Portanto, para cada instância foi gerado um conjunto referência *Ref*.

A avaliação do desempenho das abordagens propostas foi realizada por meio da versão unária da métrica *Additive ϵ -Indicator*. Para cada instância de teste, o cálculo $I_{\epsilon^+}(A, Ref)$ foi efetuado para os 10 conjuntos aproximados obtidos por cada abordagem, sendo *A* o conjunto analisado e *Ref* o conjunto referência associado a instância em questão.

Para os testes estatísticos, foi realizado um experimento em delineamento inteiramente casualizado com apenas um fator, o qual é o tipo do algoritmo e 19 níveis (abordagens). A análise ANOVA associada a esse tipo de experimento é chamada de análise de variância de simples entrada (no inglês, *One Way ANOVA*), na qual são verificadas diferenças entre os valores das médias de cada nível. Para cada nível são associadas 10 repetições, equivalentes às 10 execuções da abordagem associada ao ní-

vel. O valor médio obtido considerando todas as instâncias é definido como a variável resposta do par abordagem/repetição.

5.5.1 Análise dos Resultados

A primeira análise realizada considera os resultados obtidos pelas abordagens para os dois conjuntos de instâncias de teste utilizados (“*Gravit100*” e “*Rnd100*”). Sendo assim, esta análise fornece uma visão geral do desempenho das abordagens propostas. Para isso, o cálculo da média associada ao par abordagem/repetição abrange as variáveis respostas obtidas para todas instâncias de teste consideradas. De acordo com os resultados dos testes estatísticos, os quais são apresentados na Tabela 5.2, verificou-se que os pressupostos da análise ANOVA foram atingidos (P-valor > 0.01), e a mesma provou que existe diferença significativa entre as médias (P-valor < 0.01).

Tabela 5.2. Resultados dos Testes Estatísticos - Conjuntos “*Gravit100*” e “*Rnd100*”, com Nível de Confiança de 99%

Testes		
Shapiro-Wilk W	Levene	ANOVA
0,5542	0,0989	0,0000

Para descobrir quais abordagens obtiveram os melhores desempenhos foi realizado o teste de *Scott-Knott*, que é apresentado na Tabela 5.3.

Conforme a Tabela 5.3, independente de qual MOEA for utilizado, as abordagens que utilizam o operador de recombinação 4 obtiveram desempenhos superiores em relação às demais abordagens avaliadas, destacando-se as abordagens que usaram os algoritmos NSGA-II e SPEA2 (Grupo n). Entretanto, as médias provenientes destas 4 abordagens não possuem diferenças significativas. No outro extremo da tabela estão as abordagens nas quais foram utilizadas a combinação dos operadores de recombinação 2 e de mutação 2. O baixo desempenho destas abordagens é devido principalmente aos resultados ruins que obtiveram quando aplicadas às instâncias de teste do conjunto “*Rnd100*”, que são caracterizadas pela distribuição homogênea das requisições.

De forma geral, considerando apenas os MOEAs utilizados, é possível constatar que, exceto para a combinação de operadores (C(2)M(2)), o NSGA-II e o SPEA2 obtiveram desempenhos superiores em relação ao IBEA. Além disso, é verificado que a abordagem proposta em [Viana et al., 2015] (NSGA2-C(1)M(1)) obteve desempenho intermediário, sendo superior às abordagens que utilizam os operadores de recombinação 2 e 3 combinados ao operador de mutação 2. A Figura 5.1 apresenta o gráfico *boxplot* (diagrama de caixa e bigode), no qual é possível verificar o desempenho geral e

a variância dos resultados obtidos pelas abordagens propostas. Assim como na Tabela 5.3, o gráfico apresenta a acentuada diferença de desempenho entre as abordagens nas quais foram utilizadas a combinação dos operadores de recombinação 2 e de mutação 2 e as demais abordagens analisadas e também comprova a superioridade das abordagens que utilizam o operador de recombinação 4.

Tabela 5.3. Teste de *Scott-Knott*, com Nível de Confiança de 99% - Conjuntos “*Gravit100*” e “*Rnd100*”

Grupos	Níveis	Médias
a	SPEA2-C(2)M(2)	0,6195
b	IBEA-C(2)M(2)	0,5993
c	NSGA2-C(2)M(2)	0,5781
d	IBEA-C(3)M(2)	0,1738
e	SPEA2-C(3)M(2)	0,1520
f	NSGA2-C(3)M(2)	0,1360
f	NSGA2-C(1)M(1)	0,1339
f	IBEA-C(3)M(1)	0,1315
g	IBEA-C(2)M(1)	0,1190
h	SPEA2-C(3)M(1)	0,1119
i	NSGA2-C(3)M(1)	0,1067
j	SPEA2-C(2)M(1)	0,0999
k	NSGA2-C(2)M(1)	0,0932
l	IBEA-C(4)M(2)	0,0817
m	IBEA-C(4)M(1)	0,0750
n	SPEA2-C(4)M(2)	0,0699
n	NSGA2-C(4)M(1)	0,0686
n	SPEA2-C(4)M(1)	0,0674
n	NSGA2-C(4)M(2)	0,0639

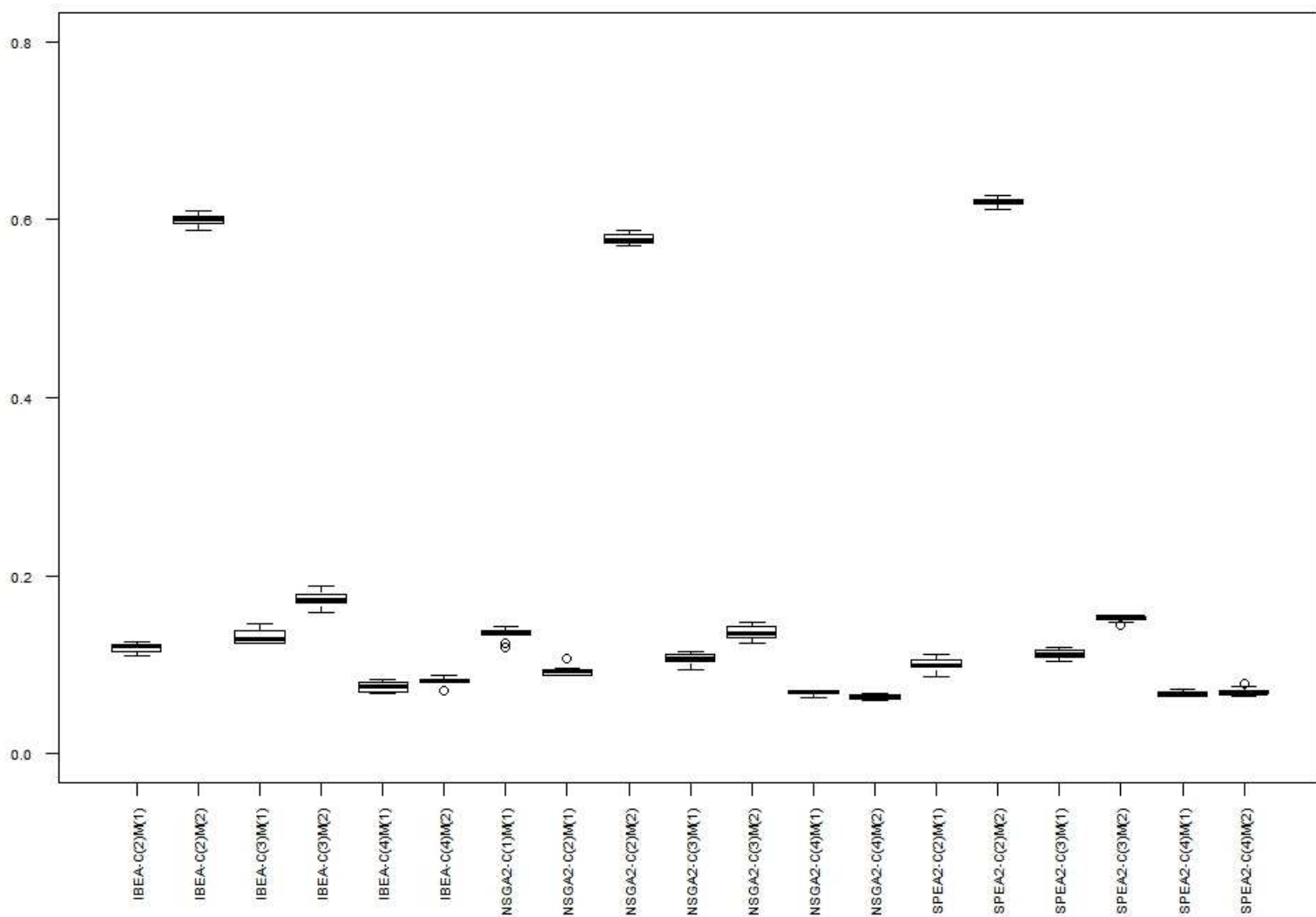


Figura 5.1. Gráfico de Médias - Comparação das Abordagens Desenvolvidas para Resolução do Caso Estático - Conjuntos "Gravit100" e "Rnd100"

Uma vez apresentado o desempenho das abordagens considerando os dois conjuntos de instâncias (desempenho geral), agora serão apresentadas duas análises, nas quais cada um dos dois conjuntos de instâncias é considerado separadamente.

De acordo com a Tabela 5.4, quando aplicados somente aos resultados associados às instâncias do conjunto “*Gravit100*”, os testes estatísticos indicaram que os pressupostos da análise ANOVA são atingidos (P-valor > 0.01), e a mesma provou que existe diferença significativa entre as médias (P-valor < 0.01).

Tabela 5.4. Resultados dos Testes Estatísticos - Conjunto “*Gravit100*”, com Nível de Confiança de 99%

Testes		
Shapiro-Wilk W	Levene	ANOVA
0,5179	0,0345	0,0000

De forma similar à análise anterior, para constatar quais abordagens obtiveram os melhores desempenhos também foi realizado o teste de *Scott-Knott*, o qual é apresentado na Tabela 5.5.

Tabela 5.5. Teste de *Scott-Knott*, com Nível de Confiança de 99% - Conjunto “*Gravit100*”

Grupos	Níveis	Médias
a	IBEA-C(3)M(2)	0,2894
b	SPEA2-C(2)M(2)	0,2390
c	IBEA-C(3)M(1)	0,2075
c	NSGA2-C(3)M(2)	0,2061
c	IBEA-C(2)M(2)	0,1986
c	NSGA2-C(1)M(1)	0,1982
c	SPEA2-C(3)M(2)	0,1960
d	IBEA-C(2)M(1)	0,1708
d	SPEA2-C(3)M(1)	0,1649
d	NSGA2-C(3)M(1)	0,1633
e	NSGA2-C(2)M(2)	0,1562
f	SPEA2-C(2)M(1)	0,1322
f	NSGA2-C(2)M(1)	0,1263
f	IBEA-C(4)M(2)	0,1260
f	IBEA-C(4)M(1)	0,1176
g	NSGA2-C(4)M(1)	0,1006
g	SPEA2-C(4)M(2)	0,0942
g	SPEA2-C(4)M(1)	0,0927
g	NSGA2-C(4)M(2)	0,0861

De acordo com os resultados do teste de *Scott-Knott*, considerando apenas instâncias do conjunto “*Gravit100*”, as abordagens que utilizam o operador de recombinação 4 obtiveram desempenhos superiores em relação às demais abordagens avaliada, prevalecendo a superioridade das abordagens que utilizam os algoritmos NSGA-II e SPEA2 (Grupo g). Comportamento também verificado se considerado o conjunto geral de instâncias na análise.

Outra importante observação está relacionada às abordagens que utilizam a combinação dos operadores de recombinação 2 e de mutação 2. Nesta análise as mesmas não obtiveram médias tão inferiores quanto na análise geral, ou seja, para este conjunto de instâncias de testes estas abordagens conseguem ter um desempenho próximo das demais.

A Figura 5.2 apresenta o gráfico *boxplot*, no qual é possível verificar o desempenho e a variância das abordagens propostas quando utilizado somente o conjunto de instâncias de teste “*Gravit100*”.

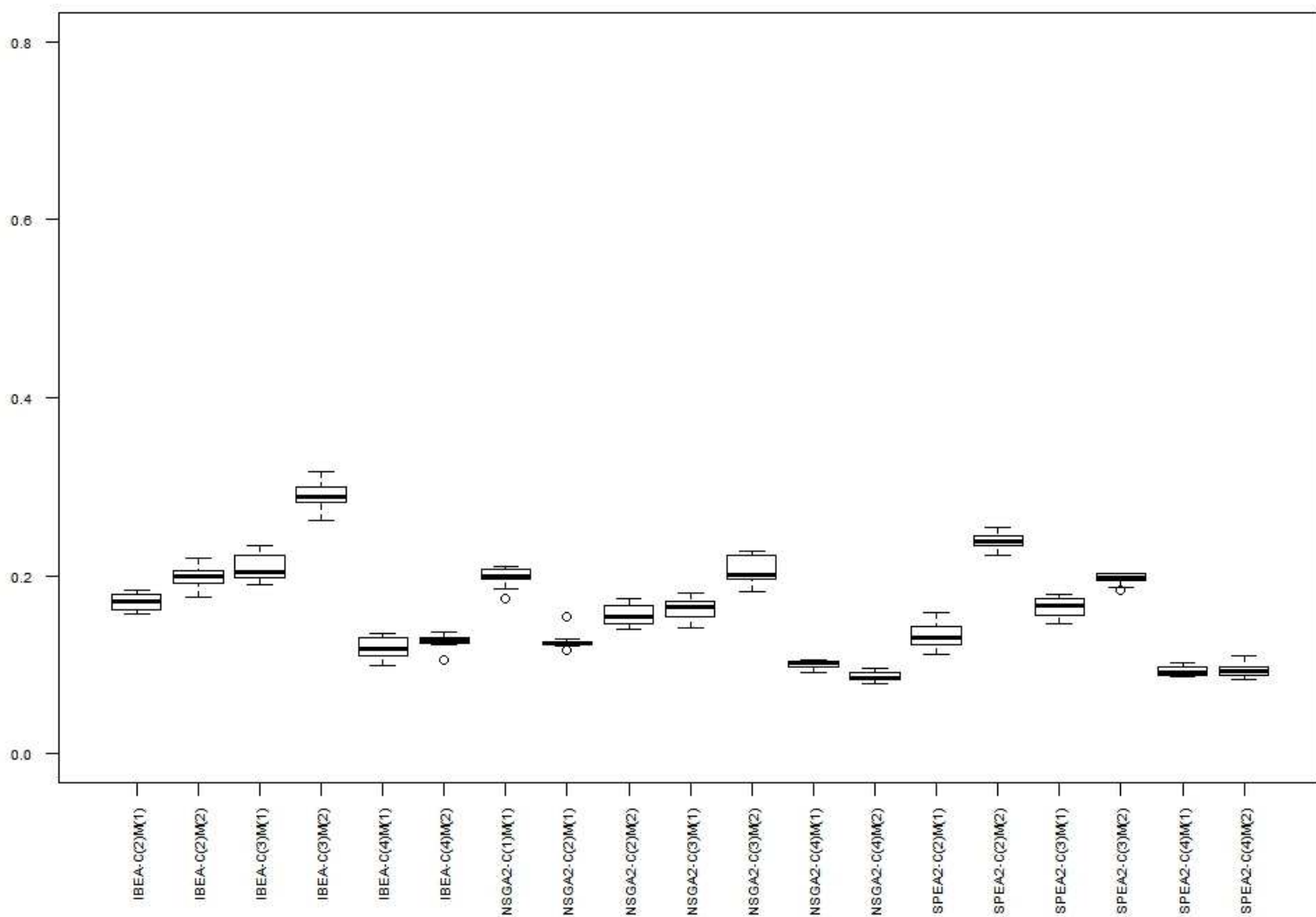


Figura 5.2. Gráfico de Médias - Comparação das Abordagens Desenvolvidas para Resolução do Caso Estático - Conjunto “Gravit100”

Para encerrar as análises individuais, a próxima análise a ser apresentada considera os resultados obtidos pelas abordagens quando utilizado somente o conjunto de instâncias de teste “*Rnd100*”. Conforme apresentado na Tabela 5.6, também para estes dados, os pressupostos da análise ANOVA foram atingidos (P-valor > 0.01), e a mesma provou que existe diferença significativa entre as médias (P-valor < 0.01).

