

DISTRITAMENTO APLICADO AO PROBLEMA DE FATURAMENTO EM REDES DE SERVIÇO

Augusto Magalhães Pinto de Mendonça

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Orientadora: Laura Silva de Assis
Co-orientador: Luís Domingues Tomé Jardim
Tarrataca

Rio de Janeiro,
Dezembro de 2020

Distritamento Aplicado ao Problema de Faturamento em Redes de Serviço

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação, do Centro Federal de Educação Tecnológica Celso Suckow da Fonseca, CEFET/RJ, como parte dos requisitos necessários à obtenção do título de mestre.

Augusto Magalhães Pinto de Mendonça

Banca Examinadora:

Laura Silva de Assis

Presidente, Professora. D.Sc. Laura Silva de Assis (CEFET/RJ)

Luís Domingues Tomé Jardim Tarrataca

Professor Ph.D. Luís Domingues Tomé Jardim Tarrataca (CEFET/RJ)

Diego Nunes Brandão

Professor D.Sc. Diego Nunes Brandão (CEFET/RJ)

Fábio Luiz Usberti

Professor D.Sc. Fábio Luiz Usberti (IC - UNICAMP)

Rio de Janeiro,
Dezembro de 2020

Ficha catalográfica elaborada pela Biblioteca Central do CEFET/RJ

M539 Mendonça, Augusto Magalhães Pinto de
Distritamento aplicado ao problema de faturamento em redes de
serviço / Augusto Magalhães Pinto de Mendonça – 2020.
67f : il. color. , enc.

Dissertação (Mestrado) Centro Federal de Educação
Tecnológica Celso Suckow da Fonseca, 2020.

Bibliografia : f. 61-67

Orientadora: Laura Silva de Assis

Co-orientador: Luís Domingues Tomé Jardim Tarrataca

1. Otimização combinatória. 2. Redes de serviço. 3. Problema de
Distritamento Capacitado. 4. Energia elétrica - Distribuição.
I. Assis, Laura Silva de (Orient.). II. Tarrataca, Luís Domingues
Tomé Jardim (Co-orient.). III. Título.

CDD 519.4

Elaborada pelo bibliotecário Leandro Mota de Menezes – CRB/7 nº 5281

DEDICATÓRIA

Este trabalho é dedicado a Deus, pelas pessoas
iluminadas que tive pelo caminho e a minha filha
Elise, pelo amor sem fim.

AGRADECIMENTOS

Agradeço a Professora Laura Silva de Assis pela dedicação incansável, por ser o exemplo de servidor público que aspiro alcançar e por ter andado comigo cada degrau deste longo processo. Agradeço ao Professor Luís Domingues Tomé Jardim Tarrataca, pela amizade e por orientar o melhor caminho, pelo companheirismo e toda paciência.

Agradeço aos demais professores do CEFET/RJ por estarem sempre de portas abertas. Em especial, aos Professores Jurair Rosa, Eduardo Ogasawara, Eduardo Bezerra, Kele Belloze e Diego Brandão que, foram muito além da missão de ensino.

Por fim, agradeço a minha esposa Dominique Mendonça e a minha mãe Rosângela por toda paciência e amor nesses anos.

RESUMO

Distritamento Aplicado ao Problema de Faturamento em Redes de Serviço

Esta dissertação tem o objetivo de investigar o Problema de Distritamento Capacitado (PDC). O PDC é um problema de otimização combinatória que consiste em particionar uma determinada região em uma quantidade de distritos predeterminados, considerando um ou mais critérios de decisão. A definição dos distritos deve respeitar suas capacidades, as quais são definidas conforme as especificidades do problema. Existem diversas aplicações para o PDC, como o distritamento político, cobertura de vendas, entrega de correspondências, coleta de lixo e serviços de emergência, dentre tantas outras. Esta pesquisa tem como foco resolver o PDC aplicado ao problema de definir lotes de trabalhos para leitores de medidores de redes de serviço tais como energia, água, gás, considerando os critérios de compacidade e homogeneidade para definição de tais territórios. Um novo método de solução baseado em Algoritmo Genético (AG) comparando duas estruturas distintas é apresentado, respeitando restrições de contiguidade, um número predefinido de distritos, dentre outras. Um método de otimização de hiperparâmetros é proposto para determinar um conjunto de valores que forneça soluções de qualidade com certa confiabilidade. Para validar a abordagem de solução proposta foram realizados experimentos computacionais utilizando instâncias de grande porte com características distintas. Os resultados alcançados mostram a eficiência da abordagem proposta para o PDC em estudo.

Palavras-chave: Otimização Combinatória; Problema de Distritamento; Algoritmo Genético.

ABSTRACT

Districting Applied to the Billing of Service Networks

This dissertation has the objective of investigating the Capacitated Districting Problem (CDP). The CDP is a combinatorial optimization problem that consists in partitioning a determined region into districts considering one or more decision criteria. The districts design must respect their capacities, which are defined according to problem requirements. There are several different applications for CDP, such as political districting, sales design, mail delivery, garbage collecting and medical emergency services, among others. This research focuses on solving a CDP applied to the problem of defining work territories for energy meter readers, considering both the compactness and the homogeneity of the zones. A new solution method based on a Genetic Algorithm (GA) with two distinct structures is presented which considers, among others, contiguity restrictions alongside a predefined number of districts. A hyperparameter optimization method is proposed to determine a set of values that provide quality solutions. In order to validate the proposed solution approach, computational experiments were performed using large instances with different characteristics and the results obtained show the efficiency of the proposed approach.

Key-words: Combinatorial Optimization; Districting Problems; Genetic Algorithm.

Lista de Ilustrações

3.1	Mapa original sinalizado ©2020 Microsoft.	18
3.2	Representação do mapa através de grafo.	18
3.3	Transformação do grafo (de Assis et al., 2014).	19
4.1	Fluxo de execução do AG.	22
4.2	Representação do cromossomo de um indivíduo para o PDC.	22
4.3	Representação mais realística do cromossomo de um indivíduo para o PDC em estudo.	22
4.4	Representação de uma população não estruturada.	24
4.5	Representação da população estruturada.	24
4.6	Seleção por roleta.	27
4.7	Operador de <i>crossover</i> de um ponto.	28
4.8	Exemplo de geração de filhos inactiváveis após o cruzamento.	28
4.9	Calculando a similaridade entre dois indivíduos pela distância de Hamming.	29
4.10	Procedimento de atualização dos rótulos dos distritos.	31
4.11	Exemplo de solução com os nós de fronteira destacados.	33
5.1	Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto C	38
5.2	Valor médio de FO das combinações de hiperparâmetros C - AGPNE.	40
5.3	Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto C_1 - <i>reset</i>	41
5.4	Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto E	43
5.5	Valor médio de FO das combinação de hiperparâmetros E - AGPE.	44
5.6	Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto E_1 - extensão do g_{max}	45
5.7	Exemplos da topologia de instâncias com diferentes esparsidades.	47
5.8	Instância RedeSP de 1659 nós.	48
5.9	Instâncias de 512 nós totalmente conectadas.	53
5.10	Instâncias de 512 nós parcialmente esparsas.	53
5.11	Instâncias de 512 nós minimamente conectadas.	54
5.12	Instâncias de 1024 nós totalmente conectadas.	54

5.13	Instâncias de 1024 nós parcialmente esparsas.	55
5.14	Instâncias de 1024 nós minimamente conectadas.	55
5.15	Resultado para a RedeSP.	55
5.16	Resultados para as instâncias com diferentes esparsidades.	57
5.17	Resultado da instância RedeSP.	58

Lista de Tabelas

2.1	Critérios de inclusão de artigos no mapeamento sistemático.	3
2.2	Critérios de exclusão de artigos no mapeamento sistemático	3
2.3	Lista de artigos com informações gerais resumidas.	4
5.1	Conjunto de hiperparâmetros e espaço de busca para o AGPNE.	37
5.2	Cinco melhores combinações de hiperparâmetros e resultados para o AGPNE.	37
5.3	Cinco melhores combinações de hiperparâmetros e resultados - <i>reset</i> - AGPNE.	40
5.4	Conjunto de hiperparâmetros e espaço de busca para o AGPE.	42
5.5	Cinco melhores combinações de hiperparâmetros e resultados para o AGPE.	42
5.6	Cinco melhores combinações de hiperparâmetros e seus resultados.	44
5.7	Tamanho das instâncias.	46
5.8	Conjunto de Hiperparâmetros C_{11} e E_{11}	48
5.9	Evolução do AGPNE.	49
5.10	Evolução do AGPE.	50
5.11	Resultados: AGPE $\times$ AGPNE.	51
5.12	Amplitude máxima de evolução dos Algoritmos.	52

Lista de Abreviações

AG	Algoritmo Genético
AGPE	Algoritmo Genético Com População Estruturada
AGPNE	Algoritmo Genético Com População Não Estruturada
DI	Distância Interquartil
FO	Função Objetivo
PD	Problema De Distritamento
PDC	Problema De Distritamento Capacitado

Sumário

1	Introdução	1
2	Trabalhos Relacionados	3
3	Apresentação e Formulação do Problema	16
3.1	Problema de Otimização das Leituras de Consumo em Redes de Serviço	16
3.2	Representação do PDC em Estudo	17
3.3	Modelo Matemático	19
4	Metodologia de Solução para o PDC	21
4.1	Indivíduo	21
4.2	Cálculo do <i>Fitness</i>	22
4.3	População	23
4.3.1	População não Estruturada	23
4.3.2	População Estruturada	24
4.4	Geração da População Inicial	25
4.5	Seleção	26
4.5.1	Operador Híbrido	26
4.5.2	Operador Determinístico	27
4.6	Crossover	27
4.6.1	Distância de <i>Hamming</i>	29
4.6.2	Similaridade Geométrica	30
4.7	Mutação	33
4.8	Algoritmo Genético	34
4.8.1	Reset	35
5	Resultados	36
5.1	Otimização de Hiperparâmetros	36
5.1.1	Algoritmo Genético com População não Estruturada - AGPNE	37
5.1.2	Algoritmo Genético com População Estruturada - AGPE	42

5.2	Experimentos Computacionais	44
5.2.1	Descrição das Instâncias	45
5.2.2	Estudos de Caso	46
5.2.3	Estudo de Caso 4: Análise da Evolução do AG	52
5.2.4	Visualização de Soluções	54
6	Considerações Finais	59
	Referências Bibliográficas	61

Capítulo 1

Introdução

Problema de Distritamento (PD), também conhecido como problema de agrupamento de unidades territoriais, é um problema de otimização combinatória que tem o objetivo de agrupar unidades territoriais menores em áreas geográficas (distritos) maiores, contíguos e sem sobreposição (de Assis et al., 2014). Critérios adicionais compõem cada problema de distritamento específico, que apesar de frequentemente compartilharem objetivos e restrições em comum, também possuem seus próprios requerimentos particulares que os tornam únicos (Ríos-Mercado, 2020). Existem diversas aplicações que podem ser classificadas como um PD, por exemplo, *design* de territórios (Ríos-Mercado and López, 2013), distritamento eleitoral (Levin and Friedler, 2019), definição de zonas de trabalho (Ahmad et al., 2010), coberturas de serviços de emergência (Steiner et al., 2014), logística de comércio e serviços (Ruvalcaba et al., 2011), entre muitas outras. O distritamento eleitoral é a aplicação mais frequentemente encontrada na literatura, na qual uma determinada área é dividida em um número de zonas eleitorais que precisam ser contíguas, compactas e cada uma deve possuir um número equivalente de eleitores (Pereira et al., 2007).

