



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

RENÊ DOUGLAS NOBRE DE MORAIS

**ALGORITMO INTELIGENTE PARA GERAÇÃO DE
ROTAS EM *SMART CITIES***

DISSERTAÇÃO DE MESTRADO

Recife
2018

RENÊ DOUGLAS NOBRE DE MORAIS

**ALGORITMO INTELIGENTE PARA GERAÇÃO DE
ROTAS EM *SMART CITIES***

Dissertação de Mestrado apresentado ao Programa de Pós-Graduação em Informática Aplicada, como parte dos requisitos necessários à obtenção do título de Mestre.

Orientador: Cícero Garrozi

Recife
2018

Dados Internacionais de Catalogação na Publicação (CIP)
Sistema Integrado de Bibliotecas da UFRPE
Biblioteca Central, Recife-PE, Brasil

M827a Morais, Renê Douglas Nobre de.
 Algoritmo inteligente para geração de rotas em Smart Cities / Renê Douglas
 Nobre de Morais. – Recife, 2018.
 113 f.: il.

 Orientador(a): Cícero Garrozi.
 Dissertação (Mestrado) – Universidade Federal Rural de
 Pernambuco, Programa de Pós-Graduação em Informática Aplicada, Recife, BR-PE,
 2018.

 Inclui referências e apêndice(s).

 1. Engenharia de tráfego 2. Trânsito - Fluxo 3. Trânsito congestionamento
 5. Computação evolucionária 6. Openstreetmap I. Garrozi, Cícero II. Título

CDD 004



UNIVERSIDADE FEDERAL RURAL DE PERNAMBUCO
DEPARTAMENTO DE ESTATÍSTICA E INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA

ALGORITMO INTELIGENTE PARA GERAÇÃO DE ROTAS EM
SMART CITIES

RENÊ DOUGLAS NOBRE DE MORAIS

Dissertação julgada adequada para
obtenção do título de Mestre em
Informática Aplicada, defendida e
aprovada por unanimidade em
27/08/2018 pela Banca Examinadora.

Banca Examinadora:

Prof. Dr. Cícero Garrozi
Universidade Federal Rural de Pernambuco
Orientador

Prof. Dr. Tiago Alessandro Espínola Ferreira
Universidade Federal Rural de Pernambuco

Prof. Dr. Aluizio Fausto Ribeiro Araújo
Universidade Federal de Pernambuco

*Dedico este trabalho a Deus, que foi o meu maior apoio nos momentos mais difíceis.
Dedico à minha mãe por sempre apoiar e acreditar em mim e ao meu pai a quem
sentirei eternas saudades. Dedico aos meus tios que por boa vontade me acolheram
como um filho por todo este tempo.*

Agradecimentos

Agradeço ao Senhor, por ter me dado forças espiritual e mental que permitiram a realização deste sonho.

Agradeço ao meu orientador e professor Cícero Garrozi por toda a sua dedicação, preocupação e amizade ao longo de toda esta trajetória. Dentre outras qualidades que herdarei para a vida como um futuro docente, se Deus quiser.

Agradeço à minha família por todo apoio dado, em especial aos meus tios pelo carinho, acolhimento e compreensão durante todo o tempo que precisei estar.

Agradeço ao meu amor, Janaina, por todo o companheirismo, carinho e paciência que nos fizeram superar inúmeras barreiras. Por ser a minha principal motivação de nunca desistir e sempre seguir em frente.

Agradeço aos meus amigos que estiveram presentes por toda esta trajetória e que me ajudaram e me incentivaram todo este tempo.

Agradeço a todos os docentes do PPGIA envolvidos, contribuindo imensamente no meu crescimento profissional.

Agradeço, por fim, ao DEINFO e a UFRPE pelo fornecimento dos recursos e ambiente de estudos necessários e a todos os funcionários pelo apoio administrativo prestado.

*“The cave you fear to enter holds the treasure
you seek.”*

— Joseph Campbell

Resumo

O congestionamento nos grandes centros urbanos é um dos principais problemas para as pessoas que necessitam se locomover diariamente seja utilizando o transporte público ou individual. O surgimento do conceito de cidades inteligentes tornou possível o desenvolvimento de inúmeras soluções inovadoras que, através da tecnologia, compartilham e disseminam diversas informações em tempo real para a população. Porém, à medida que o problema é tomado por inúmeras variáveis, torna-se cada vez mais difícil oferecer soluções viáveis e em tempo hábil. Desta forma, este trabalho propõe um algoritmo que utiliza Pontos de Referência para determinação de passeios em redes viárias de larga-escala considerando 3 objetivos (tempo de viagem, distância e número de semáforos) em tempo hábil utilizando um algoritmo evolucionário multiobjetivo. Além disso, um novo operador de mutação que garante apenas pequenas alterações nos indivíduos é proposto. A abordagem proposta é testada realizando simulações em um mapa de uma cidade grande com diferentes níveis de congestionamento utilizando os dados do *Openstreetmap*. Por fim, os resultados das simulações são comparados com soluções geradas pelo algoritmo de Dijkstra que representam o menor valor teórico para cada objetivo. Foram realizados 5 grupos de simulações com 3 cenários cada com diferentes níveis de congestionamento. Os resultados mostraram que o algoritmo encontrou diversos *trade-offs*, destacando os melhores resultados para os maiores níveis de congestionamento com perdas médias que não ultrapassaram os 20% em 2 dos 3 objetivos em grande parte das simulações. Quanto ao tempo computacional gasto, em 2 grupos o algoritmo demorou em média 3 segundos para encontrar as melhores rotas e 6 segundos em média para os outros 3 grupos. Desta forma, concluiu-se que o algoritmo foi capaz de gerar *trade-offs* viáveis em um bom tempo computacional considerando o ambiente em que foi executado. Estes resultados também refletiram a eficiência do operador de mutação proposto.

Palavras-chave: Cidades Inteligentes; Congestionamento de Veículos; Otimização Multiobjetivo; Computação Evolucionária; *Openstreetmap*; Problema do Menor Caminho Multiobjetivo.

Abstract

The traffic congestion in large urban centers is one of the main problems for people who need to drive daily. The concept of smart cities has made possible the development of innumerable innovative solutions that, through technology, share and disseminate various information in real time to the population. However, these problems have numerous variables that make it difficult to resolve them in satisfactory time. In this way, this work proposes an algorithm that uses Points of Reference to determine rides in large-scale road networks considering 3 objectives (travel time, distance and number of traffic lights) in a satisfactory time using an evolutionary multiobjective algorithm. In addition, a new mutation operator that guarantees only minor changes in individuals is proposed. The proposed approach is tested by performing simulations on a map of a large city with different levels of traffic congestion using Openstreetmap data. Finally, the results of the simulations are compared with solutions generated by the Dijkstra's algorithm that represent the lowest theoretical value for each objective. Five sets of simulations were carried out with 3 scenarios each with different levels of traffic congestion. The results showed that the algorithm found viable trade-offs, highlighting the best results for the highest levels of congestion with average losses that did not exceed 20% in 2 of the 3 objectives in most of the simulations when compared with the theoretical minimum values. As for the computational time spent, in 2 groups the algorithm took on average 3 seconds to find the best routes and 6 seconds on average for the other 3 groups. In this way, it was concluded that the algorithm was able to generate viable trade-offs in a good computational time considering the environment in which it was executed. These results also reflected the efficiency of the proposed mutation operator.

Keywords: Smart Cities; Traffic Congestion; Multi-objective Optimization; Evolutionary Computing; Openstreetmap; Multi-objective Shortest Path Problem.

Lista de ilustrações

Figura 1 – Exemplo de um cromossomo por Representação Binária.	28
Figura 2 – Exemplo de um cromossomo utilizando Representação por Inteiro. .	29
Figura 3 – Exemplo de um processo de mutação binária	30
Figura 4 – Exemplo de um processo de mutação por <i>Random Resetting</i>	31
Figura 5 – Exemplo de um processo de mutação por <i>Creep Mutation</i>	31
Figura 6 – Exemplo de <i>Swap Mutation</i>	32
Figura 7 – Exemplo de <i>Insert Mutation</i>	33
Figura 8 – Exemplo de <i>Scramble Mutation</i>	33
Figura 9 – Exemplo de <i>Inversion Mutation</i>	33
Figura 10 – Exemplo de <i>one-point crossover</i>	34
Figura 11 – Exemplo de <i>N-Point Crossover</i>	34
Figura 12 – Exemplo de <i>Uniform Crossover</i>	35
Figura 13 – Exemplo de <i>Whole Arithmetic Recombination</i>	36
Figura 14 – Exemplo de <i>Order Crossover</i>	36
Figura 15 – Exemplo de mapeamento dos indivíduos no método da Roleta. . . .	38
Figura 16 – Exemplo de um mapa exportado diretamente do OSM.	49
Figura 17 – Mapa da Figura 16 utilizando o código 3.1.	50
Figura 18 – Exemplo de restrição de conversão e solução proposta	52
Figura 19 – Representação das listas de caminhos ordenadas por métrica entre dois Pontos de Referência.	54
Figura 20 – Representação do cromossomo no MOEA.	56
Figura 21 – Fluxograma do processo de preenchimento dos cromossomos selevi- cionados para preenchimento privilegiando os 10 menores caminhos de cada lista ordenada.	56
Figura 22 – Representação em pseudocódigo do operador de mutação proposto. .	58
Figura 23 – Exemplo de como um caminho pode estar indexado nas tabelas de acordo com cada métrica durante o processo de mutação.	59
Figura 24 – Pontos de Referência distribuídos pelo mapa OSM da cidade de Recife. .	61
Figura 25 – Mapa viário de parte da cidade do Recife representado no JOSM utilizado para realização das simulações.	63
Figura 26 – Configuração dos 5 passeios utilizados nas simulações.	64
Figura 27 – Comparação entre a Frente de Pareto e os menores caminhos com- putados pelo Dijkstra para o grupo 1 sem congestionamento.	65
Figura 28 – Comparação entre a Frente de Pareto e os menores caminhos com- putados pelo Dijkstra para o grupo 1 com 50% de congestionamento.	66

Figura 29 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 1 com 75% de congestionamento.	67
Figura 30 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 75% de congestionamento do grupo 1.	68
Figura 31 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 2 sem congestionamento.	70
Figura 32 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 2 com 50% de congestionamento.	71
Figura 33 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 2 com 75% de congestionamento.	72
Figura 34 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 75% de congestionamento do grupo 2.	72
Figura 35 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 3 sem congestionamento.	75
Figura 36 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 3 com 50% de congestionamento.	76
Figura 37 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 50% de congestionamento do grupo 3.	76
Figura 38 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 3 com 75% de congestionamento.	77
Figura 39 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 4 sem congestionamento.	79
Figura 40 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 4 com 50% de congestionamento.	80
Figura 41 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 4 com 75% de congestionamento.	81
Figura 42 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 75% de congestionamento do grupo 4.	81
Figura 43 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 4.	82
Figura 44 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 5 sem congestionamento.	83
Figura 45 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 5 com 50% de congestionamento.	84
Figura 46 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 50% de congestionamento do grupo 5.	85
Figura 47 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 5 com 75% de congestionamento.	85
Figura 48 – Exemplo de um passeio gerado no cenário 1 do grupo 1.	100

Figura 49 – Exemplo de um passeio gerado no cenário 2 do grupo 1.	100
Figura 50 – Exemplo de um passeio gerado no cenário 3 do grupo 1.	100
Figura 51 – Exemplo de um passeio gerado no cenário 1 do grupo 2.	103
Figura 52 – Exemplo de um passeio gerado no cenário 2 do grupo 2.	103
Figura 53 – Exemplo de um passeio gerado no cenário 3 do grupo 2.	103
Figura 54 – Exemplo de um passeio gerado no cenário 1 do grupo 3.	106
Figura 55 – Exemplo de um passeio gerado no cenário 2 do grupo 3.	106
Figura 56 – Exemplo de um passeio gerado no cenário 3 do grupo 3.	106
Figura 57 – Exemplo de passeio gerado no cenário 1 do grupo 4.	109
Figura 58 – Exemplo de passeio gerado no cenário 2 do grupo 4.	109
Figura 59 – Exemplo de passeio gerado no cenário 3 do grupo 4.	109
Figura 60 – Exemplo de passeio gerado no cenário 1 do grupo 5.	112
Figura 61 – Exemplo de passeio gerado no cenário 2 do grupo 5.	112
Figura 62 – Exemplo de passeio gerado no cenário 3 do grupo 5.	112

Lista de quadros

Quadro 1 – Classificação das vias e limites de velocidade definidos pelo OSM.	62
Quadro 2 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 1.	69
Quadro 3 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 1.	69
Quadro 4 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 2.	73
Quadro 5 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 2.	74
Quadro 6 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 3.	78
Quadro 7 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 3.	78
Quadro 8 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 4.	82
Quadro 9 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 5.	86
Quadro 10 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 5.	86
Quadro 11 – Média e desvio padrão gerais e específicos dos custos excedidos pelas soluções de Pareto dos grupos e cenários para o Tempo de Viagem.	88
Quadro 12 – Média e desvio padrão gerais e específicos dos custos excedidos pelas soluções de Pareto dos grupos e cenários para a Distância. .	89
Quadro 13 – Média e desvio padrão gerais e específicos dos custos excedidos pelas soluções de Pareto dos grupos e cenários para o Número de Semáforos.	89
Quadro 14 – Tempo computacional gasto pelo NSGA-III para cada grupo e seus respectivos cenários.	89
Quadro 15 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 1 (parte 1).	98
Quadro 16 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 1 (parte 2).	99

Quadro 17 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 2 (parte 1).	101
Quadro 18 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 2 (parte 2).	102
Quadro 19 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 3 (parte 1).	104
Quadro 20 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 3 (parte 2).	105
Quadro 21 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 4 (parte 1).	107
Quadro 22 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 4 (parte 2).	108
Quadro 23 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 5 (parte 1).	110
Quadro 24 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 5 (parte 2).	111

Lista de abreviaturas e siglas

AE	Algoritmo Evolucionário
AG	Algoritmo Genético
CE	Computação Evolucionária
CVRP	Capacitated Vehicle Routing Problem
DENATRAN	Departamento Nacional de Trânsito
JOSM	OpenStreetMap Editor
MOEA	Multiobjective Evolutionary Algorithm
MOP	Multiobjective Optimization Problem
MOSP	Multi-objective Shortest Path Problem
NSGA	Nondominated Sorting Genetic Algorithm
OSM	OpenStreetMap
PR	Ponto de Referência
SPEA	Strength Pareto Evolutionary Algorithm
STDEV	Desvio Padrão
TSP	Travelling Salesman Problem
VEGA	Algoritmo Genético de Avaliação Vetorial
VRP	Vehicle Routing Problem

Sumário

1	INTRODUÇÃO	19
1.1	Problema de Pesquisa	22
1.2	Justificativa	23
1.3	Objetivos	24
1.3.1	Geral	24
1.3.2	Específicos	24
1.4	Limitações do trabalho	24
1.5	Organização da dissertação	25
2	REVISÃO DA LITERATURA	26
2.1	Formulação do Problema	26
2.2	Algoritmos Evolucionários	27
2.2.1	Representação dos Indivíduos	27
2.2.1.1	Representação Binária	28
2.2.1.2	Representação por Inteiro	28
2.2.1.3	Representação por Valores Reais ou Ponto-Flutuante	29
2.2.1.4	Representação por Permutação	29
2.2.2	Mutação	29
2.2.2.1	Mutação em Cromossomos de Representação Binária	30
2.2.2.2	Mutação em Cromossomos de Representação Inteira	30
2.2.2.3	Mutação em Cromossomos de Representação por Ponto-Flutuante	31
2.2.2.4	Mutação em Cromossomos de Representação por Permutação	32
2.2.3	Recombinação	33
2.2.3.1	Recombinação em Cromossomos de Representação Inteira	35
2.2.3.2	Recombinação em Cromossomos de Representação por Ponto-Flutuante	35
2.2.3.3	Recombinação em Cromossomos de Representação por Permutação	36
2.2.4	Modelos de População	36
2.2.5	Seleção de Pais	37
2.2.5.1	Seleção por Classificação	37
2.2.5.2	Seleção Proporcional à Aptidão	37
2.2.5.3	Seleção por Torneio	38
2.2.6	Seleção de Sobreviventes	39
2.2.6.1	Seleção Baseada na Idade dos Indivíduos	39
2.2.6.2	Seleção Baseada na Aptidão dos Indivíduos	39
2.3	Algoritmos Evolucionários Multiobjetivo (MOEA)	40
2.3.1	Operações Básicas	40

2.3.2	Classificação dos MOEAs	41
2.3.2.1	Funções de Agregação	41
2.3.2.2	Abordagem Baseada na População	41
2.3.2.3	Abordagem Baseada em Pareto	42
2.3.3	<i>The Nondominated Sorting Genetic Algorithm</i> (NSGA-III)	43
2.3.4	MOEAs aplicados a problemas de roteamento de veículos	44
3	METODOLOGIA	48
3.1	Tratamento dos dados geoespaciais do <i>OpenStreetMap</i>	48
3.2	Implementação de um algoritmo para criação do grafo de rodovias utilizando dados do <i>OpenStreetMap</i>	50
3.2.1	Solucionando o problema do nó compartilhado e sua representação em um grafo	51
3.3	Pré-processamento dos k-menores caminhos no grafo de rodovias	52
3.4	Representação do problema e configuração dos operadores no MOEA	54
3.4.1	Função de Aptidão	55
3.4.2	População inicial	55
3.4.3	Operadores Genéticos	57
3.4.3.1	Operador de Mutação Proposto	57
3.4.3.2	Operadores de Recombinação, Seleção de Pais e Sobreviventes	59
3.5	Simulações	59
3.5.1	Configuração dos passeios para as simulações e definição dos cenários de congestionamento	60
4	RESULTADOS	63
4.1	Análise do primeiro grupo de simulações	64
4.1.1	Cenário sem congestionamento	65
4.1.2	Cenário com 50% de congestionamento	66
4.1.3	Cenário com 75% de congestionamento	67
4.1.4	Análise do desempenho do algoritmo para todos os cenários do primeiro grupo de simulações	68
4.2	Análise do segundo grupo de simulações	69
4.2.1	Cenário sem congestionamento	70
4.2.2	Cenário com 50% de congestionamento	70
4.2.3	Cenário com 75% de congestionamento	71
4.2.4	Análise do desempenho do algoritmo para todos os cenários do segundo grupo de simulações	73
4.3	Análise do terceiro grupo de simulações	74
4.3.1	Cenário sem congestionamento	74

4.3.2	Cenário com 50% de congestionamento	75
4.3.3	Cenário com 75% de congestionamento	77
4.3.4	Análise do desempenho do algoritmo para todos os cenários do terceiro grupo de simulações	77
4.4	Análise do quarto grupo de simulações	78
4.4.1	Cenário sem congestionamento	78
4.4.2	Cenário com 50% de congestionamento	79
4.4.3	Cenário com 75% de congestionamento	80
4.4.4	Análise do desempenho do algoritmo para todos os cenários do quarto grupo de simulações	82
4.5	Análise do quinto grupo de simulações	83
4.5.1	Cenário sem congestionamento	83
4.5.2	Cenário com 50% de congestionamento	83
4.5.3	Cenário com 75% de congestionamento	85
4.5.4	Análise do desempenho do algoritmo para todos os cenários do quinto grupo de simulações	86
4.6	Análise geral do desempenho do algoritmo para todos os grupos	86
5	CONCLUSÃO	90
5.1	Trabalhos futuros	92
	REFERÊNCIAS	94
	APÊNDICES	97
	APÊNDICE A – SOLUÇÕES DAS FRENTES DE PARETO PARA O GRUPO 1 E EXEMPLOS GERADOS DOS SEUS CENÁRIOS DE CONGESTIONAMENTO	98
	APÊNDICE B – SOLUÇÕES DAS FRENTES DE PARETO PARA O GRUPO 2 E EXEMPLOS GERADOS DOS SEUS CENÁRIOS DE CONGESTIONAMENTO	101
	APÊNDICE C – SOLUÇÕES DAS FRENTES DE PARETO PARA O GRUPO 3 E EXEMPLOS GERADOS DOS SEUS CENÁRIOS DE CONGESTIONAMENTO	104
	APÊNDICE D – SOLUÇÕES DAS FRENTES DE PARETO PARA O GRUPO 4 E EXEMPLOS GERADOS DOS SEUS CENÁRIOS DE CONGESTIONAMENTO	107

APÊNDICE E – SOLUÇÕES DAS FRENTES DE PARETO PARA O GRUPO 5 E EXEMPLOS GERADOS DOS SEUS CENÁRIOS DE CONGESTIONAMENTO	110
--	------------

1 Introdução

Os congestionamentos nos grandes centros urbanos têm se tornado pauta nas discussões sobre mobilidade urbana, atualmente, na tentativa de encontrar soluções inteligentes para um problema que parece crescer cada vez mais com o adensamento de veículos. Encontrar soluções para monitorar e gerenciar esse tráfego da melhor forma é necessário para evitar que as cidades estejam em constante congestionamento.

No Brasil, segundo dados do DENATRAN (2016), cerca de 93 milhões de veículos compõem a frota nacional em dezembro de 2016, o que representa 3 milhões a mais de veículos em comparação com o mês de dezembro de 2015. Na medida em que o número de veículos aumenta, as suas consequências também se tornam cada vez mais evidentes, como acidentes de trânsito e congestionamentos. Segundo a TomTom Traffic Index (2016a), um ranking mundial de nível de congestionamento de trânsito, a cidade do Rio de Janeiro é a oitava cidade mais congestionada do mundo apresentando um nível geral de congestionamento de 47% com um pico de 63% pela manhã e 81% ao final da tarde de ocorrer alguma obstrução nas vias. Já a cidade do Recife se encontra na 43ª posição com um nível de congestionamento de 37%, 60% de pico pela manhã e 65% ao final da tarde.

Problemas de congestionamento não são exclusivos do Brasil, muitas cidades ao redor do mundo também enfrentam esse problema. Os modelos de cidade e de gerenciamento de trânsito do começo deste século e do final do século passado não conseguem mais acompanhar a complexidade do problema que exige cada vez mais um gerenciamento mais sofisticado e tecnológico. Em vista disso, o conceito de cidades inteligentes tem ajudado estudiosos a encontrarem soluções sustentáveis, integradas e inteligentes para a mobilidade urbana ao redor do mundo.

As cidades inteligentes se tornaram o melhor cenário para encontrar soluções para a mobilidade tal como o controle de tráfego de veículos dada a disponibilidade de informações sobre o trânsito através de tecnologias sofisticadas. Segundo Hollands (2008), uma cidade não pode ser rotulada como inteligente apenas por ter adotado uma infraestrutura de tecnologia da informação sofisticada, mas obter a contribuição de vários grupos de pessoas, assumindo riscos muito maiores com essa tecnologia, transferir o poder, combater as desigualdades e redefinir conceitos de forma inteligente.

Para Caragliu, Bo e Nijkamp (2011), as cidades inteligentes são investimentos sociais que incentivam o uso de tecnologias de informação e comunicação viabilizando o crescimento econômico sustentável, melhorias na qualidade de vida do cidadão, dentre outros, com participação atuante do governo. O autor ainda define 6 características

principais que tornam uma cidade inteligente: haver uma infraestrutura de conectividade que tenha a capacidade de oferecer e compartilhar diversos serviços nas mais diversas áreas como negócios, lazer, mobilidade, etc; o desenvolvimento da cidade deve ser orientado à negócios, estimulando novos negócios e atraindo mais empresas; uma forte inclusão de todas as classes sociais a se beneficiarem dos serviços públicos oferecidos na cidade; o crescimento cada vez mais de empresas ou indústrias de alta tecnologia e com culturas empresariais que estimulam a criatividade; o desenvolvimento de uma cultura social e relacional urbana onde as pessoas aprendem a se beneficiarem da tecnologia e dos serviços dispostos e, por último, uma cidade inteligente deve ter uma cultura voltada para a sustentabilidade social e ambiental.

Um exemplo de um sistema gerenciamento de tráfego em *smart cities* pode ser observado em um estudo realizado na cidade de Santander, Espanha, onde o governo elaborou um plano estratégico com um prazo de 10 anos (2010 a 2020), investindo em soluções tecnológicas inteligentes para solucionar problemas de mobilidade urbana e congestionamentos. Uma das soluções encontradas foi a utilização de 12000 sensores que ficam escondidos no solo ou em patrimônios públicos para a captura de diversas medições, dentre elas, verificar se há vagas em estacionamento, geolocalização de ônibus, dentre outros. Todos esses dispositivos estão interligados a uma central de computadores para onde eles transmitem os dados a serem processados (BAYO, 2016).

Outro estudo que pode servir como exemplo feito por Lee et al. (2016), realizado na cidade de Orlando (148ª posição no ranking mundial da TomTom Traffic Index (2016b) com menos de 20% de congestionamentos), na Flórida, mostrou como a cidade desenvolveu abordagens baseadas numa cidade inteligente para lidar com problemas de transporte, segurança e, principalmente, de emergência devido as temporadas de furacões do Atlântico que são comuns de ocorrerem na região. Uma das maiores características da cidade inclui redes de fibra ótica que compartilham informações entre transporte, segurança, acidentes rodoviários e um centro de operações de emergência que é responsável por gerenciar catástrofes naturais.

No gerenciamento de transportes e mobilidade, a cidade conta com um sistema chamado LYNX que conecta mais de 4000 paradas de ônibus provendo informações como os horários em cada parada. O sistema também oferece diversos outros serviços como definição de rotas, tempo estimado de chegada dos ônibus para a parada selecionada e tempo até o destino, situação do trânsito, dentre outros. Além disso, Orlando possui um centro de gerenciamento de tráfego que controla todos os semáforos. Quanto ao gerenciamento de catástrofes climáticas, a cidade possui um sistema chamado OCAAlert.net que possui aproximadamente 14000 assinantes registrados que permitem ao governo entrar em contato com os cidadãos durante uma catástrofe

enviando mensagens contendo instruções sobre onde ir, o que fazer e o que não fazer através de vários meios como e-mail, celular ou smartphone.

Encontrar métodos para melhor gerenciar e otimizar o tráfego de veículos têm sido o foco de muitos estudiosos. Porém, resolver problemas que envolvam uma frota de veículos em cenários característicos de grandes cidades e lidar com problemas de congestionamento não é trivial e requer que muitas variáveis sejam otimizadas simultaneamente caracterizando-os como problemas de otimização multiobjetivo. A grande maioria dos trabalhos encontrados nomeiam este campo como *Vehicle Routing Problem* (VRP), porém todos estes trabalhos, apesar de abordarem o mesmo campo trabalho nesta dissertação, possuem características adventas do próprio VRP e suas variantes que dificultaram qualquer comparação. A principal delas é a abordagem voltada para a otimização de uma frota de veículos. Porém, o problema aqui trabalhado se especificou na otimização de um conjunto de rotas para apenas um veículo. Isto será melhor explicado nas próximas seções.

Este trabalho propõe um algoritmo para determinação de um passeio considerando muitos objetivos em uma rede viária com congestionamento de veículos. Para tal, é proposto uma etapa de pré-processamento de dados baseado em Pontos de Referência no mapa com o objetivo de diminuir a complexidade do problema. Além disto, um novo operador de mutação que trabalha em um espaço-objetivo de menor tamanho e ordenado foi proposto com o objetivo de privilegiar a exploração dos vizinhos mais próximos e também agilizar o processo de busca. Para isto, foi utilizado um algoritmo evolucionário multiobjetivo (MOEA) para realizar a otimização dos caminhos.

Neste trabalho, considera-se um passeio feito por um veículo como um conjunto de rotas sequenciais bem definidas, ou seja, as rotas não podem ser trocadas entre si dentro de um mesmo passeio. A razão pela qual de optar pela otimização de um passeio e não de uma rota de cada vez a cada execução do algoritmo está em oferecer ao usuário a possibilidade de visitar vários Pontos de Referência de uma cidade sem que seja necessário realizar várias solicitações no decorrer do roteiro.

Para a realização deste trabalho, foram utilizados dados provenientes de uma plataforma de mapeamento geoespacial colaborativo, o *OpenStreetMap* (OSM). O OSM é mantido por uma comunidade voluntária de colaboradores ao redor do mundo; conta com informações provenientes de fontes governamentais assim como da própria comunidade de contribuintes que mantém as informações frequentemente atualizadas (HAKLAY; WEBER, 2008; CURRAN; CRUMLISH; FISHER, 2012).

A plataforma armazena dados descritivos sobre os locais através de *tags* que podem conter as mais variadas informações das vias (semáforos, pavimentação, tipo de via, etc.) e construções de uma cidade (pública, privada, finalidade, etc). O OSM possui diversas ferramentas para desenvolvedores e para este trabalho utilizaremos

o JOSM que também é *open source* e ainda permite a instalação de diversos *plugins* constantemente desenvolvidos pela comunidade para os mais diversos fins.

1.1 Problema de Pesquisa

O problema do menor caminho é um dos mais tradicionais quando se quer encontrar o menor caminho, a partir de um nó inicial até outros nós de um grafo. Ele pode ser aplicado em muitos tipos de problemas como transporte, logística, redes de computadores, dentre outros. Geralmente, trata-se de um problema de minimização e algoritmos exatos como o algoritmo de Dijkstra ou Bellman-Ford conseguem encontrar a melhor solução quando o problema possui apenas um único objetivo ou considera um único peso em cada aresta do grafo (DIJKSTRA, 1959; BELLMAN, 1958).

Porém, em problemas do mundo real como no roteamento de veículos, o melhor caminho a ser escolhido nem sempre será aquele com o menor valor de um único objetivo devido às inúmeras variáveis que podem influenciar na sua qualidade. Essa grande quantidade de variáveis faz com que as soluções geradas por métodos convencionais que usam um único critério, tal como calcular a menor distância, se tornem inviáveis para aqueles condutores que pretendem trafegar com maior qualidade pela cidade. Fatores como o nível de fluxo de tráfego, por exemplo, são de difícil previsão pelo condutor quando este está trafegando e para isso são necessárias ferramentas com métodos que sejam capazes de antecipar não apenas o fluxo de tráfego, mas também se o condutor passará por uma região perigosa ou se utilizará uma rodovia com péssimas condições de pavimentação.

Neste sentido, alguns estudos abordaram o problema do menor caminho considerando múltiplos objetivos simultaneamente no contexto do roteamento de veículos como pode ser visto em Zero et al. (2016), Kanoh e Hara (2008), Yao et al. (2017). O conjunto de problemas neste contexto é chamado de planejamento de rotas multiobjetivo (*Multi-objective Route Planning Problem*) e pertencem à classe de problemas NP-completos (TARAPATA, 2007). Este tipo de problema procura aperfeiçoar as soluções considerando vários objetivos simultaneamente a partir de um grafo de entrada com o propósito de encontrar um conjunto de caminhos ótimos. É importante destacar que os trabalhos acima mencionados não abordaram o problema do roteamento de veículos no campo do VRP, e sim procuraram solucionar problemas de roteamento multiobjetivo para um único veículo.

Desta forma, este trabalho se propõe a solucionar o seguinte problema: Como gerar rotas de adequadas considerando muitos objetivos evitando-se, na medida do possível, zonas congestionadas em tempo computacional factível em redes viárias? Para solucionar esta questão é proposto uma etapa de pré-processamento dos dados para

otimização do tempo computacional e o desenvolvimento de um novo operador de mutação que utiliza dados pré-processados a fim de explorar o espaço de busca de forma eficiente e em menos tempo utilizado em um algoritmo evolucionário multiobjetivo.

Os algoritmos evolucionários multiobjetivo são frequentemente utilizados na solução de problemas de otimização multiobjetivo, pois eles podem lidar com uma população de possíveis soluções simultaneamente permitindo encontrar várias soluções candidatas. Além disso, eles são capazes de lidar mais facilmente com descontinuidades que podem ocorrer na Frente de Pareto e conseguem lidar com grandes espaços de busca. Outras vantagens dos algoritmos evolucionários multiobjetivo consistem em requerer pouco conhecimento a cerca do problema envolvido, são robustos e podem computar problemas paralelamente diminuindo o seu tempo de execução significativamente. (COELLO-COELLO; VELDHUIZEN; LAMONT, 2002; ABRAHAM; JAIN; GOLDBERG, 2005).

1.2 Justificativa

Encontrar soluções viáveis para gerenciar o tráfego ou oferecer um roteamento mais inteligente para os usuários de tal maneira que o número de congestionamentos diminua é uma tarefa extremamente complexa e que pode envolver inúmeras variáveis aleatórias e não aleatórias.

Existe uma gama de problemas que decorrem a partir dos congestionamentos de veículos refletindo em muitos setores da sociedade. Empresas de transporte ou que dependem da estabilidade do trânsito para realizarem seus negócios com eficiência e sem ter algum tipo de prejuízo. Porém, os problemas continuam, pois uma piora no tráfego também pode refletir diretamente na saúde física e psicológica das pessoas que precisam enfrentá-lo todos os dias.

Vários trabalhos mostraram essa relação entre congestionamentos de trânsito e a saúde das pessoas que o enfrentam diariamente. Trabalhos como em Currie e Walker (2011) e Knittel, Miller e Sanders (2015) mostraram uma grande relação entre ambientes com alta densidade de tráfego de veículos e seu impacto na saúde fetal e no índice de mortalidade infantil, respectivamente. Além desses, outro estudo feito por Yiannakoulis, Bland e Svenson (2013) estudou o quanto os congestionamentos impactam na acessibilidade aos serviços de atenção básica em saúde e que esse impacto deve ser levado em consideração na disponibilização espacial desses serviços para que se tornem mais acessíveis à população.

São muitos os problemas que podem decorrer dos constantes congestionamentos e de tráfegos extremamente intensos nas grandes cidades. Abordar esse problema pode parecer apenas solucionar uma questão da mobilidade, mas ao passo

que é minimizado, diversos outros problemas tangentes a ele também poderão ser minimizados. Melhorando, assim, a qualidade de vida da população em vários aspectos.

1.3 Objetivos

1.3.1 Geral

Este trabalho tem como objetivo geral apresentar um algoritmo que utiliza Pontos de Referência (PR) para determinação de passeios multiobjetivo em redes viárias com congestionamento de veículos utilizando um algoritmo evolucionário multiobjetivo.

1.3.2 Específicos

- Selecionar as entidades do OSM que serão utilizadas como Pontos de Referência no grafo viário;
- Estabelecer as métricas a serem trabalhadas no problema;
- Selecionar API ou biblioteca que trabalhe com os dados do OSM para geração de grafos;
- Selecionar um banco de dados orientado a grafos que trabalhe com *big data* a fim de persistir as informações computadas dos caminhos entre os Pontos de Referência;
- Utilizar um algoritmo específico para geração de caminhos subótimos para cada métrica separadamente entre os Pontos de Referência definidos pelo usuário. Estes caminhos serão persistidos na base de dados gerando depósitos de caminhos;
- Simular cenários com diferentes níveis congestionamento nas principais vias da cidade;

1.4 Limitações do trabalho

Este trabalho não se propõe a realizar o controle da capacidade das vias em tempo real, pois levar em consideração o tempo para alocação de rotas não é algo trivial. Seus benefícios no controle de tráfego são grandes, podendo ser previstos picos de congestionamentos em determinados horários evitando que novas rotas passem por tais locais, oferecer rotas de maior qualidade evitando tais pontos com altas chances de congestionamento, dentre outros. Devido a sua complexidade de implementação deixaremos como proposta para trabalhos futuros.

Também não é proposto neste trabalho que as rotas terminem em locais específicos das entidades do OSM (construções públicas ou privadas) como entradas de garagens, por exemplo. Esse tipo de especificação seria de muita utilidade e até necessário para veículos autônomos. Porém o próprio OSM apresenta limitações e ele não identifica as entradas das construções no mapa. Isso é um problema que precisa ser resolvido futuramente para tornar viável a alocação de rotas para veículos autônomos, por exemplo.

Inicialmente, era esperado a criação de um grafo de sobreposição que utilizasse somente os Pontos de Referência da cidade. Porém, como já dito no parágrafo anterior, as entidades não possuem ligações com as vias, além de não indicarem locais de entrada e saída. Apesar de existirem *tags* no OSM que possam indicar a entrada e saída das construções, por algum motivo, estes dados raramente são preenchidos, inviabilizando a geração automática dos Pontos de Referência. Deste modo, todos os Pontos de Referência neste trabalho foram escolhidos manualmente levando em consideração a sua localidade e o tipo de construção como hospitais, escolas, praças, órgãos governamentais, etc.