Tabela 5.6. Resultados dos Testes Estatísticos - Conjunto “*Rnd100*”, com Nível de Confiança de 99%

Testes		
Shapiro-Wilk W	Levene	ANOVA
0,0665	0,0155	0,0000

De acordo com os resultados do teste de *Scott-Knott* apresentados na Tabela 5.7, de forma análoga à análise focada no conjunto de instâncias “*Gravit100*”, as abordagens que utilizam o operador de recombinação 4 também obtiveram desempenhos superiores em relação às demais abordagens avaliadas. Entretanto, para este conjunto de instâncias de teste, as abordagens que utilizam o algoritmo IBEA obtiveram resultados melhores do que os apresentados na análise anterior. Acredita-se que esta melhora de desempenho se deve à combinação de características do algoritmo, como por exemplo à ausência de um operador de diversidade, somadas à distribuição homogêneas das requisições. Entretanto, esta suposição não foi comprovada apenas por testes empíricos.

Embora omitido na Tabela 5.7, outro ponto de destaque desta análise é o baixo rendimento das abordagens que utilizam a combinação dos operadores de recombinação 2 e de mutação 2 quando aplicadas a este conjunto de instâncias. A homogeneidade da distribuição das requisições nas instâncias somada às características dos operadores e do procedimento de geração da população inicial resultaram no pior resultado do indicador de qualidade utilizado ($I_{\epsilon^+}(A, Ref) = 1$), em todas as 10 execuções. Foi constatado que, independente do algoritmo utilizado, não houve evolução da população inicial ao longo das gerações, ou seja, para esta situação os operadores de variação não cumpriram seus respectivos papéis. Este comportamento justifica os altos valores de média apresentados na análise geral. Além de valores altos de média, não há variância entre os valores obtidos, por isso os resultados provenientes destas abordagens não foram considerados nos testes estatísticos.

A Figura 5.3 apresenta o gráfico *boxplot*, no qual é possível verificar o desempenho e a variância das abordagens propostas quando utilizado somente o conjunto de instâncias de teste “*Rnd100*”. Os resultados obtidos pelas abordagens que utilizam a

combinação dos operadores de recombinação 2 e de mutação 2 não foram incluídos no gráfico devido aos mesmos motivos apontados anteriormente.

Tabela 5.7. Teste de *Scott-Knott*, com Nível de Confiança de 99% - Conjunto “*Rnd100*”

Grupos	Níveis	Médias
a	SPEA2-C(3)M(2)	0,1080
b	NSGA2-C(1)M(1)	0,0696
b	SPEA2-C(2)M(1)	0,0677
b	IBEA-C(2)M(1)	0,0672
b	NSGA2-C(3)M(2)	0,0660
c	NSGA2-C(2)M(1)	0,0601
c	SPEA2-C(3)M(1)	0,0588
c	IBEA-C(3)M(2)	0,0581
d	IBEA-C(3)M(1)	0,0556
e	NSGA2-C(3)M(1)	0,0501
f	SPEA2-C(4)M(2)	0,0455
g	SPEA2-C(4)M(1)	0,0420
g	NSGA2-C(4)M(2)	0,0417
h	IBEA-C(4)M(2)	0,0373
h	NSGA2-C(4)M(1)	0,0366
i	IBEA-C(4)M(1)	0,0324

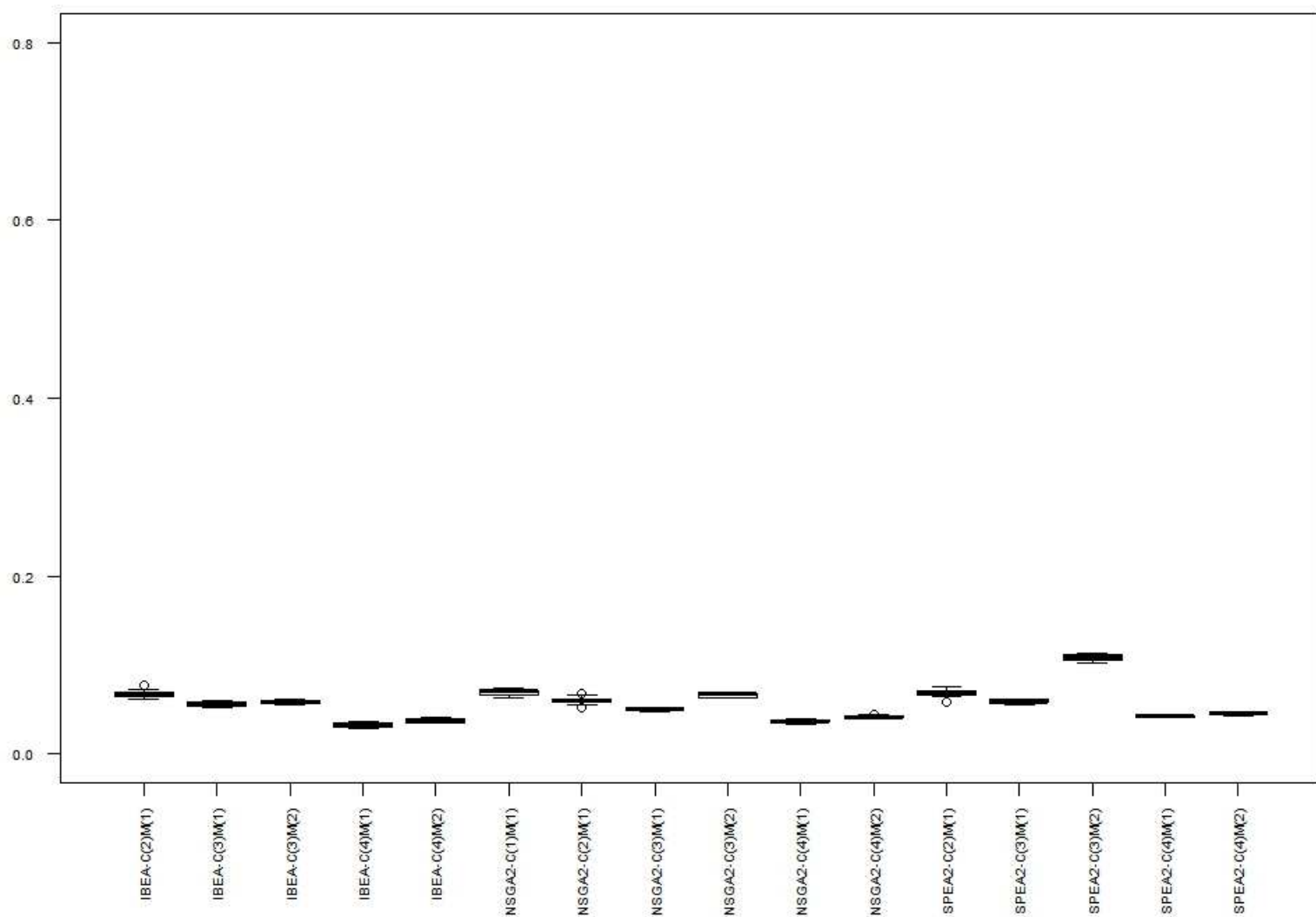


Figura 5.3. Gráfico de Médias - Comparação das Abordagens Desenvolvidas para Resolução do Caso Estático - Conjunto “*Rnd100*”

A Tabela 5.8 apresenta os resultados das três análises explicitadas anteriormente, mas ordenados de acordo com as abordagens utilizadas. Os valores e letras em negrito correspondem às abordagens que obtiveram os melhores desempenho em cada análise. Embora os testes de *Scott-Knott* realizados terem agrupado as abordagens utilizando períodos diferentes em cada análise, é possível constatar que as abordagens compostas pelos operadores de recombinação 4, no geral obtiveram os melhores resultados, independente do operador de mutação utilizado. Além disso, pode-se concluir que para instâncias do conjunto “*Rnd100*”, ou seja, instâncias com distribuição homogênea das requisições, o algoritmo IBEA é superior aos demais. Por outro lado, nas instâncias do conjunto “*Gravit100*” observa-se a superioridade das abordagens que utilizam os algoritmos NSGA-II e SPEA2.

Por meio da análise geral, a qual engloba os dois conjuntos de instâncias de teste, é possível afirmar que para instâncias nas quais não se sabe qual distribuição das requisições foi adotada, as abordagens que utilizam o operador de recombinação 4 são as mais indicadas para serem utilizadas.

Tabela 5.8. Tabela Comparativa Composta por Todas as Análises

Abordagem	Resultados					
	Conjunto “ <i>Gravit100</i> ” [a-g]		Conjunto “ <i>Rnd100</i> ” [a,i]		Conjunto Geral [a-n]	
IBEA-C2M1	0,1708	d	0,0672	b	0,1190	g
IBEA-C2M2	0,1986	c	1,0000	-	0,5993	b
IBEA-C3M1	0,2075	c	0,0556	d	0,1315	f
IBEA-C3M2	0,2894	a	0,0581	c	0,1738	d
IBEA-C4M1	0,1176	f	0,0324	i	0,0750	m
IBEA-C4M2	0,1260	f	0,0373	h	0,0817	l
NSGA2-C1M1	0,1982	c	0,0696	b	0,1339	f
NSGA2-C2M1	0,1263	f	0,0601	c	0,0932	k
NSGA2-C2M2	0,1562	e	1,0000	-	0,5781	c
NSGA2-C3M1	0,1633	d	0,0501	e	0,1067	i
NSGA2-C3M2	0,2061	c	0,0660	b	0,1360	f
NSGA2-C4M1	0,1006	g	0,0366	h	0,0686	n
NSGA2-C4M2	0,0861	g	0,0417	g	0,0639	n
SPEA2-C2M1	0,1322	f	0,0677	b	0,0999	j
SPEA2-C2M2	0,2390	b	1,0000	-	0,6195	a
SPEA2-C3M1	0,1649	d	0,0588	c	0,1119	h
SPEA2-C3M2	0,1960	c	0,1080	a	0,1520	e
SPEA2-C4M1	0,0927	g	0,0420	g	0,0674	n
SPEA2-C4M2	0,0942	g	0,0455	f	0,0699	n

5.6 Abordagens Aplicadas ao Serviço DRT Dinâmico

Nesta seção são apresentados os experimentos computacionais que envolvem as abordagens propostas para resolução do caso dinâmico do serviço DRT abordado.

5.6.1 Considerações Iniciais

Para um serviço DRT dinâmico ser considerado eficiente, além de gerar boas soluções, também é importante que execute o planejamento de forma rápida, pois é operacionalizado em tempo real. Portanto, para as abordagens que otimizam o serviço por meio da resolução de modelos matemáticos, foi definido um tempo máximo de execução para cada modelo de 2 minutos para as instâncias do conjunto “*Gravit100*” e de 4 minutos para as instâncias do conjunto “*Rnd100*”. O limite de execução definido para o segundo conjunto de instâncias é maior, devido a maior dificuldade de resolução observada durante os experimentos computacionais. Nas situações em que o limite de execução é atingido, a solução corrente, mesmo não sendo ótima, é retornada. Para as abordagens heurísticas, a primeira fase foi executada durante 10 segundos e a segunda fase durante o tempo necessário (sempre realizada em poucos segundos).

Nos experimentos computacionais, foi definido que em cada iteração, a nova demanda é composta por 10 requisições, sendo assim, em cada instância de teste ocorrerão 10 iterações. Não foram usados valores maiores devido a dificuldade de se solucionar os modelos na otimalidade com um maior número de requisições disponíveis por iteração.

A abordagem que utiliza o algoritmo NSGA-II com a combinação de operadores de recombinação 4 e de mutação 2 foi definida para compor a primeira fase da heurística proposta para resolver o caso dinâmico. Esta escolha, se deve ao fato desta abordagem ter obtido um bom desempenho no geral quando otimizado o serviço DRT na forma estática.

5.6.2 Método de Avaliação de Desempenho

Para validar o desempenho das abordagens propostas quanto a capacidade de replanejamento em ambientes dinâmicos, foi desenvolvido um algoritmo que, da mesma forma que a abordagem baseada em métodos exatos, executa em cada iteração os dois modelos matemáticos. Entretanto, cada iteração é independente das anteriores. Portanto, não é realizado o aproveitamento das rotas construídas para atender demandas passadas. Ao final da execução do algoritmo, as soluções obtidas em cada iteração são concate-

nadas com o intuito de formar uma única solução composta por todas as requisições da instância de teste. O propósito da construção desta solução é simular parcialmente um serviço DRT dinâmico, no qual requisições chegam ao longo do planejamento das rotas, mas não é realizado o aproveitamento de rotas já construídas para atender requisições de iterações anteriores, ou seja, a otimização é processada de forma estática. Desta forma, pode ser medido o ganho proporcionado pelas abordagens propostas, que aproveitam as rotas que já estão em andamento.

Para efeito de comparação e avaliação de comportamento, a abordagem heurística proposta também é avaliada quando utilizado um procedimento de seleção no qual, em cada iteração, a solução selecionada é aquela que utiliza menos veículos e caso houver empate, a que possuir menor duração total é adotada na próxima iteração. Note, que esta abordagem adota uma seleção gulosa.

5.6.3 Resultados dos Experimentos Computacionais

Para iniciar a apresentação dos resultados, as abordagens que otimizam o serviço DRT por meio da resolução dos modelos matemáticos são comparadas individualmente. As Tabelas 5.9 e 5.10 apresentam os resultados obtidos pela abordagem baseada em métodos exatos (EcA) para os conjuntos “*Gravit100*” e “*Rnd100*”, respectivamente. Para cada instância são detalhados os valores: k_{min} (F1), duração total das rotas (F2), além dos *GAP*'s fornecidos pelo CPLEX e tempos de execução de cada modelo em segundos.

Nessas tabelas, quando o valor de *GAP* não é zero, a solução ótima para a iteração em questão, ou não foi encontrada ou não pode ser provada dentro do tempo limite. Um “-” nos valores de *F2* e *GAP* indica que nenhuma solução viável foi encontrada dentro do tempo de execução limite do modelo em questão. Quando isso acontece não são mostrados os resultados das demais iterações, porque cada iteração depende de uma solução viável da iteração anterior. Para os casos, nos quais a resolução do modelo que minimiza o número de veículos necessários não encontrou solução viável dentro do prazo de execução, foi definido que o número mínimo de veículos é a soma entre o valor encontrado na iteração anterior mais o número de novas requisições que foram recebidas na iteração corrente. Assim, considera-se o pior caso, no qual é alocado um veículo para cada nova requisição. Conforme a Tabela 5.10, esta situação ocorreu na resolução das instâncias de teste *Rnd100_6* e *Rnd100_9* nas iterações 7 e 4, respectivamente.

De acordo com a Tabela 5.9, na grande maioria das execuções, tanto do primeiro quanto do segundo modelo, o tempo de resolução foi da ordem de poucos segundos. Nota-se que em apenas uma iteração da instância “*Gravit100_5*”, dentre todas as iterações apresentadas na tabela, não foi possível achar a solução ótima dentro dos 2

minutos de execução. Em todas as instâncias o número mínimo de veículos encontrado está abaixo de 32, o que mostra que a abordagem dinâmica satisfaz bem o seu propósito, agrupando o maior número possível de requisições em uma mesma rota de forma viável. Entretanto, as mesmas observações não podem ser feitas em relação aos resultados apresentados pela Tabela 5.10. Devido a distribuição homogênea das janelas de tempo dos clientes nas instâncias do conjunto “*Rnd100*”, o algoritmo não obteve soluções ótimas na mesma frequência que encontrou para as instâncias do conjunto “*Gravit100*” e para todas as instâncias o número mínimo de veículos foi próximo do número total de requisições. Para a instância de teste *Rnd100_2*, o segundo modelo não conseguiu encontrar solução viável dentro do prazo máximo estipulado de 4 minutos.