O Problema de Distritamento Capacitado (PDC) é uma generalização do problema de distritamento, o qual pode ser definido por um conjunto de objetos, onde cada um possui pesos ou demandas, sendo que tais objetos devem ser agrupados em p grupos distintos. Cada agrupamento possui uma capacidade conhecida predeterminada, de modo que a soma total dos pesos dos objetos designados a um determinado agrupamento não deve exceder a capacidade do mesmo (de Assis et al., 2014).

O PDC pertence à família de problemas de otimização combinatória que podem ser difíceis de resolver de forma ótima, para instâncias de grande porte, utilizando técnicas de programação matemática, devido o problema de distritamento ser classificado como NP-Difícil. Esta classificação foi realizada por uma redução do Problema de Particionamento (Garey and Johnson, 1979) realizada em Hojati (1996). Porém, apesar de se tratar de uma classe de problemas desafiadores, têm sido propostas abordagens de soluções bem sucedidas, baseadas principalmente em métodos heurísticos.

O foco do método proposto é a solução do PDC considerando a minimização de dois objetivos: (i) compacidade, e (ii) homogeneidade. A medida de compacidade busca obter distritos geográfica-

mente compactos e a homogeneidade procura obter uma configuração de distritos com uma carga de trabalho o mais semelhante tanto quanto possível.

Esta dissertação tem como objetivo propor um método de solução baseado em Algoritmo Genético (AG) para problemas de definição de distritos no contexto de faturamento em redes de serviço que seja bem ajustado e eficiente. A escolha pelo AG se deve por se tratar de uma técnica de computação evolutiva que oferece vantagens para os pesquisadores, tanto pela sua simplicidade quanto pela capacidade de obtenção de respostas robustas em ambientes dinâmicos, demonstrando sua alta flexibilidade (Grosan et al., 2007). Para resolver o problema em estudo foram realizados diversas melhorias nos operadores genéticos. Foi também desenvolvido e comparado duas estruturas populacionais distintas. Uma das estruturas da população é baseada na abordagem tradicional, onde a população não possui nenhum tipo de organização. A segunda abordagem utiliza uma população hierarquicamente estruturada e é, até o momento de confecção desta dissertação, o primeiro trabalho a considerar esta alternativa para este PDC. Populações estruturadas foram aplicadas anteriormente em metodologias de computação evolutiva como uma opção de conseguir explorar o espaço de busca com esforço computacional aceitável, na busca por soluções ótimas ou subótimas (Moscato, 1989; Moscato et al., 1992; Toledo et al., 2014; de Assis, 2014).

Buscando garantia de desempenho da metodologia proposta e uma maior qualidade nos experimentos, foi realizada uma longa etapa de otimização dos hiperparâmetros, com mais de 16000 testes, com o intuito de definir um conjunto de valores de qualidade. Com o objetivo de comparar as duas abordagens de solução foram realizados experimentos computacionais para verificar o impacto de cada tipo de população no AG para o problema em estudo. Foram também analisadas a qualidade das soluções obtidas e as diferenças no tempo computacional. Após a definição dos valores para os hiperparâmetros, foram executados experimentos computacionais em 24 instâncias geradas aleatoriamente (com características e tamanhos diferentes) e mais uma instância com características reais de uma região da cidade de São Paulo. Tais instâncias foram apresentadas e utilizadas anteriormente no trabalho apresentado em de Assis et al. (2014).

Esta dissertação está organizada nos seguintes capítulos. No Capítulo 2 são apresentados os trabalhos relacionados. A descrição do problema e modelo matemático para o PDC aplicado ao problema de leituristas de medidores de redes de serviço são apresentados no Capítulo 3. O Capítulo 4 apresenta a metodologia de solução proposta para o PDC em estudo. O Capítulo 5 aborda o processo de otimização de hiperparâmetros, os experimentos realizados aplicando o AG com ambas as estruturas, assim como a apresentação e as análises inerentes aos resultados obtidos. Conclusões sobre o trabalho realizado, suas principais contribuições e trabalhos futuros são discutidos no Capítulo 6.

Capítulo 2

Trabalhos Relacionados

Existem diversos trabalhos na literatura que abordam aplicações diferentes do PD abordado, propondo métodos de solução distintos. Buscando por trabalhos que têm sido desenvolvidos nessa área foi realizado um mapeamento sistemático e uma busca com *snowballing* por artigos sobre distritamento e redistritamento. Para o mapeamento sistemático, foi estabelecida uma *string* de busca a qual foi utilizada no repositório do Scopus, de forma a identificar referências relacionadas e atualizadas. O propósito da *string* foi retornar um conjunto robusto de artigos que tratem de distritamento e que sejam pertinentes à área de Computação e afins. Após alguns testes a *string* definida foi: TITLE-ABS-KEY ((“districting” OR “redistricting”)) AND (LIMIT-TO (SUBJAREA , “COMP”) OR LIMIT-TO (SUBJAREA , “MATH”) OR LIMIT-TO (SUBJAREA , “ENGI”) OR LIMIT-TO (SUBJAREA , “DECI”)) AND (LIMIT-TO (DOCTYPE , “ar”)). Essa busca retornou 203 resultados que foram analisados com base nos seguintes critérios de inclusão e exclusão apresentados nas Tabelas 2.1 e 2.2, respectivamente. A análise teve foco inicial nos resumos e a pesquisa foi finalizada em maio de 2020.

Tabela 2.1: Critérios de inclusão de artigos no mapeamento sistemático.

Código	Critério
CI1	Artigos que apresentem diferentes aplicações ou variações do PDC.
CI2	Trabalhos que utilizem heurísticas ou métodos exatos para problemas análogos ao PDC tratado.

Tabela 2.2: Critérios de exclusão de artigos no mapeamento sistemático

Código	Critério
CE1	Artigos não relacionados com a Ciência da Computação.
CE2	Trabalhos específicos para redes de computadores.
CE3	Problemas de distritamento sem restrições ou capacidade.
CE4	Não foi possível obter o trabalho completo.
CE5	O conteúdo do trabalho de fato é distinto do resumo.
CE6	O trabalho não apresentou resultados condizentes com a proposta.

Dos artigos selecionados pelas duas metodologias, 50 foram escolhidos para uma leitura completa

e 41 foram selecionados para fazer parte da referência bibliográfica. A Tabela 2.3 relaciona estes artigos, citando algumas informações de forma resumida como os estudos de caso, os métodos de solução e o domínio de cada aplicação.

Tabela 2.3: Lista de artigos com informações gerais resumidas.

#	Referência	Estudo de caso	Método	Domínio
1	Shanker et al. (1975)	Estados Unidos	Programação linear	Distritamento comercial
2	Mulvey and Beck (1984)	-	Heurística primal e subgradiente	Distritamento comercial
3	Bozkaya et al. (2003)	Edmonton, Canadá	Busca tabu e memória adaptativa	Distritamento eleitoral
4	Yang et al. (2004)	Pennsylvania, Estados Unidos	Programação não linear	Cobertura de bombeiros
5	Bacao et al. (2005)	Portugal	Algoritmo Genético	Distritamento eleitoral
6	Marianov and Fresard (2005)	Chile	Programação linear	Distritamento prisional
7	Pereira et al. (2007)	Paris, França	Algoritmo evolucionário com busca local	Distritamento de transporte público
8	Ricca et al. (2007)	Itália	Computação geométrica	Distritamento eleitoral
9	Ricca and Simeone (2008)	Itália	Gradiente Descendente, Busca Tabu, <i>Simulated Annealing</i> e <i>Old Bachelor Acceptance</i>	Distritamento eleitoral
10	Ríos-Mercado and Fernández (2009)	Monterrey, México	GRASP (<i>Greedy Randomized Adaptive Search Procedure</i>)	Distritamento comercial
11	Joshi et al. (2012)	Estados Unidos	Constrained Polygonal Spatial Clustering	Distritamento escolar e eleitoral
12	Ricca et al. (2013)	Vários	Comparativo de métodos	Distritamento eleitoral
13	Ríos-Mercado and López (2013)	México	Evolutivo com “ <i>Branch and Bound</i> ”	Distritamento comercial
14	Rincón-García et al. (2013)	México	Simulated annealing	Distritamento eleitoral
15	de Assis et al. (2014)	São Paulo, Brasil	GRASP	Classificação de lotes para leituristas
16	G. Elizondo-Amaya et al. (2014)	-	Dual bounding scheme	Distritamento comercial para entregas
17	Hu et al. (2014)	Beijing, China	NSGA-II	Distritamento de abrigos para emergências
18	Mohammadi et al. (2014)	-	Híbrido AG com Simulated Annealing	Distritamento de serviços de emergência
19	Steiner et al. (2014)	Paraná, Brasil	NSGA-II	Distritamento de saúde
20	Camacho-Collados et al. (2015)	Madri, Espanha	Busca local com requerimentos temporais	Distritamento policial
21	Castelli et al. (2015)	Estados Unidos	Método evolucionário com informação semântica	Distritamento eleitoral
22	Fréminville et al. (2015)	-	<i>Column generation heuristic</i>	Distritamento para precificação
23	Gutiérrez-Gutiérrez and Vidal (2015)	Colômbia	Ordenamento lexiográfico com controle de deterioração e priorização	Distritamento de cobertura domiciliar de saúde
24	García-Ayala et al. (2016)	-	CPLEX	Distritamento para roteamento de arcos

#	Referência	Estudo de caso	Método	Domínio
25	Liberatore and Camacho-Collados (2016)	Madri, Espanha	Diferentes buscas locais	Distritamento policial
26	Ríos-Mercado and Escalante (2016)	-	GRASP com <i>path relinking</i>	Distritamento de territórios comerciais
27	González-Ramírez et al. (2017)	Monterrey, México	GRASP com busca tabu	Distritamento comercial para entregas
28	Lin et al. (2017)	Hong Kong, China	Programação inteira mista	Distritamento de serviços
29	Vanneschi et al. (2017)	Estados Unidos	NSGA-II e SPEA-II	Distritamento eleitoral
30	Costa et al. (2018)	Maranhão, Brasil	RCMeans	Classificação de lotes para leituristas
31	Kazazi and Bashiri (2018)	-	Programação inteira mista	Distritamento de serviços de emergência
32	Lin et al. (2018)	China	NSGA-II	Distritamento de serviços para idosos
33	Regis-Hernández et al. (2018)	Montréal, Canadá	ALDP+FB	Distritamento de serviços médicos de emergência
34	Ríos-Mercado and Bard (2018)	Alemanha	Algoritmo exato	Distritamento para descarte de eletrônicos
35	Chen et al. (2019)	Londres, Reino Unido	Particionamento de grafos e Busca Tabu	Distritamento policial
36	Farughi et al. (2019)	South Khorasan, Irã	Otimizador Grey Wolf	Distritamento de cobertura de saúde
37	Levin and Friedler (2019)	Estados Unidos	Divisão e Conquista	Distritamento eleitoral
38	Lara-Caballero et al. (2019)	México	Simulated Annealing	Distritamento político
39	Yanik et al. (2019)	Istambul, Turquia	Programação inteira mista	Distritamento de cobertura de saúde
40	Hirose et al. (2020)	Santa Catarina, Brasil	CPLEX	Distritamento de sedes eleitorais
41	Wang et al. (2020)	China	Programação linear binária com um algoritmo de geração de colunas	Distritamento ferroviário

Conforme o retorno da pesquisa realizada, diversos trabalhos encontrados propõem soluções para problemas de distritamento variados. Um dos primeiros trabalhos que estudou PD, propôs uma abordagem de solução para resolver o problema de definição de territórios de vendas, na qual busca-se determinar a divisão do território de venda em sub-regiões, as quais devem ser percorridas por vendedores (Shanker et al., 1975). A abordagem de solução proposta considerou a frequência de vendas para determinar as fronteiras que demarcam a divisão do trabalho. A proposta foi resolvida por um algoritmo de programação linear, no qual cada um dos três territórios da solução é coberto por um vendedor específico, evitando a sobreposição, minimizando o tempo de espera dos clientes e consequentemente aumentando os lucros.