1.5 Organização da dissertação

Esta dissertação está organizada conforme segue. No Capítulo 2, o problema é formulado matematicamente. É tratado, também, a fundamentação teórica deste trabalho abordando algoritmos evolucionários (AE), AEs multiobjetivo e suas aplicações em problemas de roteamento de veículos. O Capítulo 3 apresenta como foi realizada a etapa de pré-processamento dos dados, representação do problema e o esquema dos grupos de simulações e cenários de congestionamento. No Capítulo 4 são apresentados os resultados desta dissertação analisando todos os grupos de simulações. As conclusões deste trabalho bem como os trabalhos futuros estão dispostos no Capítulo 5. Nos Apêndices A, B, C, D, E são mostrados os resultados obtidos e exemplos de passeios gerados do primeiro ao quinto grupo de simulações, respectivamente.

2 Revisão da Literatura

2.1 Formulação do Problema

O problema do planejamento de rotas multiobjetivo pode ser definido a partir de um grafo direcionado de vias $G = \{N, A, C\}$ onde $N = \{N_1, N_2, N_3, \dots, N_n\}$ representa um conjunto de nós, $A = \{A_1, A_2, A_3, \dots, A_a\}$ representa um conjunto de arestas e $C = (C_1, C_2, C_3, \dots, C_m)$ representa um vetor de custos ou pesos associados a cada aresta. Deste modo, podemos representar uma rota $r = \{A_r, C_r\}$ onde cada rota é formada por um conjunto de arestas entre uma origem e um destino e um conjunto de pesos.

Cada nó possui, dentre outras informações, a sua latitude e longitude. No OSM, os nós relacionados às vias incluem um conjunto de propriedades adicionais: além da latitude e longitude, possuem a informação se ele representa uma faixa de pedestres, um sinal de tráfego, uma parada de ônibus, dentre outros. As arestas também possuem suas próprias propriedades que identificam o tipo da via, o tipo de pavimentação, a velocidade máxima permitida, se o sentido da via é de mão única e muitas outras propriedades.

Assim, o vetor de custos para cada aresta utilizado neste trabalho considera três objetivos onde t é o custo relacionado ao Tempo de Viagem estimado para a aresta, d é a Distância entre os nós da aresta calculada a partir das suas respectivas latitudes e longitudes através da fórmula de Haversine (Equação 2.2) (KORN; KORN, 2000) e s é o custo relacionado ao Número de Semáforos existentes. Deste modo, as funções t , d e s foram definidas como segue:

$$t = \frac{d}{v \cdot x} \quad (2.1)$$

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\phi_2 - \phi_1}{2}\right) + \cos(\phi_1)\cos(\phi_2)\sin^2\left(\frac{\lambda_2 - \lambda_1}{2}\right)}\right) \quad (2.2)$$

$$s = n \quad (2.3)$$

Na equação 2.1, v é a velocidade permitida na rodovia e x é a variável que indica o seu nível de congestionamento. Na equação 2.2, r é o raio da Terra, ϕ é a latitude e λ é a longitude dos nós. A Equação 2.3 representa a quantidade total de semáforos dentro de uma rota.

No decorrer desta dissertação, é mostrado em detalhes como o vetor de custos de um passeio é trabalhado no MOEA bem como sua utilização na construção da Função de Aptidão de cada indivíduo no algoritmo.

2.2 Algoritmos Evolucionários

A Computação Evolucionária (CE) é um ramo de pesquisa da Inteligência Artificial para solucionar problemas realizando processos de busca e otimização inspiradas na Teoria da Seleção Natural de Darwin. A ideia de se inspirar na evolução natural tornam estes algoritmos bastante eficientes em resolver problemas devido as evidências da evolução nas espécies ao redor do mundo. No processo de evolução, uma população de indivíduos, em um determinado ambiente, reproduzem e competem pela sobrevivência dos mais aptos para, então, transferirem suas características para as próximas gerações. Desta forma, estes algoritmos inspirados pela evolução natural conseguem encontrar, através de tentativa e erro, as melhores soluções para um determinado problema (EIBEN; SMITH, 2003; YU; GEN, 2010).

Os Algoritmos Genéticos (AG) e a Programação Genética (PG) são dois dos principais ramos da CE. Os AGs, propostos por Holland (1975), possuem larga aplicação em diversas áreas científicas como problemas de otimização, aprendizagem de máquina, desenvolvimento de fórmulas matemáticas, análise de modelos, etc. Desta forma, estes algoritmos trabalham com uma população de indivíduos que irão reproduzir, sofrer mutações e competir pela sua sobrevivência. Todos os indivíduos da população são avaliados pela sua aptidão (*fitness*), ou seja, os indivíduos mais aptos terão maiores chances de sobreviverem do que os menos aptos. Porém, indivíduos menos aptos ainda têm chances de seguirem para as próximas gerações mesmo que baixas. Isto garante a diversidade da população e impede que o algoritmo convirja muito rapidamente, parando em locais ótimos do espaço de busca.

2.2.1 Representação dos Indivíduos

Nos AGs, cada indivíduo da população, chamado de cromossomo, possui uma cadeia de genes onde cada gene contém uma informação e um valor, chamado de *fitness*, que representa a qualidade destas informações naquele indivíduo. Quanto maior a qualidade ou a nota das informações em um cromossomo, maior serão suas chances de sobreviver. São nos cromossomos do AG onde o problema a ser trabalho é representado.

Porém, representar o problema da maneira correta é crucial para se obter os melhores resultados. Eiben e Smith (2003) em seu livro “Introduction to Evolutionary Computing” identifica quatro tipos possíveis de se representar um problema

nos indivíduos da população do AG: Representação Binária, Representação Inteira, Representação em Valores Reais ou Ponto-Flutuantes e Representação por Permutação. Nenhuma destas representações é melhor que a outra, escolher a representação correta para um determinado problema é um dos estágios iniciais do AG e um dos mais difíceis.

2.2.1.1 Representação Binária

A Representação Binária é uma das mais simples e foi uma das primeiras representações a serem utilizadas nos AGs. Em Yu e Gen (2010), os AGs foram originalmente propostos para trabalharem com esta representação para imitar a codificação genética em organismos biológicos. Neste tipo de representação, os indivíduos são representados por cadeias de dígitos binários como mostra o exemplo na Figura 1.

A representação Binária não é recomendada para interpretar informações não-binárias (números reais e inteiros). Os números reais, por exemplo, não podem ser representados corretamente por completo. Seria necessário definir algum tipo de tolerância para lidar com este problema. De qualquer maneira, os melhores resultados para um determinado problema podem ser obtidos usando representações Inteira ou Ponto-Flutuantes diretamente (YU; GEN, 2010; EIBEN; SMITH, 2003).

Figura 1 – Exemplo de um cromossomo por Representação Binária.

0	1	1	1	1	0	0	0	1	0
---	---	---	---	---	---	---	---	---	---

Elaborado pelo Autor

2.2.1.2 Representação por Inteiro

Este tipo de representação pode ser muito útil para problemas que, naturalmente, possuem variáveis inteiras que podem ser trabalhadas. A representação por inteiro pode conter valores irrestritos ou podem ser restritos a um conjunto e que pode representar um conjunto de informações, por exemplo, o conjunto {0, 1, 2, 3} pode representar {Norte, Sul, Leste, Oeste} (EIBEN; SMITH, 2003). A Figura 2 apresenta um outro exemplo de uma representação inteira em um cromossomo.

Figura 2 – Exemplo de um cromossomo utilizando Representação por Inteiro.

R	0 → 255	← Tabela de cores RGB	
G	0 → 255		
B	0 → 255		
R	G	B	← Cromossomo
30	144	255	

Elaborado pelo Autor

2.2.1.3 Representação por Valores Reais ou Ponto-Flutuante

Quando um determinado problema apresenta uma distribuição contínua ao invés de uma distribuição discreta é necessário representá-lo por meio de valores reais ou ponto-flutuantes. Este tipo de representação é bastante utilizado em problemas de otimização de parâmetros onde se deseja encontrar a melhor combinação de parâmetros para um determinado algoritmo, por exemplo (EIBEN; SMITH, 2003; JONG, 2006).

2.2.1.4 Representação por Permutação

Existem problemas que para serem solucionados deve-se obedecer uma determinada ordem ou sequência de eventos. Segundo Eiben e Smith (2003), existem duas classes de problemas que podem ser representadas por permutação. Na primeira delas, a ordem em que o evento ocorre é importante. Isto pode acontecer quando os recursos e/ou tempo são limitados. Um exemplo de problema é o *job-shop*, onde vários trabalhos a serem processados em uma única máquina devem ser organizados de tal forma para que o tempo de processamento seja otimizado o máximo possível. Na segunda, a ordem depende da adjacência e o principal exemplo conhecido para este tipo de problema é o TSP. No TSP, o objetivo é encontrar a melhor combinação de caminhos, ou seja, com menor custo em um grafo contendo n cidades. Então, a diferença desta última classe de problema é que o ponto de partida da *tour* não é importante, ao contrário da classe de problemas mencionada inicialmente.

2.2.2 Mutação

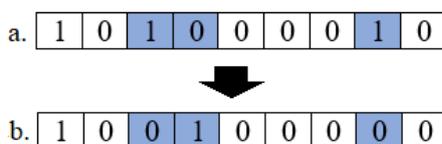
O operador de mutação é responsável por realizar pequenas variações nos genes dos indivíduos da população do AG. Estas variações são muito importantes no processo de busca, pois permitem que o AG consiga encontrar novas soluções em potencial explorando locais muito próximos no espaço de busca. A mutação também

garante a diversidade da população, pois o operador insere novas informações conforme realiza estas pequenas alterações nos genes dos cromossomos (EIBEN; SMITH, 2003; JONG, 2006).

2.2.2.1 Mutação em Cromossomos de Representação Binária

Na mutação em cromossomos de representação binária, o processo se dá bit-a-bit, ou seja, cada gene do cromossomo tem uma pequena probabilidade p de inverter o seu valor (de 1 para 0 ou 0 para 1) independentemente do resultado nos outros genes do indivíduo. É importante ressaltar que p deve ser configurado com uma taxa de probabilidade muito pequena, pois, do contrário, o operador de mutação acabará mudando muito a informação contida no indivíduo e, conseqüentemente, atrapalhando o processo de evolução do algoritmo (EIBEN; SMITH, 2003). A Figura 3 ilustra um caso onde um cromossomo a binário sofre mutação em 3 dos seus genes, tornando-se um novo cromossomo b .

Figura 3 – Exemplo de um processo de mutação binária



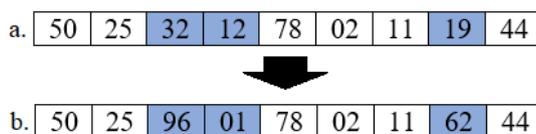
Adaptado de Eiben & Smith (2003)

2.2.2.2 Mutação em Cromossomos de Representação Inteira

Para cromossomos de representação inteira, segundo Eiben e Smith (2003) existem duas formas principais de se realizar o processo de mutação:

- **Random Resetting:** Esta operação ocorre de maneira similar a mutação em cromossomos de representação binária. Cada gene do cromossomo é capaz de mutar de forma independente obtendo um novo valor aleatório dentro de um dado conjunto de valores possíveis. A Figura 4 mostra um exemplo deste operador onde o cromossomo a possui uma cadeia de genes inteiros com valores que possam variar entre 0 e 100. O cromossomo b é o resultado do processo de mutação onde 3 genes de a tiveram seus valores mudados respeitando os limites mínimo e máximo definidos.

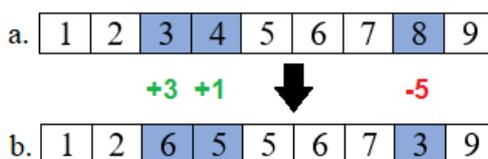
Figura 4 – Exemplo de um processo de mutação por *Random Resetting*.



Elaborado pelo Autor

- ***Creep Mutation***: Esta operação foi criada para lidar com problemas que trabalham com números ordinais. Aqui, cada gene também é capaz de mutar de forma independente. Desta forma, uma pequena alteração é feita no valor de cada gene adicionando ou subtraindo um pequeno valor aleatório do seu valor atual. Este tipo de mutação requer cuidado, pois não é fácil ajustar os parâmetros da função que irá gerar o valor aleatório. Um valor muito alto, por exemplo, pode acarretar em saltos muito grandes dificultando a exploração por novas soluções próximas. A Figura 5 ilustra um exemplo de um cromossomo *a* que sofre este processo de mutação onde 3 dos seus genes tem seus valores adicionados ou subtraídos decorrentes de uma função geradora aleatória, resultando em um novo cromossomo *b*.

Figura 5 – Exemplo de um processo de mutação por *Creep Mutation*.



Elaborado pelo Autor

2.2.2.3 Mutação em Cromossomos de Representação por Ponto-Flutuante

Para cromossomos de representação por valores reais, o processo de mutação ocorre separadamente para cada gene do cromossomo onde um limite inferior e superior devem ser definidos. Segundo Eiben e Smith (2003), existem dois tipos que se diferenciam de acordo com a probabilidade da distribuição utilizada para mutar um novo gene:

- **Mutação Uniforme:** Nesta operação, o valor do gene é substituído por um novo valor randomico uniforme selecionado dentro dos limites inferior e superior definidos para este gene.
- **Mutação Não-Uniforme com Distribuição Fixa:** Esta operação ocorre de forma semelhante ao *Creep Mutation*. É adicionado um pequeno valor aleatório ao gene que está sendo mutado através de uma distribuição gaussiana com média 0 e um desvio padrão pré-definido obedecendo os limites inferior e superior de valores possíveis para aquele gene. Este operador garante que a maioria das mudanças sejam muito pequenas, porém há uma probabilidade diferente de 0 que uma mudança grande ocorra.

2.2.2.4 Mutação em Cromossomos de Representação por Permutação

Para cromossomos de representação por permutação, não é possível considerar cada gene de forma independente, pois haveria uma desconfiguração do indivíduo e, conseqüentemente, este se tornaria uma solução inválida no processo evolutivo. Realizar mutações válidas, ou seja, explorando indivíduos que realmente existam no espaço de busca é mais vantajoso, pois não corre o risco de invalidá-los e perder informações que poderiam ser importantes atrapalhando o processo de busca do algoritmo. Desta forma, os operadores descritos a seguir realizam mudanças na ordem dos genes no cromossomo. Porém, estes operadores não são recomendados para aplicação em problemas baseados na adjacência dos genes, pois muitas conexões poderiam ser interrompidas. Para este tipo de problema o operador mais indicado é o *inversion mutation*.

- **Swap Mutation:** Este operador escolhe aleatoriamente dois genes do cromossomo para, então, trocá-los de posição. A Figura 6 apresenta um exemplo de *Swap Mutation* onde os genes 2 e 5 são selecionados e trocados.

Figura 6 – Exemplo de *Swap Mutation*.



Adaptado de Eiben & Smith (2003)

- **Insert Mutation:** Este operador escolhe aleatoriamente dois genes do cromossomo para, em seguida, mover um para perto do outro arrastando os demais genes para que isto ocorra. A Figura 7 apresenta um exemplo de *Insert Mutation* onde os genes 2 e 5 são selecionados e, em seguida, o gene 5 é movido ao lado do gene 2, arrastando todos os outros genes.

Figura 7 – Exemplo de *Insert Mutation*.

Adaptado de Eiben & Smith (2003)

- ***Scramble Mutation***: Este operador seleciona uma cadeia de genes aleatoriamente do cromossomo e embaralha todos os genes dentro dela. A Figura 8 apresenta um exemplo de *Scramble Mutation* onde uma cadeia de genes entre 2 e 5 é selecionada para, em seguida, ocorrer o embaralhamento desta cadeia.

Figura 8 – Exemplo de *Scramble Mutation*.

Adaptado de Eiben & Smith (2003)

- ***Inversion Mutation***: Este operador seleciona uma cadeia de genes aleatoriamente e inverte a sua ordem. Esta é a mudança que ocorre o menor impacto possível em problemas baseados em adjacência quebrando apenas duas conexões, o primeiro gene da cadeia e o último. A Figura 9 apresenta um exemplo de *Inversion Mutation* onde uma cadeia entre os genes 2 e 5 é selecionada para, então, a sua ordem ser invertida. Desta forma, apenas as ligações entre os genes 1-2 e 5-6 são quebradas.

Figura 9 – Exemplo de *Inversion Mutation*.

Adaptado de Eiben & Smith (2003)

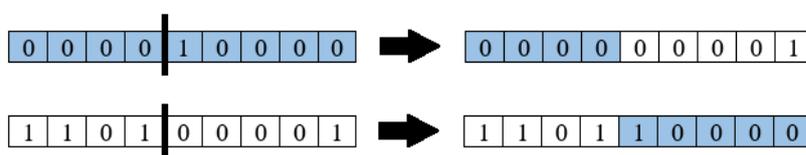
2.2.3 Recombinação

A recombinação é o processo pelo qual um novo indivíduo é gerado a partir da combinação entre dois indivíduos ou mais. Este processo é considerado um dos mais importantes nos AGs, e a mutação como um operador de busca auxiliar. A recombinação também pode ser chamada de cruzamento fazendo uma analogia ao processo de meiose que ocorre nas células. Porém, o termo cruzamento normalmente se refere a uma combinação entre um par de indivíduos. A Recombinação nem sempre irá ocorrer

em todos os indivíduos da população, normalmente existe uma taxa de probabilidade (entre 0.5 e 1.0) que determina se um par de indivíduos irão ou não recombinar-se formando dois novos descendentes. Caso a recombinação não ocorra, os descendentes viram cópias dos que iriam recombinar-se. Desta forma, uma nova população é gerada contendo os descendentes dos indivíduos e as cópias daqueles que não sofreram o processo de recombinação.

- **One-Point Crossover:** O cruzamento de um ponto funciona escolhendo uma única posição aleatória nos cromossomos a serem recombinados e partindo-os neste ponto. Após partidos, dois descendentes são criados trocando as caudas como é possível ver na Figura 10.

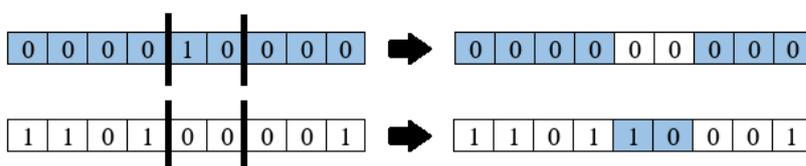
Figura 10 – Exemplo de *one-point crossover*.



Adaptado de Eiben & Smith (2003)

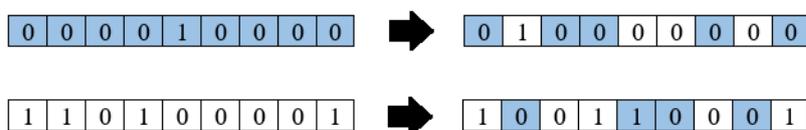
- **N-Point Crossover:** Enquanto o *one-point crossover* escolhe apenas um ponto de corte entre os dois cromossomos, o *n-point crossover* escolhe dois ou mais pontos trabalhando da mesma forma que o primeiro, ou seja, escolhendo aleatoriamente os pontos de corte levando em consideração o tamanho total dos cromossomos. Os descendentes são gerados combinando os segmentos dos dois indivíduos como é possível observar na Figura 11.

Figura 11 – Exemplo de *N-Point Crossover*.



Adaptado de Eiben & Smith (2003)

- **Uniform Crossover:** No cruzamento uniforme não existem pontos de corte como nos dois operadores anteriores. Neste operador, cada gene é tratado separadamente onde o operador escolhe de forma aleatória quais genes irão ser trocados entre os dois indivíduos. A Figura 12 apresenta um exemplo deste operador.

Figura 12 – Exemplo de *Uniform Crossover*.

Adaptado de Eiben & Smith (2003)

2.2.3.1 Recombinação em Cromossomos de Representação Inteira

A recombinação em cromossomos com valores inteiros funciona do mesmo modo que para cromossomos de representação binária. Desta forma, os operadores mostrados anteriormente também são válidos para este caso. Misturar os valores em si dos genes não é recomendável, pois, por exemplo, se uma média entre um gene contendo um valor inteiro par e o outro com valor ímpar pode gerar um gene com um resultado não inteiro, quebrando o estilo de representação definido para o problema (EIBEN; SMITH, 2003).

2.2.3.2 Recombinação em Cromossomos de Representação por Ponto-Flutuante

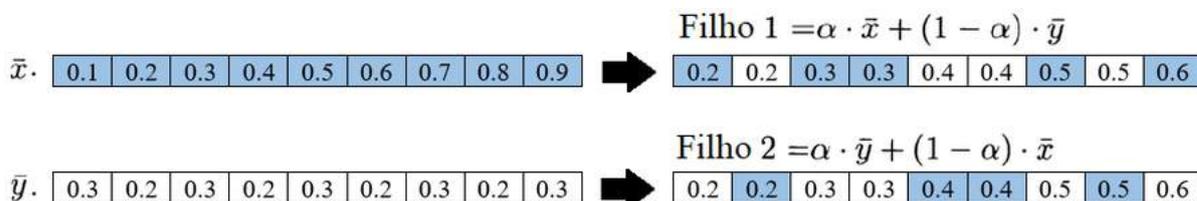
Segundo Eiben e Smith (2003), existem duas opções para este tipo de representação. A primeira opção consiste onde o operador funciona de forma semelhante aos operadores das representações por inteiro e binária, ou seja, os valores dos genes não são alterados em si, apenas a mutação é quem se encarrega de alterá-los. Neste operador, os cromossomos apenas irão recombinar-se formando novos descendentes. Os operadores deste tipo para representações por valores reais são conhecidos como **Recombinação Discreta**.

Na segunda opção, os operadores modificam os valores nos genes dos cromossomos no processo de recombinação criando descendentes com valores diferentes. Estes tipos de operadores podem se utilizar de equações matemáticas para gerar novos valores para os descendentes utilizando o material genético dos pais. Desta forma, o processo de recombinação consegue inserir indivíduos com informações totalmente novas na população. Porém, a desvantagem destes tipos de operadores é que o conjunto de valores de cada gene na população é reduzido. Estes operadores são conhecidos como **Recombinação Intermediária** ou **Aritmética**. O exemplo a seguir é um dos mais utilizados e ele funciona realizando um cálculo para cada gene tomando como base os valores dos genes dos pais.

- **Whole Arithmetic Recombination:** Neste operador, o valor de cada gene dos

descendentes é calculado utilizando uma soma ponderada utilizando os valores de cada gene dos pais (x e y). No exemplo da Figura 13 o valor de $\alpha = 1/2$.

Figura 13 – Exemplo de *Whole Arithmetic Recombination*.



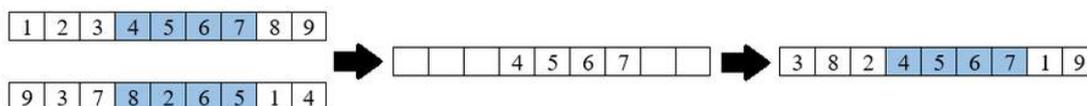
Adaptado de Eiben & Smith (2003)

2.2.3.3 Recombinação em Cromossomos de Representação por Permutação

O processo de recombinação nas representações por permutação é mais difícil devido, pois não se pode simplesmente trocar os genes dos indivíduos sem quebrar as conexões entre os genes existentes neste tipo de representação. No entanto, quando todos os indivíduos representam uma ordem bem definida fica mais fácil de criar operadores para esta classe de problemas. O exemplo a seguir foi criado para problemas de permutação baseado na ordem das informações.

- **Order Crossover:** Neste operador, o descendente é criado a partir de uma parte de um dos pais escolhida aleatoriamente para, então, ter sua cadeia genética completada com as informações do outro pai seguindo a ordem dos genes deste último. Na medida em que o descendente é preenchido, se ocorrer de um valor contido no segundo pai já existir no descendente, este é descartado e, então, o próximo valor passa a ser considerado contanto que também não exista. A Figura 14 apresenta um exemplo deste operador.

Figura 14 – Exemplo de *Order Crossover*.



Adaptado de Eiben & Smith (2003)

2.2.4 Modelos de População

Um dos elementos mais importantes na computação evolucionária é o seu processo evolutivo, ou seja, a forma como os indivíduos da população irão competir

entre si para a sobrevivência dos mais aptos. Existem dois modelos evolutivos para os AGs: o **modelo generacional** e o **modelo *steady-state***.

No modelo generacional, a cada geração do algoritmo, toda a população de indivíduos é substituída por seus descendentes gerados através dos processos de recombinação e mutação. Neste caso, o processo de recombinação somente se encerra quando o quantidade total de descendentes é igual a quantidade da população atual. A nova população, então, fará parte da próxima geração substituindo totalmente a atual.

No modelo *steady-state*, nem todos os indivíduos da população são substituídos para a próxima geração, apenas uma porcentagem da população é substituída por seus descendentes. Isto se dá no processo de recombinação onde apenas os indivíduos que entraram na probabilidade definida para o operador criam seus descendentes. Os indivíduos que não puderam se recombinar nesta geração são copiados para a próxima juntamente com descendentes.

2.2.5 Seleção de Pais

A seleção de pais é o processo pelo qual o algoritmo irá escolher aleatoriamente quais indivíduos irão recombinar criando descendentes para a próxima geração. Porém, este processo pode não ser totalmente aleatório.

2.2.5.1 Seleção por Classificação

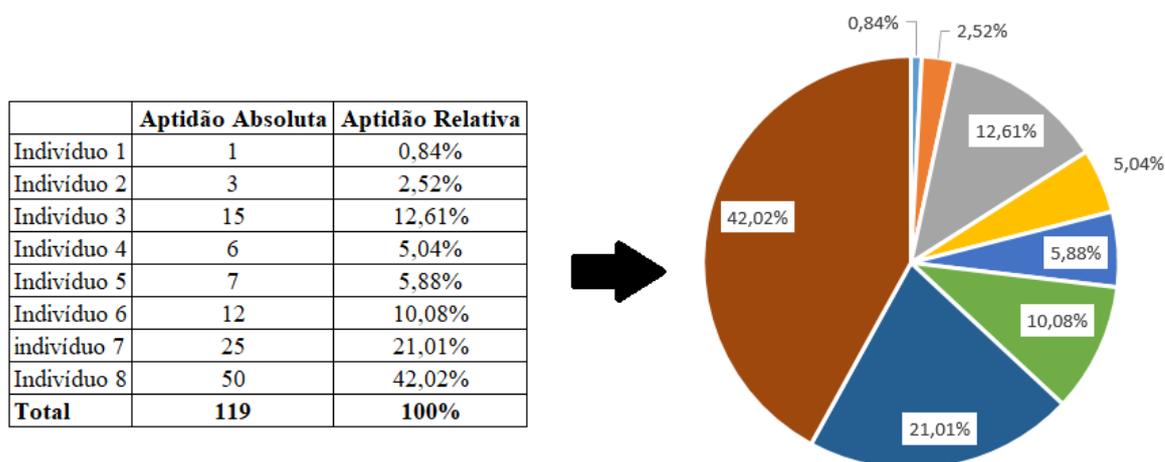
Na seleção por classificação, do inglês *Ranking Selection*, todos os indivíduos da população são classificados ordinalmente de acordo com sua aptidão (*fitness*). Os indivíduos melhor classificados terão maior probabilidade de realizarem a recombinação. É importante salientar este tipo de seleção não garante que os melhores classificados irão sempre se recombinar. Eles apenas terão maiores chances por estarem no topo da tabela de classificação, ou seja, indivíduos de baixa classificação ainda têm chances, mesmo que pequena, de entrarem no processo de recombinação.

2.2.5.2 Seleção Proporcional à Aptidão

Neste tipo de seleção, a chance de um indivíduo ser selecionado é proporcional a sua aptidão, ou seja, indivíduos mais aptos terão maiores chances do que os menos aptos. O método mais conhecido é o **método da Roleta**. Neste método, como o nome sugere, os indivíduos são mapeados em uma espécie de roleta onde cada indivíduo ocupa uma fatia proporcional a sua aptidão. Para isto, soma-se a aptidão de todos os indivíduos da população e divide o valor total pelo valor de cada indivíduo. Desta forma, quanto maior o valor da aptidão, maior será a fatia. Após todos os indivíduos

serem mapeados, roda-se a roleta para realizar a seleção de um indivíduo. Este processo se repete até que todos os indivíduos necessários sejam selecionados. A Figura 15 apresenta um exemplo do mapeamento da aptidão dos indivíduos na roleta.

Figura 15 – Exemplo de mapeamento dos indivíduos no método da Roleta.



Elaborado pelo Autor

2.2.5.3 Seleção por Torneio

Nos métodos anteriores, a seleção dos indivíduos dependiam do conhecimento de toda a população. Porém, existem problemas em que o tamanho da população é muito grande ou a população é distribuída de forma paralela. Desta forma, este processo do algoritmo pode acabar atrasando muito a execução do algoritmo e inviabilizando a sua computação. Desta forma, a seleção por Torneio vem para solucionar casos que não requerem um conhecimento global da população ou que tornam esse conhecimento inviável. No Torneio, um número k de indivíduos (geralmente pequeno) são escolhidos aleatoriamente para participarem dele. A probabilidade de um indivíduo ser escolhido no torneio dependem de quatro fatores:

- A classificação do indivíduo de acordo com os indivíduos escolhidos para o torneio;
- O tamanho do torneio, pois quanto maior, mais chances tem de serem escolhidos indivíduos acima da média;
- A taxa de probabilidade que o membro mais qualificado do torneio é selecionado. Quanto maior esta taxa maior é a pressão de seleção;
- Os indivíduos são escolhidos com ou sem reposição. Se o torneio não apresentar reposição, os indivíduos menos aptos quase nunca serão escolhidos. Por outro

lado, com reposição, existem chances de até o indivíduo menos apto da população ser escolhido.

2.2.6 Seleção de Sobreviventes

O processo de seleção de sobreviventes é a operação responsável por definir os indivíduos que irão compor a próxima geração. Como já dito anteriormente, existem dois modelos de população que podem ser utilizados em um AG. No modelo generacional, os indivíduos vivem apenas uma geração e então todos são substituídos pelos seus descendentes.

No modelo *steady-state* os indivíduos podem sobreviver mais de uma geração, pois a seleção de sobreviventes é baseada na aptidão dos indivíduos da população. Desta forma, os modelos de seleções de pais descrito acima podem atuar na seleção de sobreviventes selecionando os indivíduos para a próxima geração do algoritmo. Assim, um indivíduo que seja muito bom pode sobreviver a várias gerações do algoritmo.

2.2.6.1 Seleção Baseada na Idade dos Indivíduos

Neste tipo de seleção, a aptidão dos indivíduos não é levada em consideração. Os indivíduos possuem uma idade definida e todos duram a mesma quantidade de gerações. Apesar da aptidão não ser levada em conta, este método não impede totalmente a continuidade das melhores soluções na população, porém isso depende se eles conseguirem, ao menos uma vez, serem selecionados na seleção de pais e sobreviverem aos estágios de recombinação e mutação.

2.2.6.2 Seleção Baseada na Aptidão dos Indivíduos

Na seleção de sobreviventes baseada na aptidão, os operadores que se baseiam na aptidão na etapa de seleção de pais também podem ser utilizados nesta etapa. Além destes, outro dois métodos são apresentados a seguir.

- **Substituição dos piores:** Neste esquema, os piores indivíduos da população são escolhidos para serem substituídos. Este método melhora rapidamente a média de aptidão da população, porém uma convergência muito rápida pode prender o processo evolutivo em um ótimo local. Este esquema é recomendado quando se trabalha com uma população muito grande;
- **Elitismo:** Neste esquema, o melhor indivíduo da geração atual é mantido para seguir para a próxima geração substituindo um descendente qualquer que seja menos apto. Este método é comumente utilizado junto com seleções baseada na idade na tentativa de evitar a perda total de indivíduos muito bons.

2.3 Algoritmos Evolucionários Multiobjetivo (MOEA)

Muitos problemas do mundo real naturalmente envolvem diversos objetivos que necessitam ser otimizadas simultaneamente e, que na maioria das vezes, são conflitantes entre si. O espaço de busca nestes problemas são tão grandes que torna muito difícil resolvê-los sem utilizar técnicas avançadas de otimização. Além de um grande espaço de busca, muitas incertezas dificultam ainda mais o processo de otimização pela presença de ruídos (também existentes em problemas com um único objetivo) e componentes que podem variar no tempo.

Ao contrário dos problemas que envolvem a otimização de apenas um único objetivo, nos problemas de MOP não existe uma única e melhor solução ótima para o problema, mas um conjunto de melhores soluções que representam trocas viáveis (*trade-offs*) no espaço objetivo. Este conjunto é chamado de Frente de Pareto e nele contém todas as soluções que não podem ser mais otimizadas, ou seja, elas são melhores ou iguais a todas as outras soluções no espaço de busca. Estas soluções são conhecidas também como não-dominadas (COELLO; LAMONT, 2004).

Os algoritmos evolucionários se tornaram uma das principais técnicas para resolver problemas de otimização multiobjetivo por causa da sua capacidade de lidar com uma população de possíveis soluções simultaneamente permitindo encontrar muitos membros da Frente de Pareto em uma única execução do algoritmo. Além disso, estes algoritmos também conseguem lidar facilmente com as descontinuidades que podem ocorrer na Frente de Pareto ao contrário das outras técnicas de otimização (GOH; TAN, 2009; DEB, 2001).

2.3.1 Operações Básicas

O principal objetivo de um MOEA é convergir as soluções para encontrar a Frente de Pareto ótima. A cada execução do algoritmo uma Frente de Pareto é encontrada na população de indivíduos daquela geração. Desta forma, conforme as gerações vão avançando, a Frente de Pareto é atualizada até que o conjunto ótimo é encontrado.

Apesar do MOEA ser uma extensão dos algoritmos evolucionários, segundo Coello e Lamont (2004), duas questões devem ser consideradas:

- Como realizar o processo de seleção dos indivíduos dando preferência as soluções não-dominadas às que são dominadas?
- Como manter a diversidade na população para conseguir o máximo possível de soluções no conjunto ótimo de Pareto?

Quanto a seleção de indivíduos, a abordagem mais utilizada pelos MOEAs classifica a população baseado na dominância dos indivíduos do conjunto de Pareto. Desta forma, todos os indivíduos não dominados são tratados com o mesmo grau de importância, ou seja, os indivíduos não-dominados possuem a mesma probabilidade de reproduzirem e esta probabilidade deve ser maior que a definida para os indivíduos dominados.

Quanto a questão da diversidade, existem diversas abordagens que, dentre elas, incluem compartilhamento da aptidão, uso de esquemas para distribuição das soluções geograficamente, uso da entropia, restrição de recombinação e relaxamento da dominância de Pareto. O uso de uma população secundária em esquemas elitistas também tem sido muito comum. Neste esquema, a população secundária pode interagir de diferentes maneiras com a população principal do MOEA, armazenando soluções não-dominadas encontradas ao longo do processo. A população secundária também pode ser utilizada para melhorar a distribuição da população e também para regular a pressão de seleção.

2.3.2 Classificação dos MOEAs

Existem diversas formas de classificar um MOEA. A seguir serão apresentados três classes de MOEAs classificados tomando como base o tipo de seleção adotado.

2.3.2.1 Funções de Agregação

Neste tipo de MOEA, os valores de todos os objetivos do problema são combinados (ou agregados) em um único valor. As funções de agregação podem ser lineares ou não. Esta classe de MOEA tem sido muito subestimada pelos pesquisadores devido as limitações existentes nas funções de agregação lineares (elas não podem gerar porções não convexas da Frente de Pareto), porém, pesquisas mostraram que as funções não-lineares não apresentam as mesmas limitações, além de mostrarem bons resultados para problemas de otimização combinatória (COELLO; LAMONT, 2004).