As Tabelas 5.11 e 5.13 mostram a quantidade de veículos necessários nas soluções encontradas por cada abordagem, para os conjuntos de instâncias de teste “*Gravit100*” e “*Rnd100*”, respectivamente. A abordagem baseada em métodos exatos, a abordagem baseada em métodos exatos a qual não aproveita rotas já existentes durante o processo de otimização, a heurística com seleção multi-objetivo e a heurística com seleção gulosa são representadas na tabela pelas siglas EcA, EsA, HMO e HG, respectivamente. Os valores relacionados às abordagens HMO e HG são médias dos resultados obtidos em 10 execuções.

Note que para as instâncias “*Gravit100*” houve uma grande economia de veículos. Para algumas instâncias o número de veículos utilizados se estabilizou após algumas iterações. Por exemplo, na instância *Gravit100_8*, enquanto a abordagem na qual a otimização é processada de forma estática usa 28 veículos na 3ª iteração, essa quantidade foi atingida na abordagem proposta apenas da 4ª iteração, quando já existiam 40 requisições. E essa quantidade, 28 veículos, se manteve até o final da simulação, contemplando todas as 100 requisições. O gráfico da Figura 5.4 ilustra o aumento do número de veículos a cada 10 requisições, para as abordagens EcA e EsA utilizando a instância *Gravit100_0*. Percebe-se que a abordagem baseada em métodos exatos consegue acomodar muitas das novas requisições nos veículos que já estão atendendo as requisições anteriores, provando ser vantajoso realizar o reaproveitamento de veículos.

Entretanto, para o conjunto de instâncias “*Rnd100*” a economia foi mínima. O gráfico da Figura 5.5 mostra o resultado para a instância *Rnd100_0*. Nestas instâncias as requisições possuem horários bastante homogêneos, o que dificulta o compartilhamento de veículos. Na instância *Rnd100_4*, por exemplo, o número de veículos obtidos pelas abordagens EcA e EsA foram exatamente os mesmos.

De forma semelhante, as Tabelas 5.12 e 5.14 comparam as abordagens, mas desta vez em relação a duração total das soluções em cada iteração. De forma análoga às tabelas que apresentam os resultados associados ao número de veículos, os valores re-

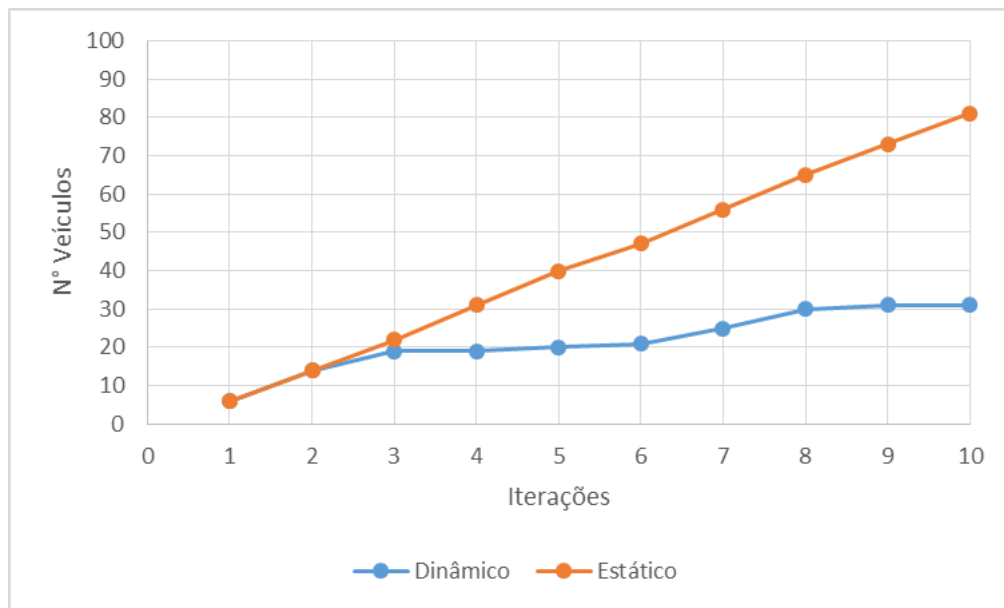


Figura 5.4. Comparação do Aumento do Número de Veículos com a Chegada de Novas Requisições para a Instância Gravit100_0

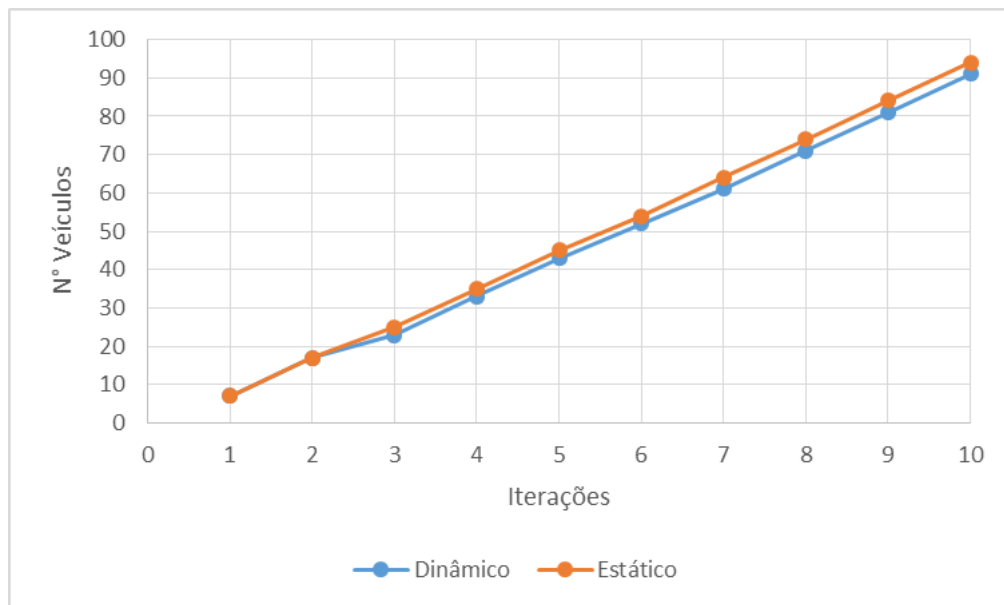


Figura 5.5. Comparação do Aumento do Número de Veículos com a Chegada de Novas Requisições para a Instância Rnd100_0

lacionados às abordagens HMO e HG são médias de 10 execuções. Por meio destas tabelas é possível perceber que também para este objetivo a abordagem EcA, a qual permite o aproveitamento das rotas ao longo das iterações atingiu resultados com qualidade superior em relação à abordagem que não permite. Em ambos conjuntos de

instância, as durações totais quando há o aproveitamento obtiveram valores inferiores, provando a efetividade da abordagem também para este objetivo.

Considerando as abordagens HMO e HG na análise comparativa, quando avaliado o número de veículos utilizados pelas soluções obtidas para as instâncias do conjunto “*Gravit100*”, tanto a abordagem que utiliza seleção multi-objetiva quando a que utiliza seleção gulosa obtiveram desempenho próximo ou equivalente à abordagem EcA. Em 50% das instâncias deste conjunto, pelo menos uma das duas heurísticas (HMO e HG) obtiveram valores inferiores à abordagem EcA. Para estas situações nas quais as heurísticas obtiveram valores de melhor qualidade que a abordagem EcA, mesmo quando os gaps obtidos pela mesma sejam zero, deve-se ao fato de não procurar uma solução ótima global, a otimalidade está relacionada à iteração. Para instâncias de teste do conjunto “*Rnd100*” somente na instância *Rnd100_4* pelo menos uma das abordagens heurísticas obteve resultado equivalente à EcA. Em 9 das 10 instâncias deste conjunto a abordagem heurística HG obteve resultados superiores aos obtidos pela HMO. E a única situação na qual não foi melhor houve empate. Analisando todas as iterações nessas 9 instâncias, pode-se perceber que a diferença visualizada na solução final ocorre desde a primeira iteração. Portanto, a seleção gulosa da abordagem HG permite à mesma obter resultados melhores dos que os obtidos pela heurística HMO, na qual a seleção é realizada considerando os três objetivos.

Quanto ao objetivo que minimiza a duração total das rotas, para o conjunto “*Gravit100*” as durações obtidas pelas abordagens HMO e HG foram na maioria superiores se comparadas às obtidas pela abordagem EcA, mas conforme pode ser visto na tabela 5.12, as diferenças são pequenas. Para as instâncias *Gravit100_3*, *Gravit100_6* e *Gravit100_8* as abordagens HMO e HG obtiveram resultados finais com melhor qualidade em relação aos obtidos pela abordagem EcA. Entretanto para as instâncias do conjunto “*Rnd100*” não houve casos, nos quais a abordagem EcA foi superada, apesar da diferença dos valores obtidos não ser elevada.

Tabela 5.9. Resultados Obtidos pela Abordagem EcA para Cada Instância do Conjunto “*Gravit100*”

ITERACÕES	1	2	3	4	5	6	7	8	9	10
Gravit100_0										
F1	6	14	19	19	20	21	25	30	31	31
F2	10301,6	22455,6	29698	36068	43772	50363	58651	69326	75986	82810
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	6,3	2,0	0,9	1,3	2,5	4,3	6,3	9,0	18,1	36,7
Tempo2	1,2	1,1	0,8	0,6	0,9	1,3	2,4	3,2	4,2	10,9
Gravit100_1										
F1	8	14	15	21	26	26	26	26	28	30
F2	9091,2	18341,2	27367,2	38721,8	49645,8	56922,8	64941,8	74127,0	82847,0	92156,0
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,3	0,7	1,2	1,5	2,7	4,6	6,8	8,1	16,8	35,5
Tempo2	0,5	0,8	0,5	1,0	1,9	1,9	3,0	2,5	4,6	10,5
Gravit100_2										
F1	9	17	22	23	23	26	26	26	26	26
F2	10724	20614	32080	40954,4	49127,4	56110	64491	72424	80783	86824
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,4	0,7	1,1	2,1	2,8	10,2	7,6	9,2	17,1	37,2
Tempo2	0,5	0,9	1,4	1,1	1,5	2,5	5,0	3,4	4,0	5,8
Gravit100_3										
F1	10	16	19	22	23	25	26	27	27	27
F2	11590	22120	32070	42581,6	51875,6	60627,6	66595,6	75845,6	85381,6	93697,6
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,7	1,0	1,0	1,7	2,5	3,2	6,0	8,7	15,0	36,3
Tempo2	84,4	1,0	0,7	1,3	1,2	1,5	1,9	2,9	5,9	4,4
Gravit100_4										
F1	8	15	22	22	23	23	23	27	27	27
F2	11144	20481	30285	37803,8	43846,8	53195,8	61224,4	68980	75510	82337
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,5	0,9	1,0	1,5	2,4	4,0	5,6	9,3	16,0	29,2
Tempo2	0,7	2,8	1,0	0,6	1,4	1,7	1,7	2,7	5,5	5,3
Gravit100_5										
F1	9	16	18	18	19	22	23	23	24	26
F2	10344	19774,6	27725,4	35966,4	48929,4	59335,4	67632,4	76035,4	84585,4	92559,4
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	24	0	0	0	0	0	0	0	0	0
Tempo1	15,3	1,1	1,9	3,7	4,6	6,2	7,9	10,0	43,3	52,5
Tempo2	120,1	1,8	1,1	9,4	9,6	1,9	2,3	2,9	6,8	9,0
Gravit100_6										
F1	10	15	23	27	27	27	27	27	27	27
F2	11338	18892	28700	38712	45120	52435	61007	71638	82590	95836
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,2	2,4	1,7	4,7	2,6	4,4	6,5	9,2	21,8	39,4
Tempo2	0,4	7,5	2,8	1,8	1,5	4,0	4,1	4,6	5,4	5,8
Gravit100_7										
F1	9	16	20	20	21	21	24	27	28	28
F2	10214	21257	30340	37448	47576,6	58254,6	68428,6	75216,6	82082,6	88292,6
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,9	0,7	1,1	1,3	3,0	5,1	6,7	8,9	20,2	19,2
Tempo2	1,3	1,0	0,9	0,9	1,2	1,5	1,9	2,8	3,5	5,0
Gravit100_8										
F1	10	18	24	28	28	28	28	28	28	28
F2	16250	25290	33606,6	43178,2	49750,2	57311,2	67835,2	76089,2	84219,2	91181,2
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,3	1,4	2,3	1,5	3,5	4,7	5,9	8,1	26,4	27,0
Tempo2	0,4	2,0	1,2	1,4	2,9	8,6	5,8	3,0	14,9	6,4
Gravit100_9										
F1	9	15	19	20	21	21	21	24	29	29
F2	11488	21028,6	29682,2	35686,8	44918	52283,6	60058,6	70728,6	79504,6	86302,6
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	1,0	0,6	1,0	2,3	2,8	4,0	6,1	9,8	21,3	21,0
Tempo2	15,2	1,6	0,8	0,9	1,1	1,7	1,8	2,3	3,7	4,5

Tabela 5.10. Resultados Obtidos pela Abordagem EcA para Cada Instância do Conjunto “*Rnd100*”

ITERACÕES	1	2	3	4	5	6	7	8	9	10
Rnd100_0										
F1	7	17	23	33	43	52	61	71	81	91
F2	25914	54244	69806	91598	112518	128106	150006	167044	179286	192722
GAP1	0	24	0	0	0	15	0	0	0	0
GAP2	0	0	0	22	0	0	0	0	0	0
Tempo1	27,5	240,1	182,7	154,4	228,9	240,1	47,0	8,0	20,6	34,4
Tempo2	2,0	16,7	21,7	240,1	16,1	82,1	23,4	26,6	18,1	53,4
Rnd100_1										
F1	9	17	23	32	41	50	59	68	78	87
F2	31534	55304	70836	93850	117184	138072	155074	172190	189038	201970
GAP1	0	18	17	22	5	0	7	0	0	0
GAP2	0	33	0	0	0	0	0	0	0	0
Tempo1	5,2	240,1	240,8	240,3	240,4	49,9	242,9	15,6	59,7	47,2
Tempo2	1,6	240,2	74,0	97,1	42,3	18,2	28,4	42,2	108,5	73,9
Rnd100_2										
F1	8	13	20							
F2	27588	43476	-							
GAP1	0	15	40							
GAP2	0	0	-							
Tempo1	6,1	240,1	240,1							
Tempo2	2,5	45,4	240,2							
Rnd100_3										
F1	9	19	28	37	46	56	65	75	85	95
F2	29690,6	56832,6	79950,6	100715	120045	140237	156619	176621	193497	208907
GAP1	0	0	0	11	9	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	4,5	13,3	237,8	240,1	240,1	32,1	94,7	8,1	16,5	27,3
Tempo2	42,4	5,1	8,2	9,9	109,6	30,0	18,5	7,2	20,4	17,8
Rnd100_4										
F1	8	18	26	36	45	54	64	73	83	93
F2	27898	54662	75614	100006	119188	139108	156388	171462	189198	203536
GAP1	0	0	12	0	0	0	0	0	0	0
GAP2	0	0	0	4	0	0	0	0	0	0
Tempo1	0,9	49,6	240,2	3,9	16,1	19,0	6,7	133,5	25,4	65,5
Tempo2	1,3	23,1	106,3	240,6	7,3	9,0	8,8	15,9	69,5	78,9
Rnd100_5										
F1	9	17	26	35	43	53	62	72	82	92
F2	31268	57054	81724	104428	123084	142270	157364	175196	189134	202070
GAP1	0	0	0	6	5	7	0	0	0	0
GAP2	0	0	0	9	8	0	0	2	0	0
Tempo1	37,8	63,4	55,2	240,1	242,5	240,3	101,0	10,9	15,2	37,5
Tempo2	4,3	4,4	7,7	240,1	240,3	30,7	20,7	240,2	36,4	62,2
Rnd100_6										
F1	8	19	27	36	45	54	64	74	84	94
F2	27254	56348	78946	100138	120506	139662	158920	175262	189108	204356
GAP1	0	42,11	0	19,44	4,44	18,39	-	0	0	0
GAP2	0	81,19	0	0	0	19,38	19,65	0	0	0
Tempo1	206,2	240,1	108,1	240,2	240,4	240,7	241,0	10,0	18,3	49,9
Tempo2	3,8	240,1	3,2	17,6	12,8	240,7	240,9	13,4	23,7	234,5
Rnd100_7										
F1	8	17	26	34	43	52	62	71	81	91
F2	26786	53400	76158	98016	119434	140520	159854	177676	192328	204728
GAP1	12,5	0	30	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	3	0,2	0	0	0
Tempo1	240,1	51,7	240,1	5,4	14,3	4,7	6,9	62,0	94,4	54,1
Tempo2	3,2	10,3	70,5	3,9	10,7	240,3	240,2	19,9	50,7	52,2
Rnd100_8										
F1	9	17	26	36	46	54	64	74	84	94
F2	29352	55382	78332	103204	124890	141328	160506	175740	191612	205354
GAP1	0	0	0	0	0	13	0	0	0	0
GAP2	0	0	0	0	0	0	1	0	0	0
Tempo1	24,7	125,1	31,2	41,1	132,6	240,5	6,1	16,7	89,3	87,7
Tempo2	4,3	10,8	3,2	5,6	23,5	52,5	243,0	21,3	36,6	47,3
Rnd100_9										
F1	8	16	26	36	43	53	63	72	82	92
F2	28716	52410	77614	94738	116758	135366	154246	169992	187366	202722
GAP1	0	0	0	-	0	0	0	0	0	0
GAP2	0	0	0	34,1	0	0,7	0	0	0	0
Tempo1	78,0	46,0	192,0	240,2	3,4	4,0	7,4	9,2	64,0	53,5
Tempo2	133,8	4,8	4,3	240,2	31,2	240,5	9,9	12,6	28,7	156,3