Em Mulvey and Beck (1984) são apresentados dois algoritmos para agrupar n entidades em p grupos mutualmente exclusivos e de tamanho restrito, também aplicado ao problema de *design*

de territórios de vendas. O primeiro algoritmo proposto inicia-se com o sorteio de p posições centrais, que definem o início da construção de cada distrito, e posteriormente continua atribuindo os demais locais de venda a partir destes. A cada iteração uma busca local é executada a fim de melhorar a solução atual. O segundo algoritmo mescla a primeira abordagem com um algoritmo de subgradiente para determinar as posições centrais iniciais. Os autores afirmam que através de testes computacionais foi possível mensurar que os resultados alcançados estavam 1% abaixo da solução ótima, contudo a aplicação é restrita a problemas inferiores a 500 clientes. Um dos problemas de agrupamento clássicos encontrado na literatura é o problema de distritamento eleitoral, que visa balancear o número de eleitores por distrito, formando áreas tão compactas quanto possível. O maior desafio no distritamento eleitoral é evitar o *gerrymandering*, que é o favorecimento de determinado candidato ou partido político pela manipulação das fronteiras, fenômeno reconhecido no século XIX e com impactos até hoje (Apollonio et al., 2009). O artigo de Ricca et al. (2013) versa sobre as principais iniciativas desenvolvidas para tratar o problema de distritamento nos últimos 50 anos, com ênfase no distritamento eleitoral, reiterando que a má formação de territórios pode gerar vantagens para partidos políticos específicos. Os autores classificam as iniciativas como: (i) crescimento de múltiplos núcleos (Vickrey, 1961), (ii) localização (Hess et al., 1965), (iii) abordagens exatas (Garfinkel and Nemhauser, 1970), (iv) buscas locais (Bourjolly et al., 1981) e (v) computação geométrica (Kalcsics et al., 2005).

Em Vickrey (1961) é apresentada uma abordagem pelo crescimento de múltiplos núcleos, onde é proposto um procedimento no qual os distritos são construídos um por vez. O autor considera que o território¹ seja dividido respeitando os dados censitários, a contiguidade e compacidade dos territórios, homogeneidade populacional, e ainda fronteiras formadas por decisões administrativas. No início deste procedimento uma unidade de referência h é fixada, e a cada iteração, um novo distrito é formado iniciado por um centro que é a unidade ainda não atribuída mais distante de h . Uma vez que existem $k - 1$ territórios sendo k um número fixo, todas as unidades não atribuídas são alocadas em um último distrito centralizado em h . Em Bodin (1973), o autor também apresentou uma proposta de solução com base no crescimento de múltiplos núcleos para o problema de distritamento relativo a divisão de trabalho para o censo do estado do Arkansas nos Estados Unidos. Esta obra foi a primeira a demonstrar como o problema pode ser representado como um grafo contíguo para representação dos territórios.

O segundo conjunto de iniciativas foi denominada como localização em Ricca et al. (2013). O primeiro resultado formal com a metodologia foi apresentada em Hess et al. (1965), que trazia um modelo e uma proposta de solução para o problema de distritamento. Na localização, se considera um número total n de unidades territoriais e um número k de distritos em um problema formalizado

¹Território ou distrito são utilizados indistintamente neste trabalho como referência para uma unidade geográfica dos problemas.

como problema de alocação discreta. A base da proposta é identificar um número k de unidades que irão representar os centros dos distritos, e cada unidade territorial será alocada a um dos centros disponíveis. Para tal, são executados quatro passos principais: as unidades que serão centrais são sorteadas; as unidades mais próximas aos centros são atribuídas aos distritos; a solução é ajustada para que cada unidade territorial esteja apenas em um distrito; os centroides dos distritos são computados e se tornam os respectivos centros. Estes quatro passos são repetidos até que uma solução satisfatória seja encontrada. A convergência foi definida quando os centros não podem ser atualizados por duas iterações consecutivas.

Abordagens exatas também foram empregadas para o problema de distritamento eleitoral. Em Garfinkel and Nemhauser (1970) é proposta uma formulação para o problema de otimização linear com variáveis binárias de decisão. Como é um algoritmo exato, tal iniciativa foi aplicável apenas em instâncias menores que 50 unidades populacionais, considerando também a limitação de *hardware* da época.

Uma alternativa de solução para o problema de distritamento eleitoral é por meio de técnicas de busca local (Bourjolly et al., 1981). A abordagem apresentada pelos autores daquele trabalho parte de uma solução inicial e através de trocas de posição sucessivas das unidades territoriais, busca-se encontrar uma solução ótima. Diferentes metaheurísticas classificadas como busca local foram comparadas em Ricca and Simeone (2008), considerando o problema de distritamento eleitoral na Itália, a saber: (i) Algoritmo descendente básico, (ii) Busca tabu, (iii) *Simulated Annealing*, e (iv) *Old Bachelor Acceptance*. Segundo os autores, na comparação realizada, o algoritmo *Old Bachelor Acceptance* foi o mais flexível e obteve o melhor resultado para a maioria das regiões da Itália tratadas no trabalho.

Devido a importância das distâncias e das medidas geométricas para o problema de distritamento, surgiram iniciativas classificadas como computação geométrica, como o apresentado em Ricca et al. (2007), que propõe uma solução baseada na geração de regiões ponderadas de Voronoy, que os autores afirmam ser uma forma de se obter regiões inerentemente compactas. Para tal é utilizada a notação de grafos, sendo os nós uma unidade ponderada cujo peso é sua população e as arestas representam a conexão destas unidades para assegurar a contiguidade. A base do algoritmo é a criação de territórios baseados no diagrama de Voronoy, no qual todos os pontos pertencentes a um mesmo território estão mais próximos de seu centro do que de qualquer outro centro. Os resultados obtidos são superiores em relação à compactidade e homogeneidade comparados ao distritamento eleitoral utilizado nas eleições da Itália entre 1994 e 2001.

Considerando as mudanças climáticas e o consequente aumento de desastres naturais em Hu et al. (2014) é apresentado o problema de distritamento relacionado à área de cobertura de abrigos de emergência. O distritamento de abrigos envolve a designação de áreas para o serviço de

emergência, onde cada área deverá receber um abrigo. O trabalho propôs um modelo matemático capaz de decidir a localização dos abrigos e o desenho dos distritos. O modelo inclui múltiplos critérios como segurança, distância total para evacuação, redução de custos e a capacidade dos abrigos, respeitando a restrição de contiguidade dos distritos formados. O estudo de caso tratado foi a distribuição dos abrigos de terremoto no distrito de Chaoyang na China. Por meio de uma estratégia que combina um algoritmo genético com um operador de *crossover* híbrido, foi possível acelerar a convergência para soluções consideradas ótimas.

O distritamento para serviços de emergência também foi tratado em Kazazi and Bashiri (2018), onde os autores propõem a expansão do alcance da proteção oferecida pelos serviços de emergência dividindo o território de cobertura em distritos e alocando dois depósitos em cada distrito, um para material e um de pessoal. Para resolver esse problema é apresentado um modelo de programação linear inteira mista, que considera o valor dos recursos que devem ser protegidos e a janela de tempo para alcançar esse objetivo. O modelo foi executado usando o solver GAMS (*General Algebraic Modeling Systems*). Os resultados alcançados mostram que dividir a área total de cobertura em distritos e depois alocar equipes específicas permitiu uma cobertura de um número maior de propriedades, com uma instância com 64 propriedades. Os autores mostraram também a possibilidade de realizar a operação em uma janela menor de tempo.

Em Wang et al. (2020), os autores propõem um modelo de solução para o distritamento ferroviário. Nesta obra, um modelo de programação linear binário é apresentado juntamente com um algoritmo híbrido baseado na geração de colunas. O algoritmo obteve uma performance superior as soluções comerciais existentes, em baixo tempo computacional, sendo ainda superior que as soluções empíricas existentes, previamente desenvolvidas pelos gestores ferroviários.

O problema de distritamento aplicado à alocação ótima de unidades de bombeiros para Pensilvânia – EUA, foi abordado em Yang et al. (2004). O modelo de programação não linear multiobjetivo apresentado considera (i) a redução do tempo médio de resposta aos incidentes, e (ii) e o equilíbrio das carga de trabalho entre as unidades de atendimento. Foi utilizado o *software* de simulação ARENA sobre soluções geradas pelos autores utilizando a ferramentas TransCAD, que segundo os testes realizados, reduziram o tempo médio de atendimento das ocorrências em 120 segundos quando comparado com a distribuição vigente.

Outro problema de distritamento que tem impacto na qualidade de vida da população é o problema da cobertura da atividade policial. Em Liberatore and Camacho-Collados (2016) é proposta uma solução para o problema da divisão ótima das patrulhas em distritos, considerando (i) o isolamento da área, (ii) o número de crimes registrados, e (iii) a distribuição de trabalho entre os agentes. O trabalho utiliza uma heurística construtiva gulosa e realiza a comparação de três buscas locais distintas: (i) *Simple Hill Climbing*, (ii) *Steepest Descent Hill Climbing*, e (iii) *Tabu*

Search. Os autores afirmam ter obtido os melhores resultados com a *Tabu Search*, que na média são superiores aos utilizados na prática, considerando os dados do distrito de Madrid – Espanha.

O problema da alocação ótima de presídios no Chile foi tratado em Marianov and Fresard (2005). Segundo o artigo, um sistema prisional saudável deve prever a necessidade da construção de novas unidades partindo de um planejamento racional e de custo fixo, prevenindo a superpopulação. Tais unidades devem ser dispostas minimizando o trajeto percorrido por familiares de detentos, incentivando assim a visitação e auxiliando na reabilitação. A solução proposta busca estabelecer a necessidade de até 20 anos do redimensionamento dos presídios existentes e também do distritamento das províncias buscando a quantidade e a localização ideal de novas unidades correcionais. Para este fim, foi empregado o software de otimização CPLEX em um modelo de programação linear inteira mista em MPL (*Mathematical Programming Language*). O modelo é aplicado em dados reais e visa equilibrar os custos de abertura e expansão das unidades, penalizando a superpopulação e o custos de transporte. A solução estimou a necessidade da abertura de 10 presídios novos, além da maximização da capacidade dos existentes.