2.3.2.2 Abordagem Baseada na População

Nesta abordagem, a população de um algoritmo evolucionário é utilizada para diversificar o processo de busca, porém, no processo de seleção este algoritmo não utiliza o conceito de dominância de Pareto. O VEGA é o exemplo mais clássico deste tipo de abordagem. O VEGA é basicamente um algoritmo genético com um operador de seleção modificado onde, para um problema com k objetivos, k sub-populações de tamanho M/k (M é o tamanho total da população) são geradas a cada geração. Então, as k -subpopulações são misturadas para obter uma nova população de tamanho M , na

qual o algoritmo irá aplicar os seus operadores genéticos. Porém, o maior problema deste algoritmo é que a sua natureza de seleção se opõe ao esquema de dominância de Pareto.

2.3.2.3 Abordagem Baseada em Pareto

Nesta categoria, serão apresentados alguns algoritmos que incorporam o conceito de otimização de Pareto na etapa de seleção.

- **Goldberg's Pareto Ranking:** Goldberg sugeriu evoluir a população usando um operador de seleção que favorece as soluções não-dominadas da população atual. Ele também sugeriu o compartilhamento da aptidão e *niching* como formas de manter a diversidade da população.
- **Multi-objective Genetic Algorithm (MOGA):** Nesta abordagem, cada indivíduo da população é classificado com base em quantos outros indivíduos o dominam. Os indivíduos que são não-dominados recebem a mesma classificação e, também, a mesma aptidão. Desta forma, todos tem a mesma probabilidade de serem selecionados. Para diversificar a população o MOGA utiliza o método de formação de nichos.
- **The Nondominated Sorting Genetic Algorithm (NSGA):** Esta abordagem trabalha com várias camadas de classificação de indivíduos. A população é ordenada baseado na não-dominância. Todos os indivíduos são classificados em uma determinada categoria com um valor de aptidão apropriado e que é proporcional ao tamanho da população, proporcionando uma melhor distribuição da probabilidade no processo de reprodução. Para manter a diversidade, os indivíduos são compartilhados com seus valores de aptidão. Então, o grupo classificado é ignorado e outra camada de indivíduos não-dominados é considerada. Este processo continua até que todos os indivíduos sejam classificados. Para o processo de seleção, no NSGA é adotada a seleção remanescente proporcional estocástica. O NSGA-II, proposto por Deb et al. (2002), utiliza o elitismo e um operador de comparação que classifica a população tomando com base a dominância de Pareto e a densidade da região. Este operador de comparação proposto no NSGA-II o torna mais rápido do que sua a versão anterior.
- **Strength Pareto Evolutionary Algorithm (SPEA):** Esta abordagem procura utilizar diversas estratégias de outros MOEAs. O SPEA utiliza um “valor de força” que é calculado de maneira semelhante como ocorre na classificação do MOGA. Para cada membro da população é atribuído um valor de aptidão baseado nas soluções não dominadas que o dominam. Quanto a diversidade, uma técnica

chamada “método de ligação média” é utilizada para realizar os agrupamentos dos indivíduos. A variação do SPEA, o SPEA2, ajusta a estratégia de cálculo da aptidão e utiliza a técnica do vizinho mais próximo.

2.3.3 *The Nondominated Sorting Genetic Algorithm (NSGA-III)*

A otimização multiobjetivo tem sido largamente estudada e muitas metodologias e algoritmos desenvolvidos mostram bons resultados para problemas de otimização com dois e três objetivos. Porém, muitos problemas do mundo real envolvem muitas variáveis que precisam ser otimizadas resultando em problemas com quatro ou mais objetivos. Este tipo de problema se torna um verdadeiro desafio até mesmo para algoritmos evolucionários multiojetivo devido a vários motivos, dentre eles, a quantidade de soluções não-dominadas cresce exponencialmente conforme o número de objetivos aumenta, operadores de manutenção da diversidade tornam-se computacionalmente muito custosos e difícil visualização da Frente de Pareto em um espaço multidimensional dificultando, conseqüentemente, tomadas de decisão.

O NSGA-III, criado por Deb e Jain (2014), utiliza o *framework* NSGA-II para lidar com problemas com mais de três objetivos. Esta nova versão possui diferenças significantes no operador de seleção além de manter a diversidade da população utilizando vários pontos de referência pré-definidos. A seguir, são apresentados os procedimentos propostos no NSGA-III para lidar com problemas que incluem vários objetivos.

- **Classificação da População em níveis de não-dominância:** Todos os membros da frente não-dominada desde o nível 1 ao nível l são incluídos numa nova população S_t . Se S_t conter o mesmo número de membros N da população inicial P_t , nenhuma operação é necessária e a próxima geração começa onde a população $P_{t+1} = S_t$. Para $|S_t| > N$, os membros das frentes do nível 1 até o nível $(l - 1)$ são selecionados e o restante ($K = N - |P_{t+1}|$) são escolhidos da última frente.
- **Determinação dos pontos de referência no hiperplano:** O NSGA-III utiliza pontos de referência pré-definidos para garantir a diversidade. Estes pontos de referência podem ser definidos de maneira estrutural ou pelo próprio usuário. Desta forma, em um hiperplano normalizado de dimensão $(M - 1)$, considerando que é igualmente inclinado em todos os eixos possuindo um em cada eixo e, além disso, um número p de divisões são consideradas ao longo de cada objetivo, o número total de pontos de referência H em um problema com M objetivos é dado

$$\text{por } H = \binom{M + p - 1}{p}.$$

- **Normalização adaptativa dos membros da população:** Em primeiro lugar, o ponto ideal da população S_t deve ser determinado identificando o valor mínimo (z_i^m) pra cada função objetivo para que seja construído o ponto ideal $\bar{z} = (z_1^{min}, z_2^{min}, \dots, z_M^{min})$. Cada valor-objetivo da população S_t é normalizado subtraindo o objetivo f_i por z_i^{min} de modo que o ponto ideal da população S_t normalizada torna-se um vetor zero. Após isto, o ponto máximo de cada objetivo ($z_i^{i,max}$) é identificado que realiza a função de escalarização mínima correspondente. Isto permite que o NSGA-III resolva problemas cujo valores dos objetivos que compoem a Frente de Pareto possuem diferentes escalas.
- **Operação de Associação:** Após a normalização de todos os indivíduos da população, é preciso associar cada indivíduo a um ponto de referência. Cada ponto de referência possui uma linha de referência associada a ele que conecta cada ponto com a origem. Então, é calculado a distância perpendicular de cada indivíduo da população S_t de cada uma das linhas de referência. Desta forma, o ponto de referência pertencente a linha que esteja mais próximo do indivíduo é associado a ele.
- **Operação de preservação de nichos:** Um ponto de referência pode ter um ou mais indivíduos da população associados ou não ter nenhum. Para descobrir o número de indivíduos da população P_{t+1} deve-se dividir o total da população inicial S_t pela última frente F_t . A operação de preservação de nicho consiste em identificar um conjunto de pontos de referência que contém um determinado valor mínimo p_j e, a partir do valor encontrado o algoritmo irá escolher os indivíduos da população que estejam próximos a um ou mais (ou nenhum) ponto de referência.
- **Operações Genéticas para criação da população de descendentes:** Após a formação da nova população P_{t+1} , ela é usada para criar uma nova população de descendentes Q_{t+1} aplicando os operadores genéticos comuns na literatura. A manutenção da diversidade da população enfatiza as soluções mais próximas das linhas de referência, além de ser feita uma cuidadosa seleção elitista.

2.3.4 MOEAs aplicados a problemas de roteamento de veículos

Diversos trabalhos têm sido desenvolvidos com o objetivo de otimizar o roteamento de veículos. A grande maioria dos trabalhos encontrados procuram resolver problemas otimizando cenários inteiros compostos por uma frota definida de veículos. Estes problemas fazem parte de uma classe chamada VRP. Segundo Toth e Vigo (2014), o VRP é um problema de otimização combinatorial que procura encontrar um conjunto de rotas com o menor custo possível para uma frota de veículos a fim de servir um determinado conjunto de clientes. Existem muitas variantes do VRP que resolvem

determinados problemas no roteamento de veículos. O CVRP é uma das variantes mais famosas e ele procura resolver problemas que envolvem o transporte de produtos em veículos de carga.

Muitos dos trabalhos encontrados utilizam MOEAs para trabalhar com VRP e suas variantes, porém, apesar do problema abordado neste trabalho também se caracterizar como um problema de otimização combinatorial nenhuma das variantes do VRP encontradas são compatíveis. Isto se dá pelo fato dele sempre levar a frota de veículos em consideração. Apesar desta definição ser clara no VRP, o fato dele ser um dos problemas mais estudados na área de transporte (TOTH; VIGO, 2014) motivou a tentativa de encontrar alguma variante mais específica.

Em vista disso, o MOSPP é a classe de problemas que melhor se encaixa neste trabalho. O MOSPP é uma variação do bem conhecido *Shortest Path Problem* (SPP) que tem como objetivo encontrar o caminho com menor custo possível dado um grafo direcionado, porém com a inclusão de vários pesos que devem ser otimizados simultaneamente nas arestas de um grafo.

Alguns trabalhos envolvendo MOSPP e o roteamento de veículos foram encontrados, no entanto, nenhum destes trabalhos buscam otimizar um passeio (um conjunto de rotas) como proposto neste documento. A otimização de um passeio remete também a um problema muito bem conhecido, o *Travelling Salesman Problem* (TSP). Existem estudos que abordam o TSP de forma multiobjetiva, porém esta classe de problemas também não se adequa a este problema, em razão do TSP apenas considerar um caminho entre dois pontos.

A seguir são apresentados trabalhos similares encontrados que envolvem otimização multiobjetiva no roteamento de veículos:

- Em Zero et al. (2016), foi proposto um algoritmo que utiliza uma função-objetivo determinística e funções-objetivo *fuzzy* min-max para solucionar problemas bi-objetivo no contexto do transporte de cargas perigosas que, em caso de acidentes, elevam os riscos ao meio-ambiente e/ou sociais. O algoritmo proposto calcula um conjunto de menores caminhos minimizando a distância e o nível de exposição da carga no trajeto caso aconteça um acidente, ou seja, em zonas urbanas o nível de exposição é considerado alto. O algoritmo utiliza comparação fuzzy triangular para uma das funções-objetivo enquanto para a outra é calculado o nível de risco levando em consideração as variáveis citadas anteriormente. Os autores concluem que o algoritmo proposto alcançou soluções para lidar com o problema do menor caminho bi-objetivo por meio de provas matemáticas e que este pode ser aplicado tanto offline quanto em ambientes reais.
- Em Kanoh e Hara (2008), foi proposto um algoritmo evolucionário híbrido para o

problema do roteamento de veículos multiobjetivo com previsão de tráfego em redes rodoviárias reais. O algoritmo evolucionário é mesclado com o algoritmo de Dijkstra para garantir a entrada de algumas soluções ótimas na criação da população inicial, porém este método é limitado apenas para rotas principais (classes 1 e 2) devido a falta de informação existente no mapa utilizado pelos autores. Quanto os objetivos utilizados para otimização, o tempo de viagem, a distância e um objetivo chamado “fácil condução” que realiza um cálculo levando em consideração diversas restrições definidas pelos autores como tipo de via, número de entradas, número de sinais. Nesse trabalho, também é proposto um operador genético chamado infecção que trabalha de forma semelhante a mutação realizando mudanças nos nodes das rotas. O algoritmo híbrido multiobjetivo proposto foi comparado com um algoritmo genético mono-objetivo e multiobjetivo considerando cenários com e sem congestionamento onde ele obteve os melhores resultados para os dois cenários. Os autores concluem que o algoritmo conseguiu gerar bons *trade-offs* evitando áreas com intenso tráfego de veículos em uma rede rodoviária real e obteve melhor desempenho comparado aos AGs mono e multiobjetivo.

- Em Yao et al. (2017), foi proposto uma aprendizagem por reforço baseada numa abordagem hiper-heurística multiobjetiva para problemas de roteamento de veículos em mapas de larga-escala de cidades inteligentes. Os autores desenvolveram um framework hiper-heurístico que seleciona uma heurística simples dentre um conjunto delas. Essas heurísticas se assemelham com os operadores genéticos de um AG, porém elas foram desenvolvidas para fins específicos. A abordagem proposta gera caminhos que otimiza a distância e a segurança do condutor e para isso vários dados são levados em consideração como dados históricos de crimes, localização de postos policiais, preço de aluguel de casas, população dentre outros. Com isso, um mapa de segurança é gerado e cada via deste mapa possui um nível calculado de segurança. Os caminhos computados são apresentados em um mapa através de um aplicativo desenvolvido pelos autores. De acordo com as conclusões apresentadas, a abordagem hiper heurística foi capaz de encontrar melhores resultados tanto em tempo computacional quanto na qualidade das soluções ao ser comparada com outros algoritmos multiobjetivo.
- Em Arbex e Cunha (2015), foi proposto um algoritmo genético que realiza a otimização de frotas de transportes públicos em redes rodoviárias levando em consideração dois objetivos: o primeiro está relacionado aos passageiros, tais como o tempo total de espera pelo ônibus, o tempo total de viagem dentro do veículo, dentre outros, e o segundo objetivo é relacionado aos custos de operação tal como o tamanho da frota a ser utilizada. Os autores também desenvolvem um

repositório de caminhos utilizando o algoritmo de Yen com a finalidade de ampliar o conjunto de soluções ótimas para que o algoritmo consiga oferecer rotas com o tempo de viagem o mais próximo do tempo mínimo possível. Este repositório também é utilizado no operador de mutação e criação da população inicial, diminuindo o tempo que o algoritmo leva para convergir para a frente ótima de Pareto. Os resultados obtidos pelo algoritmo foram comparados com outros trabalhos da literatura em três cenários distintos. De acordo com as conclusões apresentadas, o algoritmo proposto mostrou-se superior às demais propostas, apresentando soluções muito eficientes em termos de tempo de viagem, satisfação da demanda e custos dos usuários.

Todos os trabalhos acima citados utilizaram computação evolucionária, com exceção de Zero et al. (2016), e consideraram mais de um objetivo na otimização de rotas. Os trabalhos mais semelhantes com o proposto nesta dissertação foram os de Kanoh e Hara (2008) e Yao et al. (2017). Porém, em todos estes trabalhos apenas uma rota entre um ponto de origem e um ponto de chegada é otimizada. Apenas trabalhos que buscavam solucionar o problema do VRP e suas variantes otimizavam um passeio (um conjunto de rotas) ao invés de uma única rota. No entanto, como já dito anteriormente, por definição, problemas de VRP sempre incluem uma frota de veículos para ser otimizada. Desta forma, não foi possível realizar comparação entre os métodos propostos devido a diferença de escopo entre as propostas citadas e a presente neste trabalho.

3 Metodologia

Neste capítulo serão explicados em detalhes como foi realizado o pré processamento dos dados extraídos da plataforma OSM e as ferramentas utilizadas. Após a etapa de pré processamento, será explicado como o problema foi representado no MOEA e, principalmente, o funcionamento do novo operador de mutação proposto neste trabalho. Por fim, serão detalhados os esquemas das simulações para obtenção dos resultados.

3.1 Tratamento dos dados geoespaciais do *OpenStreetMap*

O *OpenStreetMap* é um projeto colaborativo no qual dispõe uma plataforma de dados geoespaciais, ou seja, um mapa de todo o mundo de forma livre e editável. Este mapa é mantido por uma comunidade de mapeadores voluntários que editam as informações através de APIs apropriadas para este fim.

O site do OSM, apesar de oferecer uma função para exportação de dados, não foi possível realizá-la nele diretamente. Para exportar grandes massas de dados é necessário recorrer à API's específicas. Para isso foi utilizado, inicialmente, o JOSM, um editor *opensource* de dados OSM. A ferramenta dispõe de um exportador de mapas para grandes escalas, porém, inicialmente, isso se tornou um problema devido a imensa quantidade de dados que uma grande cidade contém.

O presente trabalho se preocupou apenas com informações relacionadas as vias para geração do grafo, mas o OSM guarda informações não só de vias, mas também de construções e isso representa uma grande parte dos dados contidos em uma cidade. Uma solução encontrada foi a utilização do *Overpass* API (disponível em: <https://overpass-turbo.eu/>), uma ferramenta que disponibiliza a exportação de dados OSM através de um editor de código utilizando uma linguagem própria chamada *Overpass QL*.

O código 3.1 apresenta as instruções necessárias para a exportação dos dados que dizem respeito as vias. Para as *tags* do OSM foram selecionadas somente as *ways* que fazem parte do grupo de rodovias. Sendo elas: vias primárias, secundárias, terciárias, autoestradas, expressas, não-classificadas, ligamentos e residenciais. Existem ainda as vias de serviço, porém elas não são trabalhadas neste estudo. A existência de vias não-classificadas, ou seja, aquelas vias que não possuem informação a cerca do seu tipo (primária, secundária, expressa, etc.) pode influenciar na definição de rotas, pois cada tipo de via possui uma velocidade máxima permitida. Porém, após uma análise preliminar, foi observado que muitas das vias se encontravam em ambientes residenciais (vias residenciais). Uma hipótese para tais casos é que o OSM utilize

esta classificação como *default* para as vias que não tiveram a *tag* correspondente preenchida por um editor. Como o valor *default* definido para o limite de velocidade em vias não-classificadas se aproxima do limite de velocidade em vias residenciais, isto não impactou nos resultados.

Código 3.1 – Código em Overpass QL utilizado para extração dos dados das rodovias no OSM

```
(
way ["highway"~"^(primary|secondary|tertiary|motorway|trunk|
  unclassified|motorway_link|trunk_link|primary_link|
  secondary_link|tertiary_link|residential)$"] ({{bbox}});

rel ["type"="restriction"] ({{bbox}});
node ["amenity"="no"] ({{bbox}});

);
out meta;
```

As Figuras 16 e 17 apresentam um exemplo de um mesmo trecho de um mapa antes e após a filtragem dos dados utilizando o código 3.1. Desta forma, foi possível otimizar a computação para geração do grafo utilizando os dados OSM eliminando todos os dados dispensáveis para a formação do mesmo.

Figura 16 – Exemplo de um mapa exportado diretamente do OSM.



Java OpenStreetMap Editor

Figura 17 – Mapa da Figura 16 utilizando o código 3.1.



Java OpenStreetMap Editor

3.2 Implementação de um algoritmo para criação do grafo de rodovias utilizando dados do *OpenStreetMap*

Para a criação do grafo utilizando os dados do OSM, criamos uma adaptação a partir de um projeto *opensource* para smartphones chamado *FootPath* que gera um grafo de rotas de pedestres para navegação no interior de construções e que oferece instruções passo-a-passo até chegar ao destino (LINK et al., 2011). Foram realizadas diversas modificações para atender as especificações deste trabalho devido a sua complexidade. A principal delas foi o desenvolvimento de um algoritmo para a criação do grafo de rodovias respeitando as regras de trânsito vigentes na cidade.

O OSM em seu arquivos de metadados fornece diversas informações como mencionado anteriormente. Além destas, ele também oferece informações sobre restrições de conversão de trânsito existentes nas vias das cidades. Estas regras não são únicas, pois cada país possui suas próprias regras e o OSM possibilita essa flexibilidade. Entretanto, o OSM não disponibiliza ferramentas para tradução destas regras para serem utilizadas em alguma linguagem de programação.

Desta forma, se fez necessário o desenvolvimento de um algoritmo próprio para criação de um grafo de rodovias que reflita a real situação do fluxo do mapa exportado. Isto é muito importante para a geração de caminhos válidos se aplicados no mundo real. Porém, apesar do OSM disponibilizar tais informações, representar estas regras em um grafo direcionado se mostrou um desafio durante todo o processo deste trabalho devido a dificuldade em validar o algoritmo desenvolvido para garantir a corretude do mesmo.

Um dos principais desafios encontrados para representar as regras de restrição de conversão existentes no mapa para um grafo direcionado é a existência de nós

compartilhados nos mapas OSM.

Os nós compartilhados possuem esse nome devido a sua presença em mais de uma via. Uma via no OSM é representada por um conjunto de nós interligados de forma sequencial. Quando duas ou mais vias se cruzam, estas vias acabam compartilhando um único nó no ponto de interseção entre todas elas. A solução para este problema é tratada com detalhes na próxima seção.

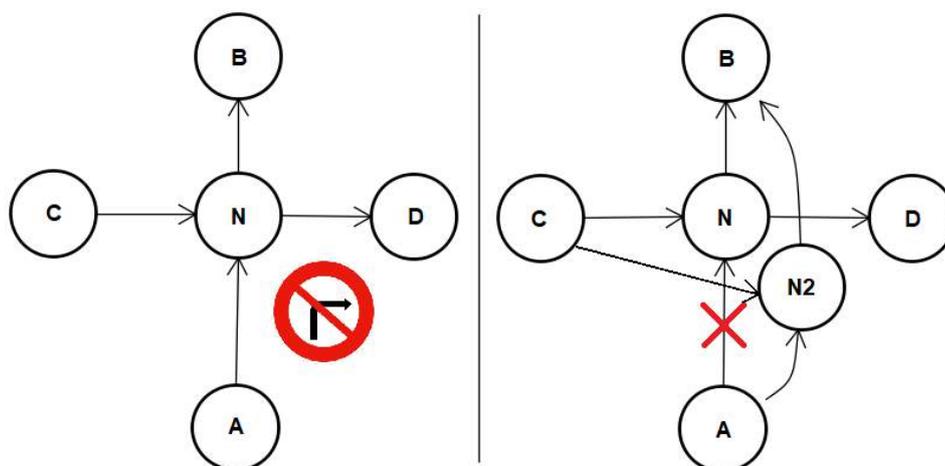
3.2.1 Solucionando o problema do nó compartilhado e sua representação em um grafo

Uma solução encontrada para resolver tal problema foi a criação de novos nós que seriam cópias dos nós compartilhados do OSM. Porém, esse processo somente é feito após a criação de todo o grafo do mapa sem o devido tratamento das regras de restrição de conversão, pois a lista de vias nos arquivos OSM não apresentam um padrão de ordenação, dificultando a computação dos dados. Então, após a criação do grafo não tratado, é aplicado o algoritmo para a correção das ligações à medida que restrições de conversão são detectadas.

Como é mostrado na Figura 18, o algoritmo desenvolvido realiza os seguintes passos para a alteração dos pontos em que é detectado uma relação de restrição de conversão em um nó compartilhado:

- 1) Um nó-via N é identificado através de uma *tag* específica;
- 2) É realizada uma busca na lista de relações de restrição de todo o mapa utilizando o nó-via N e o nó-destino D a fim de encontrar uma relação de restrição entre eles;
- 3) A relação de restrição de conversão entre o nó-origem A , o nó-via N e o nó-destino D é detectada;
- 4) Uma cópia do nó-via N é criada e gerado um novo ID artificial para a mesma;
- 5) É rompida a conexão entre o nó-via N e o nó-origem A da relação de restrição detectada;
- 6) O nó-origem A e todos os outros nós que tinham o nó-via N como destino são conectados também a sua cópia;
- 7) O nó-cópia $N2$ se conecta a todos os nós que tinham como origem o nó-via N , com exceção do nó-destino D que faz parte da relação de restrição de conversão.
- 8) Os IDs do nó-via N e de sua cópia $N2$ são salvos em um arquivo chamado Dicionário de Cópias;

Figura 18 – Exemplo de restrição de conversão e solução proposta



Elaborado pelo Autor

Todas as cópias de cada nó-via são registradas em um arquivo (Dicionário de Cópias) juntamente com seus respectivos nós-vias, pois como as cópias possuem IDs que somente são usados para identificação delas, tais IDs não existem na base OSM e ocorreriam problemas de referência ao detectar as rotas. O dicionário de cópias serve justamente para referenciar os nós-vias as suas respectivas cópias possibilitando a identificação das rotas geradas pelo MOEA quando plotadas em um mapa OSM.

Desta forma, foi possível representar as vias de tráfego de veículos de uma cidade através de um grafo, respeitando as normas de circulação de trânsito. O próximo passo foi calcular os k-menores caminhos para os Pontos de Referência definidos no grafo utilizando um algoritmo para tal e salvá-los em um banco de dados.

3.3 Pré-processamento dos k-menores caminhos no grafo de rodovias

Nas redes de grande porte, encontrar o menor caminho em aplicações dinâmicas em tempo computacional factível é um grande desafio para os algoritmos mais tradicionais, tal como para o algoritmo de menores caminhos de Dijkstra (DIJKSTRA, 1959; POTAMIAS et al., 2009).

Lidar com redes viárias em grandes centros ou metrópoles não é algo trivial devido a enorme quantidade de nós e conexões existentes. A definição de Pontos de Referência (PR) distribuídos de maneira mais uniforme possível foi a solução encontrada para diminuir o tamanho da rede viária e, conseqüentemente, a sua complexidade. Após a criação dos PRs, uma nova camada sobre o grafo de rodovias G , um grafo sobreposto $G' = \{N', P'\}$, poderia ser criado para que o algoritmo o utilizasse, onde N' representaria um conjunto de todos os nós referentes aos PRs pré-definidos na cidade tais como escolas, hospitais, universidades, parques, construções públicas, dentre

outros, e P' representa o conjunto de todos os caminhos factíveis entre cada par de PRs considerando cada objetivo separadamente.

Porém, restringir a geração de caminhos apenas entre PRs poderia reduzir demasiadamente as opções de deslocamento do usuário. Para contornar este possível problema, o usuário pode selecionar qualquer ponto de destino a partir do grafo base G e um procedimento para detectar o PR mais próximo contido em G' pode ser realizado. Desta forma, o grafo G , nesta etapa, iria se restringir apenas ao espaço entre o destino final selecionado pelo usuário e o PR mais próximo a ele. Esta etapa poderia ser feita online sem a necessidade de algum tipo de pré-processamento. Este procedimento garante que o tamanho do grafo G' seja reduzido utilizando apenas os PRs e, conseqüentemente, a complexidade computacional necessária para gerar os caminhos em G' também diminui. Isto é importante e justificável quando pretende-se atender a um grande conjunto de usuários diariamente, por exemplo.

Entretanto, como mencionado nas Limitações deste trabalho, não foi possível criar este grafo sobreposto G' de forma automática devido a complicações relacionadas ao OSM. A solução encontrada foi a identificação de PRs de forma manual e a criação do grafo se deu por demanda, ou seja, a medida que eram realizadas novas solicitações de passeios nas simulações.

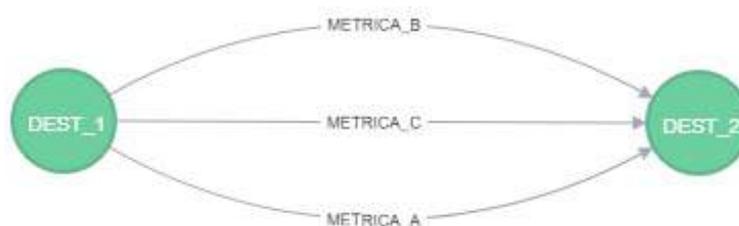
O pré-processamento de um conjunto P' de possíveis caminhos para cada objetivo entre pares de PRs pode ser considerado o ponto chave para que o algoritmo evolucionário encontre os melhores caminhos otimizando todos os objetivos simultaneamente. Para realizar esta tarefa, o algoritmo *K-shortest path* de Yen (YEN, 1971) foi utilizado para gerar os K menores caminhos entre dois PRs quaisquer, onde K é um parâmetro que indica a quantidade máxima de caminhos subótimos a serem gerados pelo algoritmo. Com isto, o espaço de busca entre cada par é reduzido, eliminando previamente os caminhos muito ruins que teriam poucas chances de serem utilizados devido à sua baixa qualidade. Por fim, todos os caminhos são armazenados em listas ordenadas (armazém de caminhos) pelo custo de cada objetivo separadamente para serem utilizados posteriormente no operador de mutação e na criação da população inicial. A persistência destas listas ordenadas se deu em um banco de dados que oferece uma estrutura suficiente para trabalhar com *big data*.

O Neo4j é uma plataforma de banco de dados orientada à grafos, ou seja, a persistência dos seus dados seguem em modelo de grafo, e foi escolhido para este trabalho. Este tipo de banco de dados consegue persistir dados com mais rapidez do que os bancos de dados transacionais comuns, além de ter uma modelagem de dados simples e intuitiva, facilitando e agilizando as consultas através do Cypher que é a linguagem nativa utilizada pela plataforma (HOLZSCHUHER; PEINL, 2013).

A Figura 19 apresenta um modelo simples de como os dados dos caminhos

foram esquematizados no Neo4j. Os nós do grafo representam os diversos Pontos de Referência da cidade e contém informações como ID, nome, latitude e longitude. As arestas contém informações do vetor de nós intermediários da rota e o seu peso total para cada métrica. Foram definidos *labels* para as arestas no intuito de classificar e ordenar as rotas por métrica. Desta forma, cada aresta representa um caminho único entre dois pontos, além de identificar por qual métrica ela foi ordenada através do *label* definido para aquela aresta. Este tipo de estrutura foi escolhido por ser mais intuitivo ao problema, em termos de código e visualização, facilitando a detecção de problemas posteriores.

Figura 19 – Representação das listas de caminhos ordenadas por métrica entre dois Pontos de Referência.



Elaborado pelo Autor

Portanto, a computação das rotas para os Pontos de Referência se faz necessário somente uma vez diminuindo o tempo de resposta do algoritmo quando houvesse solicitações de rotas. É importante lembrar que a persistências dos caminhos no banco de dados se deu de forma ordenada do menor para o maior caminho de acordo com cada métrica calculada. Isso se fez necessário, pois é um pré-requisito para a utilização do operador de mutação e criação dos indivíduos no MOEA. Após serem persistidas todas as informações, o próximo passo foi configurar os operadores do MOEA para tratar o problema, além do desenvolvimento do novo operador de mutação.

3.4 Representação do problema e configuração dos operadores no MOEA

Nesta seção, é detalhado as configurações utilizadas no NSGA-III, o MOEA escolhido disponível no *framework* JMetal (DURILLO; NEBRO; ALBA, 2010). O NSGA-III foi escolhido pela sua capacidade de lidar com problemas que possuam mais de 3 objetivos. O problema trabalhado nesta dissertação contém apenas 3 objetivos devido a poucas informações contidas no OSM que poderiam gerar outros objetivos. Mesmo

assim, o NSGA-III foi mantido, pois este trabalho não se limita apenas à plataforma OSM. Outras plataformas de dados geoespaciais podem ser utilizadas e novos objetivos podem ser gerados.

3.4.1 Função de Aptidão

Neste trabalho, o Algoritmo Evolucionário Multiobjetivo (MOEA) deve encontrar um conjunto contendo as melhores soluções (Frente de Pareto) onde cada solução representa um passeio (rotas sequenciais), minimizando os valores do vetor de funções-objetivo \mathbf{F} de cada passeio. Um passeio p é formado por um conjunto de rotas $\{r_1, r_2, r_3, \dots, r_z\}$ entre cada par de Pontos de Referência. Portanto, o vetor \mathbf{F} de funções-objetivo resultante de cada passeio p pode ser representado da seguinte maneira:

$$\mathbf{F}(p) = [T(p), D(p), S(p)] \quad (3.1)$$

onde:

$$T(p) = \sum_{i=1}^{|p|} t(r_i) \mid r_i \subset p \quad (3.2)$$

$$D(p) = \sum_{i=1}^{|p|} d(r_i) \mid r_i \subset p \quad (3.3)$$

$$S(p) = \sum_{i=1}^{|p|} s(r_i) \mid r_i \subset p \quad (3.4)$$

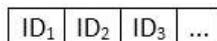
onde t , d e s são os custos de Tempo de Viagem, Distância e Número de Semáforos, respectivamente, como apresentado na Seção 2.1 deste documento.

3.4.2 População inicial

A representação do problema no NSGA-III se dá através dos indivíduos que formarão a população a ser otimizada no algoritmo multiobjetivo, ou seja, os cromossomos. Cada cromossomo da população do algoritmo corresponde a um passeio \mathbf{T} . Desta forma, cada gene contém como valor um identificador de um caminho previamente calculado que representa um trecho dentro do passeio como mostra a Figura 20. A

ordem dos genes é importante e não pode ser alterada, pois o passeio seria corrompido já que a ordem de visita aos pontos de interesse é previamente definida pelo usuário.

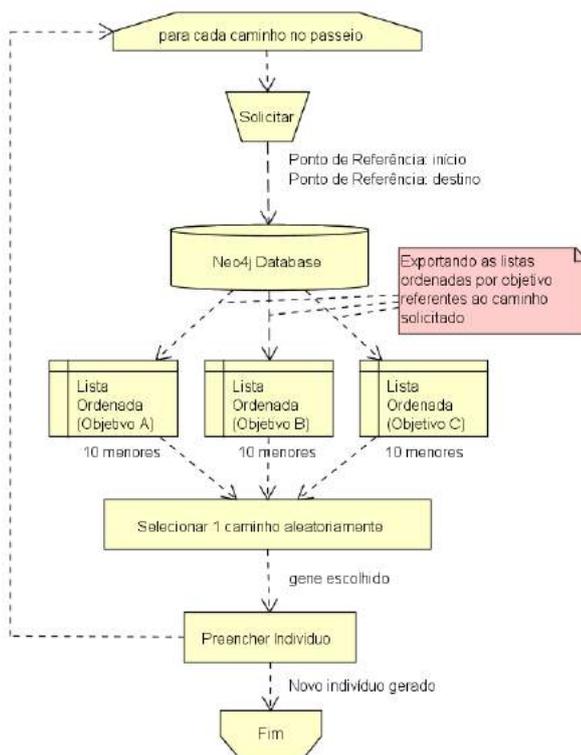
Figura 20 – Representação do cromossomo no MOEA.



Elaborado pelo Autor

Em um grafo viário de uma grande cidade ou metrópole, a quantidade de caminhos que podem existir entre dois PRs pode ser enorme, mesmo limitando o número de caminhos gerados pelo *K-shortest path*. Desta forma, a população inicial foi gerada utilizando uma heurística que privilegia a qualidade dos caminhos que foram pré-processados. Portanto, 50% dos cromossomos da população tiveram cada um dos seus genes preenchidos através de sorteio a partir de um conjunto formado pelos 30 menores caminhos existentes entre os dois pontos de interesse (10 menores caminhos de cada objetivo) como mostra a Figura 21. É importante ressaltar que cada gene possui um conjunto próprio de caminhos subótimos para o sorteio, pois se refere a um par de PRs distinto dos demais. Portanto, a introdução de caminhos ótimos na população pode agilizar o processo de busca e, ao mesmo tempo, a escolha aleatória pode evitar que o algoritmo evolucionário fique preso em soluções ótimas locais.

Figura 21 – Fluxograma do processo de preenchimento dos cromossomos selecionados para preenchimento privilegiando os 10 menores caminhos de cada lista ordenada.



Elaborado pelo Autor

3.4.3 Operadores Genéticos

3.4.3.1 Operador de Mutação Proposto

O operador de mutação desenvolvido realiza o processo mutando um ou mais caminhos dentro do passeio utilizando as listas ordenadas (depósito de caminhos) referentes a cada caminho para realizar a troca de genes garantindo, assim, que o operador trabalhe somente com os caminhos relacionados a um determinado destino dentro do passeio. Desta forma, evita-se que o operador escolha aleatoriamente qualquer caminho existente no grafo de rodovias resultando numa desconfiguração do passeio e, conseqüentemente, inviabilizando o indivíduo. A ideia principal é conceber ao operador um espaço-objetivo que contenha apenas soluções factíveis e ordenado para realização da busca. A ordenação do espaço garante que o operador privilegie a exploração de vizinhos mais próximos do indivíduo que está em processo de mutação. A seguir será apresentado com mais detalhes o funcionamento deste operador:

- 1) O operador, randomicamente, escolhe um gene do cromossomo a ser mutado;
- 2) As listas ordenadas são identificadas de acordo com este gene. Então, uma lista ordenada é randomicamente escolhida;
- 3) A posição atual r deste gene é identificada na lista ordenada escolhida;
- 4) O operador randomicamente escolhe uma nova posição na lista ordenada utilizando uma distribuição normal $N(r, s)$ onde r é a posição atual do gene na lista ordenada e s é um parâmetro do operador que representa o desvio padrão. Desta maneira, o operador pode explorar posições de caminhos próximas em relação ao atual no processo de mutação. Sendo assim, o operador favorece os vizinhos do caminho atual do processo de mutação.