Tabela 5.11. Comparação do Número de Veículos Obtidos em cada Abordagem - Conjunto “Gravit100”

Instância	Abord	Iteração									
		1	2	3	4	5	6	7	8	9	10
Gravit100_0	EcA	6	14	19	19	20	21	25	30	31	31
	EsA	6	14	22	31	40	47	56	65	73	81
	HMO	6	14	19	20	21	22	25,3	29,3	30	30
	HG	6	14	19	19,1	21	22	25	30	31	31
Gravit100_1	EcA	8	14	15	21	26	26	26	26	28	30
	EsA	8	16	26	35	44	53	61	70	80	89
	HMO	8	14	15	21	25	25	25	26	29	31
	HG	8	14	15	21	25	25	25	25	29	31
Gravit100_2	EcA	9	17	22	23	23	26	26	26	26	26
	EsA	9	19	28	38	47	56	64	73	82	91
	HMO	9,9	17,9	22	23	23	26	26	26	26	26
	HG	9	17	22	23	23	26	26	26	26	26
Gravit100_3	EcA	10	16	19	22	23	25	26	27	27	27
	EsA	10	18	28	36	46	56	65	74	83	93
	HMO	10	16	19	22	23	25	25,3	26,3	26,3	26,3
	HG	10	16	19	22	23	25	25,5	26,5	26,5	26,5
Gravit100_4	EcA	8	15	22	22	23	23	23	27	27	27
	EsA	8	16	26	36	45	55	65	75	84	93
	HMO	9	16	23	23	23	23	23	26	26	26
	HG	8	15	22	22	23	23	23	27	27	27
Gravit100_5	EcA	9	16	18	18	19	22	23	23	24	26
	EsA	9	17	25	33	42	51	61	69	78	88
	HMO	10	17	19	19	20	22,5	23	23	23,5	27
	HG	9	16	18	18	19	22	23	23	23	26
Gravit100_6	EcA	10	15	23	27	27	27	27	27	27	27
	EsA	10	19	28	37	45	53	62	71	81	90
	HMO	10	15	23	26,9	26,9	26,9	26,9	26,9	26,9	26,9
	HG	10	15	23	26	26	26	26	26	26	26
Gravit100_7	EcA	9	16	20	20	21	21	24	27	28	28
	EsA	9	18	26	35	44	53	63	72	81	89
	HMO	10	17	20	20	21	21	24	27	27	27
	HG	9	17	20	20	21	22	24	27	27	27
Gravit100_8	EcA	10	18	24	28	28	28	28	28	28	28
	EsA	10	19	28	38	46	55	64	74	82	91
	HMO	10	18	24	28	28	28	28	28	28	28
	HG	10	18	24	28	28	28	28	28	28	28
Gravit100_9	EcA	9	15	19	20	21	21	21	24	29	29
	EsA	9	17	26	33	43	52	62	71	79	87
	HMO	10	16	20	21	21	21	21	25	30	30
	HG	9	15	19	20	21	21	21	24	29	29

Tabela 5.12. Comparação da Duração Total das Rotas Obtidas em cada Abordagem - Conjunto “Gravit100”

Instância	Abord	Iteração									
		1	2	3	4	5	6	7	8	9	10
Gravit100_0	EcA	10301,6	22455,6	29698	36068	43772	50363	58651	69326	75986	82810
	EsA	10301,6	22455,6	31484,8	41560,6	54678,6	62896,4	73528,4	85896,4	94853,8	103808
	HMO	10346	22500	29844	36630,4	44976,4	51699,8	60491,1	71083,6	78174,7	84832,7
	HG	10346	22500	29834	36248,8	44601,6	50957,6	59389	69878	76515	83868
Gravit100_1	EcA	9091,2	18341,2	27367,2	38721,8	49645,8	56922,8	64941,8	74127	82847	92156
	EsA	9091,2	19457,2	32475,2	44935,8	57143,8	67459,8	78463,8	90439,8	101722	114819
	HMO	9124	18392	27346	38745	49326,2	57551,2	65569,8	74759,8	83333,8	92941,5
	HG	9124	18374	27400	38801,4	49261,4	56566,4	64583,4	74061,4	82995,2	92296,2
Gravit100_2	EcA	10724	20614	32080	40954,4	49127,4	56110	64491	72424	80783	86824
	EsA	10724	21728	34978	48930	59293,4	69956	80351,2	91823,2	102117	113674
	HMO	11543	21433	32612,8	41587,2	49683,4	57269,6	65502,5	73442,1	81889	88043,7
	HG	10724	20614	32080	40982,6	49155,6	56225,6	64330,8	72526,2	81497,6	87626,6
Gravit100_3	EcA	11590	22120	32070	42581,6	51875,6	60627,6	66595,6	75845,6	85381,6	93697,6
	EsA	11590	22656	36388	48898	61422	72472	84296	96608	109538	119740
	HMO	11590	22120	32146	42759	52190,4	60431,2	66724,4	76063,2	85587,4	93630
	HG	11590	22120	32030	42524,7	51818,7	60141,9	66330,5	75580,5	85094,5	93329,9
Gravit100_4	EcA	11144	20481	30285	37803,8	43846,8	53195,8	61224,4	68980	75510	82337
	EsA	11144	20952,6	32292,6	44324,6	54564,6	67714,6	78524,6	88338,6	98954,6	109807
	HMO	12216	21588	31742	39149	45463	54289	62138	70472	76942	83733,1
	HG	11165	20537	30451	37981	44034	52309	61707	69329	75765	82549,8
Gravit100_5	EcA	10344	19774,6	27725,4	35966,4	48929,4	59335,4	67632,4	76035,4	84585,4	92559,4
	EsA	10344	20298,6	30437,0	41217,4	52969,4	65421,4	77759,4	88553,4	101131	113219
	HMO	11140	20696	28758	37506,4	51245,4	61533	69657	77759,6	86534,6	95510,6
	HG	10344	19900	27896	36137	49286	59782	67965	75380,9	84239,3	92905,3
Gravit100_6	EcA	11338	18892	28700	38712	45120	52435	61007	71638	82590	95836
	EsA	11338	21060	31220	42892	53044	63596	76102	88384	100796	111936
	HMO	11338	18892	28700	38168,4	44603,4	52003,6	60947,8	72298,6	83985,8	95102,6
	HG	11338	18892	28700	37794	44202	51541	60425	71772	83526	94640
Gravit100_7	EcA	10214	21257	30340	37448	47576,6	58254,6	68428,6	75216,6	82082,6	88292,6
	EsA	10214	22829	33710	45534	59028	71754	84126	94124	104896	115070
	HMO	10716	22221	31252	38314	49188	59027	69375	76272	82712	89428
	HG	10214	22117	31085	38193	49212,2	59770,6	69592,2	76318,2	82928,2	89700,4
Gravit100_8	EcA	16250	25290	33606,6	43178,2	49750,2	57311,2	67835,2	76089,2	84219,2	91181,2
	EsA	16250	25872	35218,6	47306,6	58784,4	67762,4	79356,4	91074,4	100470	113232
	HMO	16250	25304,4	33409,4	42981,4	49553,4	57104,8	67645,8	74803,8	83056,8	90032,8
	HG	16250	25290	33395	42975	49547	57102	67626	74551	82476	89426
Gravit100_9	EcA	11488	21028,6	29682,2	35686,8	44918	52283,6	60058,6	70728,6	79504,6	86302,6
	EsA	11488	22058,6	33172,6	42401,6	55645,6	66793,6	78433,6	90287,6	100036	109764
	HMO	12686	22689,6	31339,6	37446,6	46443,6	54221,8	62625	72697	82020,6	89141,4
	HG	11488	21122,9	29840,3	35938,5	45452,1	52888	60859,2	70956,8	79737,8	86581,8

Tabela 5.13. Comparação do Número de Veículos Obtidos em cada Abordagem - Conjunto “Rnd100”

Instância	Abord	Iteração									
		1	2	3	4	5	6	7	8	9	10
Rnd100_0	EcA	7	17	23	33	43	52	61	71	81	91
	EsA	7	17	25	35	45	54	64	74	84	94
	HMO	9	19	27	37	47	56	66	76	86	96
	HG	7	17	25	35	45	54	64	74	84	94
Rnd100_1	EcA	9	17	23	32	41	50	59	68	78	87
	EsA	9	17	24	33	43	52	62	71	81	90
	HMO	10	18	25	34	44	53	63	72	82	91
	HG	9	17	24	33	43	52	62	71	81	90
Rnd100_2	EcA	8	13	20	-	-	-	-	-	-	-
	EsA	8	17	26	36	45	54	64	74	84	93
	HMO	9	18	27	37	46	55	65	75	85	94
	HG	8	17	26	36	45	54	64	74	84	93
Rnd100_3	EcA	9	19	28	37	46	56	65	75	85	95
	EsA	9	19	28	38	48	58	68	78	88	98
	HMO	10	20	29	39	49	59	69	79	89	99
	HG	9	19	28	38	48	58	68	78	88	98
Rnd100_4	EcA	8	18	26	36	45	54	64	73	83	93
	EsA	8	18	26	36	45	54	64	73	83	93
	HMO	9	19	27	37	46	55	65	74	84	94
	HG	8	18	26	36	45	54	64	73	83	93
Rnd100_5	EcA	9	17	26	35	43	53	62	72	82	92
	EsA	9	17	26	35	44	54	64	74	84	94
	HMO	10	18	27	36	45	55	65	75	85	95
	HG	9	17	26	35	44	54	64	74	84	94
Rnd100_6	EcA	8	19	27	36	45	54	64	74	84	94
	EsA	8	18	27	37	46	56	66	76	86	96
	HMO	8	18	27	37	46	56	66	76	86	96
	HG	8	18	27	37	46	56	66	76	86	96
Rnd100_7	EcA	8	17	26	34	43	52	62	71	81	91
	EsA	8	18	26	35	45	55	65	74	84	94
	HMO	8	18	26	35	45	55	65	74	84	94
	HG	7	17	25	34	44	54	64	73	83	93
Rnd100_8	EcA	9	17	26	36	46	54	64	74	84	94
	EsA	9	17	26	36	46	55	65	75	85	95
	HMO	10	18	27	37	47	56	66	76	86	96
	HG	9	17	26	36	46	55	65	75	85	95
Rnd100_9	EcA	8	16	26	36	43	53	63	72	82	92
	EsA	8	17	27	37	47	57	67	76	86	96
	HMO	9	18	28	38	48	58	68	77	87	97
	HG	8	17	27	37	47	57	67	76	86	96

Tabela 5.14. Comparação da Duração Total das Rotas Obtidas em cada Abordagem - Conjunto “Rnd100”

Instância	Abord	Iteração									
		1	2	3	4	5	6	7	8	9	10
Rnd100_0	EcA	25914	54244	69806	91598	112518	128106	150006	167044	179286	192722
	EsA	25914	54506	74804	96596	117516	134404	156304	173342	185584	199020
	HMO	30658	59250	79548	101340	122260	139044	160944	177979	190221	203657
	HG	25914	54506	74804	96596	117516	134300	156200	173235	185477	198913
Rnd100_1	EcA	31534	55304	70836	93850	117184	138072	155074	172190	189038	201970
	EsA	31534	55304	76076	99090	122424	143312	162556	179672	196520	209452
	HMO	34030	57800	78572	101586	124920	145808	165049	182165	199013	211994
	HG	31534	55304	76076	99090	122424	143312	162553	179669	196517	209498
Rnd100_2	EcA	27588	43476	-	-	-	-	-	-	-	-
	EsA	27588	53786	78248	101614	122472	139746	159220	177092	191954	203552
	HMO	30273	56471	80933	104299	125157	142431	161905	179777	194639	206258
	HG	27663	53861	78323	101689	122547	139821	159295	177167	192029	203648
Rnd100_3	EcA	29690,6	56832,6	79950,6	100715	120045	140237	156619	176621	193497	208907
	EsA	29690,6	56832,6	79950,6	102909	124153	144345	162445	182447	199323	214733
	HMO	32658	59800	82918	105876	127120	147312	165412	185414	202290	217700
	HG	30049	57191	80309	103267	124511	144703	162803	182805	199681	215091
Rnd100_4	EcA	27898	54662	75614	100006	119188	139108	156388	171462	189198	203536
	EsA	27898	54662	75614	100006	119188	139108	156388	171462	189198	203536
	HMO	30212	56976	77983	102375	121557	141477	158757	173831	191567	205905
	HG	27927	54691	75698	100090	119272	139192	156472	171546	189282	203620
Rnd100_5	EcA	31268	57054	81724	104428	123084	142270	157364	175196	189134	202070
	EsA	31268	57054	81724	104428	124724	143910	160232	178064	192002	204938
	HMO	34342	60128	84798	107502	127798	146984	163306	181138	195076	208012
	HG	31268	57054	81724	104428	124724	143910	160232	178064	192002	204938
Rnd100_6	EcA	27254	56348	78946	100138	120506	139662	158920	175262	189108	204356
	EsA	27254	56594	79192	102370	122738	143226	162968	179310	193156	208404
	HMO	27271	56611	79209	102387	122755	143243	162985	179327	193173	208421
	HG	27268	56608	79206	102384	122752	143240	162982	179324	193170	208418
Rnd100_7	EcA	26786	53400	76158	98016	119434	140520	159854	177676	192328	204728
	EsA	26786	55916	78674	100532	123510	146222	165556	183378	198030	210430
	HMO	26962	56092	78850	100708	123686	146398	165732	183554	198206	210606
	HG	24850	53980	76738	98596	121574	144286	163620	181442	196094	208494
Rnd100_8	EcA	29352	55382	78332	103204	124890	141328	160506	175740	191612	205354
	EsA	29352	55382	78332	103204	124890	143702	162880	178114	193986	207728
	HMO	32334	58364	81314	106186	127872	146684	165862	181096	196968	210707
	HG	29352	55382	78332	103204	124890	143702	162880	178114	193986	207725
Rnd100_9	EcA	28716	52410	77614	94738	116758	135366	154246	169992	187366	202722
	EsA	28716	54430	79634	102650	124670	143278	162158	177904	195278	210634
	HMO	31441	57155	82359	105375	127395	146003	164883	180634	198008	213364
	HG	28716	54430	79634	102650	124670	143278	162158	177909	195283	210639

Capítulo 6

Conclusões

Neste trabalho foi abordado um serviço de transporte reativo a demanda no qual o planejamento das rotas e definição do atendimento dos passageiros são realizados buscando minimizar o número de veículos necessários, a duração total e o atraso na entrega dos passageiros. Com o intuito de otimizar este serviço foram propostas abordagens evolutivas para tratar o modo de operação estático e abordagens baseadas em métodos exatos e heurísticas para resolução do modo de operação dinâmico.