Os problemas de distritamento normalmente apresentam múltiplos objetivos conflitantes que precisam ser considerados na busca pela solução ótima. Em Pereira et al. (2007) foi apresentado um método para precificação do transporte público em Paris – França, cujo preço deve ser igual dentro de um mesmo território, mas que deve aumentar em função da distância para o centro. Foi proposto um algoritmo evolucionário com busca local. Neste algoritmo uma fila armazena as soluções potencialmente eficientes geradas pelos operadores genéticos que são posteriormente selecionadas por uma busca local que classifica soluções próximas.

Em Fréminville et al. (2015) também é estudado um problema referente a precificação de produtos para distritos de uma região. O problema considera que a legislação não permite a determinação de preço específico para um cliente isolado, porém permite que um valor pré-determinado seja cobrado em um território com um conjunto de clientes. Foi proposta uma heurística gulosa que gera os territórios iniciais, e posteriormente é aplicado um procedimento que divide e reagrupa os territórios que serão recombinados por um algoritmo de geração de colunas utilizando uma relaxação linear, até que todos os territórios apresentem uma melhora no critério de homogeneidade. Os autores afirmam que esta heurística gerou soluções 35% mais homogêneas que as vigentes.

Um modelo geométrico genérico é proposto juntamente com diferentes teoremas matemáticos considerando teoria de grafos em Saracco and Saracco (2019). O trabalho teve foco no PD eleitoral, buscando o menor perímetro possível que delimite determinada área com um número esperado de eleitores.

Uma proposta baseada no agrupamento de polígonos para o redistritamento foi apresentada em Joshi et al. (2012) para dois estudos de caso: uma divisão de área estadual para determinar

número de congressistas e outra sobre a formação de distritos escolares. A proposta de solução dos autores resolveu o distritamento como um processo de busca iterativo que parte de uma semente inicial e através da recombinação de polígonos alcança um resultado ótimo. O algoritmo proposto obteve resultados superiores a soluções clássicas considerando a compacidade dos territórios e homogeneidade populacional.

No trabalho de Bozkaya et al. (2003) foi tratado um problema de distritamento eleitoral que combina vários objetivos em uma única função multicritério. O trabalho também aplicou uma metaheurística que usa busca tabu em conjunção com memória adaptativa, na qual a melhor solução em uma execução anterior é utilizada como base na geração de novas soluções. Segundo os testes, este método produziu soluções que reduziram de 25% para menos de 1% o desvio populacional para a cidade de Edmonton – Canadá.

O trabalho apresentado em Bacao et al. (2005) aborda o problema de distritamento eleitoral na região de Lisboa e propõe um AG que foi capaz de prover soluções viáveis para o mesmo com estudo para uma grande área territorial. Em Castelli et al. (2015), os autores otimizaram operadores geométricos semânticos para o problema de distritamento eleitoral. Estes operadores, inicialmente apresentados em Moraglio et al. (2012), aplicam a informação semântica do problema diretamente no processo evolucionário, medindo as similaridades entre os indivíduos do AG desenvolvido. Esta proposta aprimorou o resultado anterior, ainda que a solução tenha gerado uma solução final de difícil reprodução devido ao tamanho do seu resultado.

Em Rincón-García et al. (2013) foi realizada a comparação entre tratar o problema de distritamento eleitoral como monoobjetivo ou multiobjetivo. No problema estudado considerando apenas um objetivo, a compacidade dos distritos e a distribuição igualitária dos eleitores são ponderadas e tratadas como um único objetivo. Considerando o problema multiobjetivo é encontrada uma fronteira de Pareto contendo um conjunto de soluções factíveis não dominadas. Nos testes realizados, a solução obtida através do modelo multiobjetivo forneceu resultados melhores quando comparados com as soluções obtidas pelo modelo monoobjetivo. Ambos modelos foram implementados considerando abordagens de solução baseadas no *simulated annealing*, apresentada originalmente em Kirkpatrick et al. (1983).

Em Vanneschi et al. (2017), os autores propõe um método multiobjetivo no tratamento do problema de distritamento eleitoral. A solução combina o método elitista NSGA-II (*Non-dominated Sorting Genetic Algorithm II*) com uma estratégia de busca local. O NSGA-II é utilizado com o intuito de gerar um conjunto de soluções não-dominadas, que são aproximações da fronteira de Pareto. A busca local percorre vizinhanças de tamanho variável em busca de soluções melhores. A proposta apresentou resultados confiáveis e superiores quando comparada aos resultados obtidos pela heurística CPSC (*Constrained Polygonal Spatial Clustering*), que os autores afirmam ser uma

das melhores para o problema até então. O NSGA-II foi comparado com o SPEA-II (*Strength Pareto Evolutionary Algorithm*) em Lara-Caballero et al. (2019), também na proposta de resolver o problema de distritamento eleitoral. Os autores apresentam propostas de solução consideradas ideais para problemas sem um limite claro de distribuição populacional e compacidade. Como ambas metaheurísticas forneceram soluções viáveis na região de uma fronteira de Pareto, os autores não concluíram qual foi superior.

Uma variação do distritamento eleitoral que aborda a alocação das sedes das zonas eleitorais no estado de Santa Catarina foi tratado em Hirose et al. (2020). Considerando que as zonas eleitorais são responsáveis pela organização das eleições, registro dos eleitores, e supervisão do processo eleitoral, a distribuição homogênea dos cartórios considerando um número equilibrado de eleitores e a compacidade deste território é um avanço no uso racional de recursos públicos. Para este fim, os autores propõem um modelo de programação linear inteira mista para a alocação das sedes das zonas eleitorais considerando os endereços já existentes. Para alcançar a solução foram propostas duas fases. Na primeira fase é definido o número ideal de zonas e a sua localização. Na segunda fase, cidades que foram subdivididas com mais de uma zona tem uma atualização para uma disposição ótima das zonas de votação. Foi utilizado o solver CPLEX para o resolver o problema. Segundo os testes realizados, os resultados alcançados apresentam uma solução de melhor homogeneidade.

Algoritmos genéticos também foram aplicados a problemas de distritamento relacionados ao acesso a unidades de saúde. Como na solução proposta por Steiner et al. (2014), que aborda o problema de definição de micro regiões que maximizem a homogeneidade da população, em relação às suas necessidades de serviços médicos, e minimizem o tempo de viagem dos pacientes. Em Gutiérrez-Gutiérrez and Vidal (2015), os autores apresentam o problema de distritamento relativo ao acompanhamento médico residencial de famílias, considerando o rápido crescimento das cidades e a necessidade de redistritamento, definindo uma solução que não se deteriore em um curto período de tempo. Uma outra aplicação do PDC na área de saúde é apresentada em Lin et al. (2018), onde os autores propõem um método de solução baseado no NSGA-II para resolver o problema de alocação de agentes de saúde para o tratamento de idosos. Uma característica que diferencia o problema abordado neste artigo em comparação com os demais citados é que este não possui um número fixo predeterminado de distritos.

Outra proposta de solução para o problema de redistritamento para oferta de serviços para idosos, residentes em Hong Kong na China, foi feita em Lin et al. (2017). Considerando o aumento da expectativa de vida, um grande número de cidadãos de idade avançada se encontram em suas residências e não em unidades de tratamento e a necessidade de alimentação de muitos é atendida por um serviço de entrega de alimentos. Este serviço segue um mapa de territórios para otimizar o tempo de entrega. Um conjunto de veículos é despachado de uma cozinha central e estaciona em

um número pré fixado de distritos. Destes locais saem entregadores que vão a pé até as residências para entregar as refeições. O objetivo consiste em determinar pontos ótimos para os veículos que minimizem: (i) o caminho entre os entregadores e as residências dos idosos; e (ii) uma janela de entregas que devem se adequar aos horários das refeições. A proposta combina um método exato baseado em programação inteira mista para selecionar o local do estacionamento e uma heurística gulosa para determinar as rotas que serão realizadas a pé. Segundo os testes realizados, a proposta gerou soluções válidas em menor tempo computacional que a existente. Distritamento aplicado à saúde também foi tratado em Yanik et al. (2019), onde se considerou simultaneamente múltiplos critérios como balanço da carga de serviço, acessibilidade, compacidade das áreas, renda dos pacientes e a conformidade da manutenção dos distritos de atendimento existentes em uma aplicação para Istambul na Turquia.

Uma abordagem de solução multiobjetivo foi proposta em Farughi et al. (2019), para o monitoramento e promoção de higiene e saúde pública através da alocação ótima de times de profissionais em distritos de trabalho. Foi considerado tanto a formação heterogênea das equipes, para maximizar o atendimento das diferentes especialidades necessárias para a população, quanto a formação compacta e contígua dos distritos resultantes, considerando uma maior cobertura e o menor custo possível. Para tal, duas metaheurísticas evolutivas foram propostas e comparadas, uma baseada no comportamento de formigas e uma segunda, que de acordo com os autores obteve melhor performance, denominada GWO (*Grey Wolf Optimizer*). O GWO envolve classificar soluções não-dominadas de Pareto como lobos alpha, beta, delta e ômega, que são iterativamente comparados com novas soluções geradas, simulando o comportamento de uma alcateia de lobos, onde o lobo alpha é a solução de melhor *fitness* e cada iteração simula uma caçada.

O distritamento é aplicável também na distribuição de necessidades críticas como na alocação de ambulâncias. Regis-Hernández et al. (2018) considera a necessidade de alocação de múltiplos veículos em casos de desastres em um ambiente dinâmico. Os autores propõem que após o distritamento inicial, seja realizado iterativamente um redistritamento baseado em dados operacionais através do uso de um algoritmo de otimização que considera a interoperabilidade de metaheurísticas e técnicas de programação matemática (Speranza and Archetti, 2014). Segundo os autores, o redistritamento proposto reduziu o número de ambulâncias que precisavam deixar seus distritos de base para atendimento. O problema de distritamento para alocação de ambulâncias também foi tratado em Mohammadi et al. (2014) com uma metaheurística que combinou *Simulated Annealing* com AG.

Em Lari et al. (2009) é apresentada uma proposta de solução de programação linear para problemas de distritamento geral. A formulação permitiu agrupar dentro de um tempo considerado razoável pelos autores instâncias com até 500 entidades. Os testes realizados apresentaram um tempo

excessivo para instâncias maiores.

Soluções para o PDC relativo ao problema de logística de uma rede de distribuição de bebidas foram propostas em Ríos-Mercado and Fernández (2009) e G. Elizondo-Amaya et al. (2014), utilizando a metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*). Nestes artigos motivados por aplicações reais, as empresas buscavam formar territórios que fossem o mais compactos possível, respeitando o equilíbrio do número de clientes em cada área atendida. A distância euclidiana foi considerada como medida de dispersão. No trabalho Ríos-Mercado and Escalante (2016), o GRASP foi ajustado para permitir a construção simultânea de todos os distritos, o que os autores afirmaram ter obtido uma redução na violação da restrição de contiguidade do problema. *Path relinking* foi utilizado como mecanismo de busca, a fim de combinar trechos similares em soluções distintas que melhorassem a solução final. Já no trabalho (González-Ramírez et al., 2017), o GRASP foi combinado com uma Busca Tabu, na qual soluções geradas recentemente são designadas como “Tabu ativa” durante um período fixo de iterações e guardadas em uma memória que só é atualizada caso uma solução melhor seja gerada. A combinação de heurística gulosa com a Busca Tabu recebeu o nome RG-TS (*Random-Greedy-and-Tabu-Search*) e foi apresentada inicialmente em Camacho-Collados et al. (2015) para um distritamento policial e posteriormente, aprimorada em Chen et al. (2019) como GP-TS (*Graph-Partition-Tabu Search*), através da representação da rede de ruas como um grafo bidirecional com informações de predição de crimes.