Em alguns casos, o novo identificador gerado pelo operador de mutação pode se tornar inválido. Isto pode ocorrer quando o identificador do caminho que está para ser mutado se encontra nas bordas da lista ordenada. Então, uma técnica de espelhamento foi desenvolvida para garantir que o operador de mutação sempre irá gerar identificadores válidos. Os dois possíveis casos são tratados a seguir:

- 1) Limite Superior: Neste caso, o caminho que está para ser mutado está nas primeiras posições da lista ordenada. Se o valor da sua posição for menor que o valor do desvio padrão configurado, pode ocorrer do operador gerar uma nova posição com valor negativo. Sendo assim, o seu valor deve ser multiplicado por -1;

- 2) Limite Inferior: Neste caso, o caminho que está para ser mutado está nas últimas posições da lista ordenada e a nova posição gerada pelo operador de mutação pode extrapolar o tamanho total da lista. Desta forma, a Equação 3.5 foi utilizada para gerar uma nova posição válida.

$$I' = U - (I - U) = 2U - I \quad (3.5)$$

Onde I é a posição inválida gerada pelo operador e U é o tamanho da lista ordenada.

A Figura 22 apresenta mais detalhadamente o funcionamento do operador de mutação onde tem como variáveis iniciais definidas, o valor do desvio padrão para o cálculo da distribuição (STDEV) e a taxa de probabilidade para cada gene.

Figura 22 – Representação em pseudocódigo do operador de mutação proposto.

```

1  STDEV ← 3
2  probabilidade ← 0,2
3  para cada gene do cromossomo
4    var ← random(0,1)
5    se var <= probabilidade
6      list ← retorna uma lista de caminhos ordenada por uma métrica
7            aleatoriamente de acordo com o destino do gene
8      lowerBound ← 0
9      upperBound ← list.size
10     gaussian_index ← gerarNovoID(gene, STDEV)
11     se gaussian_index < lowerBound
12       gaussian_index ← gaussian_index * -1
13       gene ← list.get(gaussian_index)
14     fim se
15     se gaussian_index > upperBound
16       gaussian_index ← upperBound - (gaussian_index - upperBound)
17       gene ← list.get(gaussian_index)
18     fim se
19     senao
20       gene ← list.get(gaussian_index)
21     fim senao
22   fim se
23 fim para

```

Elaborado pelo Autor

Portanto, o operador de mutação consegue explorar outras soluções que estejam próximas da que se deseja mutar. É importante lembrar que o valor do desvio padrão no cálculo da distribuição deve ser um número pequeno para que a mutação explore os locais mais próximos e não execute saltos muito grandes, do contrário, o algoritmo

não conseguiria explorar todos estes locais onde poderiam haver boas soluções. A Figura 23 exemplifica como o operador trabalha entre as listas ordenadas utilizando a distribuição normal. Neste exemplo, um caminho **I** que está para ser mutado apresenta diferentes posições em cada tabela devido aos diferentes valores existentes em cada uma de suas métricas. Ele pode ser trocado por um vizinho mais próximo em um limite máximo de 3 posições ($STDEV = 3$) de forma aleatoria utilizando a distribuição normal em uma lista ordenada escolhida aleatoriamente no início do processo.

Figura 23 – Exemplo de como um caminho pode estar indexado nas tabelas de acordo com cada métrica durante o processo de mutação.

Lista Ordenada (Métrica 1)	Lista Ordenada (Métrica 2)	Lista Ordenada (Métrica 3)
...	Caminho I	...
Vizinho I - 3	Vizinho I + 1	...
Vizinho I - 2	Vizinho I + 2	...
Vizinho I - 1	Vizinho I + 3	...
Caminho I	...	Vizinho I - 3
Vizinho I + 1	...	Vizinho I - 2
Vizinho I + 2	...	Vizinho I - 1
Vizinho I + 3	...	Caminho I
...	...	Vizinho I + 1

Elaborado pelo Autor

3.4.3.2 Operadores de Recombinação, Seleção de Pais e Sobreviventes

O operador de recombinação utilizado neste trabalho foi o *crossover* uniforme. Este operador realiza a troca de cromossomos que estejam na mesma posição dentro do indivíduo. A ordem em que os genes se encontram dentro do indivíduo não pode ser alterada, pois haveria uma desconfiguração deste. Todos os indivíduos entram obrigatoriamente no processo de recombinação, porém, cada posição deles possui uma pequena probabilidade p independente de realizar a troca. Desta forma, podem existir descendentes resultantes de trocas grandes, pequenas ou até que sejam cópias dos seus pais. Quanto aos operadores de seleção de pais e sobreviventes, foi utilizado o torneio binário que está configurado como padrão para o NSGA-III no *framework* JMetal.

3.5 Simulações

As simulações têm como objetivo principal provar a eficiência do algoritmo proposto para lidar com problemas em vários casos desde passeios mais simples até passeios maiores em um mapa do mundo real e, ao mesmo tempo, lidando com situações reais de congestionamento. Desta forma, as simulações foram separadas

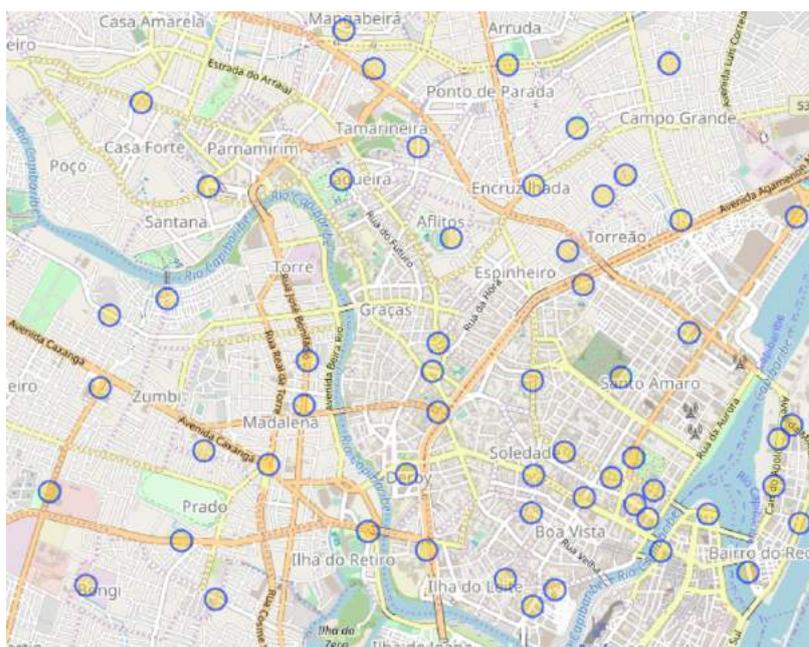
por grupos onde cada grupo possui cenários com níveis distintos de congestionamento e um passeio gerado aleatoriamente.

Em cada simulação, além da execução do algoritmo, também executou-se o algoritmo de Dijkstra para fins de comparação. As soluções obtidas representam as ótimas teóricas, porém este algoritmo é limitado apenas a um objetivo por execução. Desta forma, foram obtidas as 3 soluções com menor custo possível sendo uma para cada métrica separadamente. Em seguida, foram calculados os valores para os outros objetivos de cada uma destas soluções para fins de comparação com o MOEA.

3.5.1 Configuração dos passeios para as simulações e definição dos cenários de congestionamento

Para cada grupo de simulação foi definido um passeio de tamanho aleatório que visita certos Pontos de Referência, também escolhidos aleatoriamente, sem repetições destes pontos para o mesmo passeio. Desta forma, foram configurados passeios (um para cada grupo de simulação) que variam aleatoriamente entre 2 a 10 caminhos. Para cada grupo o algoritmo proposto calculou as melhores combinações dado os níveis de congestionamento para cada cenário.

Os Pontos de Referência foram definidos priorizando locais públicos como órgãos governamentais, escolas, universidades, hospitais, praças, parques, dentre outros. Alguns PRs que indicavam construções privadas também foram escolhidos devido o grande fluxo ou a sua importância como hospitais particulares, por exemplo. A Figura 24 apresenta a distribuição dos Pontos de Referência pelo mapa de parte da cidade de Recife, totalizando 57 PRs.

Figura 24 – Pontos de Referência distribuídos pelo mapa OSM da cidade de Recife.

Elaborado pelo Autor

Para a simulação de cenários de congestionamento, foi adotada uma estratégia utilizando o limite máximo de velocidade permitida dentro das vias. Em situações de congestionamento, o limite de velocidade nas vias tende a cair dependendo do tipo da via. Geralmente, as vias principais da cidade que possuem grande fluxo de veículos e interligam diversos pontos tendem a sofrer mais impacto do que vias residenciais, por exemplo.

Desta forma, foram criados 3 cenários com diferentes níveis de congestionamento apenas nas principais vias da cidade utilizando a classificação de vias disponibilizada pelo OSM. Sendo assim, foram selecionadas as vias classificadas como primárias e secundárias. Estas vias, normalmente, ficam destacadas em laranja e amarelo nos mapas OSM, respectivamente. O Quadro 1 apresenta a classificação e o limite máximo de velocidade definidos pelo OSM. É importante lembrar que estes valores presentes na tabela para o limite de velocidade somente são utilizados caso este tipo de dado não esteja preenchido no arquivo exportado referente ao mapa.

Deste modo, foi reduzido o limite máximo permitido nestas vias gradativamente de um cenário para o outro a fim de analisar o desempenho do algoritmo proposto para diferentes níveis de congestionamento:

- Cenário 1: Sem congestionamentos, ou seja, não há redução nos limites de velocidade;
- Cenário 2: Diminuição de 50% do limite de velocidade de tráfego;

- Cenário 3: Diminuição de 75% do limite de velocidade de tráfego.

Inicialmente, foi definido um quarto cenário onde a diminuição era de 25% do limite de velocidade de tráfego nas vias. Porém, de acordo com resultados preliminares, foi possível perceber que este valor era muito baixo e não influenciava o suficiente no cenário como um todo. Como apenas as vias primárias e secundárias são afetadas pela diminuição, elas ainda teriam seu limite de velocidade maior que os demais tipos de vias onde o limite de velocidade é muito baixo.

Por fim, as melhores soluções encontradas pelo NSGA-III foram comparadas com os menores passeios calculados para cada peso ou objetivo separadamente utilizando o algoritmo de Dijkstra.

Quadro 1 – Classificação das vias e limites de velocidade definidos pelo OSM.

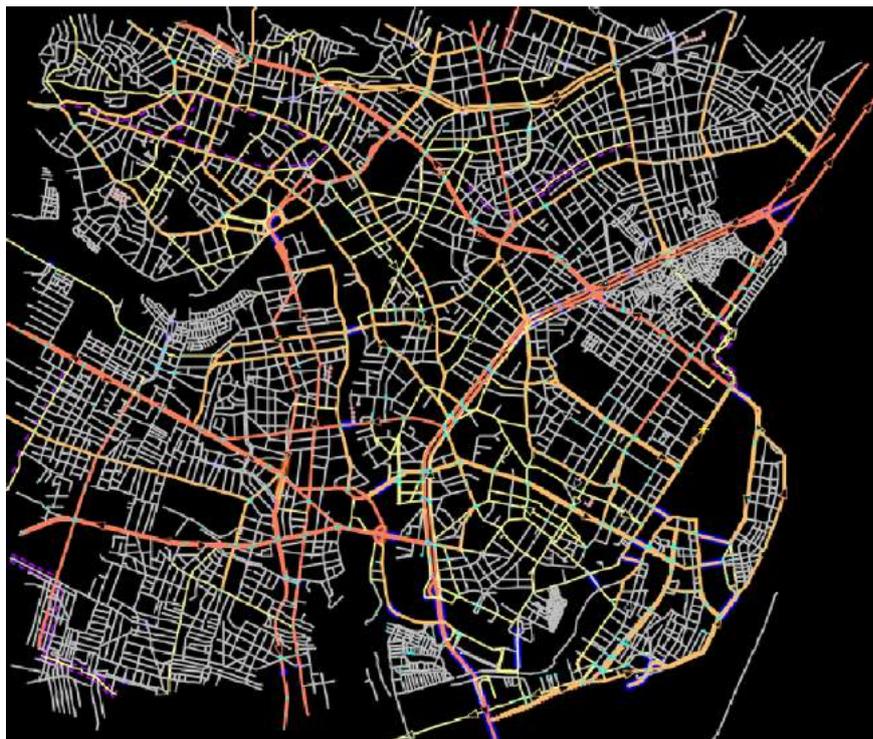
Tipo de via	Limite Máximo (Brasil)
Rodovia	110 Km/h
Expressa	80 Km/h
Primária	60 Km/h
Secundária	60 Km/h
Terciária	40 Km/h
Não-classificada	40 Km/h
Residencial	40 Km/h
Serviço	20 Km/h

wiki.openstreetmap.org

4 Resultados

Para a realização das simulações, foi utilizado um mapa de parte da cidade de Recife proveniente do OSM com as seguintes coordenadas: lat. min.: -8.0222, lat. max.: -8.0730; lon. min.: -34.9291, lon. max.: -34.8645. O mapa inclui 13527 nós e 26311 arestas relacionados às vias de tráfego de veículos, ou seja, após a realização da filtragem detalhada na Seção 3.1 e contém todas as informações necessárias para calcular a distância, o tempo de viagem estimado a partir da velocidade média determinada na via, e a quantidade de sinais de trânsito. A Figura 25 apresenta o mapa filtrado, onde as vias laranjas e amarelas representam as vias primárias e secundárias, respectivamente. As vias em cinza representam as demais vias como terciárias, residenciais, não-classificadas e de serviço.

Figura 25 – Mapa viário de parte da cidade do Recife representado no JOSM utilizado para realização das simulações.

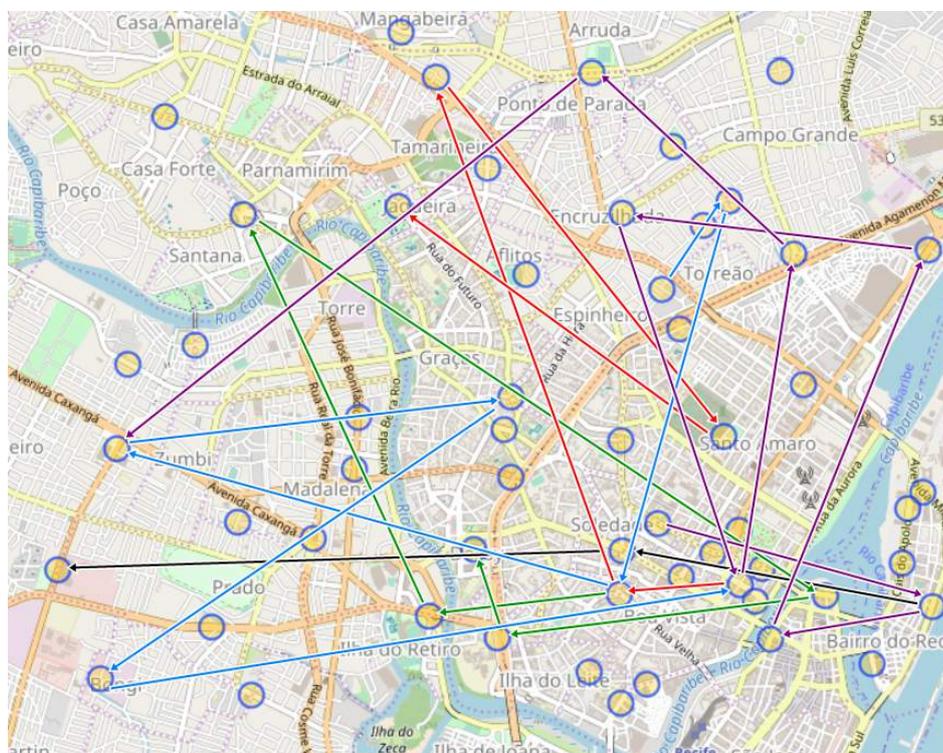


Java OpenStreetMap Editor (JOSM)

Foram definidos 5 grupos de simulações totalizando 15 simulações distintas se levado em consideração os 3 cenários de congestionamento de cada grupo. Todos os passeios foram construídos de forma aleatória em termos de número de rotas (valor mínimo igual a 2 e máximo igual a 10) e Pontos de Referência visitados, como detalhado na seção anterior. Desta forma, cada passeio gerado se configurou como apresenta a Figura 26. As setas pretas representam o passeio com 2 destinos, as

vermelhas o passeio com 4 destinos, as verdes o passeio com 5 destinos, as azuis o passeio com 6 destinos e as magentas o passeio com 8 destinos.

Figura 26 – Configuração dos 5 passeios utilizados nas simulações.



Adaptado pelo Autor

Nas seções seguintes, cada cenário de cada grupo será analisado separadamente bem como a análise geral de cada grupo. Para os cenários com os melhores resultados em cada grupo, estes cenários serão analisados de forma mais detalhada através de histogramas normalizados que ajudarão a compreender a distribuição de frequência das Frentes de Pareto encontradas. Por fim, será analisado o desempenho do algoritmo comparando os resultados de todos os grupos e discutido a sua eficiência dado os resultados apresentados.

Todas as legendas de todos os gráficos de linhas deste Capítulo estão limitadas a mostrar apenas 15 soluções da Frente de Pareto, pois a apresentação da legenda completa, contendo todas as 92 soluções, tornaria inviável a visualização de tais gráficos. Desta forma, a legenda foi limitada, porém, dentro dos gráficos, ainda é constado todas as soluções obtidas pelo algoritmo.

4.1 Análise do primeiro grupo de simulações

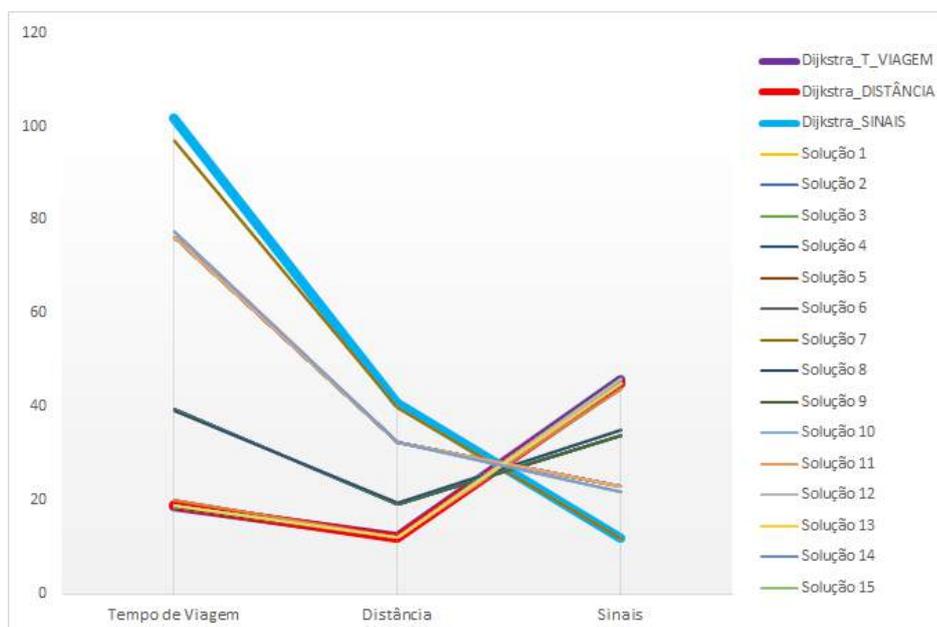
Nesta seção, serão discutidos os resultados do primeiro grupo de simulações que foi executado utilizando o menor dos passeios gerados aleatoriamente, possuindo

apenas dois destinos. Em seguida, para cada cenário de congestionamento, as soluções encontradas pelo algoritmo proposto foram comparadas com as melhores soluções encontradas pelo algoritmo de Dijkstra em cada cenário. Para que fosse possível realizar tal comparação, as soluções encontradas pelo Dijkstra também tiveram seus valores para os outros objetivos calculados.

4.1.1 Cenário sem congestionamento

Em termos de diversidade das soluções, a Figura 27 apresenta as soluções obtidas e os menores caminhos encontrados pelo algoritmo de Dijkstra com seus valores calculados para os outros objetivos. Cada linha representa uma solução da Frente de Pareto e seus respectivos valores para todos os objetivos após a execução do algoritmo. Desta forma, o gráfico consegue prover informação sobre a execução do algoritmo, demonstrando a sua capacidade de encontrar diversas soluções e *trade-offs* adequados para problemas com mais de dois objetivos.

Figura 27 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 1 sem congestionamento.



Elaborado pelo Autor

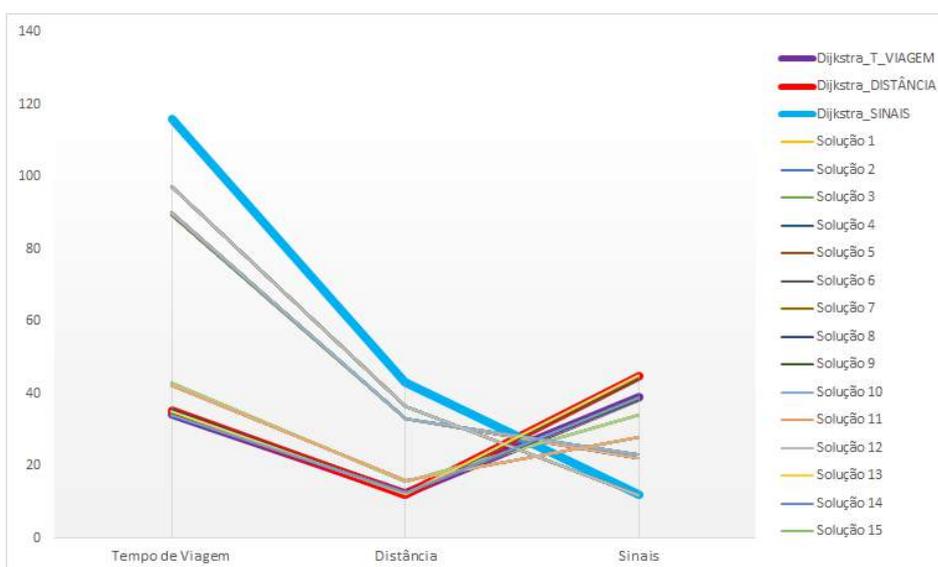
As soluções encontradas pelo algoritmo de Dijkstra, a princípio, podem parecer mais atraentes devido ao fato dele garantir o menor caminho para um determinado objetivo. Porém, esse caminho pode ser muito ruim quando os demais objetivos são considerados. Isto pode ser observado em todas as simulações com o caminho Dijkstra_SINAIS onde o Dijkstra sempre consegue encontrar o caminho com menor número de sinais em relação aos demais, mas torna-se muito inadequado ao considerar os outros dois objetivos.

Para este cenário, as soluções encontradas pelo algoritmo proposto estão agrupadas em grupos pequenos e o melhor deles, visivelmente, encontra-se com valores próximos das soluções Dijkstra_T_VIAGEM e Dijkstra_DISTÂNCIA. De fato, para um cenário sem congestionamentos, as melhores soluções levando em consideração o tempo de viagem ou a distância serão as computadas pelo algoritmo de Dijkstra. Porém, o algoritmo proposto também foi capaz de oferecer soluções muito parecidas. Quanto ao objetivo correspondente ao número de semáforos, o algoritmo encontrou trocas interessantes onde, apesar de perder um pouco neste objetivo, ganha-se consideravelmente nos dois outros objetivos.

4.1.2 Cenário com 50% de congestionamento

Para o cenário com redução de metade do limite de velocidade nas principais vias da cidade o algoritmo proposto foi capaz de oferecer *trade-offs* mais adequados em relação ao cenário anterior, ou seja, oferecendo uma menor perda em determinado objetivo para ganhar em outro. É possível notar ao olhar as soluções próximas das Dijkstra_T_VIAGEM e Dijkstra_DISTÂNCIA como mostra a Figura 28, onde algumas delas empatam ou perdem em valor irrisório quanto aos objetivos relacionados ao tempo de viagem e distância, mas apresentam um ganho quanto ao número de semáforos. Ainda existem algumas soluções acima de Dijkstra_DISTÂNCIA que, apesar de ser mais perceptível a perda quanto aos dois primeiros objetivos, compensam ganhando consideravelmente no último objetivo.

Figura 28 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 1 com 50% de congestionamento.

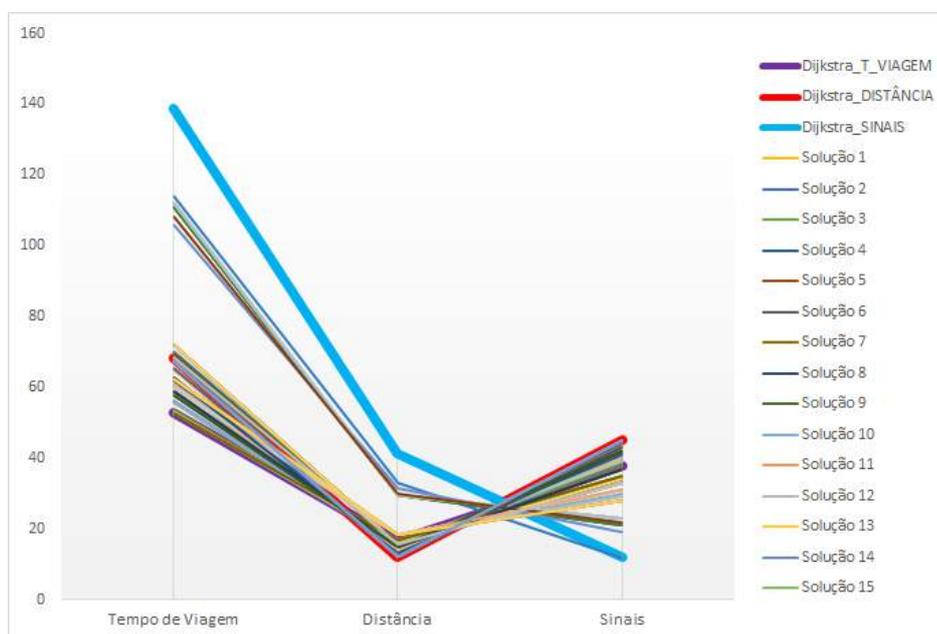


Elaborado pelo Autor

4.1.3 Cenário com 75% de congestionamento

Para o cenário com redução de 75% do limite de velocidade nas principais vias a Figura 29 apresenta as soluções da Frente de Pareto mais espalhadas em relação aos dois cenários anteriores, onde é possível ver alguns agrupamentos. Com isto, mais *trade-offs* interessantes se tornam presentes quando o nível de congestionamento aumenta. Isto se torna claro pela quantidade de soluções encontradas entre Dijkstra_T_VIAGEM e Dijkstra_DISTÂNCIA observando a coluna relacionada ao tempo de viagem. É interessante notar que a solução que garante o menor caminho (Dijkstra_DISTÂNCIA), em termos de distância, perde para a maior parte das soluções da Frente de Pareto quanto ao tempo de viagem e para todas quanto ao número de semáforos. Ganhando somente em termos de distância o que já era esperado, porém, de forma sutil para muitas soluções. Isto evidencia uma ineficácia por parte de aplicativos utilizados no dia-a-dia por motoristas que utilizam algoritmos tradicionais como o Dijkstra para traçar rotas em cenários onde são comuns ocorrer congestionamentos.

Figura 29 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 1 com 75% de congestionamento.

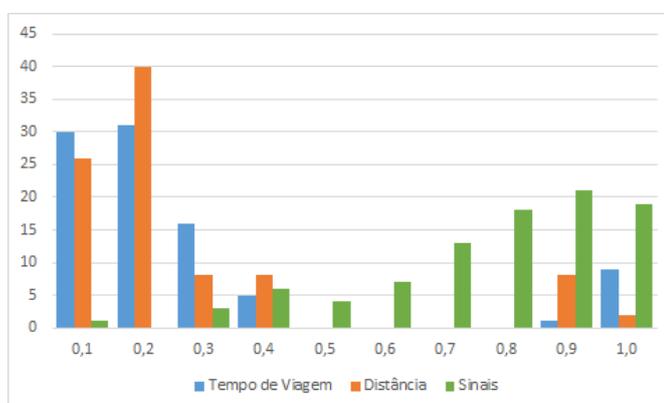


Elaborado pelo Autor

A Figura 30, apresenta a frequência das soluções de Pareto da Figura 29 em um histograma normalizado para os três objetivos. Como é possível observar, a maioria das soluções para os objetivos Tempo de Viagem e Distância, agrupam-se nos dois primeiros intervalos. Isso implica dizer que a maioria dos *trade-offs* encontrados pelo NSGA-III possuem valores muito próximos aos valores mínimos teóricos encontrados pelo algoritmo de Dijkstra em ao menos dois objetivos. Porém, em contra partida,

a maioria das soluções obtiveram valores muito distantes do mínimo teórico para o objetivo Número de Sinais (ou Semáforos).

Figura 30 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 75% de congestionamento do grupo 1.



Elaborado pelo Autor

4.1.4 Análise do desempenho do algoritmo para todos os cenários do primeiro grupo de simulações

O Quadro 3 apresenta alguns resultados dos cenários do grupo 1, o quadro completo pode ser visto no Apêndice A - Quadros 15 e 16. As soluções obtidas pelo algoritmo proposto são comparadas com as soluções computadas pelo algoritmo de Dijkstra para cada cenário. Nesta tabela, os percentuais foram obtidos através da divisão do valor obtido pela respectiva solução pelo menor valor da coluna. Estes valores em percentual significam o custo excedido de cada solução em relação aos resultados obtidos pelo Algoritmo de Dijkstra. Em cada coluna, o menor valor (não percentual) sempre será obtido pelo algoritmo de Dijkstra ao otimizar tal objetivo individualmente, ele foi utilizado como base para as comparações com as demais soluções da respectiva coluna. Portanto, a comparação de todas as soluções de uma coluna permite observar a sua proximidade percentual do menor valor teórico. Por outro lado, a análise das linhas, ou seja, ao considerar também os outros objetivos, surgem soluções mais adequadas de acordo com a flexibilidade desejada se comparadas com as soluções Dijkstra_T_VIAGEM, Dijkstra_DISTÂNCIA e Dijkstra_SINAIS.

Por exemplo, a solução 8 no cenário 2 se aproxima bastante do valor ótimo dos objetivos tempo de viagem e distância e apresenta um ganho quanto ao número de semáforos. Possivelmente, em uma aplicação prática, seria uma opção mais adequada do que aquelas obtidas pelo Dijkstra.

No cenário 1, muitas das soluções possuem resultados parecidos refletindo os agrupamentos das soluções vistas na Figura 27 da Seção 4.1.1. Porém, a medida que os níveis de congestionamento aumentam, o algoritmo apresenta melhores resultados,

ou seja, a média total da Frente de Pareto, em ao menos dois objetivos, vai diminuindo. A diminuição destes valores significa que os *trade-offs* estão cada vez mais próximos dos valores mínimos teóricos como é possível ver no Quadro 2 que apresenta a média e o desvio padrão da Frente de Pareto de cada cenário.

No apêndice A - Figuras 48, 49 e 50 são apresentados exemplos de soluções para cada cenário a fim de mostrar, na prática, a capacidade do algoritmo de gerar novas rotas adaptando-se conforme os níveis de congestionamento aumentam.

Quadro 2 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 1.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Média	128,68%	70,94%	196,11%	73,80%	80,48%	146,92%	26,10%	38,68%	199,64%
Desvio Padrão	144,09%	79,32%	89,28%	81,29%	87,05%	94,77%	31,18%	41,02%	56,66%

Quadro 3 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 1.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	18,78	1,26%	283,33%	34,09	3,59%	225,00%	52,67	38,63%	216,67%
Dijkstra Distância	1,32%	12,06	275,00%	3,83%	12,06	275,00%	29,56%	12,06	275,00%
Dijkstra Sinais	442,44%	239,66%	12	240,07%	257,62%	12	163,39%	242,02%	12
Solução 1	0,71%	0,19%	275,00%	185,57%	201,67%	0,00%	10,57%	24,09%	208,33%
Solução 2	110,18%	60,63%	183,33%	161,90%	173,55%	91,67%	1,04%	31,76%	225,00%
Solução 3	0,71%	0,19%	275,00%	26,36%	28,30%	183,33%	29,49%	1,20%	258,33%
Solução 4	307,06%	169,06%	91,67%	0,00%	3,59%	225,00%	36,31%	29,06%	183,33%
Solução 5	0,71%	0,19%	275,00%	3,83%	0,00%	275,00%	5,98%	24,99%	216,67%
Solução 6	109,47%	61,70%	191,67%	0,00%	3,59%	225,00%	18,09%	15,75%	241,67%
Solução 7	313,20%	170,08%	83,33%	164,13%	174,39%	83,33%	0,00%	38,63%	216,67%
Solução 8	0,71%	0,19%	275,00%	2,23%	4,42%	216,67%	112,54%	143,63%	91,67%
Solução 9	306,45%	169,25%	91,67%	23,67%	31,70%	133,33%	29,83%	29,52%	175,00%
Solução 10	1,32%	0,00%	275,00%	185,57%	201,67%	0,00%	14,27%	52,11%	150,00%

Elaborado pelo Autor

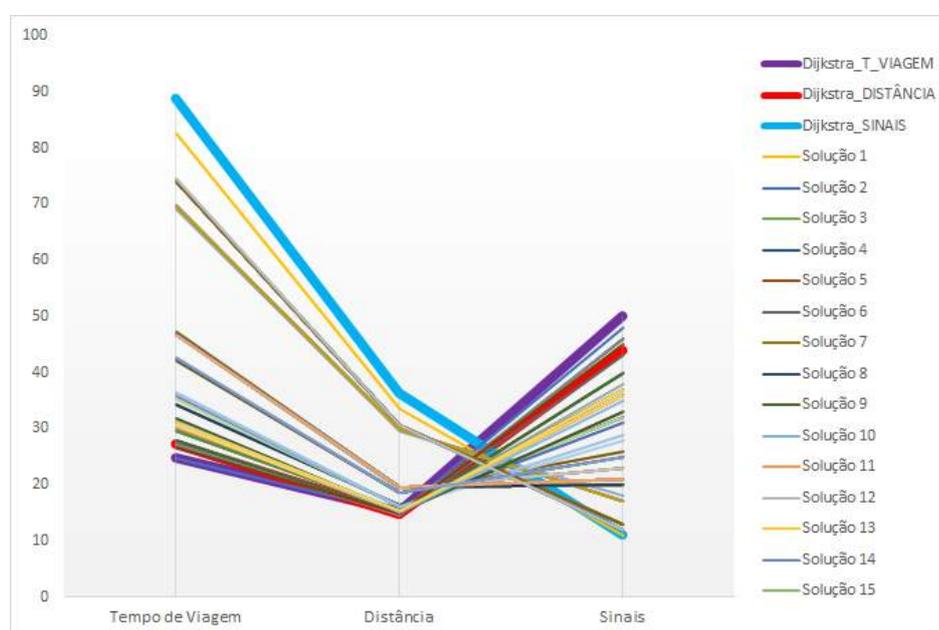
4.2 Análise do segundo grupo de simulações

Nesta seção, é analisado os cenários do segundo grupo de simulações que utilizou o segundo menor passeio configurado com 4 destinos gerados aleatoriamente.

4.2.1 Cenário sem congestionamento

Para o cenário sem congestionamento, como mostra a Figura 31, o desempenho do algoritmo proposto se mostrou melhor quanto a quantidade de *trade-offs* encontrados se comparado com o mesmo cenário do grupo 1. Ainda como esperado, as soluções de Dijkstra_T_VIAGEM e Dijkstra_DISTÂNCIA apresentam o melhor valor teórico para seus respectivos objetivos, porém, estas soluções perdem para a maioria das soluções da Frente quanto ao número de semáforos onde algumas delas perdem pouco em tempo de viagem para ganhar consideravelmente neste objetivo.

Figura 31 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 2 sem congestionamento.