Para compor as abordagens evolutivas foram utilizados três algoritmos evolutivos multi-objetivo (NSGA-II, IBEA, SPEA2), os quais foram avaliados para diferentes combinações de operadores de variação. No geral, os experimentos computacionais associados ao caso estático, envolveram 19 abordagens as quais antes de serem avaliadas foram calibradas para obterem máximo desempenho.

A avaliação das abordagens evolutivas foi realizada por meio da versão unária da métrica *Additive ϵ -Indicator* e validada por meio de testes estatísticos. Foi conduzida uma avaliação geral do desempenho das abordagens considerando os dois conjuntos de instâncias de teste e duas avaliações individuais, uma para cada conjunto. No geral, os resultados destacaram o bom desempenho das abordagens que utilizam o operador de recombinação 4, que se diferencia dos demais pelo fato de construir as soluções filhas utilizando uma quantidade maior que uma rota proveniente do segundo pai e completando a solução por meio do procedimento que viabiliza a inserção das requisições com o menor acréscimo na duração total.

Outra observação, é sobre o desempenho semelhante dos operadores de mutação. O operador de mutação 2 o qual sempre gera uma nova rota para uma requisição selecionada de forma aleatória, mesmo sendo um procedimento mais simples que o operador de mutação 1, se comportou de forma satisfatória. Talvez por ser rápido, o tempo ganho permita a realização de mais gerações, conseqüentemente contribui na

evolução das soluções. Além disso a rota acrescentada à solução pelo operador pode ser facilmente desfeita por meio dos operadores que realizam o procedimento de inserção que busca o menor acréscimo na duração total.

Em relação às abordagens propostas para otimização do serviço abordado de forma dinâmica, é possível afirmar que tanto a abordagem baseada em métodos exatos quanto as abordagens heurísticas obtiveram desempenhos satisfatórios quando utilizada a simulação dinâmica proposta. Por meio dos resultados obtidos pela abordagem baseada em métodos exatos que não realiza o aproveitamento das rotas criadas nas iterações anteriores, é possível notar que, principalmente para as instâncias que possuem distribuição heterogênea das requisições, as abordagens propostas são capazes de realizar um bom aproveitamento das rotas. Embora inferiores em algumas situações, as heurísticas propostas são capazes de lidar com grandes conjuntos de novas requisições, ao contrário da abordagem baseada em métodos exatos.

Como trabalhos futuros, uma proposta seria estudar uma maneira de aumentar a capacidade de resolução dos modelos matemáticos para que possa obter uma solução ótima para cada um dos objetivo que são otimizados em conjunto pelas abordagens evolutivas multi-objetivo. Outra opção é incorporar buscas locais guidas por indicadores de qualidade às abordagens evolutivas propostas neste trabalho.

6.1 Publicações

Viana, R. J. d. S., Santos, A. G. e Arroyo J. E. C. (2015). Multi-objective evolutionary approach for optimizing a demand responsive transport. *XLVII Simpósio Brasileiro de Pesquisa Operacional (SBPO 2015)* pp. 1918-1929.

Viana, R. J. d. S. e dos Santos, A. G (2015). Dynamic approach for a demand responsive transport service. *XLI Conferencia Latinoamericana en Informática (CLEI 2015)* pp. 1-12.

Apêndice A

Calibração das Abordagens Heurísticas

Neste apêndice são apresentados os resultados obtidos por meio dos experimentos de calibração das abordagens propostas neste trabalho. Tais resultados indicam quais as melhores probabilidades de execução dos operadores de recombinação e de mutação, ou seja, as configurações de parâmetros que permitem o melhor desempenho das abordagens. Para cada um dos três algoritmos analisados (NSGA-II, IBEA e SPEA-II), os experimentos de calibração foram realizados considerando todas as combinações entre os operadores de recombinação e de mutação propostos. Os resultados obtidos pela calibração das abordagens foram validados seguindo o mesmo planejamento experimental e utilizando os mesmos testes estatísticos apresentados na Seção 5.

A.1 Calibração das Abordagens Propostas

Com o intuito definir a melhor abordagem dentre todas as propostas neste trabalho, foi realizado um conjunto de experimentos computacionais que tem como objetivo definir para cada uma, as configurações dos parâmetros que permitem as mesmas alcançarem seus melhores resultados.

Os experimentos foram realizados utilizando uma amostra composta por 40% do conjunto total de instâncias. Cada algoritmo foi testado com 6 combinações diferentes de operadores, resultando em 18 abordagens diferentes. Em cada abordagem, as probabilidades de execução dos operadores de recombinação e de mutação foram testadas utilizando os seguintes conjunto de valores: $Pc \in \{0.2, 0.5, 0.8, 1.0\}$ e $Pm \in \{0.2, 0.5, 0.8, 1.0\}$. Note-se que ao todo foram analisadas 16 configurações de parâmetros diferentes ou tratamentos. A Tabela A.1 apresenta o identificador de

cada uma das configurações de parâmetros analisadas. Para uma análise confiável, as abordagens foram executadas 10 vezes para cada tratamento.

Tabela A.1. Configurações de Parâmetros Analisadas

Tratamento	Pc	Pm
T1	0.2	0.2
T2	0.2	0.5
T3	0.2	0.8
T4	0.2	1.0
T5	0.5	0.2
T6	0.5	0.5
T7	0.5	0.8
T8	0.5	1.0
T9	0.8	0.2
T10	0.8	0.5
T11	0.8	0.8
T12	0.8	1.0
T13	1.0	0.2
T14	1.0	0.5
T15	1.0	0.8
T16	1.0	1.0

A avaliação do desempenho dos tratamentos foi realizada por meio da versão unária da métrica *Additive ϵ -Indicator*. Portanto, para avaliação dos 10 conjuntos aproximados obtidos por cada tratamento em cada instância, foi construído um conjunto referência composto por todos pontos não-dominados encontrados. Utilizando a métrica, cada execução é avaliada de acordo com o respectivo conjunto referência. Para aplicação dos testes estatísticos, o valor médio obtido considerando todas as instâncias é definido como a variável resposta do par tratamento/execução.

Os resultados provenientes dos experimentos de calibração e das análises estatísticas serão apresentados a seguir da seguinte maneira: separados em três subseções (uma para cada algoritmo analisado), inicialmente será apresentada uma tabela composta pelos resultados de todos testes estatísticos aplicados aos dados provenientes dos experimentos de calibração das 6 abordagens analisadas. Nesta tabela estão dispostos o P-valor dos testes de *Shapiro-Wilk*, *Levene*, ANOVA e *Kruskal-Wallis*, utilizado quando algum dos dois primeiros testes não forem satisfeitos (situação indicada por valores na cor vermelha). Posteriormente, para cada abordagem são apresentados os resultados obtidos pelo teste de *Scott-Knott* e o gráfico *boxplot*.

A.1.1 Calibração NSGA2

Nesta subseção são apresentadas as configurações de parâmetros que obtiveram melhor desempenho nos experimentos de calibração para as abordagens que utilizam o algoritmo NSGA-II. De acordo com a Tabela A.2, no qual apresenta o P-valor obtidos pelos testes estatísticos, exceto para a combinação de operadores Recombinação 2/Mutação 1 não há diferença significativa entre os tratamentos. Para o restante das abordagens, um ou mais tratamentos obtiveram resultados estatisticamente superiores em relação aos demais. Para a combinação de operadores Recombinação 2/Mutação 2, o pressuposto de normalidade não foi atingido (destacado em vermelho), portanto foi utilizado o teste de *Kruskal-Wallis* para realização da análise estatística.

Tabela A.2. Experimentos Estatísticos - Algoritmo NSGA-II

Abordagem	Testes Aplicados			
	<i>Shapiro-Wilk</i>	<i>Levene</i>	ANOVA	<i>Kruskal-Wallis</i>
NSGA-II/Recombinação 2/Mutação 1	0,2016	0,8923	0,2340	-
NSGA-II/Recombinação 3/Mutação 1	0,6474	0,6061	0,0000	-
NSGA-II/Recombinação 4/Mutação 1	0,1092	0,1098	0,0000	-
NSGA-II/Recombinação 2/Mutação 2	0,0003	0,5140	-	0,0000
NSGA-II/Recombinação 3/Mutação 2	0,0348	0,9011	0,0000	-
NSGA-II/Recombinação 4/Mutação 2	0,0742	0,5367	0,0000	-

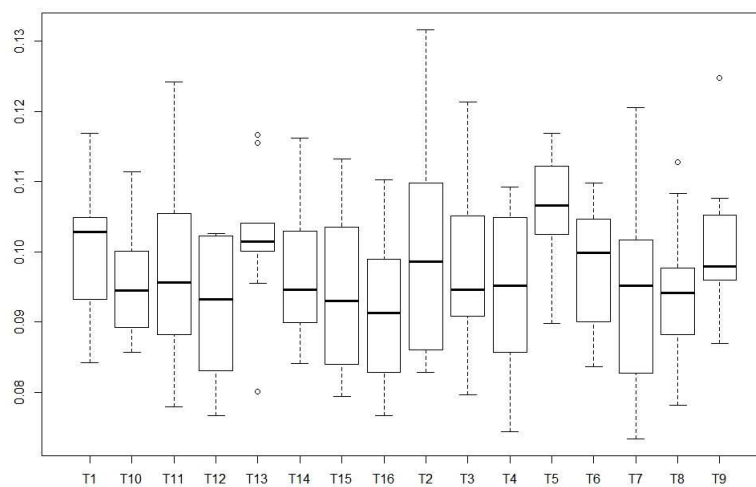
A seguir são explicitadas de forma detalhada as análises estatísticas realizadas para definição do melhor tratamento em cada abordagem que utiliza o algoritmo NSGA-II.

A.1.1.1 NSGA2 - Recombinação 2 e Mutação 1

Para a combinação de operadores composta pela recombinação 2 e mutação 1, a hipótese nula não foi rejeitada, logo não há diferença significativa entre os tratamentos. Desta forma o tratamento (T12), o qual obteve a menor média de acordo com a Tabela A.3 foi selecionado para a execução desta abordagem para as demais instâncias. Note, que para esta situação não há definição de grupos. Em seguida, a Figura A.1 apresenta o gráfico *boxplot*.

Tabela A.3. Teste de *Scott-Knott* - NSGA2 - Recombinação 2 e Mutação 1

Grupos	Níveis	Médias
-	T1	0,101629
-	T10	0,095959
-	T11	0,098992
-	T12	0,092118
-	T13	0,101836
-	T14	0,096556
-	T15	0,093771
-	T16	0,09216
-	T2	0,099886
-	T3	0,098535
-	T4	0,095249
-	T5	0,106382
-	T6	0,097708
-	T7	0,093915
-	T8	0,093653
-	T9	0,100205

**Figura A.1.** Gráfico *boxplot* - Recombinação 2 e Mutação 1

A.1.1.2 NSGA2 - Recombinação 3 e Mutação 1

Para a combinação de operadores composta pela recombinação 3 e mutação 1, a hipótese nula foi rejeitada, ou seja, há diferença significativa entre os tratamentos. De acordo com a Tabela A.4, para esta abordagem existe um conjunto de configurações estatisticamente melhores que as demais, entretanto dentro deste grupo não existe di-

ferença estatística entre os respectivos membros. Portanto, o tratamento com menor média dentro do conjunto ($T4$) foi selecionado para execução da abordagem para todas as instâncias. Em seguida, a Figura A.2 apresenta o gráfico *boxplot*.

Tabela A.4. Teste de *Scott-Knott* - NSGA2 - Recombinação 3 e Mutação 1

Grupos	Níveis	Médias
a	T13	0,363251
b	T15	0,347226
b	T14	0,346705
b	T16	0,337998
c	T9	0,137203
d	T10	0,111528
d	T5	0,111383
d	T1	0,107143
e	T11	0,098469
e	T2	0,095379
e	T6	0,092716
e	T12	0,091116
f	T7	0,086256
f	T8	0,084142
f	T3	0,082762
f	T4	0,081632

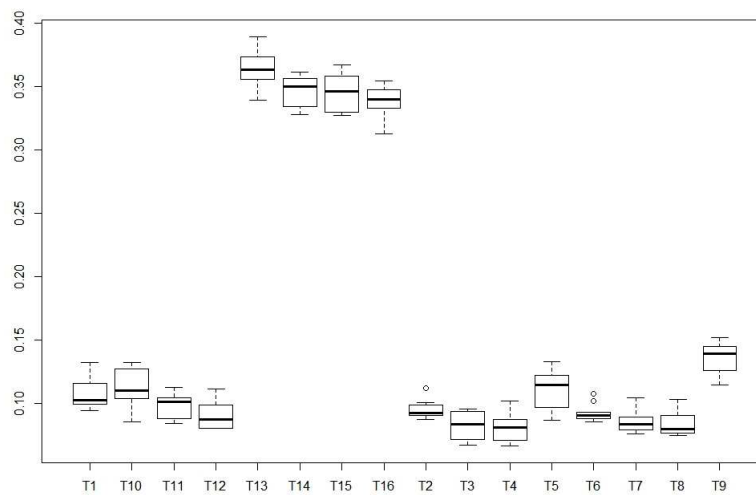


Figura A.2. Gráfico *boxplot* - Recombinação 3 e Mutação 1

A.1.1.3 NSGA2 - Recombinação 4 e Mutação 1

Para a combinação de operadores Recombinação 4/Mutação 1 foi constatado por meio da análise ANOVA que existe diferença significativa entre as médias. De forma análoga a abordagem anterior, existe um conjunto de configurações estatisticamente melhores que as demais, conforme apresentado na Tabela A.5. Portanto, o tratamento com menor média dentro do conjunto (*T16*) foi selecionado para execução da abordagem para todas as instâncias. Em seguida, a Figura A.3 apresenta o gráfico *boxplot*.

Tabela A.5. Teste de *Scott-Knott* - NSGA2 - Recombinação 4 e Mutação 1

Grupos	Tratamentos	Médias
a	T4	0.08797152
a	T2	0.08749458
a	T3	0.08471784
b	T1	0.07771876
b	T5	0.07576272
b	T14	0.07537182
b	T8	0.07533774
b	T6	0.07410774
b	T12	0.0735844
b	T9	0.07315493
b	T11	0.07298017
b	T7	0.07284325
b	T10	0.07272629
b	T13	0.07257475
b	T15	0.07241045
b	T16	0.07142039

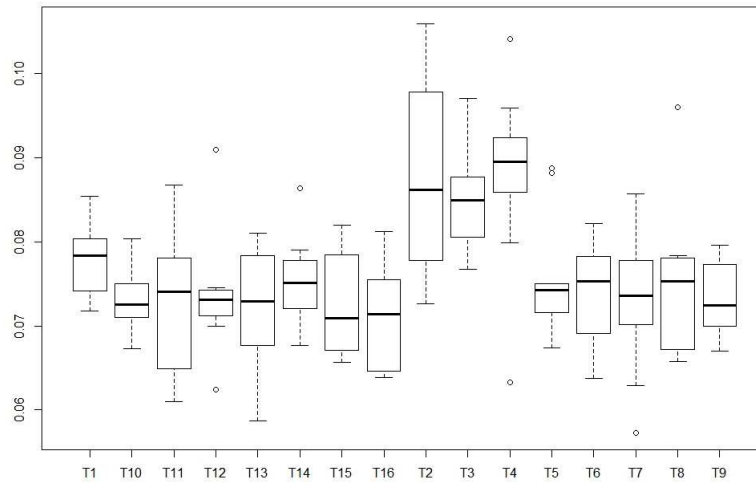


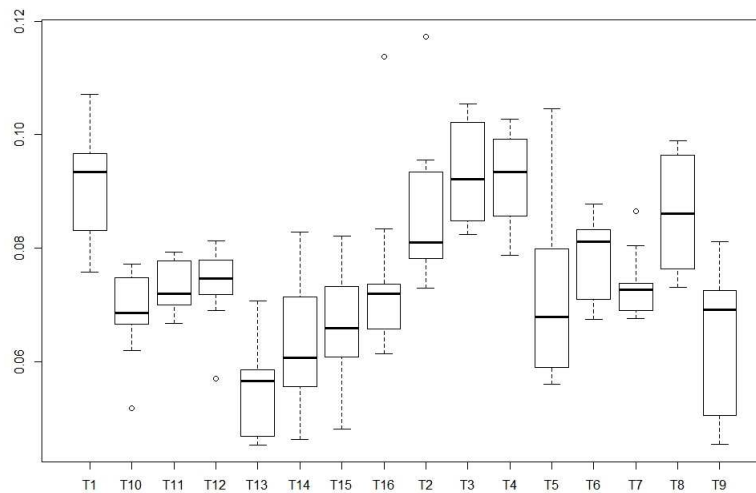
Figura A.3. Gráfico *boxplot* - Recombinação 4 e Mutação 1

A.1.1.4 NSGA2 - Recombinação 2 e Mutação 2

Para a abordagem composta pelos operadores de recombinação 2 e mutação 2, o pressuposto que verifica a normalidade dos dados não foi atingido ($P\text{-valor} < 0.01$). Portanto, foi realizado o teste de *Kruskal-Wallis* no lugar da análise ANOVA, no qual foi provado que há diferença significativa entre os tratamentos. De acordo com o teste de *Scott-Knott* apresentado na Tabela A.6, para esta abordagem existe um conjunto de configurações estatisticamente melhores que as demais. Portanto, o tratamento com menor média dentro do conjunto (*T13*) foi selecionado para execução da abordagem para todas as instâncias. Em seguida, a Figura A.4 apresenta o gráfico *boxplot*.