Um modelo de distritamento para serviços foi proposto em García-Ayala et al. (2016). O modelo produziu partições que são contíguas e com carga de trabalho balanceadas, combinando um algoritmo *Branch and Bound*, com a estratégia de corte denominado *cut generation*, que reduz iterativamente o tamanho da região da solução. A solução proposta conseguiu encontrar resultados ótimos para instâncias de até 491 nós e 763 arestas, gerando distritos adequados para áreas de trabalho.

Outra abordagem de solução aplicou o método *Branch and Bound* em Ríos-Mercado and López (2013). O algoritmo foi aplicado juntamente com uma estratégia de corte que eliminava soluções que violavam as restrições de contiguidade e permitiu instâncias de até 10000 clientes. Este trabalho, motivado pelo problema de distribuição de bebidas, consistia em dividir uma cidade em territórios de acordo com: o número de consumidores, a demanda de bebidas e o esforço realizado pelas entregas. Sendo necessário satisfazer o critério de gerar uma solução com territórios balanceados nesses quesitos.

O problema de distritamento com dispersão máxima foi abordado em Ríos-Mercado and Bard (2018) aplicado à necessidade de se ter unidades para o descarte e posterior coleta de lixo elétrico e eletrônico. Os autores, partindo de um modelo de programação inteira mista propuseram um modelo que visa a máxima dispersão particionando o território existente e distribuindo os pontos

de coleta. A solução parte de um pré-processamento que reduzia o espaço de busca do problema através de um algoritmo exato desenvolvido e posteriormente considerando um limite superior para a dispersão das soluções aplica uma busca binária direcionada que determina a solução ótima. Testes extensos realizados demonstraram que o algoritmo proposto teve um desempenho 23% superior a solução existente e o relaxamento permitiu tratar a distribuição de 1400 unidades de coleta contra 100 unidades da solução anterior.

Um estudo sobre diferentes propostas de solução para o problema de distritamento foi apresentado em Kalcsics (2015). Nesta obra os autores apresentam diferentes abordagens de solução para as quatro principais aplicações classificadas como problemas de distritamento: (i) o distritamento político, (ii) a divisão de território de vendas, (iii) a divisão de distritos para áreas de serviço e (iv) a divisão de territórios para cobertura de entregas. Foram apresentadas várias metaheurísticas para o tratamento destes problemas, a saber, Busca Tabu, GRASP e *Simulated Annealing*, tendo como conclusão que novas pesquisas devem ter como objetivo tratar os problemas de distritamento de forma generalizada, propondo soluções abrangentes e aplicáveis entre estudos de caso similares.

Outro estudo abrangente foi realizado em Levin and Friedler (2019) para o problema de distritamento eleitoral, com ênfase no novo ciclo de redistritamento para os Estados Unidos em 2020. O trabalho comparou a configuração de distritos existente e uma simulação com *simulated annealing* e propõe um novo algoritmo recursivo de divisão e conquista. O algoritmo combina particionamento e recombinação dos componentes do problema buscando uma solução ótima. Os resultados obtidos mostraram ganho de performance no tratamento de desvio populacional.

O trabalho desta tese de mestrado baseia-se no problema apresentado em de Assis et al. (2014), onde é apresentado o problema de agrupamento capacitado relacionado à definição de regiões de trabalho para leituristas de medidores de energia elétrica. A solução proposta pelos autores é baseada na metaheurística GRASP. A análise multiobjetivo realizada, busca distritos compactos e homogêneos e ainda considerou uma restrição de conformidade. Esta restrição considera os custos operacionais de se modificar os distritos existentes. O trabalho apresentou métodos construtivos e de busca local que asseguravam a conectividade dos distritos. Busca local que obtém rápido reconhecimento de boas vizinhanças, o que permite a solução de grandes redes em baixos tempos de execução. Outro trabalho recente próximo a esta pesquisa, realizado em parceria com a Companhia de Energia Elétrica do Maranhão, também abordou este problema em Costa et al. (2018), onde utilizaram uma variante da heurística K-Means nomeada RCMeans para a qual os autores afirmam ter obtido resultados superiores ao mapeamento original utilizado pela empresa.

Como pode ser observado, o PD permanece pertinente mesmo após décadas de estudo e é tema de vários artigos com metodologias diversas. As principais aplicações do PD consistem no distritamento eleitoral, e seu impacto no resultado das eleições, além da crescente necessidade

das empresas de fornecer serviços e logística bem adaptados aos desafios de cidades em constante evolução. Expansão esta que pode levar a cenários complexos com muitas restrições e instâncias cada vez maiores que necessitam de metodologias eficientes para obter soluções de qualidade em tempo computacional aceitável.

A abordagem dos AGs é uma opção válida para tratar problemas com grande espaço de busca, além de possuir semelhança com a evolução natural que o torna de fácil compreensão. A principal contribuição deste trabalho é o desenvolvimento de uma metaheurística com características distintas das vistas na literatura pois apresenta: *(i)* um método construtivo otimizado, *(ii)* operadores genéticos ajustados, *(iii)* um novo operador de *crossover*, *(iv)* um comparativo inédito na literatura entre duas estruturas populacionais para este problema, *(v)* uma proposta de otimização de hiperparâmetros, *(vi)* resultados para 25 redes de tamanhos e características distintas. A solução também foi capaz de encontrar soluções para uma grande instância com 2408 nós e 5324 arestas em baixo tempo computacional.

Capítulo 3

Apresentação e Formulação do Problema

Este capítulo apresenta detalhadamente o Problema de Distritamento Capacitado (PDC) estudado. Para o bom entendimento do problema é necessário abordar de forma detalhada suas características, restrições e critérios considerados.

3.1 Problema de Otimização das Leituras de Consumo em Redes de Serviço

Problemas de distritamento tem como objetivo particionar um conjunto de n unidades de uma determinada região em p distritos (com $p < n$) na busca pela melhor solução possível de uma função objetivo. Esta função emprega normalmente um ou mais critérios que são definidos de acordo com a aplicação em estudo (Ríos-Mercado and Fernández, 2009). Devido ao elevado número de combinações possíveis de distritos o espaço de busca cresce tipicamente de forma exponencial.

O PDC tratado no escopo deste trabalho é descrito como o agrupamento em p distritos de um dado conjunto de n unidades territoriais, onde cada unidade tem associada um ou mais valores. Cada distrito tem uma capacidade conhecida, e a soma total dos pesos dessas unidades atribuídas a um determinado distrito não deve exceder sua capacidade total (de Assis et al., 2014).

O problema em estudo neste trabalho (apresentado em Assis (2009) e em de Assis et al. (2014)) é um PDC aplicado à definição de distritos para leituristas de medidores de energia elétrica. Os autores abordam a necessidade mensal que empresas de fornecimento de energia possuem de medir o consumo dos clientes nas áreas de concessão. Considerando a evolução e crescimento natural das cidades, os lotes de trabalho destas concessionárias precisam de constante atualização e a falta de um plano racional pode causar um grande impacto nos custos operacionais de leitura. Este processo deve redefinir as fronteiras geográficas dos distritos na busca de equalizar as cargas de trabalho e otimizar a compacidade dos distritos. O PDC em estudo deve ainda respeitar as seguintes restrições:

- Os clientes devem ser divididos (ou particionados) em um número fixo de p distritos;
- Cada cliente deve ser atribuído a um único distrito;
- Os distritos resultantes precisam ser conexos;

- Todos os nós precisam ser atribuídos a um distrito;
- A soma dos pesos dos clientes associados a um determinado distrito não deve ultrapassar a capacidade desse distrito.

Para o problema em estudo neste trabalho serão considerados os seguintes critérios:

1. **Critério de compacidade:** busca uma baixa dispersão entre os clientes em um mesmo distrito, favorecendo uma posterior construção de rotas. O objetivo é alcançar uma operação de leitura que seja a mais eficiente possível (veja Eq. (3.1)).
2. **Critério de homogeneidade:** busca equilibrar a carga de trabalho nos distritos. No PDC em estudo consideramos duas medidas para mensurar a carga de trabalho: *(i)* o número de medidores de energia elétrica, e *(ii)* o tempo de leitura. A homogeneidade possui o objetivo de minimizar os custos operacionais de mão de obra promovendo uma carga de trabalho equivalente entre os distritos. (veja Eq. (3.2)).

É importante realçar que, apesar do PDC em estudo ter sido aplicado ao problema de leituras de medidores de energia elétrica, tanto o modelo quanto o método de solução propostos podem ser facilmente adaptados para dados que representem outras redes como gás, água e outros serviços. Assim como acrescentar novos critérios ao problema, conforme os apresentados em de Assis et al. (2014), por exemplo.

3.2 Representação do PDC em Estudo

Utilizando a terminologia da teoria dos grafos (Bondy and Murty 1982; Bondy and Murty 2008), a representação do PDC apresentado pode ser feita através de um grafo conexo não-orientado $G(V, E)$ no qual V é o conjunto não-vazio dos n nós e E representa o conjunto das m arestas do grafo. Cada nó desse grafo possui um conjunto não vazio A de atividades associadas e w_i^a significa a atividade $a \in A$ referente ao nó $i \in V$.

No contexto do problema das concessionárias de energia elétrica, a representação geográfica de atuação de cada empresa é realizada associando uma esquina do mapa a um nó do grafo, e a cada rua desta região uma aresta. A Figura 3.1 exemplifica um mapa de uma região com os nós demarcados em vermelho e a Figura 3.2 ilustra a representação desta região através do grafo.

Tipicamente, as abordagens de solução realizam o agrupamento de nós. No entanto, os valores das atividades (tempo de leitura e número de medidores) estão, neste caso, associadas às arestas. Por esta razão foi proposto em de Assis et al. (2014) uma transformação do grafo denominada $G \rightarrow G'$, onde cada aresta do grafo original G é equivalente a um nó no grafo resultante G' . Adicionalmente em G' , dois nós adjacentes representam duas arestas adjacentes em G . Considerando



Figura 3.1: Mapa original sinalizado ©2020 Microsoft.

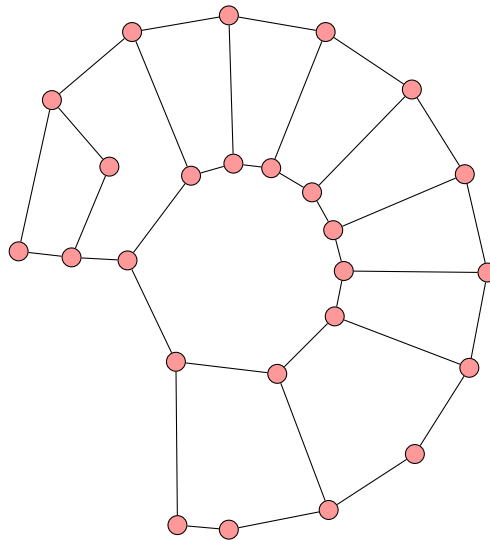


Figura 3.2: Representação do mapa através de grafo.

ainda que as coordenadas geográficas dos nós em G são conhecidas, é possível calcular a distância euclidiana entre cada par de nós $i, j \in V$. As coordenadas geográficas em G' são dadas pelo ponto médio da aresta que este nó representa. A Figura 3.3 ilustra a transformação do grafo ($G \rightarrow G'$).