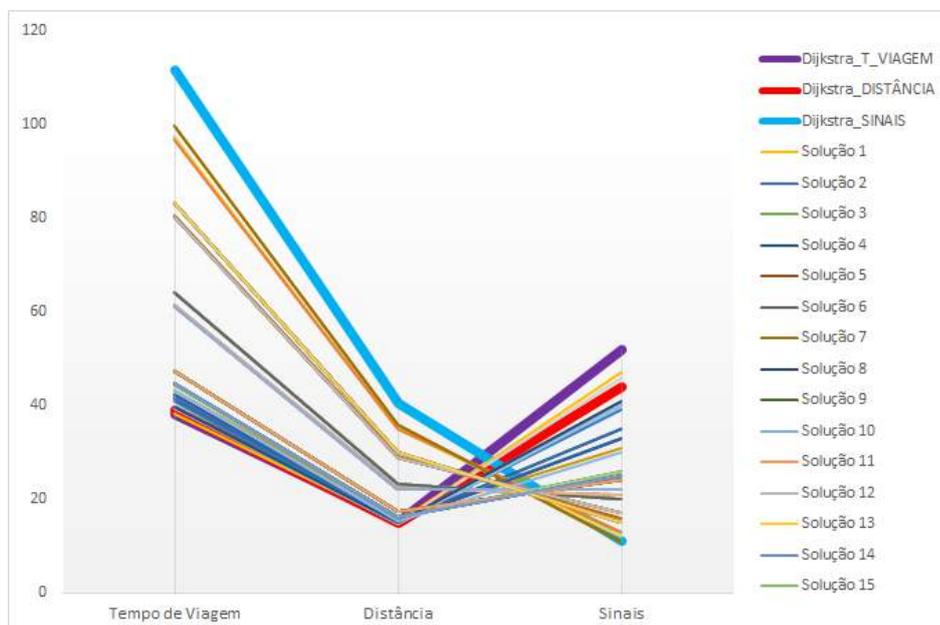


Elaborado pelo Autor

4.2.2 Cenário com 50% de congestionamento

Para o cenário com 50% de congestionamento, a Figura 32 mostra um desempenho semelhante ao cenário anterior, onde a maioria das soluções da Frente de Pareto apresentam trocas viáveis entre os objetivos tempo de viagem e número de semáforos. Ainda é possível observar alguns agrupamentos da Frente, porém, grande partes das soluções tendem a se aproximar das soluções ótimas teóricas nos dois primeiros objetivos.

Figura 32 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 2 com 50% de congestionamento.

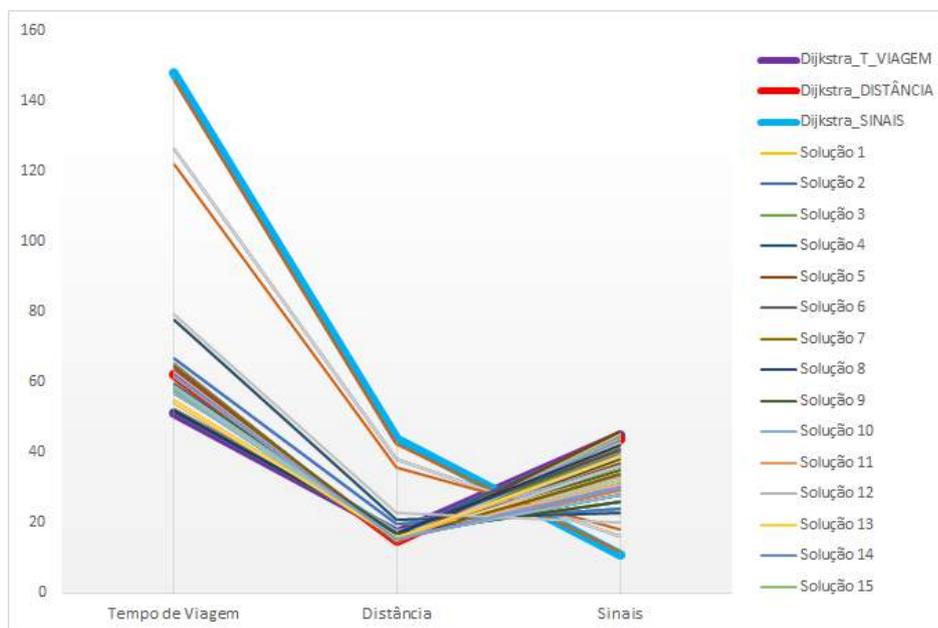


Elaborado pelo Autor

4.2.3 Cenário com 75% de congestionamento

Para o cenário com 75% de congestionamento, como mostra a Figura 33, o padrão parece se repetir se comparado ao mesmo cenário do grupo 1. O algoritmo proposto encontrou diversos *trade-offs* onde a maioria ganhou nos objetivos tempo de viagem e número de semáforos se comparados com a solução Dijkstra_DISTÂNCIA que representa as soluções convencionais utilizadas no dia-a-dia pelos motoristas.

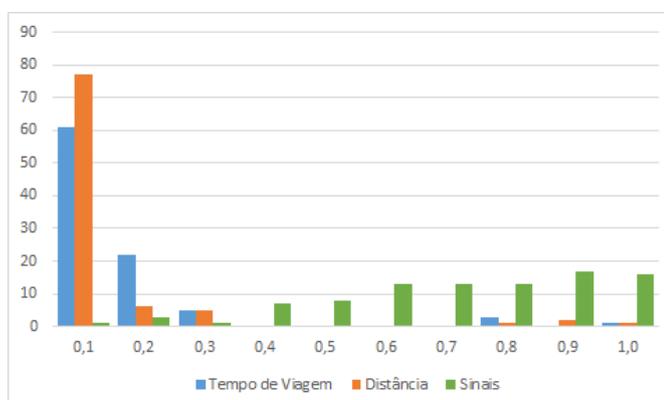
Figura 33 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 2 com 75% de congestionamento.



Elaborado pelo Autor

Na Figura 34, é apresentada a frequência das soluções da Frente de Pareto da Figura 33 em um histograma normalizado para os três objetivos. É possível observar que, assim como no cenário 3 do grupo 1, na seção 4.1.3, a grande maioria das soluções encontradas pelo NSGA-III possuem valores muito próximos aos mínimos teóricos para os objetivos Tempo de Viagem e Distância. É importante notar que, quanto ao objetivo Número de Sinais, as soluções deste grupo estão melhor distribuídas, em termos de frequência, se comparado ao mesmo cenário do grupo 1.

Figura 34 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 75% de congestionamento do grupo 2.



Elaborado pelo Autor

4.2.4 Análise do desempenho do algoritmo para todos os cenários do segundo grupo de simulações

O Quadro 5 também apresenta um padrão semelhante de desempenho do algoritmo proposto em relação ao primeiro grupo, onde a média e o desvio padrão do objetivo que representa o tempo de viagem diminui a cada cenário enquanto os outros dois objetivos variam. Por exemplo, a média da Frente para a Distância tem uma média de 46,8% de perda em relação aos valores mínimos teóricos computados pelo Dijkstra no cenário 1, em seguida, essa média sobe para 51,58%, mas volta a descer para apenas 16,76% no último cenário. Quanto ao objetivo que representa o número de sinais, os valores para a média subiram. Provavelmente, consequência de rotas maiores visto que em um cenário muito congestionado evita as vias expressas e mais rápidas em termos de distância. A tabela completa está disponível no apêndice B, Quadros 17 e 18.

Porém, mesmo assim com estas variações, o NSGA-III apresentou melhores *trade-offs* a medida que os níveis de congestionamento vão aumentando de um cenário para o outro como pode ser observado no Quadro 4. Os exemplos de caminhos gerados para este grupo está disponível no Apêndice B, Figuras 51, 52, 53.

Quadro 4 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 2.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Média	100,66%	46,80%	122,63%	60,98%	51,58%	107,41%	21,63%	16,76%	216,50%
Desvio Padrão	71,37%	43,31%	94,46%	50,20%	46,21%	73,40%	30,67%	33,10%	71,57%

Quadro 5 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 2.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	24,75	3,38%	354,55%	38,21	3,29%	372,73%	51,31	15,32%	309,09%
Dijkstra Distância	10,75%	14,72	300,00%	2,24%	14,72	300,00%	21,58%	14,72	300,00%
Dijkstra Sinais	259,26%	146,13%	11	192,05%	175,64%	11	188,70%	198,07%	11
Solução 1	234,11%	128,23%	0,00%	0,47%	1,47%	327,27%	15,70%	3,17%	218,18%
Solução 2	0,15%	2,24%	336,36%	117,68%	103,74%	36,36%	13,33%	5,64%	272,73%
Solução 3	90,75%	32,05%	81,82%	110,95%	96,93%	45,45%	29,98%	34,65%	118,18%
Solução 4	89,38%	31,78%	90,91%	17,18%	10,23%	127,27%	1,65%	15,24%	227,27%
Solução 5	199,93%	108,35%	9,09%	117,68%	103,74%	36,36%	20,67%	1,99%	236,36%
Solução 6	9,87%	1,20%	290,91%	23,91%	17,04%	118,18%	16,16%	1,86%	263,64%
Solução 7	198,56%	108,07%	18,18%	117,68%	103,74%	36,36%	11,07%	21,49%	172,73%
Solução 8	88,64%	32,69%	90,91%	8,44%	4,38%	218,18%	146,45%	158,50%	45,45%
Solução 9	38,60%	10,50%	190,91%	60,79%	51,10%	90,91%	12,20%	7,16%	218,18%
Solução 10	181,01%	102,09%	54,55%	23,91%	17,04%	118,18%	19,98%	4,74%	172,73%

Elaborado pelo Autor

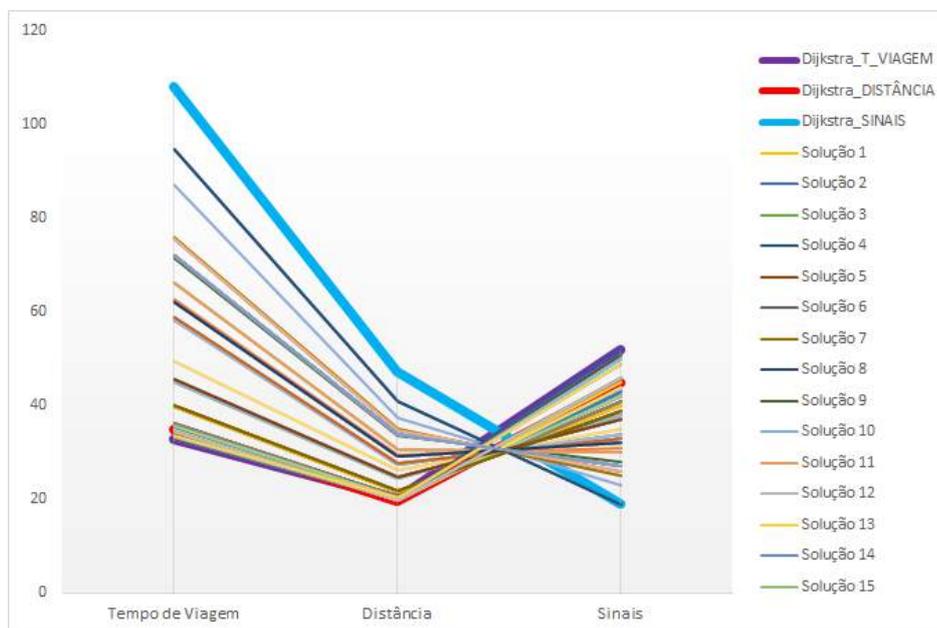
4.3 Análise do terceiro grupo de simulações

Nesta seção, é analisado os cenários do terceiro grupo de simulações que utilizou o passeio configurado com 5 destinos gerados aleatoriamente.

4.3.1 Cenário sem congestionamento

Para o cenário sem congestionamento, a Figura 35 apresenta uma Frente de Pareto melhor espalhada se comparada com os dois grupos anteriores para este mesmo cenário, principalmente, quanto aos dois primeiros objetivos. Com isto, até aqui, é possível notar que o algoritmo proposto vem mostrando uma certa tendência em apresentar mais *trade-offs* adequados a medida que o passeio cresce.

Figura 35 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 3 sem congestionamento.

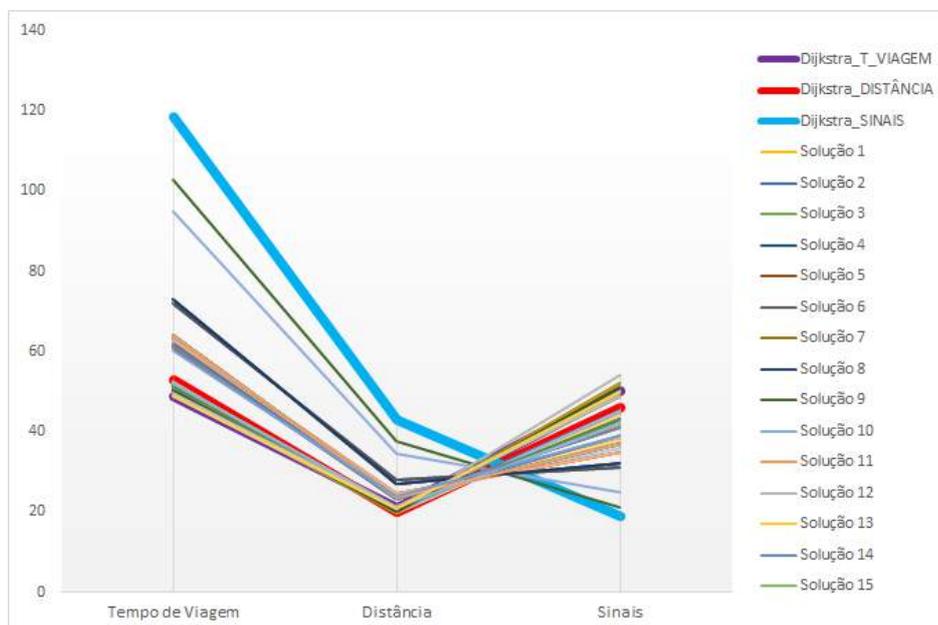


Elaborado pelo Autor

4.3.2 Cenário com 50% de congestionamento

Para o cenário com 50% de congestionamento, a Figura 36 apresenta a Frente de Pareto com menos agrupamentos se comparado com os dois primeiros grupos. A maior parte das soluções permanecem próximas as soluções do Dijkstra repetindo o mesmo comportamento para os mesmos cenários anteriores. Além disto, no ponto de vista da solução Dijkstra_DISTÂNCIA que oferece o menor caminho teórico, grande parte das soluções ganham nos outros dois objetivos ao mesmo tempo que perdem pouco comparado ao menor caminho.

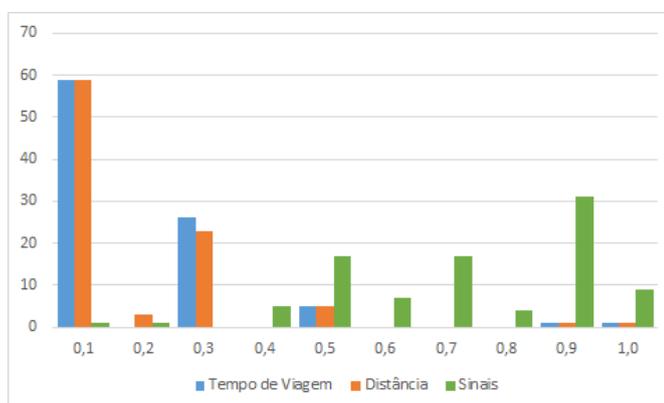
Figura 36 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 3 com 50% de congestionamento.



Elaborado pelo Autor

Diferentemente dos grupos anteriores que tiveram os melhores resultados para o cenário com 75% de congestionamento, este grupo apresentou melhores resultados no cenário com 50% de congestionamento. Isto se dá pelo fato de que a grande maioria das soluções da Frente de Pareto encontrada pelo NSGA-III agruparam-se no primeiro intervalo como é mostrado na Figura 37. Mais de 60% das soluções de Pareto possuem valores muito próximos dos valores ótimos teóricos encontrados pelo Dijkstra para os objetivos Tempo de Viagem e Distância.

Figura 37 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 50% de congestionamento do grupo 3.

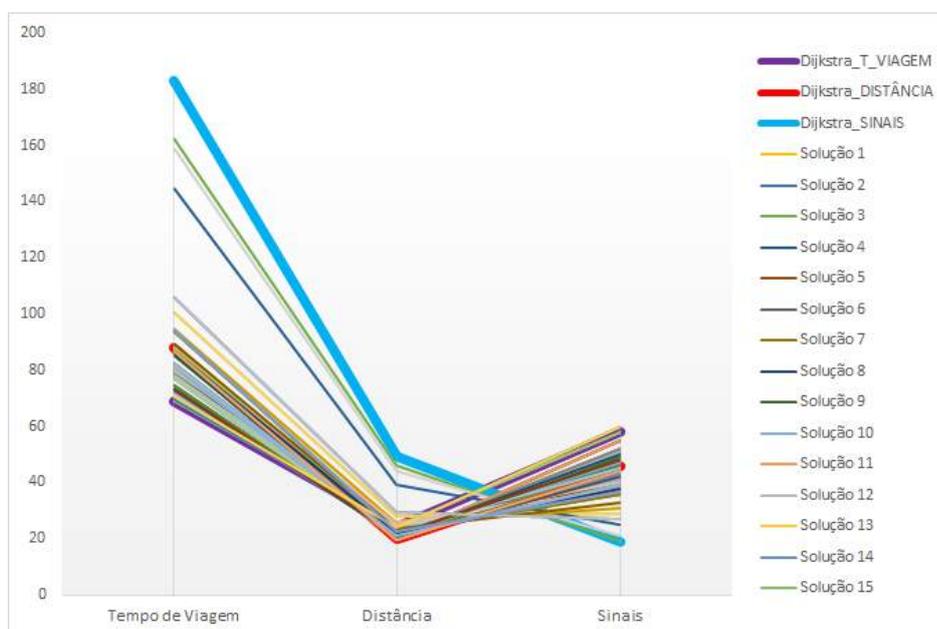


Elaborado pelo Autor

4.3.3 Cenário com 75% de congestionamento

Para o cenário com 75% de congestionamento, a Figura 38 apresenta o mesmo padrão de comportamento da Frente comparado aos mesmos cenários anteriores, onde grande partes das soluções se concentram próximas as soluções computadas pelo Dijkstra proporcionando diversos *trade-offs*.

Figura 38 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 3 com 75% de congestionamento.



Elaborado pelo Autor

4.3.4 Análise do desempenho do algoritmo para todos os cenários do terceiro grupo de simulações

Como mostra o Quadro 6 o terceiro grupo fugiu um pouco ao padrão de desempenho dos dois grupos anteriores. As melhores médias obtidas foram para o segundo cenário e não para o terceiro como ocorreu anteriormente. Porém, os resultados para os cenários com níveis de congestionamento se apresentaram satisfatórios e muitos *trade-offs* viáveis podem ser percebidos em ambos cenários como pode ser visto no Quadro 7. As frentes de Pareto completas para este grupo estão disponíveis no apêndice C - Quadros 19 e 20. Os exemplos de caminhos gerados nos cenários de congestionamento estão disponíveis neste mesmo Apêndice nas Figuras 54, 55 e 56.

Quadro 6 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 3.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Media	47,23%	26,65%	102,57%	15,08%	12,18%	128,09%	23,05%	14,43%	125,63%
Desvio Padrao	52,27%	30,45%	45,89%	19,43%	14,38%	36,12%	21,76%	21,99%	42,40%

Quadro 7 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 3.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	32,89	1,60%	173,68%	48,90	5,24%	163,16%	69,06	22,90%	205,26%
Dijkstra Distância	5,86%	19,76	136,84%	7,84%	19,86	142,11%	27,68%	19,86	142,11%
Dijkstra Sinais	229,04%	139,47%	19	141,87%	116,36%	19	165,48%	149,02%	19
Solução 1	37,35%	24,12%	100,00%	2,05%	0,13%	173,68%	13,20%	15,68%	136,84%
Solução 2	8,66%	2,29%	121,05%	46,98%	40,68%	63,16%	24,74%	3,54%	121,05%
Solução 3	101,45%	54,48%	57,89%	1,43%	6,25%	157,89%	134,86%	132,65%	0,00%
Solução 4	77,33%	39,01%	78,95%	49,07%	35,76%	68,42%	25,06%	-0,26%	136,84%
Solução 5	20,49%	9,46%	110,53%	30,78%	19,13%	94,74%	25,84%	11,80%	100,00%
Solução 6	1,36%	1,86%	163,16%	1,29%	6,30%	157,89%	13,67%	1,76%	163,16%
Solução 7	5,00%	0,61%	142,11%	2,05%	0,13%	173,68%	12,08%	2,23%	189,47%
Solução 8	0,33%	1,07%	168,42%	5,15%	4,58%	126,32%	25,06%	2,04%	136,84%
Solução 9	117,93%	69,71%	47,37%	110,16%	89,30%	10,53%	35,32%	17,26%	89,47%
Solução 10	38,48%	25,80%	94,74%	7,24%	-0,33%	131,58%	11,65%	4,20%	168,42%

Elaborado pelo Autor

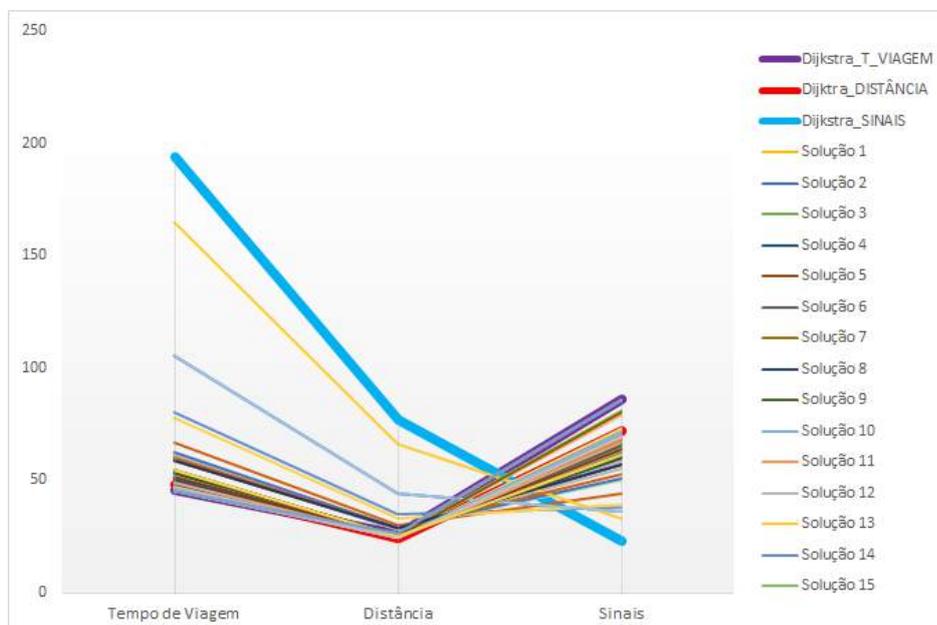
4.4 Análise do quarto grupo de simulações

Nesta seção, é analisado o grupo de simulações com o segundo maior passeio gerado aleatoriamente.

4.4.1 Cenário sem congestionamento

Para o cenário sem congestionamento, a Figura 39, comparando com os mesmos cenários dos grupos anteriores, a maior parte das soluções da Frente estão próximas dos valores teóricos ótimos para os dois primeiro objetivos. O algoritmo ainda conseguiu se desempenhar melhor mesmo com o tamanho do passeio e, consequentemente, sua complexidade aumentando. No objetivo que representa o número de sinais, a maior parte das soluções apresentam melhores valores comparado ao Dijkstra_T_VIAGEM e Dijkstra_DISTÂNCIA.

Figura 39 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 4 sem congestionamento.

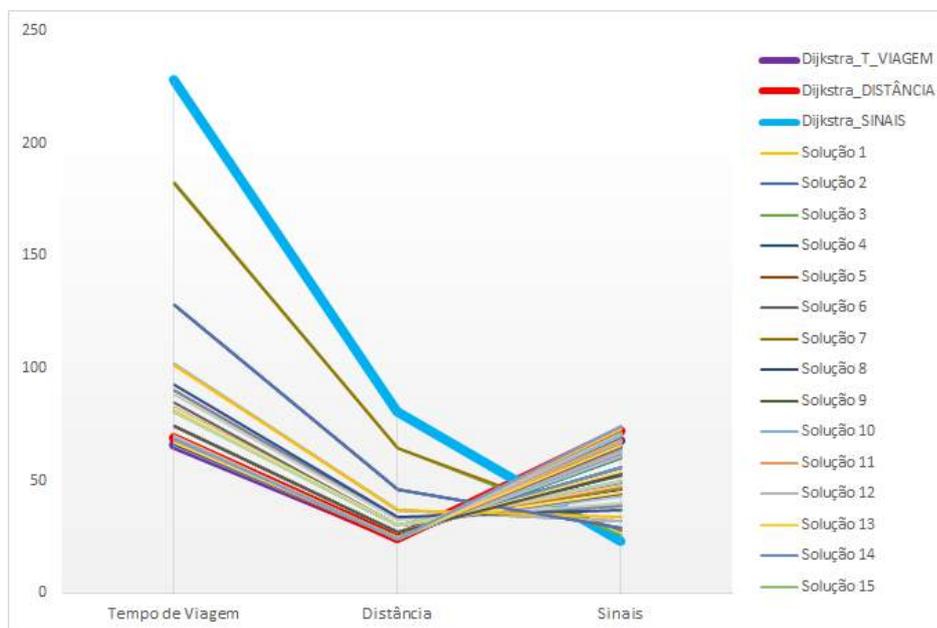


Elaborado pelo Autor

4.4.2 Cenário com 50% de congestionamento

Para o cenário com 50% de congestionamento, o algoritmo proposto também apresentou uma melhora da Frente em relação ao mesmo cenário dos grupos anteriores como mostra a Figura 40. As soluções encontram-se espalhadas e mais próximas aos valores teóricos para os dois primeiros objetivos. É interessante notar também que, quanto ao número de semáforos, as soluções da Frente apresentam-se mais espalhadas e, muitas delas, próximas ao valor teórico para este objetivo. Em nenhum dos outros grupos anteriores as soluções chegaram tão próximas, com exceção do grupo 2, porém, as soluções perdem bastante nos dois primeiros objetivos.

Figura 40 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 4 com 50% de congestionamento.

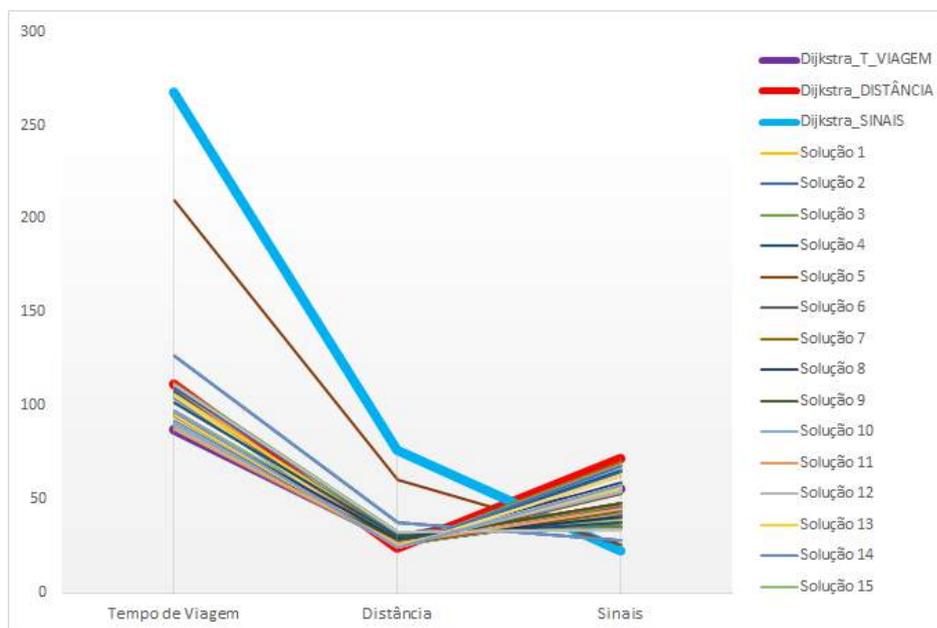


Elaborado pelo Autor

4.4.3 Cenário com 75% de congestionamento

Para o cenário com 75% de congestionamento, o algoritmo continuou apresentando melhoramento da Frente comparando os mesmos cenários dos grupos anteriores, onde grande parte das soluções da Frente estão localizadas próximo aos valores teóricos ótimos, como mostra a Figura 41. É importante ressaltar novamente a qualidade das soluções de Pareto comparadas ao menor caminho (Dijkstra_DISTÂNCIA) que representa os métodos convencionais utilizados em aplicativos GPS ou de roteamento.

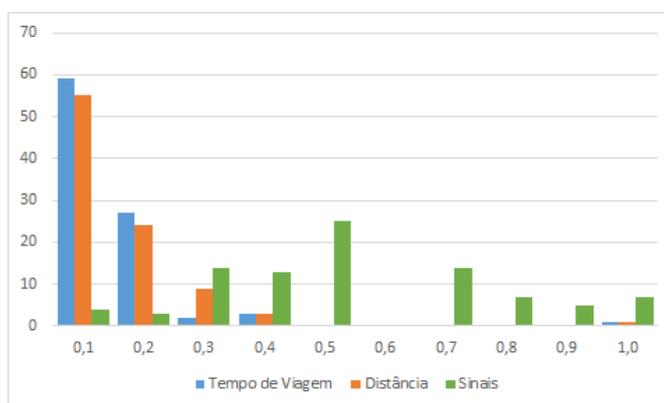
Figura 41 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 4 com 75% de congestionamento.



Elaborado pelo Autor

Para este grupo, os melhores resultados se deram no cenário com 75% de congestionamento, assim como nos dois primeiros grupos. A Figura 42, apresenta o mesmo padrão de agrupamento das soluções de Pareto, onde a maioria dos *trade-offs* apresentam valores muito próximos dos valores de Dijkstra nos objetivos Tempo de Viagem e Distância. Porém, ao contrário dos dois primeiros grupos, os resultados quanto ao objetivo Número de Sinais mostram ser melhores, pois os maiores agrupamentos estão com seus valores mais próximos do mínimo teórico (Dijkstra_SINAIS).

Figura 42 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 75% de congestionamento do grupo 4.



Elaborado pelo Autor

4.4.4 Análise do desempenho do algoritmo para todos os cenários do quarto grupo de simulações

O Quadro 43 apresenta a média e seu desvio padrão das frentes de Pareto obtidas dos três cenários do quarto grupo. Como é possível perceber, o algoritmo proposto ainda continua melhorando a qualidade das soluções conforme aumentam os níveis de congestionamento, mesmo para um passeio ainda maior que os anteriores. Com exceção da distância que sobe no cenário 2, mas volta a cair no último cenário. O Quadro 8 apresenta 10 soluções das frentes dos seus respectivos cenários e o ganho de cada uma por objetivo em relação ao valor mínimo teórico. As frentes completas estão disponíveis no apêndice D, Quadros 21 e 22. Os exemplos de caminhos gerados nos cenários de congestionamento estão disponíveis neste mesmo apêndice nas figuras 57, 58 e 59.

Figura 43 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 4.

	cenário 1			cenário 2			cenário 3		
	Tempo de Viagem	Distância	Nº de Sinais	Tempo de Viagem	Distância	Nº de Sinais	Tempo de Viagem	Distância	Nº de Sinais
Média	18,33%	9,51%	185,59%	18,74%	16,99%	148,30%	13,29%	15,06%	106,95%
Desvio Padrão	34,22%	22,08%	47,82%	31,66%	30,57%	56,26%	17,10%	19,03%	45,55%

Quadro 8 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 4.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ Nº de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ Nº de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ Nº de Sinais (sum)
Dijkstra T.Viagem	45,38	8,18%	273,91%	65,90	1,96%	195,65%	87,48	17,32%	143,48%
Dijkstra Distância	5,60%	24,22	213,04%	4,94%	24,22	213,04%	27,60%	24,22	213,04%
Dijkstra Sinais	327,80%	217,02%	23	246,52%	232,91%	23	206,28%	216,75%	23
Solução 1	3,25%	0,27%	213,04%	0,31%	1,58%	195,65%	21,48%	22,65%	69,57%
Solução 2	10,99%	0,10%	191,30%	54,40%	52,83%	39,13%	44,60%	56,32%	21,74%
Solução 3	16,95%	5,06%	165,22%	177,28%	165,90%	13,04%	8,08%	5,20%	152,17%
Solução 4	0,94%	1,34%	213,04%	1,22%	1,08%	195,65%	0,00%	17,32%	143,48%
Solução 5	4,38%	0,78%	195,65%	24,73%	24,08%	91,30%	20,17%	2,35%	173,91%
Solução 6	12,93%	4,99%	178,26%	28,51%	23,45%	126,09%	2,31%	7,76%	134,78%
Solução 7	8,84%	0,68%	182,61%	176,39%	165,21%	21,74%	21,19%	1,31%	204,35%
Solução 8	33,07%	17,08%	156,52%	24,47%	24,42%	91,30%	11,37%	17,20%	78,26%
Solução 9	5,60%	0,01%	208,70%	4,12%	0,17%	213,04%	10,66%	4,12%	139,13%
Solução 10	8,47%	0,07%	200,00%	0,31%	1,58%	195,65%	2,34%	6,67%	100,00%

Elaborado pelo Autor

4.5 Análise do quinto grupo de simulações

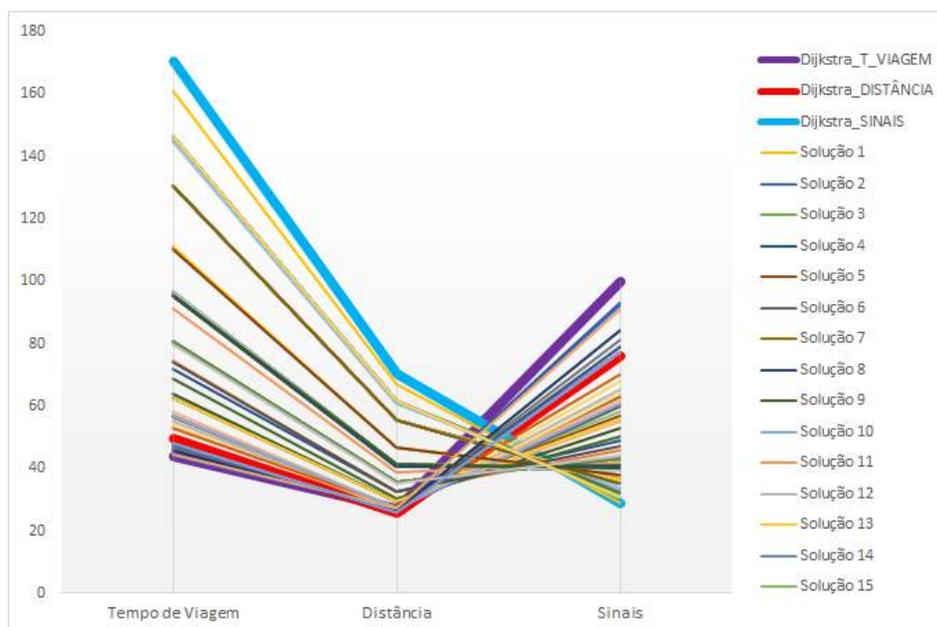
Nesta seção, é analisado o grupo de simulações com o maior passeio gerado aleatoriamente.

4.5.1 Cenário sem congestionamento

Para o cenário sem congestionamento, a Frente obtida se mostrou bem espalhada para os três objetivos como mostra a Figura 44. Comparando este cenário com os mesmos cenários anteriores, é possível observar uma semelhança com o cenário do grupo 3.

Apesar de haver muitas soluções com trocas não muito interessantes, também existem muitas delas que podem representar bons *trade-offs*. Por exemplo, para este cenário a melhor solução teórica seria a Dijkstra_DISTÂNCIA, pois em um cenário sem congestionamento, o menor caminho quase sempre será a melhor opção. Porém, existem soluções da Frente que, apesar de perderem um pouco na distância, ganham em tempo de viagem e empatam, na prática, em número de sinais.

Figura 44 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 5 sem congestionamento.