Tabela A.6. Teste de *Scott-Knott* - NSGA2 - Recombinação 2 e Mutação 2

Grupos	Níveis	Médias
a	T3	0,093214
a	T4	0,092652
a	T1	0,090491
a	T2	0,086337
a	T8	0,085206
b	T6	0,07839
b	T16	0,074381
b	T7	0,073408
b	T12	0,073376
b	T11	0,073097
b	T5	0,071081
b	T10	0,068342
c	T15	0,066149
c	T9	0,064652
c	T14	0,062353
c	T13	0,056261

**Figura A.4.** Gráfico *boxplot* - Recombinação 2 e Mutação 2

A.1.1.5 NSGA2 - Recombinação 3 e Mutação 2

No experimento de calibração do algoritmo NSGA-II com os operadores recombinação 3 e mutação 2 foi constatado por meio da análise ANOVA que existe diferença significativa entre as médias. De acordo com os resultados do Teste de *Scott-Knott* apresentados na Tabela A.7, para esta abordagem existe um conjunto de configurações estatisticamente melhores que as demais. Portanto, o tratamento com menor média

dentro do conjunto (T_{12}) foi selecionado para execução da abordagem para todas as instâncias. Em seguida, a Figura A.5 apresenta o gráfico *boxplot*.

Tabela A.7. Teste de *Scott-Knott* - NSGA2 - Recombinação 3 e Mutação 2

Grupos	Níveis	Médias
a	T13	0,36505
a	T14	0,361665
a	T15	0,357899
a	T16	0,350591
b	T9	0,121928
c	T5	0,102893
c	T1	0,099657
c	T2	0,096251
c	T10	0,096145
c	T8	0,095859
c	T7	0,09257
c	T3	0,092262
d	T6	0,090761
d	T11	0,086006
d	T4	0,085896
d	T12	0,081823

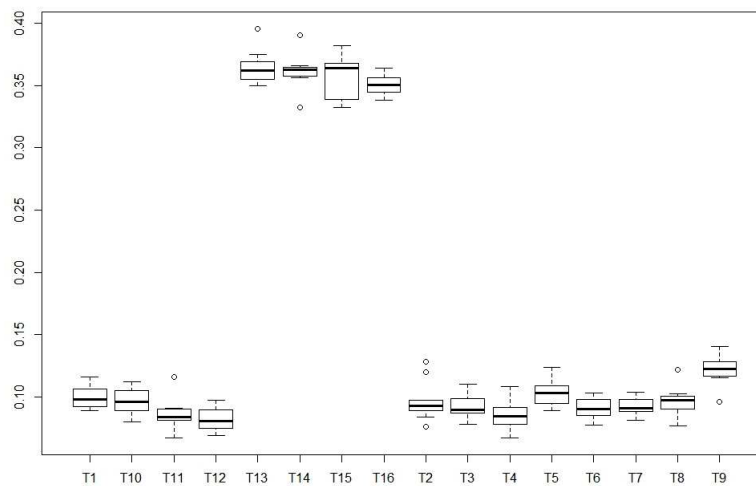


Figura A.5. Gráfico *boxplot* - Recombinação 3 e Mutação 2

A.1.1.6 NSGA2 - Recombinação 4 e Mutação 2

No experimento de calibração do algoritmo NSGA-II com os operadores recombinação 4 e mutação 2 foi constatado por meio da análise ANOVA que existe diferença significativa entre as médias. De acordo com os resultados do Teste de *Scott-Knott* apresentados na Tabela A.8, para esta abordagem existe um conjunto de configurações estatisticamente melhores que as demais. Portanto, o tratamento com menor média dentro do conjunto (*T13*) foi selecionado para execução da abordagem para todas as instâncias. Em seguida, a Figura A.6 apresenta o gráfico *boxplot*.

Tabela A.8. Teste de *Scott-Knott* - NSGA2 - Recombinação 4 e Mutação 2

Grupos	Níveis	Médias
a	T4	0,09747
b	T3	0,0893
b	T2	0,085606
c	T8	0,081872
c	T12	0,077794
c	T1	0,076979
c	T7	0,076306
d	T6	0,070413
d	T16	0,069932
d	T11	0,066881
d	T5	0,066784
d	T15	0,064182
d	T10	0,064146
e	T9	0,060458
f	T14	0,055296
f	T13	0,053263

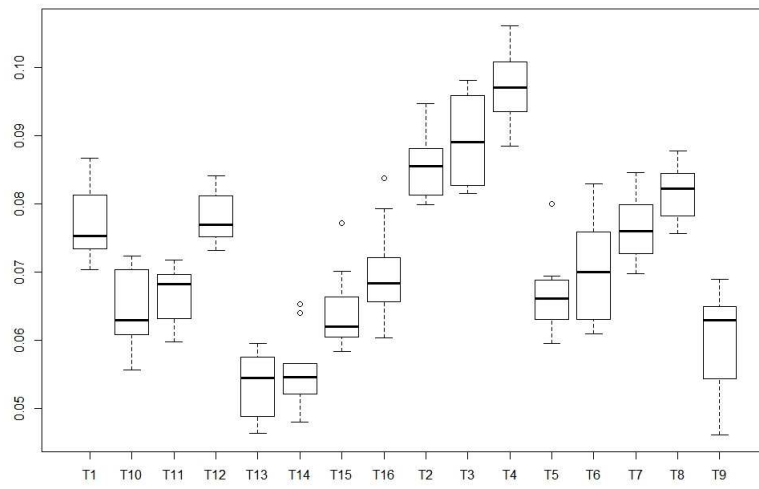


Figura A.6. Gráfico *boxplot* - Recombinação 4 e Mutação 2

A.1.2 Calibração IBEA

Nesta subseção serão apresentadas as configurações de parâmetros que obtiveram melhor desempenho nos experimentos de calibração para as abordagens que utilizam o algoritmo IBEA. De acordo com a Tabela A.9, para todas abordagens há diferença significativa entre os tratamentos, ou seja, um ou mais tratamentos se destacaram em relação aos demais.

Tabela A.9. Experimentos Estatísticos - Algoritmo IBEA

Abordagem	Testes			
	Shapiro-Wilk W	Levene	ANOVA	Kruskal-Wallis
IBEA/Recombinação 2/Mutação 1	0,2022	0,3965	0,0000	-
IBEA/Recombinação 3/Mutação 1	0,2718	0,9549	0,0000	-
IBEA/Recombinação 4/Mutação 1	0,2293	0,6750	0,0000	-
IBEA/Recombinação 2/Mutação 2	0,1124	0,0643	0,0000	-
IBEA/Recombinação 3/Mutação 2	0,4627	0,0328	0,0000	-
IBEA/Recombinação 4/Mutação 2	0,4253	0,1157	0,0000	-

Para simplificação da apresentação dos resultados associados às abordagens que utilizam o algoritmo IBEA, para cada uma somente serão apresentados o tratamento que foi selecionado para execução do conjunto de instâncias, uma tabela, na qual é apresentada o teste de *Scott-Knott* e uma figura com o gráfico *boxplot*. Para as análises nas quais um conjunto de tratamentos são estatisticamente melhores que os demais tratamentos foi selecionado o tratamento com menor média que pertence ao grupo melhor classificado.

A.1.2.1 IBEA - Recombinação 2 e Mutação 1

Para esta abordagem, o tratamento (T15) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.10 e Figura A.10, respectivamente.

Tabela A.10. Teste de *Scott-Knott* - IBEA - Recombinação 2 e Mutação 1

Grupos	Níveis	Médias
a	T1	0,125453
a	T2	0,125396
a	T5	0,123824
a	T13	0,120577
a	T9	0,118565
a	T7	0,118276
a	T10	0,114158
b	T3	0,112728
b	T4	0,110439
b	T6	0,109526
b	T14	0,107585
b	T12	0,107021
b	T11	0,103261
b	T8	0,10287
b	T16	0,099717
b	T15	0,099352

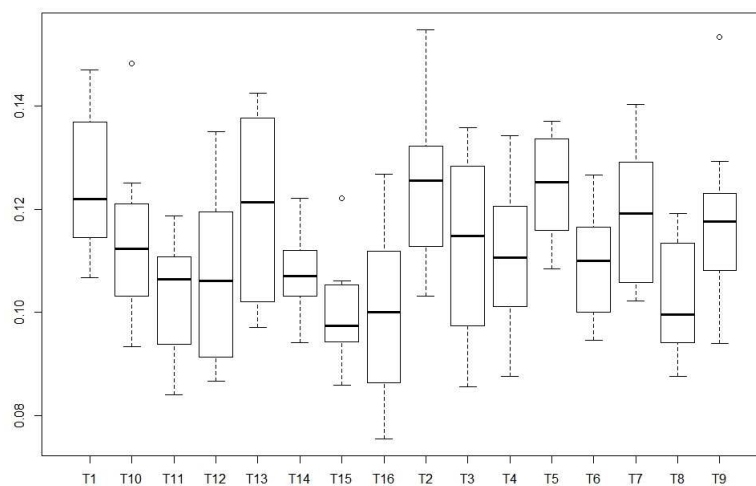


Figura A.7. Gráfico *boxplot* - Recombinação 2 e Mutação 1

A.1.2.2 IBEA - Recombinação 3 e Mutação 1

Para esta abordagem, o tratamento (T4) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.11 e Figura A.11, respectivamente.

Tabela A.11. Teste de *Scott-Knott* - IBEA - Recombinação 3 e Mutação 1

Grupos	Níveis	Médias
a	T13	0,358279
b	T14	0,343222
b	T15	0,338611
b	T16	0,336222
c	T9	0,152565
d	T5	0,133602
d	T1	0,126924
d	T10	0,119129
e	T6	0,113582
e	T2	0,109063
e	T11	0,106466
f	T3	0,101707
f	T8	0,099096
f	T12	0,098153
f	T7	0,095783
f	T4	0,090805

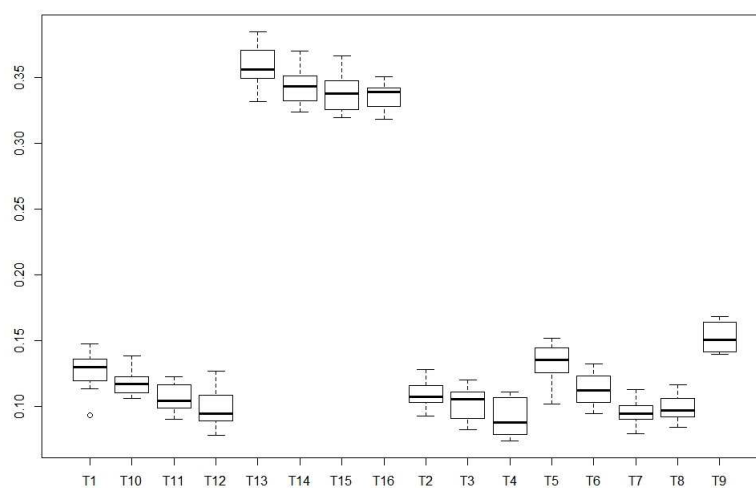


Figura A.8. Gráfico *boxplot* - Recombinação 3 e Mutação 1

A.1.2.3 IBEA - Recombinação 4 e Mutação 1

Para esta abordagem, o tratamento (T16) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.12 e Figura A.12, respectivamente.

Tabela A.12. Teste de *Scott-Knott* - IBEA - Recombinação 4 e Mutação 1

Grupos	Níveis	Médias
a	T2	0,10381
a	T1	0,102639
a	T3	0,100402
a	T4	0,099299
a	T8	0,098743
a	T6	0,095026
a	T5	0,094986
a	T7	0,092253
b	T10	0,086839
b	T12	0,08637
b	T11	0,084223
b	T15	0,080793
b	T9	0,080365
b	T14	0,079419
b	T13	0,076648
b	T16	0,072031

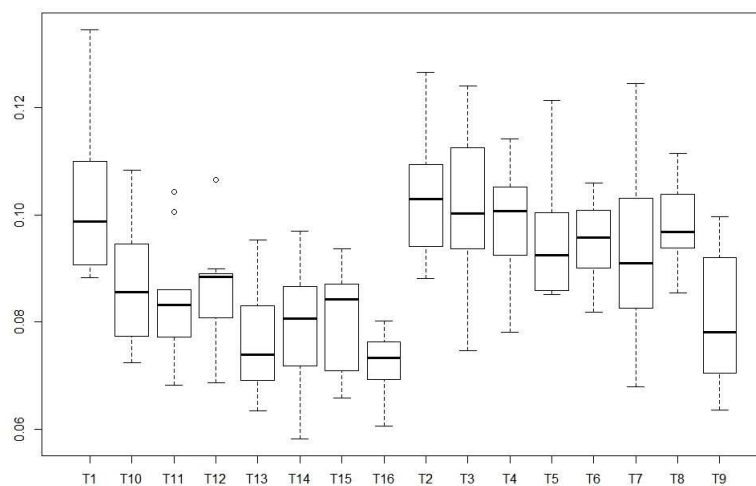


Figura A.9. Gráfico *boxplot* - Recombinação 4 e Mutação 1

A.1.2.4 IBEA - Recombinação 2 e Mutação 2

Para esta abordagem, o tratamento (T13) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.13 e Figura A.10, respectivamente.

Tabela A.13. Teste de *Scott-Knott* - IBEA - Recombinação 2 e Mutação 2

Grupos	Níveis	Médias
a	T2	0,09746
a	T3	0,093437
a	T1	0,091989
b	T4	0,084635
b	T6	0,0799
c	T7	0,07648
c	T8	0,075064
c	T5	0,073532
d	T9	0,066035
d	T12	0,065919
d	T11	0,063699
d	T16	0,063311
d	T10	0,062889
d	T14	0,061949
d	T15	0,059917
d	T13	0,054512

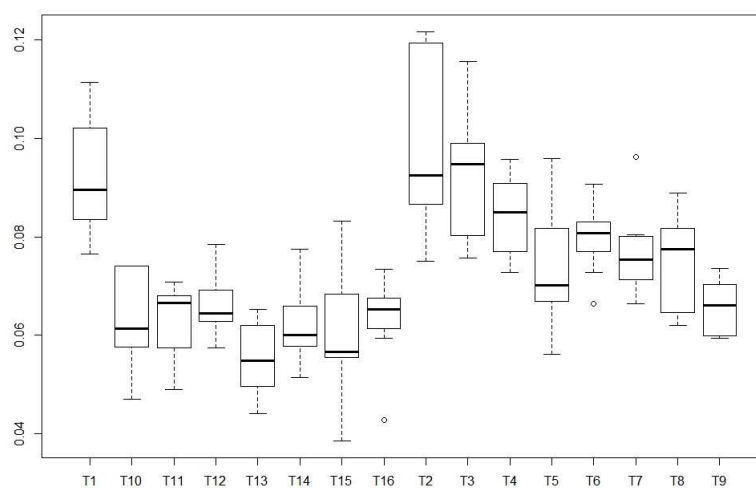


Figura A.10. Gráfico *boxplot* - Recombinação 2 e Mutação 2

A.1.2.5 IBEA - Recombinação 3 e Mutação 2

Para esta abordagem, o tratamento (T12) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.14 e Figura A.11, respectivamente.