Um distrito k pode ser considerado como um subgrafo $V_k \subseteq V$. Um conjunto artificial de nós $V_0 = \{1, 2, \dots, p\}$ representa o centro dos distritos com p representando o número de distritos totais.

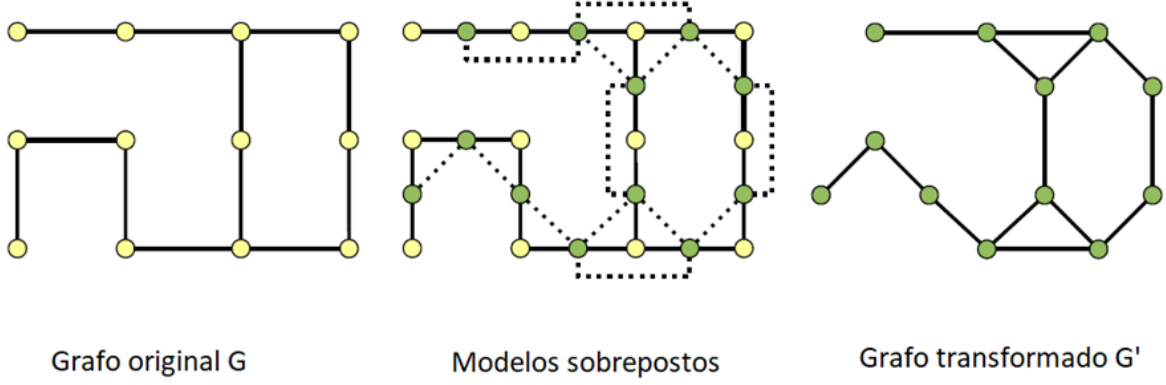


Figura 3.3: Transformação do grafo (de Assis et al., 2014).

3.3 Modelo Matemático

O modelo matemático empregue nesta dissertação foi descrito em de Assis et al. (2014). O modelo usa dois conjuntos de variáveis de decisão, h_{ki} e z_{ijk} . Se o nó i for atribuído ao distrito k , então $h_{ki} = 1$, e caso contrário $h_{ki} = 0$. Similarmente, se um nó i e um nó j são designados a um distrito k , então $z_{ijk} = 1$, e caso contrário $z_{ijk} = 0$.

As funções objetivos consideradas, de compacidade e de homogeneidade, são apresentadas nas Equações (3.1) e (3.2), respectivamente.

$$F(S) = \sum_{k=1}^p \max_{ij \in V} d_{ij} z_{ijk} \quad (3.1)$$

$$G(S) = \sum_{k=1}^p \sum_{a \in A} \sum_{i=1}^n |w_i^a h_{ki} - \mu^a| \quad (3.2)$$

Onde d_{ij} é a distância euclidiana entre os nós $i, j \in V$ que é definida pela fórmula: $d_{i,j} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$, sendo que (x_i, y_i) são as coordenadas dos nós i e (x_j, y_j) do nó j . O valor médio da atividade a considerando os p distritos é definido pela fórmula $\mu^a = \sum_{i=1}^n w_i^a / p$. Onde a é o identificador da atividade, i representa o nó e p é o número pré-determinado de distritos. Em uma solução com carga de trabalho perfeitamente homogênea, todos os distritos deveriam ter seu valor de carga de trabalho igual a μ^a para cada atividade a , i.e. $G(S) = 0$.

Sendo S uma solução para o problema em estudo, a função $F(S)$ minimiza a soma das distâncias máximas entre um par de nós de cada distrito e $G(S)$ minimiza a soma dos desvios das cargas de trabalho do valor médio μ^a desejado. Conforme apresentado em de Assis et al. (2014), o modelo matemático para o PDC aplicado ao problema de leituristas de medidores de energia elétrica pode ser formulado como segue.

$$\min \quad \lambda F(S) + (1 - \lambda)G(S) \quad (3.3)$$

S.a.

$$\sum_{k=1}^p h_{ki} = 1 \quad i \in V \quad (3.4)$$

$$\sum_{i \in \cup_{v \in D} N^V \setminus D} h_{ki} - \sum_{i \in D} h_{ki} \geq 1 - |D| \quad k \in V_0, D \subset V \quad (3.5)$$

$$z_{ijk} \leq h_{ik} \quad i, j \in V, \quad k \in V_0 \quad (3.6)$$

$$z_{ijk} \leq h_{jk} \quad i, j \in V, \quad k \in V_0 \quad (3.7)$$

$$z_{ijk} \geq h_{ik} + h_{jk} - 1 \quad i, j \in V, \quad k \in V_0 \quad (3.8)$$

$$h_{ki}, z_{ijk} \in \{0, 1\} \quad i, j \in V, \quad k \in V_0 \quad (3.9)$$

A função objetivo apresentada na Equação (3.3) minimiza a combinação convexa da soma dos dois critérios introduzidos nas Equações (3.1) e (3.2), onde $\lambda \in [0, 1]$, assim o valor de λ pondera os dois objetivos do problema. O conjunto de restrições em (3.4) garante que cada nó deve ser atribuído apenas a um único distrito. As restrições apresentadas na Equação (3.5) garantem a conectividade dos distritos, quando selecionado um subconjunto D de nós de uma solução, se todos os elementos deste subconjunto pertencerem ao mesmo distrito, algum outro nó desse distrito será vizinho a D , ou seja, cada distrito será formado por um subgrafo conexo de V , ou seja, $V_k \subseteq V$. Informações mais detalhadas sobre estas restrições podem ser consultadas em Assis (2009). As Equações (3.6)–(3.8) garantem que quando ambos os nós i e j são atribuídos a um mesmo distrito ($h_{ik} = h_{jk} = 1$) então $z_{ijk} = 1$. As restrições (3.9) definem o domínio das variáveis de decisão.

Capítulo 4

Metodologia de Solução para o PDC

A abordagem proposta para resolver o PDC parte de conceitos bem estabelecidos na literatura de AG (Holland, 1975; Goldberg, 1989; Whitley, 1994; Lopes et al., 2013; Gong et al., 2015). De acordo com Bacao et al. (2005), com o tempo devido, um AG bem codificado pode encontrar uma solução ótima. Encontrar uma maneira adequada de representar o indivíduo da população (uma solução para o problema) em um cromossomo que será recombinado e poderá sofrer mutação é um desafio. Os AGs são métodos de otimização que trabalham de maneira “aleatoriamente orientada”, de acordo com regras probabilísticas que são baseadas em uma analogia com os princípios de Darwin para a evolução das espécies e genética (Whitley, 1994). Existem diversas propostas e variações, porém a ideia básica de implementação de um AG é semelhante: considerando uma população de indivíduos dentro de um ambiente com recursos limitados, a competição por esses recursos causa uma seleção natural, ou seja, a sobrevivência do mais forte, que leva a um aumento da aptidão da população restante (Eiben and Smith, 2015). Os AGs utilizam uma população inicial, que é evoluída através de operadores genéticos que visam melhorar a aptidão (*fitness*) dos indivíduos. Adicionalmente, AG são flexíveis, adaptáveis e normalmente são utilizados para encontrar soluções de problemas de otimização quando se trabalha com instâncias de grande porte (Linden, 2006). O fluxo de execução do AG é apresentado na Figura 4.1.

4.1 Indivíduo

AG mantém uma população de indivíduos que representam potenciais soluções e que evoluem através de um processo de seleção e dos operadores genéticos (Michalewicz, 1996). A função da população é conter a representação de soluções possíveis (Eiben and Smith, 2015). Cada indivíduo é representado por um cromossomo. No caso do PDC em estudo, o cromossomo é representado através de um vetor de inteiros de tamanho n , sendo n o número de nós do grafo, ou seja, as unidades territoriais que deseja-se agrupar. A Figura 4.2 exemplifica o cromossomo de um indivíduo para o PDC abordado. Os índices de cada coluna do vetor representam os nós do grafo $G(V, E)$. Os números localizados na primeira linha indicam em qual distrito o nó i está atribuído. As cores

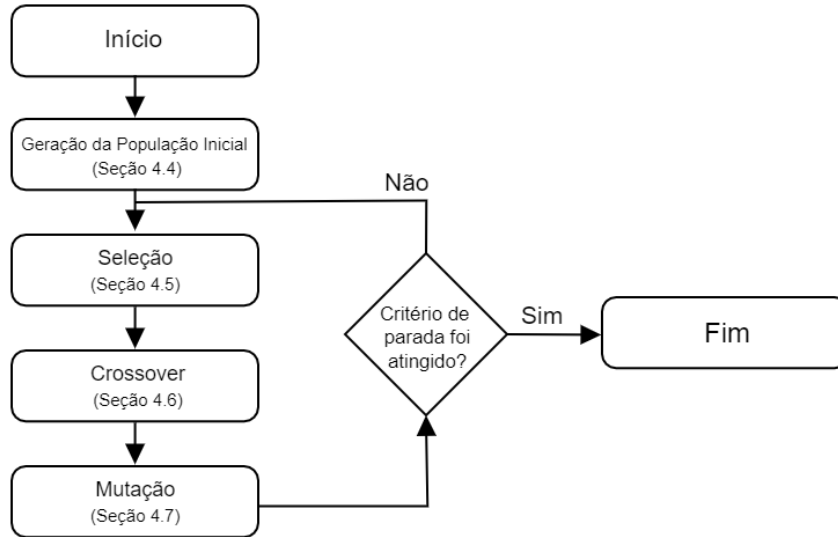


Figura 4.1: Fluxo de execução do AG.

apresentadas são apenas para ilustrar os distritos e facilitar a visualização de uma solução.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	—> Índice
1	1	1	1	1	2	2	2	2	2	3	3	3	3	3	—> Distrito

Figura 4.2: Representação do cromossomo de um indivíduo para o PDC.

É importante ressaltar que, no processo evolutivo o cromossomo dos indivíduos não necessariamente permanecem ou são criados com os territórios bem ordenados como ilustrado na Figura 4.2. Este formato de distribuição é praticamente impossível de ocorrer, pois a atribuição dos nós aos distritos não é sequencial, porém é mais didático. Comumente, a codificação do indivíduo ocorre de forma semelhante ao apresentado na Figura 4.3.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	—> Índice
1	3	1	2	3	1	3	2	2	1	2	3	3	1	2	—> Distrito

Figura 4.3: Representação mais realística do cromossomo de um indivíduo para o PDC em estudo.