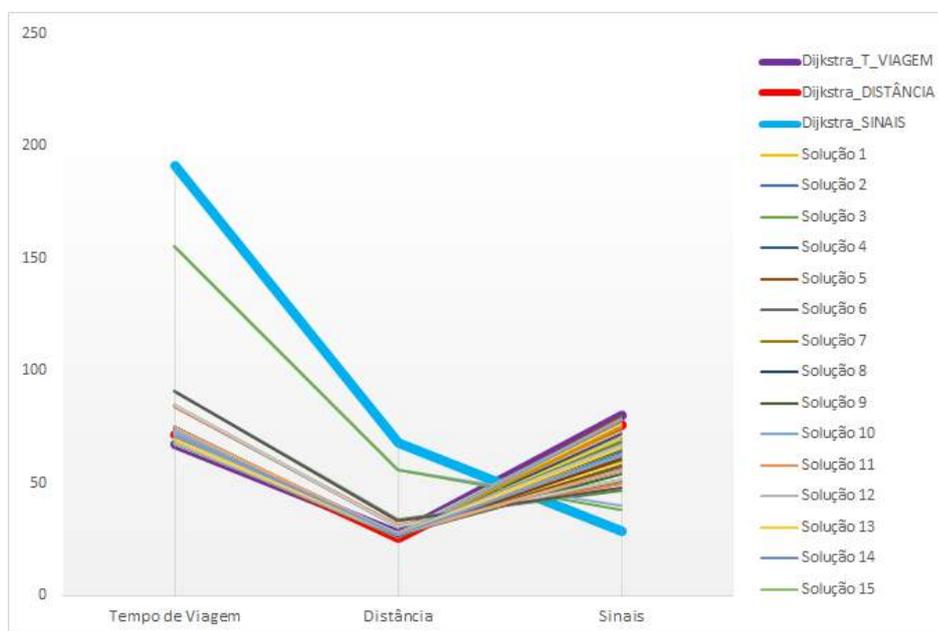
Elaborado pelo Autor

4.5.2 Cenário com 50% de congestionamento

Para o cenário com 50% de congestionamento, como mostra a Figura 45, o algoritmo proposto ainda consegue manter grande parte das soluções de Pareto

próximas dos valores ótimos teóricos em, ao menos, dois objetivos. Este foi o melhor cenário para este nível de congestionamento em todos os grupos de simulação.

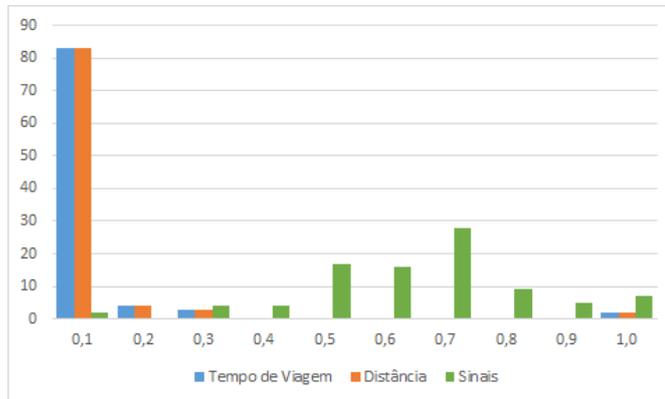
Figura 45 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 5 com 50% de congestionamento.



Elaborado pelo Autor

Assim como no grupo 3, este grupo obteve os melhores resultados no cenário com 50% de congestionamento como é mostrado na Figura 46. Porém, se comparado com o grupo 3, a frequência das soluções de Pareto para o primeiro intervalo alcança quase 90% de toda a Frente de Pareto, enquanto o grupo 3 apresenta em torno de 60% para os objetivos Tempo de Viagem e Distância. Entre todos os grupos, este foi o que apresentou a maior frequência de soluções no primeiro intervalo. Porém, quanto ao objetivo relacionado ao Número de Sinais, os resultados não apresentaram melhoras e continuaram a se distribuir nas classes mais afastadas do valor mínimo teórico para este objetivo.

Figura 46 – Frequência das soluções da Frente de Pareto normalizada para os três objetivos no cenário com 50% de congestionamento do grupo 5.

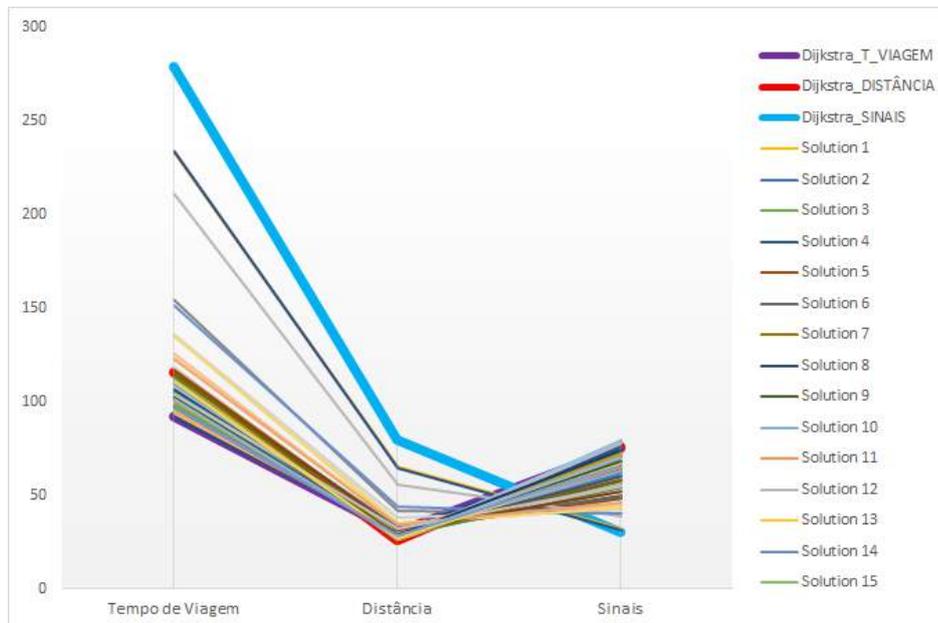


Elaborado pelo Autor

4.5.3 Cenário com 75% de congestionamento

Para o cenário com 75% de congestionamento, ocorre a mesma situação como nos cenários com 50% de congestionamento. A Frente de Pareto, em sua maior parte, ainda se manteve próxima aos valores ótimos teóricos em ao menos dois objetivos, como mostra a Figura 47.

Figura 47 – Comparação entre a Frente de Pareto e os menores caminhos computados pelo Dijkstra para o grupo 5 com 75% de congestionamento.



Elaborado pelo Autor

4.5.4 Análise do desempenho do algoritmo para todos os cenários do quinto grupo de simulações

No Quadro 9, é possível observar que os melhores resultados para este grupo de simulações se deu no cenário 2, onde são encontrados os menores valores em ao menos dois objetivos. Além deste grupo, apenas o grupo 3 obteve um comportamento semelhante. O Quadro 10 apresenta com mais detalhes este comportamento mostrando algumas soluções da Frente de Pareto para cada cenário além de compará-las com os valores mínimos teóricos. As Frentes de Pareto completas estão disponíveis no apêndice E, Quadros 23 e 24. Os exemplos de caminhos gerados são apresentados nas figuras 60, 61 e 62.

Quadro 9 – Média e Desvio Padrão dos custos excedidos das soluções de Pareto para todos os cenários do grupo 5.

	cenário 1			cenário 2			cenário 3		
	T. de Viagem	Distância	Nº de Sinais	T. de Viagem	Distância	Nº de Sinais	T. de Viagem	Distância	Nº de Sinais
Média	112,70%	58,75%	63,19%	10,36%	8,59%	118,63%	22,14%	18,13%	101,05%
Desvio Padrão	77,98%	48,91%	54,96%	19,41%	17,18%	27,67%	25,64%	25,25%	35,59%

Quadro 10 – Comparação do custo excedido em porcentagem entre as soluções de Dijkstra com 10 soluções de Pareto para cada cenário do grupo 5.

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ Nº de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ Nº de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ Nº de Sinais (sum)
Dijkstra T.Viagem	43,69	4,88%	244,83%	67,30	7,15%	175,86%	91,99	17,60%	150,00%
Dijkstra Distância	13,37%	25,87	162,07%	6,01%	25,87	162,07%	25,47%	25,87	153,33%
Dijkstra Sinais	289,70%	171,82%	29	184,32%	163,94%	29	202,77%	206,68%	30
Solução 1	268,32%	158,96%	0,00%	11,34%	4,65%	93,10%	152,77%	151,66%	6,67%
Solução 2	0,89%	4,62%	217,24%	2,09%	7,45%	124,14%	24,33%	10,93%	80,00%
Solução 3	120,88%	60,39%	34,48%	11,17%	3,22%	106,90%	9,08%	6,79%	146,67%
Solução 4	198,99%	115,15%	20,69%	8,56%	1,55%	131,03%	24,74%	0,04%	153,33%
Solução 5	28,78%	4,95%	117,24%	1,59%	0,67%	175,86%	12,32%	19,18%	93,33%
Solução 6	120,88%	60,39%	34,48%	25,19%	19,76%	75,86%	10,97%	13,96%	100,00%
Solução 7	84,74%	37,20%	48,28%	5,95%	7,50%	106,90%	21,19%	3,81%	113,33%
Solução 8	198,67%	114,88%	24,14%	1,52%	8,38%	127,59%	7,21%	8,16%	120,00%
Solução 9	6,87%	3,25%	172,41%	10,71%	5,97%	86,21%	21,18%	1,81%	126,67%
Solução 10	231,84%	135,17%	13,79%	5,47%	5,08%	127,59%	3,37%	12,23%	143,33%

Elaborado pelo Autor

4.6 Análise geral do desempenho do algoritmo para todos os grupos

Nesta seção, será discutido o desempenho geral do algoritmo proposto sob uma perspectiva mais ampla, a fim de identificar a sua capacidade de gerar bons resultados independente do tamanho do passeio em diferentes níveis de congestionamento.

Os Quadros 11, 12 e 13, apresentam as médias e os desvios padrões de cada grupo de simulação e seus respectivos cenários de congestionamento para cada objetivo. A média de cada Frente de Pareto reflete o quão próximas as soluções encontradas estão próximas dos valores ótimos teóricos computados pelo Algoritmo de Dijkstra e o desvio padrão dá uma noção do espaçamento entre as soluções da Frente de Pareto. A divisão das tabelas por objetivo se deu necessário para melhorar a visibilidade dos resultados.

Como é possível observar, o algoritmo proposto apresentou os melhores resultados para as simulações do grupo 3 e 4 para os objetivos Tempo de Viagem e a Distância onde os valores das Frontes de Pareto de cada cenário para Média e o Desvio Padrão decrementam à medida que os cenários se tornam mais congestionados. Este decréscimo significa que, a cada cenário, o algoritmo consegue encontrar *trade-offs* mais próximos dos valores mínimos teóricos já que os valores percentuais significam o custo excedido das soluções de Pareto em relação as soluções de Dijkstra. Quanto ao Número de Semáforos, os melhores resultados se deram para os grupos 3 e 5 observando, também, um decréscimo do custo excedido das soluções encontradas em relação ao valor mínimo teórico.

Em relação aos cenários de congestionamento, o algoritmo proposto se mostrou eficiente a medida que os níveis de congestionamento vão subindo não importando o tamanho do passeio, como é possível observar nos Quadros 11, 12 e 13, os melhores resultados para os objetivos Tempo de Viagem e Distância se deram nos cenários 2 e 3, enquanto que, para o Número de Semáforos, os cenários 1 e 2 apresentaram os melhores resultados.

É importante destacar que o grupo 5, o que possui o maior passeio e, consequentemente, maior complexidade, é o que possui o melhor resultado após os grupos 2 e 3 para os dois primeiros objetivos. O grupo 1, que contém o menor dos passeios, apresentou o pior dos resultados em todos os cenários. Estes resultados podem parecer contraditórios dado um grupo que possui um passeio com apenas dois destinos. Porém, como é possível observar no Quadro 26, o passeio gerado possui pontos muito distantes um dos outros e dado a complexidade do grafo isto pode ter influenciado nos seus resultados. Provavelmente, um passeio com Pontos de Referência mais próximos teria gerado resultados melhores. Uma outra explicação pode estar na quantidade máxima de caminhos definido no *K-shortest path* que não teria sido o suficiente para guardar uma quantidade viável de caminhos subótimos dado a grande distância entre os PRs e a extensividade do grafo.

No Quadro 13, apesar dos valores serem altos em relação ao mínimo teórico, isto não influencia na qualidade das soluções. Na prática, não seria lógico que este último objetivo diminuísse a ponto de chegar próximo do valor teórico, pois as principais

e mais rápidas vias das cidades sempre serão controladas por uma boa quantidade de semáforos para garantir a estabilidade do tráfego de veículos na cidade.

No Quadro 14, apresenta o tempo computacional gasto pelo NSGA-III após a etapa de pré-processamento dos dados para cada cenário de cada grupo de simulação realizado, bem como as suas respectivas médias e desvios padrão. Como é possível observar, a medida que o tamanho do passeio aumenta, maior é o espaço de busca e mais tempo é demandado pelo algoritmo em encontrar a Frente Ótima de Pareto. Em aplicações dinâmicas, um tempo de execução que ultrapasse 3 segundos pode não ser muito interessante, porém, é importante lembrar que a configuração da máquina utilizada reflete diretamente no desempenho do algoritmo. Mesmo assim, em 2 dos 5 grupos o tempo de processamento ficou em torno de 3 segundos enquanto que a média para os outros 3 grupos variou entre 5 e 6 segundos.

Apesar do limitado ambiente computacional, os resultados se mostraram satisfatórios quanto ao tempo de execução em pelo menos 2 grupos. Em relação ao seu desempenho quanto a qualidade dos trade-offs gerados, o algoritmo mostrou ser eficiente em encontrar boas soluções alternativas destacando os melhores resultados onde os níveis de congestionamento são mais altos. Estes resultados também mostram que o operador de mutação desenvolvido atendeu bem as expectativas encontrando bons *trade-offs* em bom tempo computacional para um problema deste nível de complexidade.

Quadro 11 – Média e desvio padrão gerais e específicos dos custos excedidos pelas soluções de Pareto dos grupos e cenários para o Tempo de Viagem.

	Tempo de Viagem					
	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Média
Cenário 1	128,68% (144,09%)	100,66% (71,37%)	47,23% (52,27%)	18,33% (34,22%)	112,70% (77,98%)	81,52% (75,98%)
Cenário 2	73,80% (81,29%)	60,98% (50,20%)	15,08% (19,43%)	18,74% (31,66%)	10,36% (19,41%)	35,79% (40,40%)
Cenário 3	26,10% (31,18%)	21,63% (30,67%)	23,05% (21,76%)	13,29% (17,10%)	22,14% (25,64%)	21,24% (25,27%)
Média	76,19% (85,52%)	61,09% (50,75%)	28,45% (31,15%)	16,79% (27,66%)	48,40% (41,01%)	

Quadro 12 – Média e desvio padrão gerais e específicos dos custos excedidos pelas soluções de Pareto dos grupos e cenários para a Distância.

	Distância					
	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Média
Cenário 1	70,94% (79,32%)	46,80% (43,31%)	26,65% (30,45%)	9,51% (22,08%)	58,75% (48,91%)	42,53% (44,81%)
Cenário 2	80,48% (87,05%)	51,58% (46,21%)	12,18% (14,38%)	16,99% (30,57%)	8,59% (17,18%)	33,97% (39,08%)
Cenário 3	38,68% (41,02%)	16,76% (33,10%)	14,43% (21,99%)	15,06% (19,03%)	18,13% (25,25%)	20,61% (28,08%)
Média	63,36% (69,13%)	38,38% (40,87%)	17,75% (22,27%)	13,86% (23,89%)	28,49% (30,45%)	

Quadro 13 – Média e desvio padrão gerais e específicos dos custos excedidos pelas soluções de Pareto dos grupos e cenários para o Número de Semáforos.

	Número de Semáforos					
	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Média
Cenário 1	196,11% (89,28%)	122,63% (94,46%)	102,57% (45,89%)	185,59% (47,82%)	63,19% (54,96%)	134,02% (66,48%)
Cenário 2	146,92% (94,77%)	107,41% (73,40%)	128,09% (36,12%)	148,30% (56,26%)	118,63% (27,67%)	129,87% (57,64%)
Cenário 3	199,64% (56,66%)	216,50% (71,57%)	125,63% (42,40%)	106,95% (45,55%)	101,05% (35,59%)	149,95% (50,35%)
Média	180,89% (80,24%)	148,85% (79,81%)	118,76% (41,47%)	146,94% (49,88%)	94,29% (39,40%)	

Quadro 14 – Tempo computacional gasto pelo NSGA-III para cada grupo e seus respectivos cenários.

	Tempo de Execução do NSGA-III (ms)				
	Cenário 1	Cenário 2	Cenário 3	Média	D. Padrão
1º Grupo	2344	2683	2520	2516	138
2º Grupo	2891	3328	3261	3160	192
3º Grupo	3541	5284	6547	5124	1232
4º Grupo	6016	6278	5755	6016	214
5º Grupo	4898	6303	5810	5670	582
Média	3938	4775	4779		
D. Padrão	1344	1505	1584		

5 Conclusão

Esta dissertação apresentou uma abordagem utilizando um algoritmo evolucionário multiobjetivo (MOEA) para solucionar um problema de roteamento de veículos em cenários com altos níveis de congestionamento em mapas viários. Neste problema, o usuário deseja realizar um passeio, ou seja, visitar vários pontos em uma determinada sequência. O desafio deste problema está em computar um conjunto de passeios de ótima qualidade em tempo factível utilizando um MOEA levando em consideração três métricas: tempo de viagem, distância e número de semáforos. Para tal, foi necessário realizar um pré processamento dos dados e criado um repositório de caminhos utilizando uma base de dados orientada a grafo no intuito de agilizar a persistência e as solicitações ao banco de dados. O repositório foi preenchido de caminhos subótimos calculados para cada métrica separadamente utilizando o algoritmo *k-shortest path* de Yen. O objetivo principal do repositório consistiu em melhorar o processo de otimização, proporcionando soluções ótimas na criação da população inicial além de diminuir e ordenar o espaço-objetivo para o operador de mutação desenvolvido.

Foram utilizados os dados geoespaciais da plataforma de mapeamento colaborativa *OpenStreetMap* (OSM), porém o algoritmo desenvolvido independe de plataforma. Um pré-processamento dos dados utilizando o *Overpass* API foi feito no intuito de agilizar o processo de conversão dos dados da plataforma OSM para a criação do grafo trabalhado no algoritmo desenvolvido. Porém, o grafo criado deveria obedecer as regras de circulação de trânsito existentes nos mapas OSM e não foram encontrados API's ou projetos *open source* que criassem um mapa viário a partir desses dados. Foi necessário, então, o desenvolvimento de um grafo de rodovias (mapa viário) que obedecesse as regras de circulação de trânsito existentes nos mapas extraídos do OSM.

A escolha de um algoritmo evolucionário multiobjetivo se deu pelo fato de apresentarem o melhor desempenho nestes tipos de problemas, segundo a literatura. O NSGA-III, especificamente, foi escolhido pela sua capacidade de trabalhar com problemas com mais de três objetivos (YUAN; XU; WANG, 2014). Apesar do problema trabalhado nesta dissertação não apresentar mais de três devido as limitações de dados disponíveis no próprio OSM, este tipo de problema pode ser facilmente escalável para mais objetivos se uma outra fonte com mais dados for utilizada. Desta forma, a abordagem proposta não limita-se apenas a problemas com até três objetivos.

Para este trabalho, foi desenvolvido um operador de mutação que garantisse a não desconfiguração dos indivíduos utilizando o repositório de caminhos. Ao mesmo tempo, o operador priorizou os vizinhos mais próximos no processo de busca garantindo

pequenas mudanças nos indivíduos através de uma distribuição gaussiana. Outra vantagem da utilização do repositório se deu na diminuição do espaço-objetivo em que o operador deveria buscar, pois com a utilização do *k-shortest path* de Yen, foi possível definir um número máximo de caminhos subótimos garantindo soluções muito boas teoricamente através da ordenação pelo valor de cada métrica destes caminhos no repositório. Desta forma, evitou-se que o operador de mutação explorasse subespaços muito ruins ao longo do processo evolutivo. Uma técnica de espelhamento também foi desenvolvida para evitar que o operador gerasse indivíduos inválidos.

A estratégia da criação da população inicial também utilizou o repositório de caminhos. Nesta estratégia, metade da população foi criada utilizando aleatoriamente um conjunto de indivíduos de menor custo para cada métrica no repositório. Quanto a etapa de recombinação, foi utilizado o cruzamento uniforme para garantir que os indivíduos não sofressem desconfiguração dos caminhos e nas etapas de seleção de pais e sobrevivente foi utilizado o torneio binário.

Foram realizados cinco grupos de simulações onde cada grupo conteve três cenários com diferentes níveis de congestionamento. Um passeio gerado aleatoriamente obedecendo um limite mínimo de tamanho de 2 e máximo de 10 percorrendo pontos de interesse (escolas, hospitais, praças, etc.) pré-definidos foi estabelecido para cada grupo. Os níveis de congestionamento foram simulados diminuindo os valores do objetivo tempo de viagem nas vias primárias e secundárias da cidade. Desta forma, estabeleceu-se três níveis de congestionamento: 0%, 50% e 75%. Os resultados obtidos das simulações foram comparados com as soluções computadas pelo algoritmo de Dijkstra para cada métrica separadamente. Além disso, cada uma das soluções de Dijkstra teve seus valores para os outros objetivos computados para fins de comparação com as soluções ótimas de Pareto encontradas.

Os resultados das simulações mostraram, no geral, que o algoritmo evolucionário mostrou *trade-offs* mais adequados e em maior quantidade à medida que os níveis de congestionamento aumentaram. No primeiro grupo de simulações, com um passeio constituído de 2 rotas, os melhores resultados se deram quando o nível de congestionamento é mais alto devido aos menores valores obtidos na média e desvio padrão da Frente de Pareto obtida em relação as soluções ótimas teóricas de Dijkstra para este grupo. Esse mesmo padrão também pôde ser visto no segundo grupo onde a perda média das soluções da Frente de Pareto em relação a Dijkstra ficou em torno de 20% para 2 dos 3 objetivos calculados. No terceiro grupo, esse padrão é quebrado e os melhores resultados foram obtidos no segundo cenário (50% de congestionamento) onde a média da Frente de Pareto obtida não ultrapassou os 15% para os dois primeiros objetivos. O quarto grupo voltou a seguir o padrão de desempenho detectado nos dois primeiros obtendo valores em torno de 15% para a média e o último grupo voltou a

quebrar, apresentando os melhores resultados para o segundo cenário obtendo valores em torno de 10% na média da Frente de Pareto.

Quanto ao tempo computacional despendido pelo NSGA-III após a etapa de pré-processamento, em vista de um limitado ambiente computacional, a representação no algoritmo em conjunto com o operador de mutação desenvolvido mostrou ser eficiente obtendo um tempo máximo de até 6 segundos no pior dos casos para encontrar a Frente Ótima de Pareto.

Desta forma, além de um desempenho adequado em termos de tempo de execução, 3 grupos apresentaram os melhores resultados para o cenário com 75% de congestionamento e 2 grupos para o cenário com 50% de congestionamento. Isto evidenciou que o algoritmo proposto se desempenhou melhor nos cenários onde o problema do congestionamento é presente encontrando *trade-offs* adequados em tempo factível.

5.1 Trabalhos futuros

Como trabalhos futuros, mais objetivos importantes podem ser aplicados na otimização dos passeios. Um fator muito importante nas grandes cidades é a segurança e proporcionar soluções que também evitem zonas perigosas é algo que a maioria dos usuários desejam. A reordenação de rotas dentro do passeio de modo a otimizar outros objetivos como a distância e tempo de viagem também é algo a ser considerado, pois existem problemas específicos em que esta reordenação poderia ser importante. No entanto, isto aumentaria bastante a complexidade do problema. Trabalhar com tempo, mais especificamente janelas temporais, é de grande valia para problemas de roteamento de veículos, pois os níveis de fluxo de tráfego não são constantes tornando este um objetivo crítico a ser tratado para trazer soluções que sejam válidas em cenários reais.

Encontrar um método para escolha da melhor solução da Frente de Pareto é algo importante a ser trabalhado, levando em consideração as necessidades do condutor. Além disto, um método para realizar um espalhamento das soluções de Pareto também é interessante, pois muitas soluções das Frontes de Pareto em todos grupos de simulação se agruparam em 1 ou 2 intervalos como foi mostrado nos histogramas.

O desenvolvimento de um algoritmo para conectar corretamente os Pontos de Referência na via correta proporcionaria a criação de um grafo de sobreposição que não foi possível ser criado neste trabalho devido as limitações do *OpenStreetMap*. Este grafo de sobreposição proporcionaria uma maior praticidade em se utilizar a ferramenta e viabilizar a sua utilização prática através de um aplicativo de roteamento.

Escalar o problema proposto nesta dissertação para um problema de *Vehicle Routing Problem* (VRP) seria umas das propostas mais desafiadoras se utilizada em conjunto com janelas temporais, pois o controle de todo ou grande parte do tráfego de veículos proporcionaria alocações de rotas que evitassem o estresse em determinadas zonas ou vias provocando os congestionamentos. Os usuários utilizariam um aplicativo para registrar seus passeios e este controlaria o tráfego alocando rotas que tentassem favorecer as necessidades individuais destes usuários e, ao mesmo tempo, manter o bom fluxo de tráfego para toda a comunidade em uma cidade inteligente.

Referências

- ABRAHAM, A.; JAIN, L.; GOLDBERG, R. *Evolutionary Multiobjective Optimization: Theoretical advances and applications*. [S.l.]: Springer, Berlin, 2005.
- ARBEX, R. O.; CUNHA, C. B. da. Efficient transit network design and frequencies setting multi-objective optimization by alternating objective genetic algorithm. *Transportation Research Part B: Methodological*, v. 81, p. 355 – 376, 2015. ISSN 0191-2615.
- BAYO, J. G. *International Case Studies of Smart Cities: Santander, Spain*. [S.l.]: Inter-American Development Bank, 2016.
- BELLMAN, R. On a routing problem. *Quarterly of Applied Mathematics*, v. 16, n. 1, p. 87 – 90, 1958.
- CARAGLIU, A.; BO, C. D.; NIJKAMP, P. Smart Cities in Europe. *Journal of Urban Technology*, Routledge, v. 18, n. 2, p. 65 – 82, 2011. Disponível em: <<https://doi.org/10.1080/10630732.2011.601117>>.
- COELLO, C. A. C.; LAMONT, G. B. *Applications of multi-objective evolutionary algorithms*. [S.l.]: World Scientific, 2004. v. 1.
- COELLO-COELLO, C. A.; VELDHUIZEN, D. V.; LAMONT, G. B. *Evolutionary Algorithms for Solving Multi-Objective Problems*. [S.l.]: Norwell, MA: Kluwer, 2002.
- CURRAN, K.; CRUMLISH, J.; FISHER, G. OpenStreetMap. *IJICST*, v. 2, n. 1, p. 69 – 78, 2012. Disponível em: <<http://dx.doi.org/10.4018/ijicst.2012010105>>.
- CURRIE, J.; WALKER, R. Traffic Congestion and Infant Health: Evidence from E-ZPass. *American Economic Journal: Applied Economics*, v. 3, n. 1, p. 65 – 90, 2011.
- DEB, K. *Multi-Objective Optimization Using Evolutionary Algorithms*. New York, NY, USA: John Wiley & Sons, Inc., 2001. ISBN 047187339X.
- DEB, K.; JAIN, H. An Evolutionary Many-Objective Optimization Algorithm Using Reference-Point-Based Nondominated Sorting Approach, Part I: Solving Problems With Box Constraints. *IEEE Transactions on Evolutionary Computation*, v. 18, n. 4, p. 577 – 601, 2014. ISSN 1089-778X.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182 – 197, 2002. ISSN 1089-778X.
- DENATRAN. *Frota de veículos*. 2016. Disponível em: <<http://www.denatran.gov.br/index.php/estatistica/237-frota-veiculos>>. Acesso em: 24/03/2017.
- DIJKSTRA, E. W. A note on two problems in connexion with graphs. *Numerische Mathematik*, v. 1, p. 269 – 271, 1959. Disponível em: <<http://doi.acm.org/10.1145/636517.636521>>.

DURILLO, J. J.; NEBRO, A. J.; ALBA, E. The jMetal framework for multi-objective optimization: Design and architecture. In: *IEEE Congress on Evolutionary Computation IS -*. [S.l.: s.n.], 2010. p. 1 – 8.

EIBEN, A. E.; SMITH, J. E. *Introduction to evolutionary computing*. [S.l.]: Springer, 2003. v. 53.

GOH, C.; TAN, K. C. *Evolutionary multi-objective optimization in uncertain environments. Issues and Algorithms*. [S.l.]: Springer, 2009. v. 186.

HAKLAY, M.; WEBER, P. OpenStreetMap: User-Generated Street Maps. *IEEE Pervasive Computing*, v. 7, n. 4, p. 12 – 18, 2008.

HOLLAND, J. H. *Adaptation in natural and artificial systems*. [S.l.]: Ann Arbor, MI: University of Michigan Press, 1975.

HOLLANDS, R. G. Will the real smart city please stand up? *City*, Routledge, v. 12, n. 3, p. 303 – 320, 2008. Disponível em: <<https://doi.org/10.1080/13604810802479126>>.

HOLZSCHUHER, F.; PEINL, R. Performance of Graph Query Languages: Comparison of Cypher, Gremlin and Native Access in Neo4J. In: *Proceedings of the Joint EDBT/ICDT 2013 Workshops*. New York, NY, USA: ACM, 2013. (EDBT '13), p. 195 – 204. ISBN 978-1-4503-1599-9. Disponível em: <<http://doi.acm.org/10.1145/2457317.2457351>>.

JONG, K. A. de. *Evolutionary computation: a unified approach*. [S.l.]: MIT Press, 2006.

KANO, H.; HARA, K. Hybrid Genetic Algorithm for Dynamic Multi-objective Route Planning with Predicted Traffic in a Real-world Road Network. In: *Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2008. (GECCO '08), p. 657 – 664. ISBN 978-1-60558-130-9. Disponível em: <<http://doi.acm.org/10.1145/1389095.1389226>>.

KNITTEL, C. R.; MILLER, D. L.; SANDERS, N. J. Caution, Drivers! Children Present: Traffic, Pollution, and Infant Health. *The Review of Economics and Statistics*, MIT Press, v. 98, n. 2, p. 350 – 366, Nov 2015. ISSN 0034-6535. Doi: 10.1162/REST_a_00548.

KORN, G. A.; KORN, T. M. Appendix B: B9. Plane and Spherical Trigonometry: Formulas Expressed in Terms of the Haversine Function. In: _____. *Mathematical handbook for scientists and engineers: Definitions, theorems, and formulas for reference and review*. 3. ed. New York: Mineola, 2000. p. 892 – 893. ISBN 978-0-486-41147-7.

LEE, S. K. et al. *International Case Studies of Smart Cities: Orlando, United States of America*. [S.l.]: Inter-American Development Bank, 2016.

LINK, J. ã. B. et al. FootPath: Accurate map-based indoor navigation using smartphones. In: *2011 International Conference on Indoor Positioning and Indoor Navigation IS - SN - VO - VL -*. [S.l.: s.n.], 2011. p. 1 – 8.

POTAMIAS, M. et al. Fast Shortest Path Distance Estimation in Large Networks. In: *Proceedings of the 18th ACM Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 2009. (CIKM '09), p. 867 – 876. ISBN 978-1-60558-512-3. Disponível em: <<http://doi.acm.org/10.1145/1645953.1646063>>.

- TARAPATA, Z. Selected Multicriteria Shortest Path Problems: An Analysis of Complexity, Models and Adaptation of Standard Algorithms. *Applied Mathematics and Computer Science*, v. 17, n. 2, p. 269 – 287, 2007. Disponível em: <<http://dx.doi.org/10.2478/v10006-007-0023-2>>.
- TOMTOM TRAFFIC INDEX. *Full Ranking*. 2016. Disponível em: <https://www.tomtom.com/en_gb/trafficindex/list?citySize=LARGE&continent=ALL&country=ALL>. Acesso em: 25/03/2017.
- TOMTOM TRAFFIC INDEX. *Traffic congestion statistics for Orlando based on TomTom's historical database for 2016*. 2016. Disponível em: <https://www.tomtom.com/en_gb/trafficindex/city/orlando>. Acesso em: 27/03/2017.
- TOTH, P.; VIGO, D. *Vehicle routing: problems, methods, and applications*. [S.l.]: SIAM, 2014.
- YAO, Y. et al. An efficient learning-based approach to multi-objective route planning in a smart city. In: *2017 IEEE International Conference on Communications (ICC) IS - SN - VO - VL -*. [S.l.: s.n.], 2017. p. 1 – 6.
- YEN, J. Y. Finding the K Shortest Loopless Paths in a Network. *Management Science*, INFORMS, v. 17, n. 11, p. 712 – 716, Jul 1971. ISSN 0025-1909. Doi: 10.1287/mnsc.17.11.712.
- YIANNAKOULIAS, N.; BLAND, W.; SVENSON, L. W. Estimating the effect of turn penalties and traffic congestion on measuring spatial accessibility to primary health care. *Applied Geography*, v. 39, p. 172 – 182, 2013. ISSN 0143-6228.
- YU, X.; GEN, M. *Introduction to evolutionary algorithms*. [S.l.]: Springer Science & Business Media, 2010.
- YUAN, Y.; XU, H.; WANG, B. An Improved NSGA-III Procedure for Evolutionary Many-objective Optimization. In: *Proceedings of the 2014 Annual Conference on Genetic and Evolutionary Computation*. New York, NY, USA: ACM, 2014. (GECCO '14), p. 661 – 668. ISBN 978-1-4503-2662-9. Disponível em: <<http://doi.acm.org/10.1145/2576768.2598342>>.
- ZERO, L. et al. Bi-objective shortest path problem with one fuzzy cost function applied to dangerous goods transportation on a road network. In: *2016 11th System of Systems Engineering Conference (SoSE) IS - SN - VO - VL -*. [S.l.: s.n.], 2016. p. 1 – 5.