Tabela A.14. Teste de *Scott-Knott* - IBEA - Recombinação 3 e Mutação 2

Grupos	Níveis	Médias
a	T14	0,342211
a	T13	0,341193
a	T15	0,334915
a	T16	0,330222
b	T9	0,149001
c	T1	0,121392
c	T5	0,120113
c	T2	0,119358
c	T3	0,116919
d	T10	0,112409
d	T6	0,110839
d	T4	0,106651
d	T11	0,103331
d	T7	0,101762
e	T8	0,096182
e	T12	0,088884

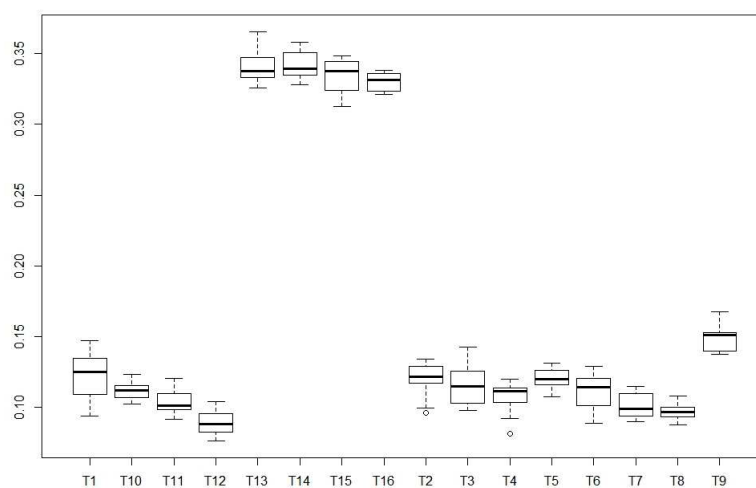


Figura A.11. Gráfico *boxplot* - Recombinação 3 e Mutação 2

A.1.2.6 IBEA - Recombinação 4 e Mutação 2

Para esta abordagem, o tratamento (T13) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.15 e Figura A.12, respectivamente.

Tabela A.15. Teste de *Scott-Knott* - IBEA - Recombinação 4 e Mutação 2

Grupos	Níveis	Médias
a	T4	0,107683
a	T1	0,101937
a	T3	0,101079
a	T2	0,100131
b	T8	0,088853
b	T6	0,087047
b	T5	0,084104
b	T12	0,083563
b	T7	0,081622
b	T10	0,078428
c	T16	0,076158
c	T11	0,073516
c	T9	0,071391
c	T15	0,070077
c	T14	0,068194
c	T13	0,067311

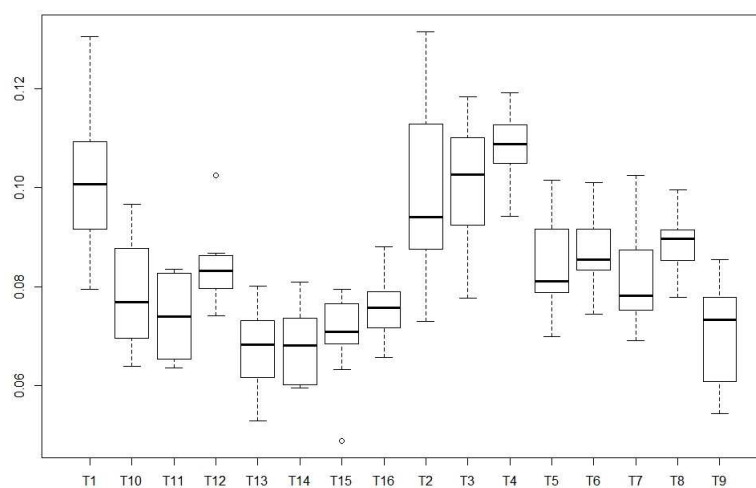


Figura A.12. Gráfico *boxplot* - Recombinação 4 e Mutação 2

A.1.3 Calibração SPEA2

Nesta subseção serão apresentadas as configurações de parâmetros que obtiveram melhor desempenho nos experimentos de calibração para as abordagens que utilizam o algoritmo SPEA2. De acordo com a Tabela A.16, para todas abordagens há diferença significativa entre os tratamentos, ou seja, um ou mais tratamentos se destacaram em relação aos demais. Em duas abordagens o pressuposto de normalidade não foi atingido (destacado em vermelho), portanto foi utilizado o teste de *Kruskal-Wallis* para validação dos resultados dessas abordagens.

Tabela A.16. Experimentos Estatísticos - Algoritmo SPEA2

Abordagem	Testes			
	Shapiro-Wilk W	Levene	ANOVA	Kruskal-Wallis
SPEA2/Recombinação 2/Mutação 1	0,2000	0,5722	0,0000	-
SPEA2/Recombinação 3/Mutação 1	0,0698	0,3064	0,0000	-
SPEA2/Recombinação 4/Mutação 1	0,0006	0,0858	-	0,0000
SPEA2/Recombinação 2/Mutação 2	0,1550	0,0119	0,0000	-
SPEA2/Recombinação 3/Mutação 2	0,0066	0,7891	-	0,0000
SPEA2/Recombinação 4/Mutação 2	0,0802	0,0642	0,0000	-

De forma similar à subseção anterior, para simplificação da apresentação dos resultados, em cada abordagem que utiliza o algoritmo SPEA2 somente serão apresentados o tratamento que foi selecionado para execução do conjunto de instâncias, uma tabela com o teste de *Scott-Knott* e uma figura com o gráfico *boxplot*. Para as análises nas quais um conjunto de tratamentos são estatisticamente melhores que os demais tratamentos foi selecionado o tratamento com menor média que pertence ao grupo melhor classificado.

A.1.3.1 SPEA2 - Recombinação 2 e Mutação 1

Para esta abordagem, o tratamento (T12) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.17 e Figura A.13, respectivamente.

Tabela A.17. Teste de *Scott-Knott* - SPEA2 - Recombinação 2 e Mutação 1

Grupos	Níveis	Médias
a	T1	0,186473
a	T5	0,175371
b	T9	0,161458
c	T13	0,147705
d	T2	0,123633
d	T6	0,117077
d	T10	0,11676
d	T14	0,111674
e	T7	0,102312
e	T3	0,101577
e	T11	0,098075
e	T16	0,096912
e	T15	0,095684
e	T4	0,090986
e	T8	0,088536
e	T12	0,08793

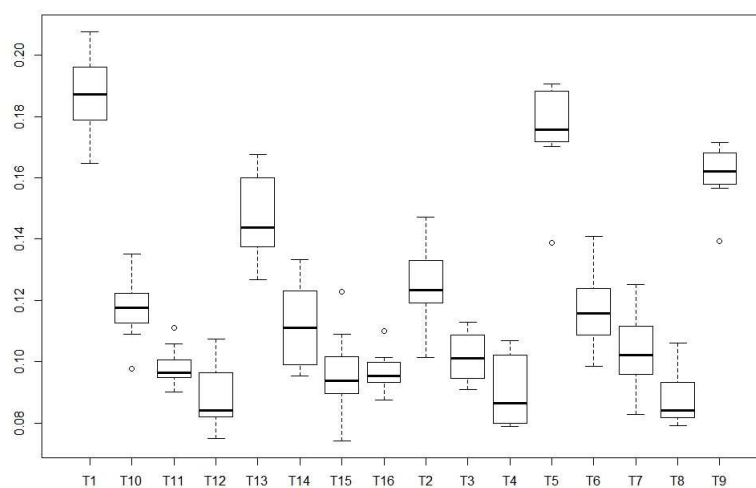


Figura A.13. Gráfico *boxplot* - Recombinação 2 e Mutação 1

A.1.3.2 SPEA2 - Recombinação 3 e Mutação 1

Para esta abordagem, o tratamento (T4) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.18 e Figura A.14, respectivamente.

Tabela A.18. Teste de *Scott-Knott* - SPEA2 - Recombinação 3 e Mutação 1

Grupos	Níveis	Médias
a	T13	0,354574
a	T14	0,342798
b	T15	0,336396
b	T16	0,327874
c	T1	0,178401
d	T9	0,164662
d	T5	0,155411
e	T10	0,130676
e	T2	0,121927
e	T11	0,118799
e	T6	0,115742
f	T12	0,103861
f	T7	0,103377
f	T3	0,09916
g	T8	0,09232
g	T4	0,088182

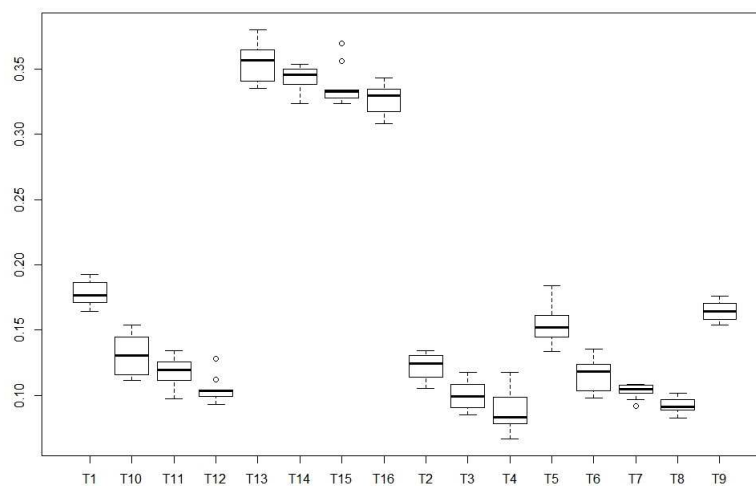


Figura A.14. Gráfico *boxplot* - Recombinação 3 e Mutação 1

A.1.3.3 SPEA2 - Recombinação 4 e Mutação 1

Para esta abordagem, o tratamento (T15) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.19 e Figura A.15, respectivamente.

Tabela A.19. Teste de *Scott-Knott* - SPEA2 - Recombinação 4 e Mutação 1

Grupos	Níveis	Médias
a	T1	0,144366
b	T2	0,107136
b	T5	0,104447
b	T3	0,102616
c	T4	0,094089
c	T6	0,093015
c	T7	0,091084
c	T9	0,090105
c	T8	0,089397
d	T10	0,086345
d	T12	0,080876
e	T14	0,077522
e	T13	0,077349
e	T16	0,076126
e	T11	0,073405
e	T15	0,073115

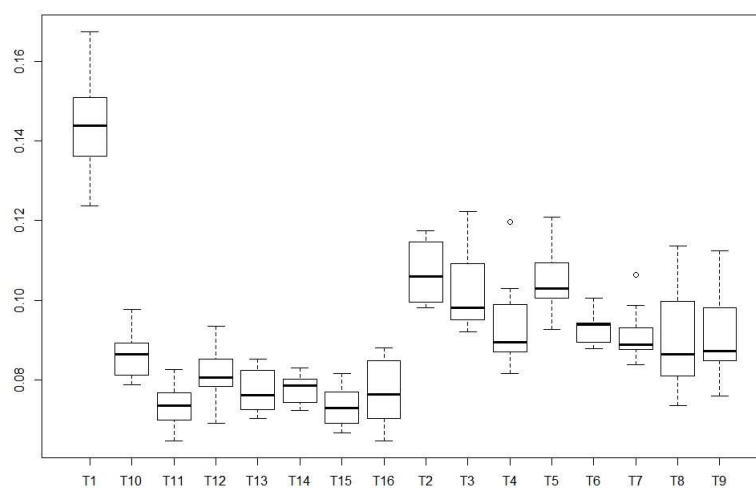


Figura A.15. Gráfico *boxplot* - Recombinação 4 e Mutação 1

A.1.3.4 SPEA2 - Recombinação 2 e Mutação 2

Para esta abordagem, o tratamento (T12) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.20 e Figura A.16, respectivamente.

Tabela A.20. Teste de *Scott-Knott* - SPEA2 - Recombinação 2 e Mutação 2

Grupos	Níveis	Médias
a	T2	0,307076
a	T1	0,270574
a	T3	0,268133
a	T6	0,249779
b	T5	0,225246
b	T9	0,215533
b	T10	0,210913
b	T7	0,202118
c	T4	0,183227
c	T14	0,182496
c	T13	0,153732
c	T15	0,143736
c	T11	0,14354
d	T8	0,127928
d	T16	0,099495
d	T12	0,098368

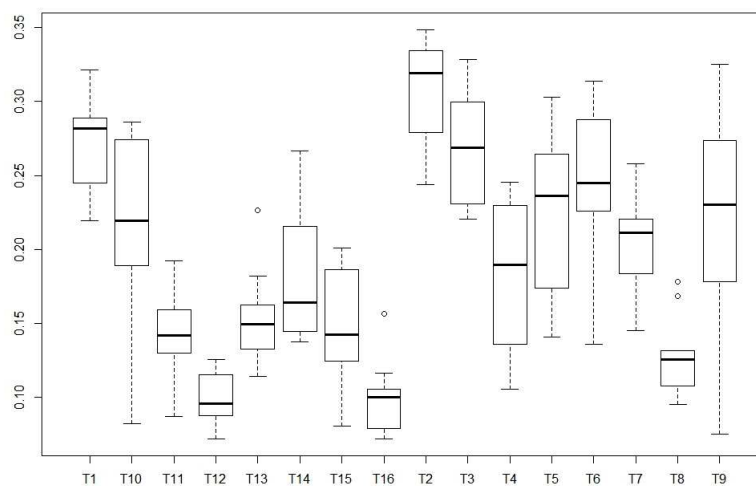


Figura A.16. Gráfico *boxplot* - Recombinação 2 e Mutação 2

A.1.3.5 SPEA2 - Recombinação 3 e Mutação 2

Para esta abordagem, o tratamento (T12) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.21 e Figura A.17, respectivamente.

Tabela A.21. Teste de *Scott-Knott* - SPEA2 - Recombinação 3 e Mutação 2

Grupos	Níveis	Médias
a	T13	0,343862
a	T14	0,341471
a	T16	0,336818
a	T15	0,331941
b	T1	0,144345
c	T5	0,133051
c	T9	0,130382
d	T2	0,119463
d	T6	0,118615
d	T10	0,116493
e	T11	0,1061
e	T3	0,105996
e	T7	0,104693
e	T4	0,102536
e	T8	0,101913
e	T12	0,097742

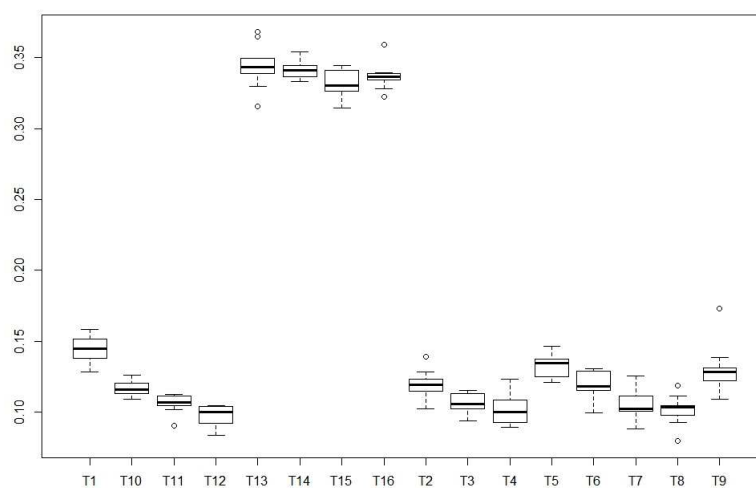


Figura A.17. Gráfico *boxplot* - Recombinação 3 e Mutação 2

A.1.3.6 SPEA2 - Recombinação 4 e Mutação 2

Para esta abordagem, o tratamento (T13) foi definido para execução do conjunto de instâncias. Teste de *Scott-Knott* e gráfico *boxplot* são apresentados na Tabela A.22 e Figura A.22, respectivamente.