4.2 Cálculo do *Fitness*

Todos os indivíduos de uma população possuem um valor de *fitness* associado que é atualizado a cada geração do AG. O *fitness* mede a qualidade de um indivíduo na população corrente e é calculado através de uma função. A função de *fitness* utilizada nesse trabalho é apresentada na Equação (4.4), a qual é composta pelas funções de compacidade e homogeneidade que são apresentadas nas Equações (4.1) e (4.2), respectivamente.

$$f_{F(S)} = \frac{1}{pd_{\max}} \sum_{k=1}^p \left\{ \max_{i,j \in V_k} d_{ij} \right\} \quad (4.1)$$

A Equação (4.1) calcula a maior distância euclideana entre dois nós $i, j \in V_k$, para cada distrito k . Este valor é então normalizado por $d_{\max}$ que é a maior distância Euclideana entre dois nós $i, j \in V$ e por p que é a quantidade de distritos.

$$f_{G(S)} = \frac{1}{p} \sum_{k=1}^p \sum_{a \in A} g^a(V_k) \quad (4.2)$$

A Equação (4.2) determina o somatório da carga de trabalho para cada distrito V_k e cada atividade a , $g^a(V_k)$, normalizado por p . O desvio da carga de trabalho média no distrito k para a atividade a é definido na Equação (4.3).

$$g^a(V_k) = \frac{|w^a(V_k) - \mu^a|}{\mu^a} \quad (4.3)$$

onde $w^a(V_k) = \sum_{i \in V_k} w_i^a$ é a quantidade da atividade a que está presente no distrito V_k .

O *fitness* do indivíduo (Equação (4.4)) é calculado pelo inverso da combinação convexa das Equações (4.1) e (4.2), onde $\lambda \in [0, 1]$ define a importância fornecida à compactidade e, consequentemente, à homogeneidade. Assim, quanto maior o valor de λ mais compactos e menos homogêneos são os distritos da solução avaliada, e quanto menor o valor de λ mais homogêneos e menos compactos.

$$Fitness = \frac{1}{\lambda f_{F(S)} + (1 - \lambda) f_{G(S)}} \quad (4.4)$$

4.3 População

Este trabalho considera a implementação e comparação de duas estruturas populacionais distintas. A primeira abordagem desenvolvida considera a utilização de uma população não estruturada (Seção 4.3.1), enquanto que no segundo caso é empregue uma população estruturada (Seção 4.3.2).

4.3.1 População não Estruturada

A população não estruturada é a forma que se encontra com maior frequência na literatura de armazenar e tratar os indivíduos da população em um algoritmo evolutivo. Considerando que cada indivíduo é uma solução para o problema, na implementação realizada, estes são dispostos em um vetor sem qualquer ordem ou hierarquia entre os mesmos. A Figura 4.4 apresenta uma população não estruturada com 78 indivíduos, sendo que cada posição nesta figura representa um indivíduo e

o valor representa seu *fitness*.

100	7	45	50	22	40	80	35	60	91	35	60	45
45	07	15	10	90	42	28	11	4	79	31	99	58
65	4	49	18	51	23	17	3	21	42	14	56	81
82	43	16	83	39	22	11	18	37	10	74	59	1
15	8	24	51	50	30	16	67	14	45	31	69	35
3	14	15	92	65	35	89	79	32	38	46	26	43

Figura 4.4: Representação de uma população não estruturada.

4.3.2 População Estruturada

A segunda estrutura proposta é uma população hierarquicamente estruturada, metodologia promissora já aplicada em problemas similares aos de agrupamento capacitado, como em Moscato (1989), Moscato et al. (1992), Toledo et al. (2014) e de Assis (2014). A população é organizada em uma estrutura de árvore $\mathcal{L}$ -ária com h -níveis. No caso deste trabalho, foi implementada uma estrutura de população que considera $h = 3$ e $\mathcal{L} = 3$. Estes valores representam um bom compromisso entre a qualidade das soluções e o tempo de execução de acordo com Moscato and Cotta e (2003) de Assis et al. (2015). Dessa forma, a população possui um total de 13 indivíduos distribuídos em uma árvore ternária com indivíduos que são classificados como líderes ou subordinados. A Figura 4.5 apresenta a população estruturada implementada nesta dissertação.

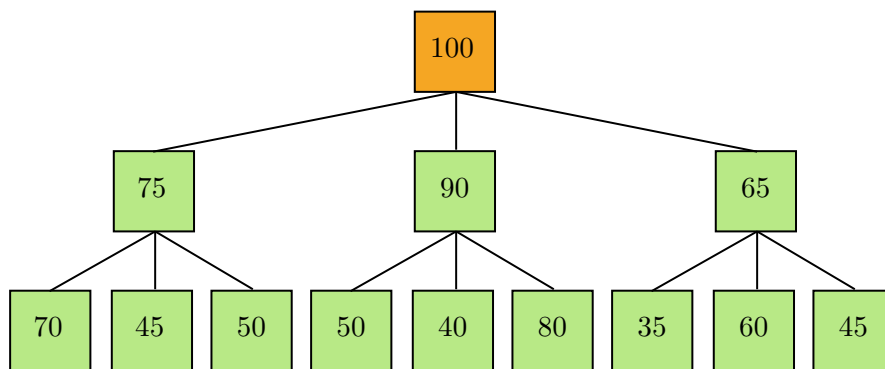


Figura 4.5: Representação da população estruturada.

Na Figura 4.5 cada nó da árvore representa um indivíduo e os seus valores o seu *fitness*. De acordo com a altura na árvore os indivíduos são classificados como líderes e subordinados. Nesta implementação, cada nó em um nível (a partir da raiz) é líder dos três nós no nível imediatamente seguinte. Como exemplo o nó raiz de *fitness* 100 é líder dos subordinados de *fitness* 75, 90 e 65, por sua vez o nó de *fitness* 75 é líder dos subordinados 70, 45 e 50. Neste tipo de população o melhor

indivíduo sempre está posicionado no nó raiz.

4.4 Geração da População Inicial

Baseada nos conceitos apresentados em Ríos-Mercado and Escalante (2016), a heurística para geração de uma população inicial viável cria exatos p distritos simultaneamente, favorecendo a concepção de distritos balanceados e reduzindo a ocorrência de nós isolados. O Algoritmo 1 apresenta a heurística proposta para gerar uma população inicial viável, isto é, uma população que respeite as restrições do problema (ver Seção 3.3). O algoritmo proposto tenta: (i) fornecer uma população bem distribuída pelo espaço de busca com o intuito de evitar uma convergência prematura; e (ii) que esta seja um ponto de partida viável para o problema considerando a função objetivo e restrições associadas.

Algorithm 1 $\text{initialPopulation}(p_{size}, d_{max})$

```

1:  $j \leftarrow 0$ 
2:  $pop \leftarrow \emptyset$ 
3: while ( $j \leq p_{size}$ ) do
4:    $I^j \leftarrow \emptyset$ 
5:   for ( $k = 0; k \leq p; k++$ ) do
6:      $v \leftarrow \text{randomCenter}(d_{max})$ 
7:      $I_k^j \leftarrow I_k^j \cup \{v\}$ 
8:   end for
9:    $\text{degreeSort}(V)$ 
10:  while ( $|I| \leq n$ ) do
11:    for ( $k = 0; k \leq p; k++$ ) do
12:       $v \leftarrow \text{bestAdjacentNode}()$ 
13:       $I_k^j \leftarrow I_k^j \cup \{v\}$ 
14:    end for
15:  end while
16:   $pop \leftarrow pop \cup I^j$ 
17: end while
18: return  $pop$ 

```

A variável p_{size} representa o tamanho da população e a variável d_{max} é relativa a maior distância Euclidiana entre dois nós $i, j \in V$. Inicialmente, a heurística começa com uma população vazia (linha 2) e em cada iteração do loop mais externo (linhas 3-17) é gerado um indivíduo da população. Para cada distrito k do indivíduo j é selecionado aleatoriamente um nó v , que além de representar o centro do distrito, é também o primeiro nó atribuído a cada distrito (linhas 5-8). A seleção do centro v considera uma distância mínima entre tais centros, favorecendo a distribuição uniforme dos distritos na região em estudo. Inicialmente são sorteadas distâncias maiores que são depois progressivamente reduzidas a fim de viabilizar a atribuição (linha 6). Em seguida, os demais nós são ordenados pelo método *degreeSort* (linha 9), que aplica um *quicksort* para que os nós de menor grau sejam considerados primeiro no processo de atribuição. O *loop* interno que compreende as

linhas (linhas 10-15) garante que cada nó $v \in V$ é atribuído a um distrito k do indivíduo j da população e o método *bestAdjacentNode* (linha 12) é que define a atribuição destes nós. Este último seleciona, a cada iteração, um nó que é adjacente a algum nó pertencente ao distrito k e que respeite o limite da atividade de acordo com o valor de $\mu^a \forall a \in A \times \tau$, sendo τ um parâmetro de tolerância fixo em 0,05. Quando o limite for ultrapassado em todos os distritos os nós restantes são atribuídos nos distritos que violem menos o valor médio. Cada indivíduo j é adicionado na população (linha 16) e finalmente a população com p_{size} é retornada (linha 18).

A complexidade do algoritmo é $O(n)$, considerando $O(n \times p_{size} \times p)$ com p_{size} e p executados por um número fixo e baixo de iterações ($O(1)$).

4.5 Seleção

O objetivo da seleção é distinguir os indivíduos, com base na sua qualidade e definir quais serão os pais naquela geração (Eiben and Smith, 2015). Este operador é responsável por direcionar a população para evolução. No escopo proposto para este trabalho, duas abordagens são comparadas na etapa de seleção, (i) uma probabilística usada na população não estruturada (Seção 4.5.1) e, (ii) outra baseada na população hierarquicamente estruturada na qual todos os pares de indivíduos (líder, subordinado) são selecionados para participarem do *crossover* a cada geração. O valor da taxa de seleção s_r , multiplicado pelo tamanho da população pop_s , determina o número de filhos que será criado a cada geração.

4.5.1 Operador Híbrido

O operador híbrido considera dois requisitos. O primeiro é que indivíduos de maior *fitness* participem do cruzamento. O segundo requisito é que continue a ocorrer diversidade genética apesar da primeira condição. Com base nestes requisitos foi desenvolvido um operador que mescla as seleções elitista e a seleção por roleta. Este operador revelou-se superior à aplicação individual do método elitista ou da estratégia por roleta. Na seleção elitista os indivíduos de maior *fitness* são selecionados para o *crossover*. Na seleção por roleta, todos os indivíduos possuem uma certa probabilidade de serem escolhidos para o cruzamento, a qual é proporcional ao seu valor de *fitness*. A Figura 4.6 ilustra a seleção por roleta considerando cinco indivíduos, onde cada letra e cor representam um indivíduo e os números a chance de serem selecionados. Para cada par de indivíduos selecionados, o operador de seleção híbrido proposto seleciona o primeiro indivíduo através da seleção elitista e o segundo por roleta, o que leva a uma maior incidência dos indivíduos de maior aptidão mas ainda favorecendo uma certa diversidade na escolha do segundo progenitor na geração de novos indivíduos.

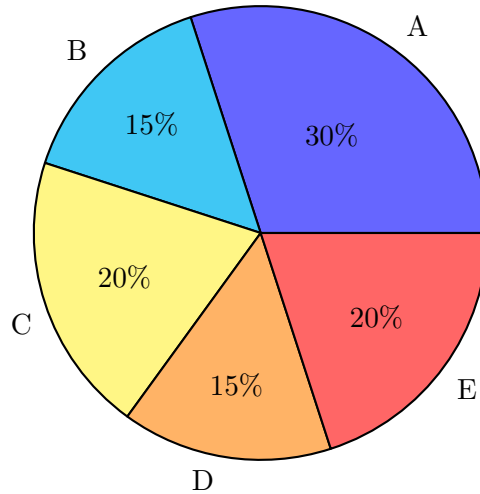


Figura 4.6: Seleção por roleta.