Apêndices

APÊNDICE A – Soluções das frentes de Pareto para o Grupo 1 e exemplos gerados dos seus cenários de congestionamento

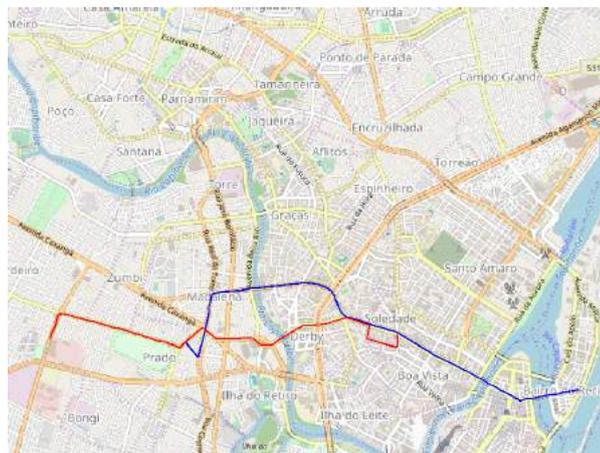
Quadro 15 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 1 (parte 1).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	18,78	1,26%	283,33%	34,09	3,59%	225,00%	52,67	38,63%	216,67%
Dijkstra Distância	1,32%	12,06	275,00%	3,83%	12,06	275,00%	29,56%	12,06	275,00%
Dijkstra Sinais	442,44%	239,66%	12	240,07%	257,62%	12	163,39%	242,02%	12
Solução 1	0,71%	0,19%	275,00%	185,57%	201,67%	0,00%	10,57%	24,09%	208,33%
Solução 2	110,18%	60,63%	183,33%	161,90%	173,55%	91,67%	1,04%	31,76%	225,00%
Solução 3	0,71%	0,19%	275,00%	26,36%	28,30%	183,33%	29,49%	1,20%	258,33%
Solução 4	307,06%	169,06%	91,67%	0,00%	3,59%	225,00%	36,31%	29,06%	183,33%
Solução 5	0,71%	0,19%	275,00%	3,83%	0,00%	275,00%	5,98%	24,99%	216,67%
Solução 6	109,47%	61,70%	191,67%	0,00%	3,59%	225,00%	18,09%	15,75%	241,67%
Solução 7	313,20%	170,08%	83,33%	164,13%	174,39%	83,33%	0,00%	38,63%	216,67%
Solução 8	0,71%	0,19%	275,00%	2,23%	4,42%	216,67%	112,54%	143,63%	91,67%
Solução 9	306,45%	169,25%	91,67%	23,67%	31,70%	133,33%	29,83%	29,52%	175,00%
Solução 10	1,32%	0,00%	275,00%	185,57%	201,67%	0,00%	14,27%	52,11%	150,00%
Solução 11	306,45%	169,25%	91,67%	1,09%	4,61%	216,67%	1,30%	31,86%	208,33%
Solução 12	0,61%	1,07%	283,33%	164,13%	174,39%	83,33%	15,60%	16,09%	233,33%
Solução 13	0,00%	1,26%	283,33%	2,69%	0,19%	275,00%	11,68%	24,07%	208,33%
Solução 14	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	23,49%	2,13%	266,67%
Solução 15	109,47%	61,70%	191,67%	23,67%	31,70%	133,33%	18,25%	51,68%	158,33%
Solução 16	0,61%	1,07%	283,33%	164,13%	174,39%	83,33%	116,21%	172,73%	0,00%
Solução 17	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	11,50%	22,62%	250,00%
Solução 18	0,00%	1,26%	283,33%	3,78%	1,21%	266,67%	18,10%	9,37%	250,00%
Solução 19	307,06%	169,06%	91,67%	1,09%	4,61%	216,67%	11,50%	22,93%	241,67%
Solução 20	1,32%	0,00%	275,00%	23,67%	31,70%	133,33%	18,80%	9,15%	250,00%
Solução 21	110,18%	60,63%	183,33%	0,00%	3,59%	225,00%	1,30%	31,86%	208,33%
Solução 22	0,71%	0,19%	275,00%	185,57%	201,67%	0,00%	10,57%	24,09%	208,33%
Solução 23	0,61%	1,07%	283,33%	26,36%	28,30%	183,33%	23,49%	2,13%	266,67%
Solução 24	313,20%	170,08%	83,33%	2,23%	4,42%	216,67%	36,31%	29,06%	183,33%
Solução 25	307,06%	169,06%	91,67%	161,90%	173,55%	91,67%	14,27%	51,80%	158,33%
Solução 26	1,32%	0,00%	275,00%	23,67%	31,70%	133,33%	15,60%	16,09%	233,33%
Solução 27	0,61%	1,07%	283,33%	161,90%	173,55%	91,67%	5,69%	24,99%	241,67%
Solução 28	110,18%	60,63%	183,33%	0,00%	3,59%	225,00%	110,72%	145,60%	75,00%
Solução 29	110,18%	60,63%	183,33%	185,57%	201,67%	0,00%	0,29%	39,03%	191,67%
Solução 30	1,32%	0,00%	275,00%	3,83%	0,00%	275,00%	17,86%	15,80%	241,67%
Solução 31	307,06%	169,06%	91,67%	161,90%	173,55%	91,67%	18,51%	51,93%	133,33%
Solução 32	0,61%	1,07%	283,33%	0,00%	3,59%	225,00%	33,23%	29,10%	183,33%
Solução 33	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	29,56%	0,00%	275,00%
Solução 34	0,71%	0,19%	275,00%	26,36%	28,30%	183,33%	11,68%	24,07%	208,33%
Solução 35	306,45%	169,25%	91,67%	3,83%	0,00%	275,00%	112,54%	143,63%	91,67%
Solução 36	0,71%	0,19%	275,00%	164,13%	174,39%	83,33%	1,00%	32,02%	225,00%
Solução 37	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	14,24%	52,05%	158,33%
Solução 38	1,32%	0,00%	275,00%	0,00%	3,59%	225,00%	33,23%	29,10%	183,33%
Solução 39	307,06%	169,06%	91,67%	26,36%	28,30%	183,33%	6,75%	24,73%	225,00%
Solução 40	1,32%	0,00%	275,00%	2,69%	0,19%	275,00%	17,86%	15,80%	241,67%
Solução 41	110,18%	60,63%	183,33%	2,69%	0,19%	275,00%	19,09%	9,14%	250,00%
Solução 42	0,61%	1,07%	283,33%	23,67%	31,70%	133,33%	1,04%	37,93%	216,67%
Solução 43	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	15,60%	16,09%	233,33%
Solução 44	7,46%	1,02%	266,67%	0,00%	3,59%	225,00%	11,50%	22,62%	250,00%
Solução 45	1,32%	0,00%	275,00%	164,13%	174,39%	83,33%	110,72%	145,60%	75,00%
Solução 46	110,18%	60,63%	183,33%	2,69%	0,19%	275,00%	1,04%	31,76%	225,00%

Quadro 16 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 1 (parte 2).

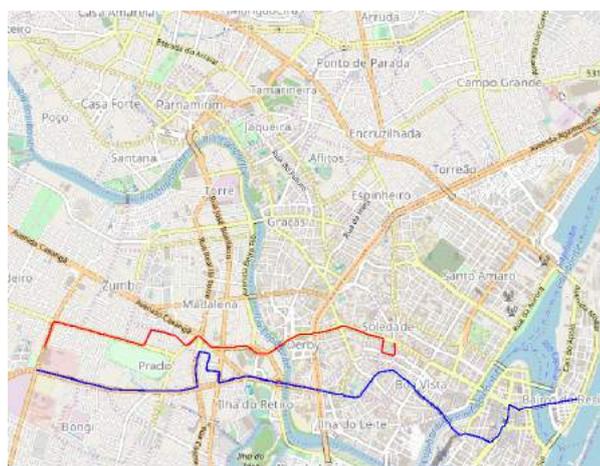
	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	18,78	1,26%	283,33%	34,09	3,59%	225,00%	52,67	38,63%	216,67%
Dijkstra Distância	1,32%	12,06	275,00%	3,83%	12,06	275,00%	29,56%	12,06	275,00%
Dijkstra Sinais	442,44%	239,66%	12	240,07%	257,62%	12	163,39%	242,02%	12
Solução 47	0,00%	1,26%	283,33%	23,67%	31,70%	133,33%	1,30%	31,86%	208,33%
Solução 48	307,06%	169,06%	91,67%	1,09%	4,61%	216,67%	10,57%	24,09%	208,33%
Solução 49	109,47%	61,70%	191,67%	185,57%	201,67%	0,00%	112,54%	143,63%	91,67%
Solução 50	0,71%	0,19%	275,00%	161,90%	173,55%	91,67%	28,08%	1,23%	258,33%
Solução 51	1,32%	0,00%	275,00%	185,57%	201,67%	0,00%	27,49%	0,20%	275,00%
Solução 52	306,45%	169,25%	91,67%	3,78%	1,21%	266,67%	18,51%	51,93%	133,33%
Solução 53	7,46%	1,02%	266,67%	23,67%	31,70%	133,33%	36,31%	29,06%	183,33%
Solução 54	415,93%	229,69%	0,00%	163,04%	173,36%	91,67%	11,68%	24,07%	208,33%
Solução 55	0,71%	0,19%	275,00%	1,14%	3,40%	225,00%	0,29%	38,62%	191,67%
Solução 56	306,45%	169,25%	91,67%	4,92%	1,02%	266,67%	1,00%	32,02%	225,00%
Solução 57	1,32%	0,00%	275,00%	23,67%	31,70%	133,33%	11,50%	22,93%	241,67%
Solução 58	110,18%	60,63%	183,33%	161,90%	173,55%	91,67%	17,39%	15,97%	241,67%
Solução 59	0,61%	1,07%	283,33%	0,00%	3,59%	225,00%	36,31%	29,06%	183,33%
Solução 60	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	23,78%	2,12%	266,67%
Solução 61	415,93%	229,69%	0,00%	26,36%	28,30%	183,33%	19,09%	9,14%	250,00%
Solução 62	0,61%	1,07%	283,33%	1,09%	4,61%	216,67%	10,57%	24,09%	208,33%
Solução 63	110,18%	60,63%	183,33%	185,57%	201,67%	0,00%	32,33%	29,18%	183,33%
Solução 64	0,71%	0,19%	275,00%	163,04%	173,36%	91,67%	15,60%	16,09%	233,33%
Solução 65	0,61%	1,07%	283,33%	2,23%	4,42%	216,67%	18,51%	51,93%	133,33%
Solução 66	306,45%	169,25%	91,67%	185,57%	201,67%	0,00%	15,43%	51,82%	141,67%
Solução 67	307,06%	169,06%	91,67%	161,90%	173,55%	91,67%	112,54%	143,63%	91,67%
Solução 68	0,00%	1,26%	283,33%	2,69%	0,19%	275,00%	101,07%	159,38%	58,33%
Solução 69	307,06%	169,06%	91,67%	26,36%	28,30%	183,33%	11,68%	24,07%	208,33%
Solução 70	0,61%	1,07%	283,33%	0,00%	3,59%	225,00%	6,75%	24,73%	225,00%
Solução 71	306,45%	169,25%	91,67%	3,83%	0,00%	275,00%	1,30%	31,86%	208,33%
Solução 72	0,00%	1,26%	283,33%	185,57%	201,67%	0,00%	29,49%	1,20%	258,33%
Solução 73	1,32%	0,00%	275,00%	1,09%	4,61%	216,67%	10,57%	24,09%	208,33%
Solução 74	306,45%	169,25%	91,67%	164,13%	174,39%	83,33%	18,09%	15,75%	241,67%
Solução 75	306,45%	169,25%	91,67%	23,67%	31,70%	133,33%	1,30%	31,86%	208,33%
Solução 76	1,32%	0,00%	275,00%	2,23%	4,42%	216,67%	112,54%	143,63%	91,67%
Solução 77	109,47%	61,70%	191,67%	164,13%	174,39%	83,33%	15,60%	16,09%	233,33%
Solução 78	0,61%	1,07%	283,33%	23,67%	31,70%	133,33%	14,24%	52,05%	158,33%
Solução 79	307,06%	169,06%	91,67%	185,57%	201,67%	0,00%	36,31%	29,06%	183,33%
Solução 80	1,32%	0,00%	275,00%	1,09%	4,61%	216,67%	6,75%	24,73%	225,00%
Solução 81	0,00%	1,26%	283,33%	26,36%	28,30%	183,33%	1,04%	31,76%	225,00%
Solução 82	109,47%	61,70%	191,67%	185,57%	201,67%	0,00%	18,80%	9,15%	250,00%
Solução 83	307,06%	169,06%	91,67%	4,92%	1,02%	266,67%	105,02%	146,52%	83,33%
Solução 84	0,00%	1,26%	283,33%	1,09%	4,61%	216,67%	28,37%	1,22%	258,33%
Solução 85	306,45%	169,25%	91,67%	161,90%	173,55%	91,67%	0,29%	38,62%	191,67%
Solução 86	0,71%	0,19%	275,00%	3,83%	0,00%	275,00%	11,68%	24,07%	208,33%
Solução 87	0,00%	1,26%	283,33%	1,14%	3,40%	225,00%	9,73%	22,81%	250,00%
Solução 88	306,45%	169,25%	91,67%	163,04%	173,36%	91,67%	27,49%	0,20%	275,00%
Solução 89	307,06%	169,06%	91,67%	23,67%	31,70%	133,33%	18,25%	51,68%	158,33%
Solução 90	0,61%	1,07%	283,33%	185,57%	201,67%	0,00%	29,83%	29,52%	175,00%
Solução 91	1,32%	0,00%	275,00%	2,69%	0,19%	275,00%	18,51%	51,93%	133,33%
Solução 92	313,20%	170,08%	83,33%	0,00%	3,59%	225,00%	27,49%	0,20%	275,00%
Média	128,68%	70,94%	196,11%	73,80%	80,48%	146,92%	26,10%	38,68%	199,64%
Desvio Padrão	144,09%	79,32%	89,28%	81,29%	87,05%	94,77%	31,18%	41,02%	56,66%

Figura 48 – Exemplo de um passeio gerado no cenário 1 do grupo 1.



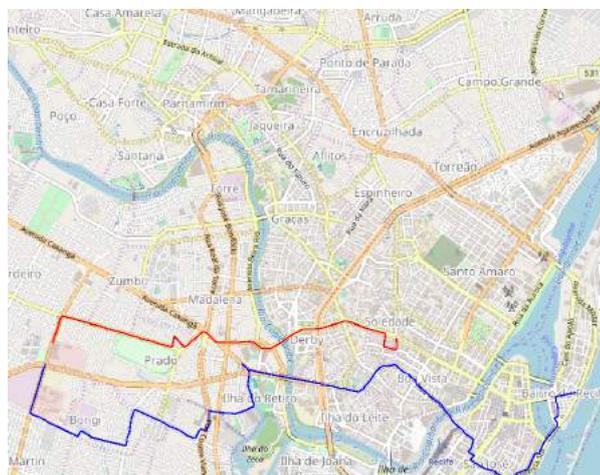
JMapView modificado pelo Autor

Figura 49 – Exemplo de um passeio gerado no cenário 2 do grupo 1.



JMapView modificado pelo Autor

Figura 50 – Exemplo de um passeio gerado no cenário 3 do grupo 1.



JMapView modificado pelo Autor

APÊNDICE B – Soluções das frentes de Pareto para o Grupo 2 e exemplos gerados dos seus cenários de congestionamento

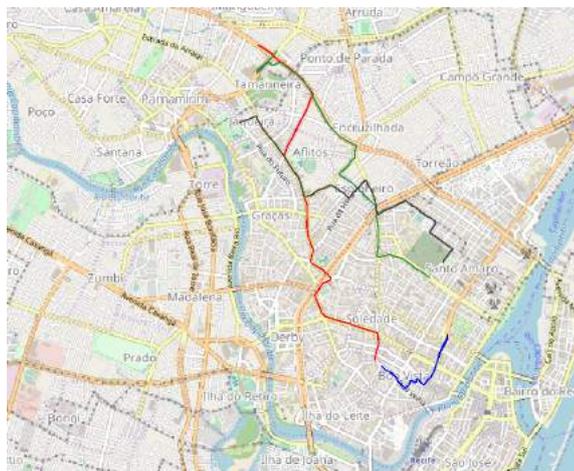
Quadro 17 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 2 (parte 1).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	24,75	3,38%	354,55%	38,21	3,29%	372,73%	51,31	15,32%	309,09%
Dijkstra Distância	10,75%	14,72	300,00%	2,24%	14,72	300,00%	21,58%	14,72	300,00%
Dijkstra Sinais	259,26%	146,13%	11	192,05%	175,64%	11	188,70%	198,07%	11
Solução 1	234,11%	128,23%	0,00%	0,47%	1,47%	327,27%	15,70%	3,17%	218,18%
Solução 2	0,15%	2,24%	336,36%	117,68%	103,74%	36,36%	13,33%	5,64%	272,73%
Solução 3	90,75%	32,05%	81,82%	110,95%	96,93%	45,45%	29,98%	34,65%	118,18%
Solução 4	89,38%	31,78%	90,91%	17,18%	10,23%	127,27%	1,65%	15,24%	227,27%
Solução 5	199,93%	108,35%	9,09%	117,68%	103,74%	36,36%	20,67%	1,99%	236,36%
Solução 6	9,87%	1,20%	290,91%	23,91%	17,04%	118,18%	16,16%	1,86%	263,64%
Solução 7	198,56%	108,07%	18,18%	117,68%	103,74%	36,36%	11,07%	21,49%	172,73%
Solução 8	88,64%	32,69%	90,91%	8,44%	4,38%	218,18%	146,45%	158,50%	45,45%
Solução 9	38,60%	10,50%	190,91%	60,79%	51,10%	90,91%	12,20%	7,16%	218,18%
Solução 10	181,01%	102,09%	54,55%	23,91%	17,04%	118,18%	19,98%	4,74%	172,73%
Solução 11	43,77%	9,50%	181,82%	3,59%	2,83%	272,73%	20,73%	0,14%	318,18%
Solução 12	69,72%	26,43%	136,36%	153,67%	137,52%	18,18%	6,63%	9,17%	181,82%
Solução 13	181,75%	101,17%	54,55%	10,78%	3,32%	200,00%	14,06%	2,40%	309,09%
Solução 14	90,01%	32,97%	81,82%	59,90%	50,83%	100,00%	5,52%	9,41%	209,09%
Solução 15	38,60%	10,50%	190,91%	154,56%	137,79%	9,09%	26,10%	0,14%	272,73%
Solução 16	181,01%	102,09%	54,55%	7,04%	1,90%	254,55%	15,39%	15,15%	136,36%
Solução 17	88,64%	32,69%	90,91%	23,91%	17,04%	118,18%	184,65%	188,80%	9,09%
Solução 18	10,13%	0,30%	318,18%	117,68%	103,74%	36,36%	15,01%	5,91%	154,55%
Solução 19	199,93%	108,35%	9,09%	13,14%	10,23%	181,82%	15,31%	4,76%	200,00%
Solução 20	25,39%	4,03%	227,27%	117,68%	103,74%	36,36%	1,29%	11,79%	309,09%
Solução 21	71,83%	25,79%	127,27%	23,91%	17,04%	118,18%	12,90%	4,41%	281,82%
Solução 22	88,64%	32,69%	90,91%	117,68%	103,74%	36,36%	25,16%	3,90%	209,09%
Solução 23	199,30%	107,15%	18,18%	23,91%	17,04%	118,18%	5,97%	6,14%	272,73%
Solução 24	13,02%	0,41%	263,64%	17,18%	10,23%	127,27%	6,43%	7,11%	254,55%
Solução 25	89,38%	31,78%	90,91%	23,91%	17,04%	118,18%	51,54%	42,21%	109,09%
Solução 26	179,65%	101,82%	63,64%	117,68%	103,74%	36,36%	0,00%	15,32%	309,09%
Solução 27	13,02%	0,41%	263,64%	17,18%	10,23%	127,27%	12,86%	7,43%	209,09%
Solução 28	13,02%	0,41%	263,64%	9,89%	5,05%	200,00%	6,63%	9,17%	181,82%
Solução 29	90,01%	32,97%	81,82%	23,91%	17,04%	118,18%	25,16%	3,90%	209,09%
Solução 30	199,30%	107,15%	18,18%	117,68%	103,74%	36,36%	15,20%	2,74%	227,27%
Solução 31	69,72%	26,43%	136,36%	16,29%	9,96%	136,36%	6,13%	8,74%	190,91%
Solução 32	38,60%	10,50%	190,91%	117,68%	103,74%	36,36%	20,73%	0,14%	318,18%
Solução 33	181,01%	102,09%	54,55%	67,52%	57,91%	81,82%	19,86%	1,88%	245,45%
Solução 34	88,59%	31,77%	109,09%	9,89%	5,05%	200,00%	13,33%	5,64%	272,73%
Solução 35	28,53%	3,24%	200,00%	110,06%	96,65%	54,55%	14,49%	1,88%	290,91%
Solução 36	199,93%	108,35%	9,09%	23,91%	17,04%	118,18%	15,31%	4,76%	200,00%
Solução 37	10,13%	0,30%	318,18%	154,56%	137,79%	9,09%	12,90%	4,41%	281,82%
Solução 38	198,56%	108,07%	18,18%	9,89%	5,05%	200,00%	29,98%	34,65%	118,18%
Solução 39	88,64%	32,69%	90,91%	67,52%	57,91%	81,82%	15,39%	15,15%	136,36%
Solução 40	10,13%	0,30%	318,18%	8,44%	4,38%	218,18%	1,89%	11,12%	290,91%
Solução 41	88,59%	31,77%	109,09%	161,29%	144,61%	0,00%	26,76%	0,42%	263,64%
Solução 42	199,30%	107,15%	18,18%	59,90%	50,83%	100,00%	11,07%	20,28%	172,73%
Solução 43	90,75%	32,05%	81,82%	110,06%	96,65%	54,55%	6,59%	6,95%	254,55%
Solução 44	199,30%	107,15%	18,18%	17,18%	10,23%	127,27%	15,01%	5,91%	154,55%
Solução 45	19,68%	4,24%	236,36%	23,91%	17,04%	118,18%	0,68%	15,32%	300,00%
Solução 46	47,45%	9,50%	154,55%	67,52%	57,91%	81,82%	19,98%	4,74%	172,73%

Quadro 18 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 2 (parte 2).

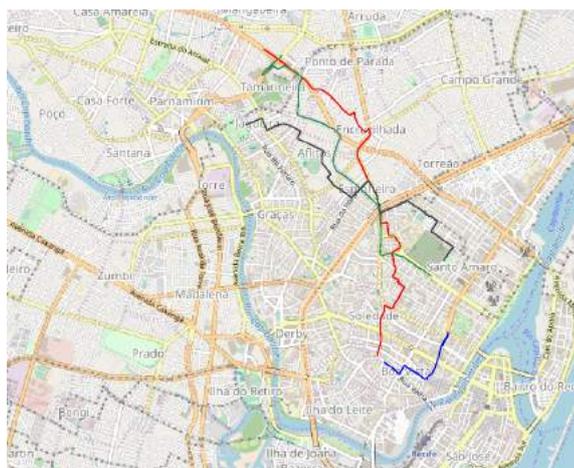
	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	24,75	3,38%	354,55%	38,21	3,29%	372,73%	51,31	15,32%	309,09%
Dijkstra Distância	10,75%	14,72	300,00%	2,24%	14,72	300,00%	21,58%	14,72	300,00%
Dijkstra Sinais	259,26%	146,13%	11	192,05%	175,64%	11	188,70%	198,07%	11
Solução 47	199,93%	108,35%	9,09%	110,06%	96,65%	54,55%	5,97%	6,14%	272,73%
Solução 48	90,01%	32,97%	81,82%	2,85%	0,94%	318,18%	146,45%	158,50%	45,45%
Solução 49	88,59%	31,77%	109,09%	17,18%	10,23%	127,27%	51,54%	42,21%	109,09%
Solução 50	181,01%	102,09%	54,55%	67,52%	57,91%	81,82%	1,65%	15,24%	227,27%
Solução 51	42,70%	9,50%	190,91%	117,68%	103,74%	36,36%	5,97%	8,90%	190,91%
Solução 52	200,67%	107,43%	9,09%	17,18%	10,23%	127,27%	12,86%	7,43%	209,09%
Solução 53	88,59%	31,77%	109,09%	153,67%	137,52%	18,18%	14,35%	5,64%	163,64%
Solução 54	22,41%	2,97%	245,45%	59,90%	50,83%	100,00%	13,33%	5,64%	272,73%
Solução 55	88,59%	31,77%	109,09%	117,68%	103,74%	36,36%	21,56%	0,28%	300,00%
Solução 56	45,29%	9,22%	181,82%	3,87%	0,81%	263,64%	29,98%	34,65%	118,18%
Solução 57	181,01%	102,09%	54,55%	67,52%	57,91%	81,82%	15,39%	15,15%	136,36%
Solução 58	199,93%	108,35%	9,09%	117,68%	103,74%	36,36%	15,04%	2,89%	227,27%
Solução 59	21,05%	4,52%	227,27%	17,18%	10,23%	127,27%	25,41%	2,89%	209,09%
Solução 60	90,01%	32,97%	81,82%	67,52%	57,91%	81,82%	14,65%	4,48%	209,09%
Solução 61	90,01%	32,97%	81,82%	161,29%	144,61%	0,00%	1,73%	11,28%	290,91%
Solução 62	199,30%	107,15%	18,18%	23,91%	17,04%	118,18%	6,43%	7,11%	254,55%
Solução 63	13,02%	0,41%	263,64%	16,29%	9,96%	136,36%	5,97%	6,14%	272,73%
Solução 64	181,75%	101,17%	54,55%	17,18%	10,23%	127,27%	11,07%	21,49%	172,73%
Solução 65	21,05%	4,52%	227,27%	117,68%	103,74%	36,36%	14,06%	2,40%	309,09%
Solução 66	88,64%	32,69%	90,91%	59,90%	50,83%	100,00%	51,54%	42,21%	109,09%
Solução 67	88,59%	31,77%	109,09%	23,91%	17,04%	118,18%	12,90%	4,41%	281,82%
Solução 68	19,68%	4,24%	236,36%	17,18%	10,23%	127,27%	6,63%	9,17%	181,82%
Solução 69	199,93%	108,35%	9,09%	117,68%	103,74%	36,36%	1,34%	15,30%	254,55%
Solução 70	199,30%	107,15%	18,18%	10,78%	3,32%	200,00%	19,98%	4,74%	172,73%
Solução 71	88,64%	32,69%	90,91%	110,95%	96,93%	45,45%	137,88%	144,18%	63,64%
Solução 72	10,13%	0,30%	318,18%	110,06%	96,65%	54,55%	26,76%	0,42%	263,64%
Solução 73	181,01%	102,09%	54,55%	117,68%	103,74%	36,36%	15,31%	2,00%	281,82%
Solução 74	88,59%	31,77%	109,09%	10,78%	3,32%	200,00%	0,02%	15,05%	309,09%
Solução 75	19,68%	4,24%	236,36%	23,91%	17,04%	118,18%	13,35%	5,37%	272,73%
Solução 76	46,08%	9,22%	163,64%	3,87%	0,81%	263,64%	22,24%	0,27%	290,91%
Solução 77	198,56%	108,07%	18,18%	60,79%	51,10%	90,91%	6,63%	9,17%	181,82%
Solução 78	88,59%	31,77%	109,09%	117,68%	103,74%	36,36%	15,04%	2,89%	227,27%
Solução 79	23,78%	3,24%	236,36%	117,68%	103,74%	36,36%	5,52%	9,41%	209,09%
Solução 80	199,93%	108,35%	9,09%	59,90%	50,83%	100,00%	12,05%	4,56%	300,00%
Solução 81	90,01%	32,97%	81,82%	16,29%	9,96%	136,36%	14,06%	2,40%	309,09%
Solução 82	199,93%	108,35%	9,09%	3,59%	2,83%	272,73%	51,54%	42,21%	109,09%
Solução 83	7,65%	0,97%	309,09%	110,06%	96,65%	54,55%	20,73%	0,14%	318,18%
Solução 84	71,09%	26,71%	127,27%	23,91%	17,04%	118,18%	0,76%	16,07%	272,73%
Solução 85	198,56%	108,07%	18,18%	23,91%	17,04%	118,18%	25,16%	3,90%	209,09%
Solução 86	90,01%	32,97%	81,82%	117,68%	103,74%	36,36%	1,27%	14,93%	281,82%
Solução 87	28,53%	3,24%	200,00%	17,18%	10,23%	127,27%	15,39%	15,15%	136,36%
Solução 88	26,15%	3,24%	218,18%	13,18%	3,46%	172,73%	15,01%	5,91%	154,55%
Solução 89	88,64%	32,69%	90,91%	23,91%	17,04%	118,18%	19,83%	2,13%	254,55%
Solução 90	200,67%	107,43%	9,09%	110,06%	96,65%	54,55%	54,83%	54,33%	81,82%
Solução 91	25,39%	4,03%	227,27%	117,68%	103,74%	36,36%	6,82%	6,00%	254,55%
Solução 92	71,83%	25,79%	127,27%	17,18%	10,23%	127,27%	19,98%	4,74%	172,73%
Média	100,66%	46,80%	122,63%	60,98%	51,58%	107,41%	21,63%	16,76%	216,50%
Desvio Padrão	71,37%	43,31%	94,46%	50,20%	46,21%	73,40%	30,67%	33,10%	71,57%

Figura 51 – Exemplo de um passeio gerado no cenário 1 do grupo 2.



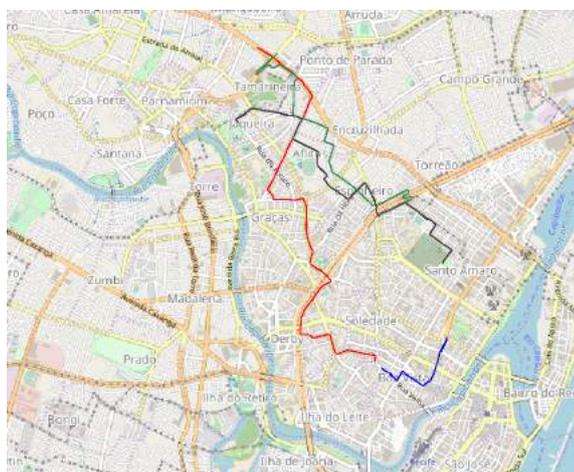
JMapView modificado pelo Autor

Figura 52 – Exemplo de um passeio gerado no cenário 2 do grupo 2.



JMapView modificado pelo Autor

Figura 53 – Exemplo de um passeio gerado no cenário 3 do grupo 2.



JMapView modificado pelo Autor

APÊNDICE C – Soluções das frentes de Pareto para o Grupo 3 e exemplos gerados dos seus cenários de congestionamento

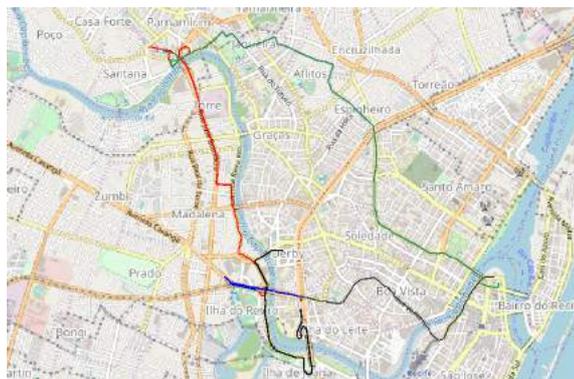
Quadro 19 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 3 (parte 1).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	32,89	1,60%	173,68%	48,90	5,24%	163,16%	69,06	22,90%	205,26%
Dijkstra Distância	5,86%	19,76	136,84%	7,84%	19,86	142,11%	27,68%	19,86	142,11%
Dijkstra Sinais	229,04%	139,47%	19	141,87%	116,36%	19	165,48%	149,02%	19
Solução 1	37,35%	24,12%	100,00%	2,05%	0,13%	173,68%	13,20%	15,68%	136,84%
Solução 2	8,66%	2,29%	121,05%	46,98%	40,68%	63,16%	24,74%	3,54%	121,05%
Solução 3	101,45%	54,48%	57,89%	1,43%	6,25%	157,89%	134,86%	132,65%	0,00%
Solução 4	77,33%	39,01%	78,95%	49,07%	35,76%	68,42%	25,06%	-0,26%	136,84%
Solução 5	20,49%	9,46%	110,53%	30,78%	19,13%	94,74%	25,84%	11,80%	100,00%
Solução 6	1,36%	1,86%	163,16%	1,29%	6,30%	157,89%	13,67%	1,76%	163,16%
Solução 7	5,00%	0,61%	142,11%	2,05%	0,13%	173,68%	12,08%	2,23%	189,47%
Solução 8	0,33%	1,07%	168,42%	5,15%	4,58%	126,32%	25,06%	2,04%	136,84%
Solução 9	117,93%	69,71%	47,37%	110,16%	89,30%	10,53%	35,32%	17,26%	89,47%
Solução 10	38,48%	25,80%	94,74%	7,24%	-0,33%	131,58%	11,65%	4,20%	168,42%
Solução 11	188,11%	107,53%	0,00%	1,29%	6,30%	157,89%	4,72%	9,30%	147,37%
Solução 12	7,29%	0,23%	131,58%	30,70%	19,27%	94,74%	19,29%	6,19%	105,26%
Solução 13	5,00%	0,61%	142,11%	1,29%	6,30%	157,89%	26,84%	-0,48%	136,84%
Solução 14	1,36%	1,86%	163,16%	3,38%	1,39%	163,16%	36,37%	14,51%	94,74%
Solução 15	6,65%	0,32%	131,58%	30,78%	19,13%	94,74%	15,72%	8,75%	115,79%
Solução 16	0,33%	1,07%	168,42%	6,44%	5,64%	121,05%	8,10%	6,80%	142,11%
Solução 17	89,24%	47,89%	68,42%	28,94%	20,95%	100,00%	13,55%	4,15%	157,89%
Solução 18	37,35%	24,12%	100,00%	1,29%	6,30%	157,89%	13,42%	0,32%	173,68%
Solução 19	9,79%	3,98%	115,79%	6,98%	-0,34%	136,84%	0,00%	22,90%	205,26%
Solução 20	130,14%	76,30%	36,84%	6,44%	5,64%	121,05%	16,07%	6,70%	136,84%
Solução 21	119,06%	71,40%	42,11%	28,94%	20,95%	100,00%	26,10%	15,01%	94,74%
Solução 22	38,48%	25,80%	94,74%	6,44%	5,64%	121,05%	45,87%	41,17%	52,63%
Solução 23	20,86%	8,88%	110,53%	1,29%	6,30%	157,89%	24,02%	30,36%	100,00%
Solução 24	50,69%	32,39%	84,21%	6,51%	2,40%	136,84%	17,41%	4,79%	121,05%
Solução 25	50,69%	32,39%	84,21%	7,24%	-0,33%	131,58%	27,77%	22,05%	73,68%
Solução 26	164,48%	89,06%	21,05%	94,34%	74,43%	31,58%	15,17%	22,75%	115,79%
Solução 27	101,45%	54,48%	57,89%	30,78%	19,13%	94,74%	5,00%	7,23%	173,68%
Solução 28	6,65%	0,32%	131,58%	1,29%	6,30%	157,89%	19,29%	6,19%	105,26%
Solução 29	8,66%	2,29%	121,05%	3,38%	1,39%	163,16%	13,48%	0,41%	168,42%
Solução 30	1,36%	1,86%	163,16%	49,07%	35,76%	68,42%	5,65%	5,74%	163,16%
Solução 31	22,00%	10,56%	105,26%	1,43%	6,25%	157,89%	24,74%	3,54%	121,05%
Solução 32	119,06%	71,40%	42,11%	5,15%	4,58%	126,32%	5,60%	9,28%	147,37%
Solução 33	38,48%	25,80%	94,74%	6,44%	5,64%	121,05%	12,08%	2,23%	189,47%
Solução 34	37,35%	24,12%	100,00%	1,29%	6,30%	157,89%	15,17%	22,75%	115,79%
Solução 35	0,33%	1,07%	168,42%	30,70%	19,27%	94,74%	13,55%	4,15%	157,89%
Solução 36	6,13%	0,40%	131,58%	0,00%	5,24%	163,16%	15,62%	7,32%	121,05%
Solução 37	38,48%	25,80%	94,74%	28,94%	20,95%	100,00%	45,87%	41,17%	52,63%
Solução 38	7,15%	1,19%	126,32%	6,44%	5,64%	121,05%	26,84%	-0,48%	136,84%
Solução 39	89,24%	47,89%	68,42%	1,29%	6,30%	157,89%	0,00%	22,90%	205,26%
Solução 40	78,84%	40,11%	73,68%	6,51%	2,40%	136,84%	25,13%	15,00%	100,00%
Solução 41	9,79%	3,98%	115,79%	24,55%	22,13%	100,00%	17,29%	4,87%	121,05%
Solução 42	119,06%	71,40%	42,11%	3,38%	1,39%	163,16%	13,20%	15,68%	136,84%
Solução 43	1,36%	1,86%	163,16%	1,29%	6,30%	157,89%	14,46%	1,75%	163,16%
Solução 44	0,86%	0,99%	168,42%	25,09%	16,15%	115,79%	25,08%	1,10%	131,58%
Solução 45	22,00%	10,56%	105,26%	3,38%	1,39%	163,16%	8,10%	6,80%	142,11%

Quadro 20 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 3 (parte 2).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	32,89	1,60%	173,68%	48,90	5,24%	163,16%	69,06	22,90%	205,26%
Dijkstra Distância	5,86%	19,76	136,84%	7,84%	19,86	142,11%	27,68%	19,86	142,11%
Dijkstra Sinais	229,04%	139,47%	19	141,87%	116,36%	19	165,48%	149,02%	19
Solução 46	5,00%	0,61%	142,11%	30,95%	19,27%	89,47%	25,07%	29,04%	100,00%
Solução 47	131,27%	77,98%	31,58%	28,87%	24,19%	84,21%	11,65%	4,20%	168,42%
Solução 48	0,33%	1,07%	168,42%	1,29%	6,30%	157,89%	129,89%	122,48%	10,53%
Solução 49	5,00%	0,61%	142,11%	28,94%	20,95%	100,00%	36,25%	28,43%	63,16%
Solução 50	77,33%	39,01%	78,95%	7,09%	-0,46%	136,84%	25,06%	-0,26%	136,84%
Solução 51	1,36%	1,86%	163,16%	1,29%	6,30%	157,89%	36,56%	14,52%	89,47%
Solução 52	7,29%	0,23%	131,58%	5,15%	4,58%	126,32%	25,84%	11,80%	100,00%
Solução 53	90,37%	49,57%	63,16%	1,29%	6,30%	157,89%	17,62%	8,63%	105,26%
Solução 54	38,48%	25,80%	94,74%	6,44%	5,64%	121,05%	13,42%	0,32%	173,68%
Solução 55	20,49%	9,46%	110,53%	2,05%	0,13%	173,68%	18,32%	6,18%	110,53%
Solução 56	188,11%	107,53%	0,00%	26,64%	17,21%	105,26%	53,65%	49,21%	42,11%
Solução 57	8,66%	2,29%	121,05%	30,78%	19,13%	94,74%	25,84%	11,80%	100,00%
Solução 58	119,06%	71,40%	42,11%	2,05%	0,13%	173,68%	109,37%	98,19%	31,58%
Solução 59	9,79%	3,98%	115,79%	1,29%	6,30%	157,89%	16,48%	8,64%	110,53%
Solução 60	7,15%	1,19%	126,32%	6,44%	5,64%	121,05%	16,07%	6,70%	136,84%
Solução 61	131,27%	77,98%	31,58%	28,87%	24,19%	84,21%	36,56%	14,52%	89,47%
Solução 62	38,48%	25,80%	94,74%	30,78%	19,13%	94,74%	12,79%	4,20%	163,16%
Solução 63	22,00%	10,56%	105,26%	1,29%	6,30%	157,89%	5,83%	5,59%	157,89%
Solução 64	119,06%	71,40%	42,11%	3,38%	1,39%	163,16%	4,72%	9,30%	147,37%
Solução 65	101,45%	54,48%	57,89%	30,70%	19,27%	94,74%	12,08%	2,23%	189,47%
Solução 66	78,84%	40,11%	73,68%	0,85%	2,49%	184,21%	11,65%	4,20%	168,42%
Solução 67	5,34%	0,08%	136,84%	6,44%	5,64%	121,05%	26,84%	-0,48%	136,84%
Solução 68	2,49%	3,54%	157,89%	1,29%	6,30%	157,89%	23,76%	3,54%	126,32%
Solução 69	0,33%	1,07%	168,42%	1,29%	6,30%	157,89%	26,10%	15,01%	94,74%
Solução 70	7,15%	1,19%	126,32%	2,05%	0,13%	173,68%	0,49%	22,14%	215,79%
Solução 71	78,84%	40,11%	73,68%	28,87%	24,19%	84,21%	25,06%	-0,26%	136,84%
Solução 72	9,79%	3,98%	115,79%	25,09%	16,15%	115,79%	13,29%	2,71%	173,68%
Solução 73	22,00%	10,56%	105,26%	2,05%	0,13%	173,68%	37,05%	27,40%	63,16%
Solução 74	0,00%	1,60%	173,68%	1,29%	6,30%	157,89%	24,74%	3,54%	121,05%
Solução 75	0,33%	1,07%	168,42%	5,15%	4,58%	126,32%	8,10%	6,80%	142,11%
Solução 76	119,06%	71,40%	42,11%	30,78%	19,13%	94,74%	13,20%	15,68%	136,84%
Solução 77	130,14%	76,30%	36,84%	28,87%	24,19%	84,21%	24,49%	29,09%	100,00%
Solução 78	38,48%	25,80%	94,74%	1,29%	6,30%	157,89%	16,01%	22,71%	110,53%
Solução 79	6,13%	0,40%	131,58%	30,70%	19,27%	94,74%	17,41%	4,79%	121,05%
Solução 80	89,24%	47,89%	68,42%	0,00%	5,24%	163,16%	17,29%	4,87%	121,05%
Solução 81	8,28%	2,87%	121,05%	1,29%	6,30%	157,89%	12,79%	4,20%	163,16%
Solução 82	188,11%	107,53%	0,00%	30,78%	19,13%	94,74%	5,65%	5,74%	163,16%
Solução 83	38,48%	25,80%	94,74%	3,20%	1,24%	168,42%	5,53%	7,95%	152,63%
Solução 84	0,33%	1,07%	168,42%	46,98%	40,68%	63,16%	24,74%	3,54%	121,05%
Solução 85	22,00%	10,56%	105,26%	1,29%	6,30%	157,89%	29,44%	19,61%	73,68%
Solução 86	89,24%	47,89%	68,42%	49,07%	35,76%	68,42%	25,84%	11,80%	100,00%
Solução 87	5,00%	0,61%	142,11%	3,12%	1,38%	168,42%	23,64%	2,27%	131,58%
Solução 88	1,36%	1,86%	163,16%	6,44%	5,64%	121,05%	19,29%	6,19%	105,26%
Solução 89	119,06%	71,40%	42,11%	30,70%	19,27%	94,74%	25,36%	-0,33%	131,58%
Solução 90	5,00%	0,61%	142,11%	1,29%	6,30%	157,89%	53,65%	49,21%	42,11%
Solução 91	2,49%	3,54%	157,89%	0,00%	5,24%	163,16%	1,92%	21,50%	215,79%
Solução 92	119,06%	71,40%	42,11%	23,26%	21,07%	105,26%	36,37%	14,51%	94,74%
Media	47,23%	26,65%	102,57%	15,08%	12,18%	128,09%	23,05%	14,43%	125,63%
Desvio Padrao	52,27%	30,45%	45,89%	19,43%	14,38%	36,12%	21,76%	21,99%	42,40%

Figura 54 – Exemplo de um passeio gerado no cenário 1 do grupo 3.