Tabela A.22. Teste de *Scott-Knott* - SPEA2 - Recombinação 4 e Mutação 2

Grupos	Níveis	Médias
a	T1	0,140297
b	T2	0,130119
b	T4	0,127748
b	T3	0,127727
c	T8	0,107529
c	T7	0,104813
c	T6	0,102976
c	T5	0,102411
d	T12	0,091114
d	T11	0,087789
e	T10	0,081103
e	T9	0,07963
e	T16	0,076065
f	T15	0,066019
f	T14	0,064015
f	T13	0,05883

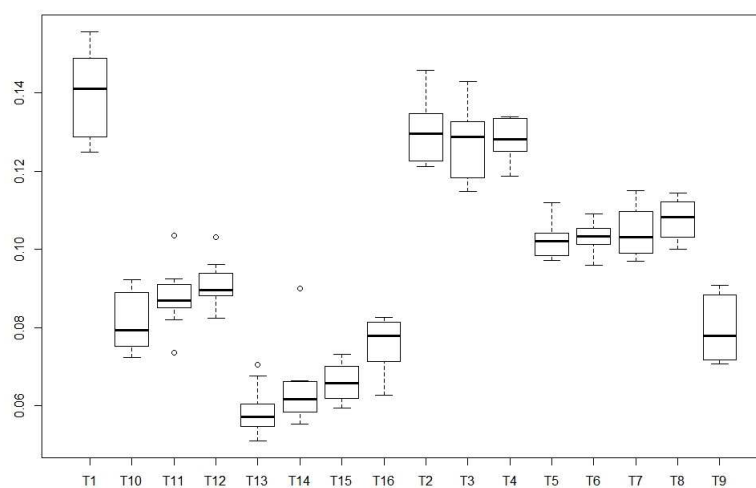


Figura A.18. Gráfico *boxplot* - Recombinação 4 e Mutação 2

Referências Bibliográficas

- Abraham, A. & Jain, L. (2005). *Evolutionary multiobjective optimization*. Springer.
- Ambrosino, G.; Nelson, J. & Romanazzo, M. (2004). *Demand responsive transport services: Towards the flexible mobility agency*. ENEA, Italian National Agency for New Technologies, Energy and the Environment.
- Attanasio, A.; Cordeau, J.-F.; Ghiani, G. & Laporte, G. (2004). Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem. *Parallel Computing*, 30(3):377--387.
- Beaudry, A.; Laporte, G.; Melo, T. & Nickel, S. (2010). Dynamic transportation of patients in hospitals. *OR spectrum*, 32(1):77--107.
- Berbeglia, G.; Cordeau, J.-F. & Laporte, G. (2012). A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem. *INFORMS Journal on Computing*, 24(3):343--355.
- Braekers, K.; Caris, A. & Janssens, G. K. (2014). Exact and meta-heuristic approach for a general heterogeneous dial-a-ride problem with multiple depots. *Transportation Research Part B: Methodological*, 67:166--186.
- Brake, J.; Nelson, J. D. & Wright, S. (2004). Demand responsive transport: towards the emergence of a new market segment. *Journal of Transport Geography*, 12(4):323--337.
- Bräysy, O. & Gendreau, M. (2005a). Vehicle routing problem with time windows, part I: Route construction and local search algorithms. *Transportation science*, 39(1):104--118.
- Bräysy, O. & Gendreau, M. (2005b). Vehicle routing problem with time windows, part II: Metaheuristics. *Transportation science*, 39(1):119--139.

- Chevrier, R. (2008). *Optimisation de Transport á la Demande dans des territoires polarisés*. PhD thesis, UMR ESPACE (CNRS 6012), Université d'Avignon et des Pays de Vaucluse.
- Chevrier, R.; Liefoghe, A.; Jourdan, L. & Dhaenens, C. (2010). On optimizing a demand responsive transport with an evolutionary multi-objective approach. Em *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*, pp. 575--580. IEEE.
- Chevrier, R.; Liefoghe, A.; Jourdan, L. & Dhaenens, C. (2012). Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport. *Applied Soft Computing*, 12(4):1247--1258.
- Chiang, T.-C. & Hsu, W.-H. (2014). A knowledge-based evolutionary algorithm for the multiobjective vehicle routing problem with time windows. *Computers & Operations Research*, 45:25--37.
- Clarke, G. & Wright, J. W. (1964). Scheduling of vehicles from a central depot to a number of delivery points. *Operations research*, 12(4):568--581.
- Coello, C. C.; Lamont, G. B. & Van Veldhuizen, D. A. (2007). *Evolutionary algorithms for solving multi-objective problems*. Springer.
- Colorni, A. & Righini, G. (2001). Modeling and optimizing dynamic dial-a-ride problems. *International transactions in operational research*, 8(2):155--166.
- Cordeau, J.-F. (2006). A branch-and-cut algorithm for the dial-a-ride problem. *Operations Research*, 54(3):573--586.
- Cordeau, J.-F. & Laporte, G. (2003). A tabu search heuristic for the static multi-vehicle dial-a-ride problem. *Transportation Research Part B: Methodological*, 37(6):579--594.
- Cordeau, J.-F. & Laporte, G. (2007). The dial-a-ride problem: models and algorithms. *Annals of Operations Research*, 153(1):29--46.
- Cordeau, J.-F.; Laporte, G.; Potvin, J.-Y. & Savelsbergh, M. W. (2007). Transportation on demand. *Transportation, handbooks in operations research and management science*, 14:429--466.
- Coslovich, L.; Pesenti, R. & Ukovich, W. (2006). A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem. *European Journal of Operational Research*, 175(3):1605--1615.

- Crainic, T. G.; Errico, F.; Malucelli, F. & Nonato, M. (2012). Designing the master schedule for demand-adaptive transit systems. *Annals of Operations Research*, 194(1):151--166.
- Daganzo, C. F. (1978). An approximate analytic model of many-to-many demand responsive transportation systems. *Transportation Research*, 12(5):325--333.
- Dantzig, G. B. & Ramser, J. H. (1959). The truck dispatching problem. *Management science*, 6(1):80--91.
- Deb, K. (2001). *Multi-objective optimization using evolutionary algorithms*, volume 16. John Wiley & Sons.
- Deb, K.; Pratap, A.; Agarwal, S. & Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: Nsga-ii. *Evolutionary Computation, IEEE Transactions on*, 6(2):182--197.
- Edgeworth, F. Y. (1881). *Mathematical psychics: An essay on the application of mathematics to the moral sciences*. Number 10. CK Paul.
- Eiben, A. E. & Smith, J. E. (2003). *Introduction to evolutionary computing*, volume 53. Springer.
- Errico, F.; Crainic, T. G.; Malucelli, F. & Nonato, M. (2013). A survey on planning semi-flexible transit systems: methodological issues and a unifying framework. *Transportation Research Part C: Emerging Technologies*, 36:324--338.
- Executive, G. M. P. T.; Enoch, M.; Potter, S.; Parkhurst, G. & Smith, M. (2004). *Intermode: Innovations in demand responsive transport*.
- Fonseca, C. M.; Knowles, J. D.; Thiele, L. & Zitzler, E. (2005). A tutorial on the performance assessment of stochastic multiobjective optimizers. Em *Third International Conference on Evolutionary Multi-Criterion Optimization (EMO 2005)*, volume 216, p. 240.
- Gomes, R.; de Sousa, J. P. & Dias, T. G. (2014). A grasp-based approach for demand responsive transportation. *Int J Transport*, 2:21--32.
- Hanne, T.; Melo, T. & Nickel, S. (2009). Bringing robustness to patient flow management through optimized patient transports in hospitals. *Interfaces*, 39(3):241--255.

- Hansen, M. P. & Jaszkiwicz, A. (1998). *Evaluating the quality of approximations to the non-dominated set*. IMM, Department of Mathematical Modelling, Technical University of Denmark.
- Horn, M. (2002a). Multi-modal and demand-responsive passenger transport systems: a modelling framework with embedded control systems. *Transportation Research Part A: Policy and Practice*, 36(2):167 – 188.
- Horn, M. E. (2002b). Fleet scheduling and dispatching for demand-responsive passenger services. *Transportation Research Part C: Emerging Technologies*, 10(1):35 – 63.
- IDEC (2013). Transporte público, insatisfação coletiva. <http://www.idec.org.br/em-acao/revista/a-espera-de-qualidade/materia/transporte-publico-insatisfaco-coletiva>. [Online; accessed 15-Maio-2016].
- Jaw, J.-J.; Odoni, A. R.; Psaraftis, H. N. & Wilson, N. H. (1986). A heuristic algorithm for the multi-vehicle advance request dial-a-ride problem with time windows. *Transportation Research Part B: Methodological*, 20(3):243--257.
- Jozefowicz, N.; Semet, F. & Talbi, E.-G. (2008). Multi-objective vehicle routing problems. *European journal of operational research*, 189(2):293--309.
- Liu, M.; Luo, Z. & Lim, A. (2015). A branch-and-cut algorithm for a realistic dial-a-ride problem. *Transportation Research Part B: Methodological*, 81:267--288.
- Madsen, O. B.; Ravn, H. F. & Rygaard, J. M. (1995). A heuristic algorithm for a dial-a-ride problem with time windows, multiple capacities, and multiple objectives. *Annals of operations Research*, 60(1):193--208.
- Mageean, J. & Nelson, J. D. (2003). The evaluation of demand responsive transport services in europe. *Journal of Transport Geography*, 11(4):255--270.
- Mendes, R. S. et al. (2016). Abordagem multiobjetivo para o problema de roteamento de veículos com transporte reativo a demanda. Master's thesis, Programa de Pós-Graduação em Modelagem Matemática e Computacional, Centro Federal de Educação Tecnológica de Minas Gerais.
- Miranda, D. S. et al. (2012). Aplicação de metaheurísticas para o problema de roteamento de veículos dinâmico para transporte reativo a demanda. Master's thesis, Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Viçosa.

- Paquette, J.; Cordeau, J.-F. & Laporte, G. (2009). Quality of service in dial-a-ride operations. *Computers & Industrial Engineering*, 56(4):1721--1734.
- Pareto, V. (1896). Cours deconomie, vol. I and II, f. Rouge, Lausanne.
- Parragh, S. N. (2011). Introducing heterogeneous users and vehicles into models and algorithms for the dial-a-ride problem. *Transportation Research Part C: Emerging Technologies*, 19(5):912--930.
- Parragh, S. N.; Cordeau, J.-F.; Doerner, K. F. & Hartl, R. F. (2012). Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints. *OR spectrum*, 34(3):593--633.
- Parragh, S. N.; Doerner, K. F. & Hartl, R. F. (2008). A survey on pickup and delivery problems. part II: transportation between pickup and delivery locations. *Journal für Betriebswirtschaft*, 58(1):81--117.
- Parragh, S. N.; Doerner, K. F. & Hartl, R. F. (2010a). Demand responsive transportation. *Wiley Encyclopedia of Operations Research and Management Science*.
- Parragh, S. N.; Doerner, K. F. & Hartl, R. F. (2010b). Variable neighborhood search for the dial-a-ride problem. *Computers & Operations Research*, 37(6):1129--1138.
- Parragh, S. N.; Doerner, K. F.; Hartl, R. F. & Gandibleux, X. (2009). A heuristic two-phase solution approach for the multi-objective dial-a-ride problem. *Networks*, 54(4):227--242.
- Parragh, S. N. & Schmid, V. (2013). Hybrid column generation and large neighborhood search for the dial-a-ride problem. *Computers & operations research*, 40(1):490--497.
- Pillac, V.; Gendreau, M.; Guéret, C. & Medaglia, A. L. (2013). A review of dynamic vehicle routing problems. *European Journal of Operational Research*, 225(1):1--11.
- Psaraftis, H. N. (1980). A dynamic programming solution to the single vehicle many-to-many immediate request dial-a-ride problem. *Transportation Science*, 14(2):130--154.
- Psaraftis, H. N. (1983). An exact algorithm for the single vehicle many-to-many dial-a-ride problem with time windows. *Transportation Science*, 17(3):351--357.
- Psaraftis, H. N.; Wen, M. & Kontovas, C. A. (2016). Dynamic vehicle routing problems: Three decades and counting. *Networks*, 67(1):3--31.

- Ropke, S.; Cordeau, J.-F. & Laporte, G. (2007). Models and branch-and-cut algorithms for pickup and delivery problems with time windows. *Networks*, 49(4):258--272.
- Schilde, M.; Doerner, K. F. & Hartl, R. F. (2011). Metaheuristics for the dynamic stochastic dial-a-ride problem with expected return transports. *Computers & operations research*, 38(12):1719--1730.
- Sexton, T. R. & Bodin, L. D. (1985a). Optimizing single vehicle many-to-many operations with desired delivery times: I. scheduling. *Transportation Science*, 19(4):378--410.
- Sexton, T. R. & Bodin, L. D. (1985b). Optimizing single vehicle many-to-many operations with desired delivery times: II. routing. *Transportation Science*, 19(4):411--435.
- Srinivas, N. & Deb, K. (1994). Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary computation*, 2(3):221--248.
- Talbi, E.-G. (2009). *Metaheuristics: from design to implementation*, volume 74. John Wiley & Sons.
- Toth, P. & Vigo, D. (1996). Fast local search algorithms for the handicapped persons transportation problem. Em *Meta-Heuristics*, pp. 677--690. Springer.
- Toth, P. & Vigo, D. (1997). Heuristic algorithms for the handicapped persons transportation problem. *Transportation Science*, 31(1):60--71.
- Toth, P. & Vigo, D. (2002). *The Vehicle Routing Problem*. Monographs on Discrete Mathematics and Applications. Society for Industrial and Applied Mathematics.
- Toth, P. & Vigo, D. (2014). *Vehicle routing: problems, methods, and applications*, volume 18. Siam.
- Viana, R. J. d. S. & dos Santos, A. G. (2015). Dynamic approach for a demand responsive transport service. Em *XLI Conferencia Latinoamericana en Informática (CLEI 2015)*, pp. 1--12. IEEE.
- Viana, R. J. d. S.; dos Santos, A. G. & Arroyo, J. E. C. (2015). Multi-objective evolutionary approach for optimizing a demand responsive transport. Em *Simpósio Brasileiro de Pesquisa Operacional, 2015*, pp. 1918--1929. SOBRAPO.
- Wilson, N.; Sussman, J.; Wong, H. & Higonnet, B. (1971). Scheduling algorithms for dial-a-ride systems. *Urban Systems Laboratory, Massachusetts Institute of Technology, Cambridge, MA*.

- Wilson, N. H. & Colvin, N. J. (1977). Computer control of the rochester dial-a-ride system. Technical report, Massachusetts Institute of Technology, Center for Transportation Studies.
- Wilson, N. H. & Wayne, W. R. (1976). Advanced dial-a-ride algorithms research project. Technical report, Massachusetts Institute of Technology, Center for Transportation Studies.
- Xiang, Z.; Chu, C. & Chen, H. (2008). The study of a dynamic dial-a-ride problem under time-dependent and stochastic environments. *European Journal of Operational Research*, 185(2):534 – 551.
- Yu, X. & Gen, M. (2010). *Introduction to evolutionary algorithms*. Springer Science & Business Media.
- Zitzler, E. & Künzli, S. (2004). Indicator-based selection in multiobjective search. Em *Parallel Problem Solving from Nature-PPSN VIII*, pp. 832--842. Springer.
- Zitzler, E.; Laumanns, M.; Thiele, L.; Zitzler, E.; Zitzler, E.; Thiele, L. & Thiele, L. (2001). SPEA2: Improving the strength pareto evolutionary algorithm. Technical report, Eidgenössische Technische Hochschule Zürich (ETH), Institut für Technische Informatik und Kommunikationsnetze (TIK).
- Zitzler, E. & Thiele, L. (1998). Multiobjective optimization using evolutionary algorithms - a comparative case study. Em *Parallel problem solving from nature - PPSN V*, pp. 292--301. Springer.
- Zitzler, E. & Thiele, L. (1999). Multiobjective evolutionary algorithms: a comparative case study and the strength pareto approach. *evolutionary computation, IEEE transactions on*, 3(4):257--271.
- Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C. M. & Da Fonseca, V. G. (2003). Performance assessment of multiobjective optimizers: An analysis and review. *Evolutionary Computation, IEEE Transactions on*, 7(2):117--132.