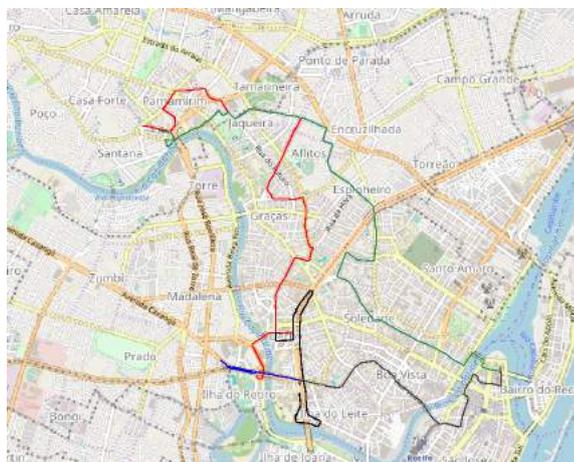
JMapView modificado pelo Autor

Figura 55 – Exemplo de um passeio gerado no cenário 2 do grupo 3.



JMapView modificado pelo Autor

Figura 56 – Exemplo de um passeio gerado no cenário 3 do grupo 3.



JMapView modificado pelo Autor

APÊNDICE D – Soluções das frentes de Pareto para o Grupo 4 e exemplos gerados dos seus cenários de congestionamento

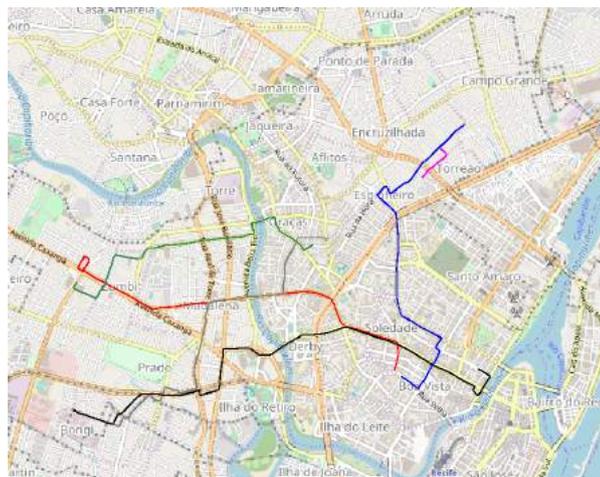
Quadro 21 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 4 (parte 1).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	45,38	8,18%	273,91%	65,90	1,96%	195,65%	87,48	17,32%	143,48%
Dijkstra Distância	5,60%	24,22	213,04%	4,94%	24,22	213,04%	27,60%	24,22	213,04%
Dijkstra Sinais	327,80%	217,02%	23	246,52%	232,91%	23	206,28%	216,75%	23
Solução 1	3,25%	0,27%	213,04%	0,31%	1,58%	195,65%	21,48%	22,65%	69,57%
Solução 2	10,99%	0,10%	191,30%	54,40%	52,83%	39,13%	44,60%	56,32%	21,74%
Solução 3	16,95%	5,06%	165,22%	177,28%	165,90%	13,04%	8,08%	5,20%	152,17%
Solução 4	0,94%	1,34%	213,04%	1,22%	1,08%	195,65%	0,00%	17,32%	143,48%
Solução 5	4,38%	0,78%	195,65%	24,73%	24,08%	91,30%	20,17%	2,35%	173,91%
Solução 6	12,93%	4,99%	178,26%	28,51%	23,45%	126,09%	2,31%	7,76%	134,78%
Solução 7	8,84%	0,68%	182,61%	176,39%	165,21%	21,74%	21,19%	1,31%	204,35%
Solução 8	33,07%	17,08%	156,52%	24,47%	24,42%	91,30%	11,37%	17,20%	78,26%
Solução 9	5,60%	0,01%	208,70%	4,12%	0,17%	213,04%	10,66%	4,12%	139,13%
Solução 10	8,47%	0,07%	200,00%	0,31%	1,58%	195,65%	2,34%	6,67%	100,00%
Solução 11	0,04%	4,88%	252,17%	23,35%	23,79%	113,04%	5,76%	5,29%	104,35%
Solução 12	31,95%	19,78%	139,13%	54,40%	52,83%	39,13%	17,16%	25,34%	65,22%
Solução 13	19,94%	4,31%	169,57%	1,38%	0,99%	182,61%	28,20%	34,53%	52,17%
Solução 14	76,57%	43,53%	65,22%	37,06%	36,53%	69,57%	25,79%	0,09%	195,65%
Solução 15	2,95%	1,22%	208,70%	5,56%	0,57%	191,30%	10,93%	2,73%	152,17%
Solução 16	0,80%	1,83%	217,39%	23,35%	23,79%	113,04%	8,84%	18,57%	78,26%
Solução 17	46,67%	23,39%	91,30%	12,90%	11,69%	130,43%	0,33%	8,01%	91,30%
Solução 18	0,00%	8,18%	273,91%	37,06%	36,53%	69,57%	4,28%	6,80%	95,65%
Solução 19	12,95%	4,99%	178,26%	3,39%	0,73%	208,70%	0,00%	16,25%	108,70%
Solução 20	131,90%	81,24%	56,52%	4,41%	0,74%	204,35%	2,73%	8,46%	82,61%
Solução 21	10,63%	0,20%	186,96%	94,54%	90,51%	26,09%	19,68%	31,04%	56,52%
Solução 22	36,75%	19,13%	117,39%	24,82%	24,08%	86,96%	11,18%	17,21%	73,91%
Solução 23	0,36%	4,47%	243,48%	5,74%	0,49%	182,61%	19,65%	1,54%	195,65%
Solução 24	30,11%	20,16%	147,83%	34,50%	34,30%	73,91%	25,37%	35,55%	52,17%
Solução 25	1,11%	1,42%	208,70%	3,70%	0,89%	204,35%	16,69%	25,52%	65,22%
Solução 26	18,02%	4,42%	169,57%	23,42%	23,75%	117,39%	3,37%	6,27%	104,35%
Solução 27	19,94%	4,31%	169,57%	1,64%	1,01%	178,26%	0,00%	17,32%	143,48%
Solução 28	4,81%	0,62%	195,65%	12,90%	11,69%	130,43%	22,97%	0,56%	186,96%
Solução 29	3,61%	0,17%	213,04%	22,93%	24,52%	104,35%	140,26%	150,15%	13,04%
Solução 30	10,23%	0,20%	191,30%	12,39%	11,44%	143,48%	2,43%	7,40%	86,96%
Solução 31	1,47%	1,32%	213,04%	24,73%	24,08%	91,30%	8,90%	18,39%	78,26%
Solução 32	4,38%	0,78%	195,65%	23,82%	24,01%	100,00%	9,46%	3,67%	156,52%
Solução 33	8,76%	0,74%	186,96%	12,90%	11,69%	130,43%	2,31%	7,76%	134,78%
Solução 34	32,63%	17,24%	156,52%	1,77%	1,24%	173,91%	0,22%	8,38%	91,30%
Solução 35	0,00%	8,18%	273,91%	23,42%	23,75%	117,39%	9,06%	4,51%	130,43%
Solução 36	1,08%	1,33%	213,04%	22,41%	24,27%	117,39%	27,51%	34,73%	47,83%
Solução 37	71,45%	37,75%	69,57%	0,55%	2,10%	178,26%	20,92%	2,30%	173,91%
Solução 38	6,21%	0,14%	204,35%	22,93%	24,52%	104,35%	4,84%	5,33%	100,00%
Solução 39	4,38%	0,78%	195,65%	12,61%	11,42%	143,48%	0,00%	16,25%	108,70%
Solução 40	2,54%	1,30%	204,35%	94,43%	90,52%	26,09%	4,28%	6,80%	95,65%
Solução 41	16,95%	5,06%	165,22%	5,90%	0,19%	200,00%	11,24%	2,64%	139,13%
Solução 42	10,23%	0,20%	191,30%	5,04%	0,00%	208,70%	21,48%	22,65%	69,57%
Solução 43	13,32%	5,00%	173,91%	22,88%	24,58%	104,35%	25,78%	35,53%	52,17%
Solução 44	7,44%	0,80%	186,96%	40,55%	40,24%	60,87%	2,73%	8,46%	82,61%
Solução 45	19,94%	4,31%	169,57%	12,39%	11,44%	143,48%	44,58%	56,34%	21,74%

Quadro 22 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 4 (parte 2).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	45,38	8,18%	273,91%	65,90	1,96%	195,65%	87,48	17,32%	143,48%
Dijkstra Distância	5,60%	24,22	213,04%	4,94%	24,22	213,04%	27,60%	24,22	213,04%
Dijkstra Sinais	327,80%	217,02%	23	246,52%	232,91%	23	206,28%	216,75%	23
Solução 46	0,28%	4,48%	252,17%	0,36%	1,87%	186,96%	9,31%	4,35%	134,78%
Solução 47	3,25%	0,27%	213,04%	23,56%	23,78%	108,70%	8,52%	3,73%	152,17%
Solução 48	0,00%	8,18%	273,91%	2,36%	1,18%	173,91%	11,24%	2,64%	139,13%
Solução 49	30,11%	20,16%	147,83%	24,46%	23,86%	95,65%	0,08%	11,69%	100,00%
Solução 50	33,07%	17,08%	156,52%	5,56%	0,57%	191,30%	0,22%	8,38%	91,30%
Solução 51	36,75%	19,13%	117,39%	22,41%	24,27%	117,39%	20,98%	2,14%	173,91%
Solução 52	10,63%	0,20%	186,96%	5,74%	0,49%	182,61%	19,65%	1,54%	195,65%
Solução 53	9,44%	0,11%	195,65%	94,43%	90,52%	26,09%	3,37%	6,27%	104,35%
Solução 54	3,01%	0,74%	204,35%	2,50%	1,06%	169,57%	27,51%	34,73%	47,83%
Solução 55	0,40%	4,45%	252,17%	54,04%	52,61%	47,83%	2,33%	7,74%	134,78%
Solução 56	37,68%	19,07%	121,74%	94,54%	90,51%	26,09%	4,84%	5,33%	100,00%
Solução 57	0,00%	8,18%	273,91%	5,52%	0,54%	195,65%	8,84%	18,57%	78,26%
Solução 58	10,63%	0,20%	186,96%	0,97%	1,37%	186,96%	23,27%	0,34%	182,61%
Solução 59	30,11%	20,16%	147,83%	4,07%	0,61%	213,04%	0,00%	16,25%	108,70%
Solução 60	5,07%	0,15%	204,35%	2,83%	1,35%	165,22%	21,48%	22,65%	69,57%
Solução 61	1,11%	1,42%	208,70%	4,44%	0,77%	200,00%	11,13%	17,44%	73,91%
Solução 62	19,59%	4,42%	165,22%	0,55%	2,10%	178,26%	4,28%	6,80%	95,65%
Solução 63	17,35%	5,06%	160,87%	0,53%	1,54%	182,61%	19,56%	24,36%	65,22%
Solução 64	13,32%	5,00%	173,91%	0,02%	1,98%	195,65%	4,84%	5,33%	100,00%
Solução 65	3,64%	0,27%	208,70%	1,29%	1,00%	186,96%	25,98%	0,09%	200,00%
Solução 66	33,07%	17,08%	156,52%	5,19%	0,23%	204,35%	2,54%	7,03%	86,96%
Solução 67	263,80%	174,59%	43,48%	3,22%	0,83%	217,39%	21,07%	22,67%	69,57%
Solução 68	3,25%	0,27%	213,04%	12,50%	11,43%	143,48%	44,58%	56,34%	21,74%
Solução 69	0,58%	3,99%	252,17%	5,52%	0,54%	195,65%	11,04%	17,59%	73,91%
Solução 70	10,63%	0,20%	186,96%	0,66%	2,09%	178,26%	19,65%	1,54%	195,65%
Solução 71	32,71%	19,07%	130,43%	4,52%	0,42%	200,00%	0,00%	16,25%	108,70%
Solução 72	0,00%	8,18%	273,91%	5,74%	0,49%	182,61%	8,84%	18,57%	78,26%
Solução 73	19,59%	4,42%	165,22%	5,74%	0,49%	182,61%	2,31%	7,76%	134,78%
Solução 74	17,35%	5,06%	160,87%	2,87%	1,28%	160,87%	25,81%	35,51%	52,17%
Solução 75	7,40%	0,71%	191,30%	2,87%	1,28%	160,87%	28,20%	34,53%	52,17%
Solução 76	131,90%	81,24%	56,52%	2,83%	1,35%	165,22%	10,93%	2,73%	152,17%
Solução 77	5,07%	0,15%	204,35%	2,36%	1,18%	173,91%	0,33%	8,01%	91,30%
Solução 78	12,93%	4,99%	178,26%	0,49%	1,50%	186,96%	8,47%	3,87%	152,17%
Solução 79	3,12%	0,72%	217,39%	12,44%	12,04%	134,78%	20,92%	2,30%	173,91%
Solução 80	1,11%	1,42%	208,70%	0,86%	1,72%	178,26%	2,45%	6,31%	100,00%
Solução 81	3,61%	0,17%	213,04%	0,61%	1,49%	186,96%	0,10%	11,29%	95,65%
Solução 82	10,63%	0,20%	186,96%	4,48%	0,39%	204,35%	16,75%	25,36%	65,22%
Solução 83	12,93%	4,99%	178,26%	1,38%	0,99%	182,61%	9,06%	4,51%	130,43%
Solução 84	5,60%	0,01%	208,70%	12,90%	11,69%	130,43%	2,40%	6,49%	100,00%
Solução 85	0,71%	4,37%	247,83%	1,39%	0,98%	186,96%	0,33%	8,01%	91,30%
Solução 86	30,11%	20,16%	147,83%	0,61%	1,49%	186,96%	8,84%	18,57%	78,26%
Solução 87	17,35%	5,06%	160,87%	5,22%	0,37%	200,00%	0,00%	16,25%	108,70%
Solução 88	1,11%	1,42%	208,70%	4,48%	0,39%	204,35%	2,31%	7,76%	134,78%
Solução 89	7,40%	0,71%	191,30%	6,23%	0,38%	195,65%	0,10%	11,29%	95,65%
Solução 90	3,47%	0,76%	200,00%	5,04%	0,00%	208,70%	27,58%	34,57%	47,83%
Solução 91	19,59%	4,42%	165,22%	1,39%	0,98%	186,96%	8,82%	4,43%	143,48%
Solução 92	0,00%	8,18%	273,91%	4,64%	0,10%	221,74%	10,93%	2,73%	152,17%
Média	18,33%	9,51%	185,59%	18,74%	16,99%	148,30%	13,29%	15,06%	106,95%
Desvio Padrão	34,22%	22,08%	47,82%	31,66%	30,57%	56,26%	17,10%	19,03%	45,55%

Figura 57 – Exemplo de passeio gerado no cenário 1 do grupo 4.



JMapView modificado pelo Autor

Figura 58 – Exemplo de passeio gerado no cenário 2 do grupo 4.

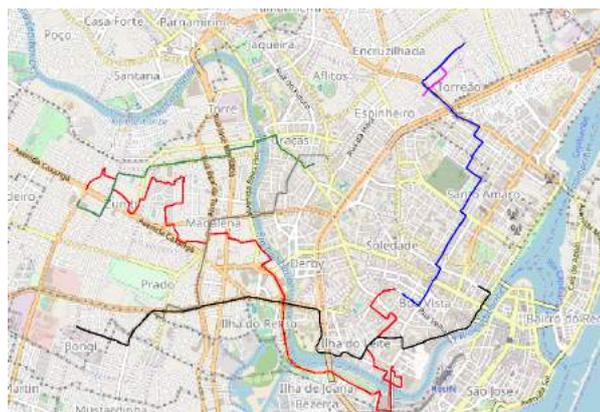
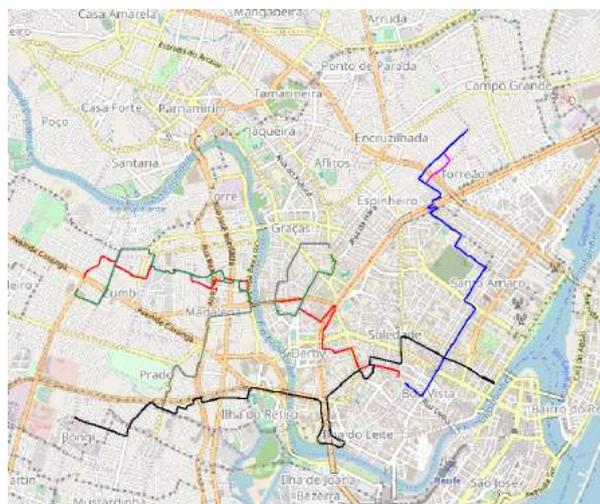


Figura 59 – Exemplo de passeio gerado no cenário 3 do grupo 4.



JMapView modificado pelo Autor

APÊNDICE E – Soluções das frentes de Pareto para o Grupo 5 e exemplos gerados dos seus cenários de congestionamento

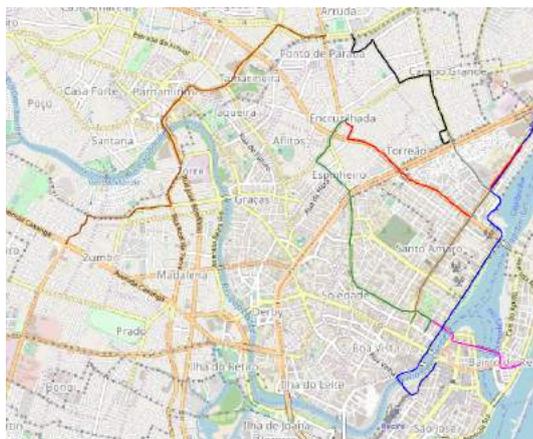
Quadro 23 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 5 (parte 1).

	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	43,69	4,88%	244,83%	67,30	7,15%	175,86%	91,99	17,60%	150,00%
Dijkstra Distância	13,37%	25,87	162,07%	6,01%	25,87	162,07%	25,47%	25,87	153,33%
Dijkstra Sinais	289,70%	171,82%	29	184,32%	163,94%	29	202,77%	206,68%	30
Solução 1	268,32%	158,96%	0,00%	11,34%	4,65%	93,10%	152,77%	151,66%	6,67%
Solução 2	0,89%	4,62%	217,24%	2,09%	7,45%	124,14%	24,33%	10,93%	80,00%
Solução 3	120,88%	60,39%	34,48%	11,17%	3,22%	106,90%	9,08%	6,79%	146,67%
Solução 4	198,99%	115,15%	20,69%	8,56%	1,55%	131,03%	24,74%	0,04%	153,33%
Solução 5	28,78%	4,95%	117,24%	1,59%	0,67%	175,86%	12,32%	19,18%	93,33%
Solução 6	120,88%	60,39%	34,48%	25,19%	19,76%	75,86%	10,97%	13,96%	100,00%
Solução 7	84,74%	37,20%	48,28%	5,95%	7,50%	106,90%	21,19%	3,81%	113,33%
Solução 8	198,67%	114,88%	24,14%	1,52%	8,38%	127,59%	7,21%	8,16%	120,00%
Solução 9	6,87%	3,25%	172,41%	10,71%	5,97%	86,21%	21,18%	1,81%	126,67%
Solução 10	231,84%	135,17%	13,79%	5,47%	5,08%	127,59%	3,37%	12,23%	143,33%
Solução 11	120,88%	60,39%	34,48%	11,03%	3,36%	103,45%	25,76%	9,29%	93,33%
Solução 12	25,86%	5,08%	124,14%	6,54%	1,78%	141,38%	14,50%	7,52%	96,67%
Solução 13	84,74%	37,20%	48,28%	0,56%	6,80%	168,97%	15,09%	6,03%	116,67%
Solução 14	235,14%	138,34%	6,90%	6,08%	4,96%	113,79%	26,77%	20,44%	60,00%
Solução 15	42,95%	14,62%	100,00%	10,15%	4,38%	103,45%	23,66%	25,95%	60,00%
Solução 16	152,06%	79,91%	31,03%	3,00%	5,67%	127,59%	13,23%	11,85%	103,33%
Solução 17	84,74%	37,20%	48,28%	8,21%	1,86%	124,14%	13,41%	23,47%	80,00%
Solução 18	9,65%	0,93%	165,52%	9,87%	2,90%	120,69%	67,22%	59,85%	33,33%
Solução 19	117,59%	57,23%	41,38%	7,28%	4,11%	113,79%	2,04%	16,66%	113,33%
Solução 20	0,78%	4,84%	220,69%	2,09%	7,45%	124,14%	16,51%	2,25%	163,33%
Solução 21	234,58%	138,14%	10,34%	35,56%	30,20%	62,07%	10,87%	3,81%	150,00%
Solução 22	231,53%	134,91%	17,24%	5,64%	6,44%	117,24%	46,49%	36,52%	43,33%
Solução 23	43,69%	14,02%	89,66%	10,98%	4,45%	96,55%	25,26%	21,32%	73,33%
Solução 24	84,74%	37,20%	48,28%	1,31%	8,11%	131,03%	28,20%	18,79%	73,33%
Solução 25	22,94%	4,20%	134,48%	2,28%	2,59%	162,07%	33,76%	34,61%	46,67%
Solução 26	84,74%	37,20%	48,28%	130,73%	117,20%	37,93%	18,36%	15,39%	90,00%
Solução 27	198,99%	115,15%	20,69%	1,36%	5,63%	141,38%	19,25%	5,14%	106,67%
Solução 28	198,67%	114,88%	24,14%	1,41%	4,54%	148,28%	0,01%	17,66%	150,00%
Solução 29	43,20%	13,78%	96,55%	5,74%	7,11%	110,34%	14,12%	4,14%	133,33%
Solução 30	84,74%	37,20%	48,28%	8,21%	1,86%	124,14%	5,56%	10,22%	110,00%
Solução 31	117,59%	57,23%	41,38%	7,48%	3,93%	113,79%	19,35%	4,99%	106,67%
Solução 32	235,14%	138,34%	6,90%	6,69%	1,63%	151,72%	12,22%	4,06%	136,67%
Solução 33	46,07%	14,75%	82,76%	7,74%	3,11%	117,24%	26,69%	20,51%	66,67%
Solução 34	84,74%	37,20%	48,28%	7,52%	1,26%	137,93%	17,71%	5,27%	110,00%
Solução 35	29,88%	5,03%	110,34%	0,95%	6,65%	151,72%	7,06%	7,90%	123,33%
Solução 36	152,06%	79,91%	31,03%	10,75%	4,88%	93,10%	16,66%	2,51%	160,00%
Solução 37	30,20%	5,31%	106,90%	5,87%	6,10%	113,79%	26,77%	20,44%	60,00%
Solução 38	198,99%	115,15%	20,69%	11,39%	3,56%	100,00%	25,39%	9,30%	93,33%
Solução 39	84,74%	37,20%	48,28%	2,14%	4,92%	137,93%	13,41%	23,47%	80,00%
Solução 40	120,88%	60,39%	34,48%	9,87%	2,90%	120,69%	19,86%	3,65%	126,67%
Solução 41	198,99%	115,15%	20,69%	3,55%	2,95%	137,93%	23,92%	10,47%	86,67%
Solução 42	6,40%	3,02%	179,31%	5,75%	0,17%	158,62%	23,24%	26,88%	63,33%
Solução 43	84,74%	37,20%	48,28%	5,25%	5,44%	131,03%	22,57%	0,75%	133,33%
Solução 44	64,05%	25,26%	68,97%	6,67%	2,10%	131,03%	153,56%	148,16%	3,33%
Solução 45	234,57%	138,45%	10,34%	26,02%	18,71%	79,31%	2,12%	16,00%	113,33%
Solução 46	119,21%	59,88%	37,93%	8,21%	1,86%	124,14%	5,16%	9,88%	120,00%

Quadro 24 – Comparação entre as soluções computadas pelo algoritmo de Dijkstra com as Frentes de Pareto geradas em cada cenário do grupo 5 (parte 2).

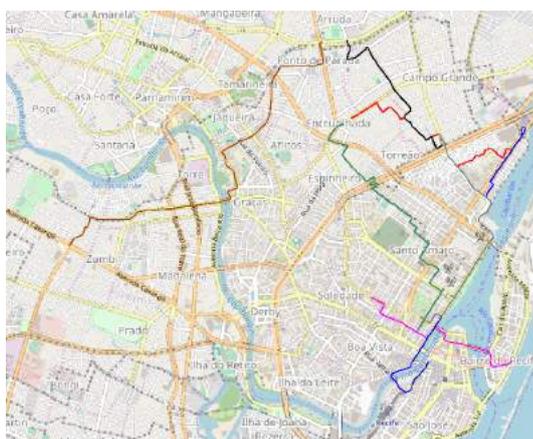
	cenário 1			cenário 2			cenário 3		
	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)	Σ T. de Viagem (min)	Σ Distância (Km)	Σ N° de Sinais (sum)
Dijkstra T.Viagem	43,69	4,88%	244,83%	67,30	7,15%	175,86%	91,99	17,60%	150,00%
Dijkstra Distância	13,37%	25,87	162,07%	6,01%	25,87	162,07%	25,47%	25,87	153,33%
Dijkstra Sinais	289,70%	171,82%	29	184,32%	163,94%	29	202,77%	206,68%	30
Solução 47	198,99%	115,15%	20,69%	2,09%	7,45%	124,14%	36,75%	28,86%	50,00%
Solução 48	30,20%	5,31%	106,90%	1,52%	8,38%	127,59%	47,79%	46,18%	40,00%
Solução 49	29,97%	4,34%	120,69%	7,13%	3,73%	117,24%	22,98%	2,12%	120,00%
Solução 50	120,88%	60,39%	34,48%	3,00%	5,67%	127,59%	18,51%	15,65%	86,67%
Solução 51	152,06%	79,91%	31,03%	0,56%	6,80%	168,97%	9,39%	6,37%	140,00%
Solução 52	84,74%	37,20%	48,28%	5,99%	6,01%	110,34%	1,73%	17,43%	126,67%
Solução 53	70,74%	26,09%	58,62%	24,79%	20,65%	72,41%	12,59%	22,23%	90,00%
Solução 54	235,14%	138,34%	6,90%	6,60%	1,84%	137,93%	13,29%	7,25%	110,00%
Solução 55	154,07%	81,02%	27,59%	6,62%	3,19%	124,14%	12,44%	12,80%	106,67%
Solução 56	30,20%	5,31%	106,90%	11,03%	3,36%	103,45%	15,47%	12,96%	93,33%
Solução 57	84,74%	37,20%	48,28%	131,06%	117,50%	31,03%	6,52%	10,12%	116,67%
Solução 58	120,88%	60,39%	34,48%	1,58%	4,50%	141,38%	10,81%	4,80%	150,00%
Solução 59	198,67%	114,88%	24,14%	1,42%	1,84%	172,41%	26,77%	20,44%	60,00%
Solução 60	30,20%	5,31%	106,90%	10,14%	5,50%	93,10%	23,24%	26,88%	63,33%
Solução 61	84,74%	37,20%	48,28%	1,31%	8,11%	131,03%	21,66%	0,96%	143,33%
Solução 62	198,99%	115,15%	20,69%	11,03%	3,36%	103,45%	14,12%	4,14%	133,33%
Solução 63	57,42%	17,34%	72,41%	11,03%	3,36%	103,45%	19,25%	5,14%	106,67%
Solução 64	235,14%	138,34%	6,90%	3,00%	5,67%	127,59%	19,53%	3,95%	106,67%
Solução 65	108,61%	50,22%	44,83%	1,31%	8,11%	131,03%	15,09%	6,03%	116,67%
Solução 66	9,65%	0,93%	165,52%	35,20%	30,00%	65,52%	128,97%	115,25%	30,00%
Solução 67	43,54%	14,34%	93,10%	6,54%	1,78%	141,38%	3,54%	11,54%	140,00%
Solução 68	198,99%	115,15%	20,69%	7,13%	3,73%	117,24%	64,75%	69,17%	33,33%
Solução 69	84,74%	37,20%	48,28%	7,28%	4,11%	113,79%	21,44%	2,01%	123,33%
Solução 70	235,14%	138,34%	6,90%	10,98%	4,45%	96,55%	14,90%	7,17%	106,67%
Solução 71	20,59%	6,47%	141,38%	1,31%	8,11%	131,03%	2,03%	16,65%	120,00%
Solução 72	84,74%	37,20%	48,28%	1,49%	1,80%	168,97%	9,40%	4,98%	143,33%
Solução 73	235,14%	138,34%	6,90%	1,20%	3,06%	158,62%	9,39%	6,37%	140,00%
Solução 74	8,83%	0,91%	172,41%	9,92%	6,63%	93,10%	12,37%	18,47%	93,33%
Solução 75	84,74%	37,20%	48,28%	1,66%	5,89%	134,48%	24,33%	10,93%	80,00%
Solução 76	235,14%	138,34%	6,90%	7,74%	3,11%	117,24%	16,24%	3,44%	163,33%
Solução 77	33,61%	4,82%	113,79%	1,53%	6,99%	131,03%	1,73%	17,43%	126,67%
Solução 78	120,88%	60,39%	34,48%	26,02%	18,71%	79,31%	16,52%	16,56%	83,33%
Solução 79	198,44%	114,95%	24,14%	10,15%	4,38%	103,45%	46,49%	36,52%	43,33%
Solução 80	29,88%	5,03%	110,34%	1,42%	1,84%	172,41%	5,56%	10,22%	110,00%
Solução 81	119,21%	59,88%	37,93%	3,00%	5,67%	127,59%	12,51%	23,65%	90,00%
Solução 82	117,59%	57,23%	41,38%	1,53%	6,99%	131,03%	15,47%	12,96%	93,33%
Solução 83	152,06%	79,91%	31,03%	10,76%	3,45%	110,34%	25,52%	20,32%	73,33%
Solução 84	69,56%	25,90%	62,07%	35,20%	30,00%	65,52%	10,97%	13,96%	100,00%
Solução 85	198,99%	115,15%	20,69%	9,97%	5,54%	100,00%	25,39%	9,30%	93,33%
Solução 86	5,22%	2,42%	189,66%	10,14%	5,50%	93,10%	5,10%	10,23%	146,67%
Solução 87	119,21%	59,88%	37,93%	2,09%	7,45%	124,14%	11,07%	7,01%	126,67%
Solução 88	235,14%	138,34%	6,90%	9,92%	6,63%	93,10%	9,67%	6,98%	133,33%
Solução 89	1,65%	4,99%	213,79%	11,34%	4,65%	93,10%	33,33%	33,85%	53,33%
Solução 90	83,06%	36,69%	51,72%	1,31%	8,11%	131,03%	13,26%	23,20%	83,33%
Solução 91	235,14%	138,34%	6,90%	3,00%	5,67%	127,59%	20,27%	2,71%	123,33%
Solução 92	9,65%	0,93%	165,52%	6,91%	4,86%	117,24%	5,41%	9,08%	120,00%
Média	112,70%	58,75%	63,19%	10,36%	8,59%	118,63%	22,14%	18,13%	101,05%
Desvio Padrão	77,98%	48,91%	54,96%	19,41%	17,18%	27,67%	25,64%	25,25%	35,59%

Figura 60 – Exemplo de passeio gerado no cenário 1 do grupo 5.



JMapView modificado pelo Autor

Figura 61 – Exemplo de passeio gerado no cenário 2 do grupo 5.



JMapView modificado pelo Autor

Figura 62 – Exemplo de passeio gerado no cenário 3 do grupo 5.



JMapView modificado pelo Autor