4.5.2 Operador Determinístico

No GA com a população estruturada, o operador de seleção é determinístico, considerando que todos os indivíduos são selecionados para participarem do *crossover*. Os indivíduos líderes na hierarquia de níveis da árvore da população são selecionados para participarem do cruzamento com seus subordinados, conforme a estrutura da árvore ternária apresentada na Figura 4.5.

4.6 Crossover

Realizada a seleção é possível executar o operador de *crossover*. Este operador é utilizado para combinar a informação genética de dois indivíduos gerando assim um filho. A expectativa é que os genes dos indivíduos de maior *fitness* apareçam com maior frequência na população, auxiliando na convergência para uma boa solução (Lopes et al., 2013). Inicialmente, para este trabalho, o operador de *crossover* de um ponto foi implementado, no qual um único ponto de corte é selecionado aleatoriamente para dividir a informação genética dos pais e combinar seus cromossomos. A Figura 4.7 ilustra a ideia geral deste operador.

Apesar do AG já ter sido aplicado com sucesso em Problemas de Distritamento como em (Bacao et al., (2005) e (Vanneschi et al., (2017), experimentos realizados nesta dissertação demonstraram que é extremamente difícil de se produzir filhos viáveis para o PDC devido às restrições: (i) os distritos gerados precisam ser um subgrafo conexo (Eq. (3.5)), (ii) cada nó deve estar designado a um único distrito (Eq. (3.4)), e (iii) todos os nós devem ser designados a algum distrito (3.6)).

Os problemas para garantir a conectividade surgem quando o operador de *crossover* começa a copiar os genes do segundo indivíduo. Durante os experimentos foi possível observar que pela natureza aleatória da heurística construtiva, a localização geográfica dos distritos e respectivos nós alocados a eles, não se reflete na posição desses nós no cromossomo. A Figura 4.8 ilustra como o

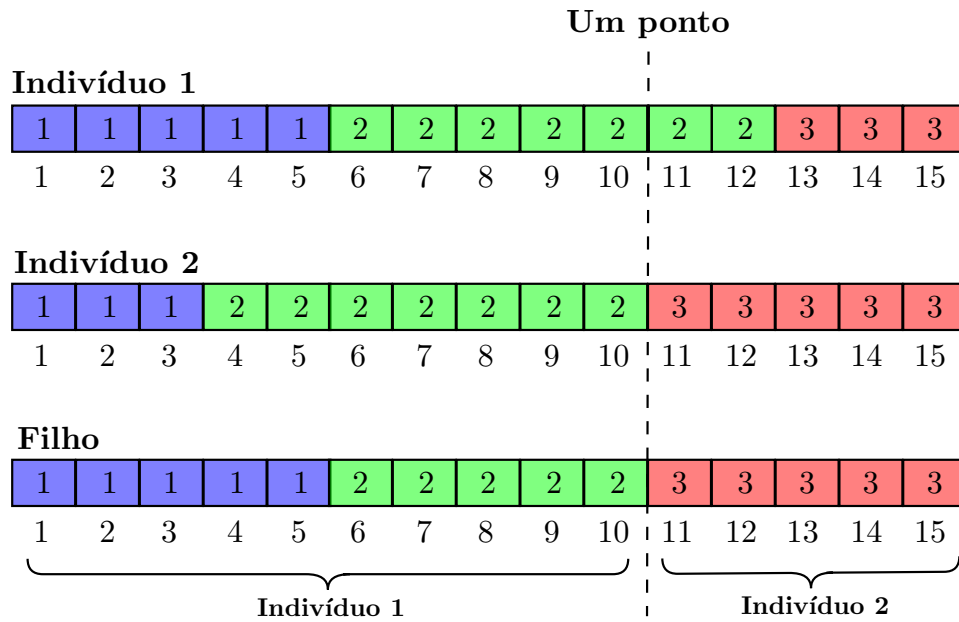


Figura 4.7: Operador de *crossover* de um ponto.

fato, por exemplo, dos rótulos dos distritos serem distintos, pode fazer com que o *crossover* gere filhos inactíveis. No exemplo apresentado na Figura 4.8, o distrito 3 desaparece tornando o filho inviável.

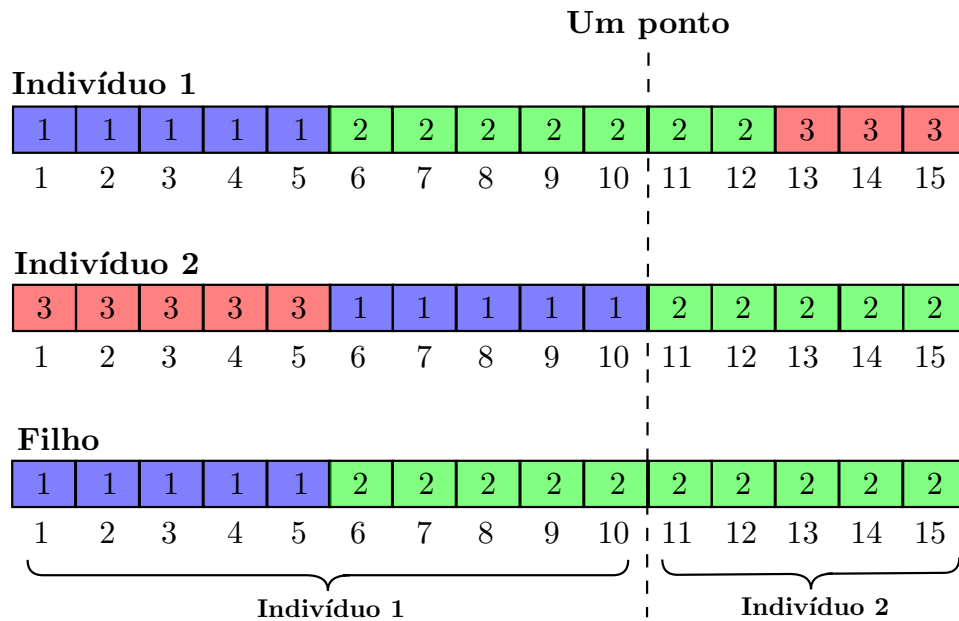


Figura 4.8: Exemplo de geração de filhos inactíveis após o cruzamento.

Devido a este motivo, foi necessário desenvolver e avaliar dois novos operadores, baseados no operador padrão *crossover* de um ponto, os quais são apresentados nas próximas seções.

4.6.1 Distância de *Hamming*

Esta abordagem tenta quantificar a diferença entre dois indivíduos selecionados para participarem do cruzamento. Uma possível forma de medir esta diferença é através da distância de Hamming, que quantifica o número de posições diferentes entre dois vetores de tamanho igual (MacKay, 2002). No caso do PDC deste trabalho, a distância de Hamming calcula quantas posições dos cromossomos dos progenitores, *i.e.*, quantos nós do grafo, estão atribuídos a um distrito diferente. O valor obtido foi normalizado, pelo número total de nós do grafo (n), conforme apresentado na Equação (4.5).

$$dh_{I_1, I_2} = \frac{\sum_{i=1}^n Dif(I_1^i, I_2^i)}{n}, \quad (4.5)$$

onde dh_{I_1, I_2} significa a distância de Hamming entre os indivíduos, n é a quantidade de nós do grafo, bem como, o tamanho do cromossomo, I_1^i e I_2^i são os dois indivíduos que foram selecionados para o *crossover*, e Dif é uma função que retorna 1 se a posição i dos indivíduos pais são diferentes e 0 se são iguais.

A Figura 4.9 ilustra uma comparação entre dois indivíduos. Os índices se referem ao rótulo dos nós, os valores nos alelos dos cromossomos são referentes à qual distrito cada nó está atribuído (assim como as cores), as setas vermelhas ressaltam as posições onde o mesmo nó se encontra em distritos diferentes nos dois indivíduos e as setas verdes as posições com alocação deste nó no mesmo distrito.

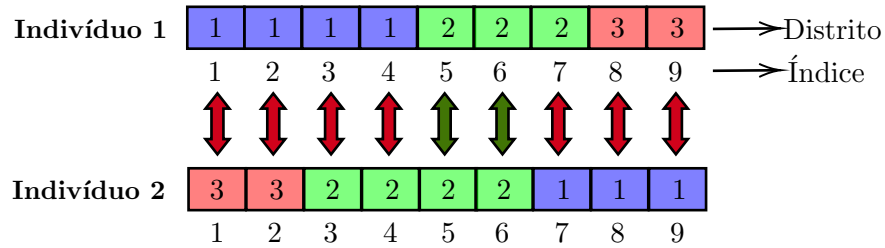


Figura 4.9: Calculando a similaridade entre dois indivíduos pela distância de Hamming.

A distância de Hamming inicialmente foi utilizada para ajustar a probabilidade do ponto de corte de acordo com a similaridade dos indivíduos. Quanto maior a diferença entre progenitores, menor a aleatoriedade na escolha deste ponto, *e.g.* se o segundo progenitor partilha 30% de alelos de distritos com o primeiro progenitor, então isto significa que o ponto de corte pode ocorrer a partir dos 70% do cromossomo. Este procedimento é feito desta forma porque quanto menos semelhantes mais o primeiro progenitor colabora com sua informação genética, facilitando a obtenção de um descendente viável. Após serem realizados diversos testes percebeu-se que a alteração no operador de *crossover* proposta através da distância de *Hamming* não trouxe melhorias significativas

na viabilidade dos filhos gerados. Adicionalmente, verificou-se ainda um aumento do tempo de execução.

4.6.2 Similaridade Geométrica

Este operador de *crossover* é baseado na similaridade geométrica de acordo com a posição dos centroides dos distritos de cada progenitor. Neste caso, é preciso considerar que, em soluções distintas pode-se ter distritos rotulados de forma diferentes, mas que representam posições geográficas semelhantes. Torna-se assim interessante comparar a similaridade de dois indivíduos através do cálculo das diferenças entre os rótulos dos distritos e a possível atualização dos mesmos.

Esta medida de similaridade considera os centroides dos distritos. O centroide de um polígono consiste no centro geométrico baseado em suas coordenadas (Deakin et al., 2002). Uma vez calculados os centroides de cada distrito é possível atualizar os rótulos dos distritos do segundo indivíduo. Esta atualização é feita com base na disposição geométrica dos centroides dos distritos do primeiro indivíduo comparado ao segundo. Para realizar o *crossover* de um ponto baseado na similaridade geométrica é necessário:

1. Encontrar os centroides de cada distrito para ambos indivíduos ($2p$ centroides);
2. Calcular a distância entre os centroides do primeiro indivíduo em relação ao segundo;
3. Encontrar o par de centroides de menor distância entre os dois indivíduos. Atribuir um novo rótulo de distrito no segundo indivíduo;
4. Repetir o processo até que todos os distritos do segundo indivíduo sejam atualizados.

A Equação (4.6) determina o cálculo do valor do centroide k , sendo $k = 1, \dots, p$, e k um determinado distrito de uma solução, $i \in V_k$ um determinado nó pertencente ao distrito k , (x_i, y_i) as coordenadas do nó i e, $|V_k|$ é a quantidade de nós presente no distrito k .

$$C_k = \left(\sum_{i \in V_k} \frac{x_i}{|V_k|}, \sum_{i \in V_k} \frac{y_i}{|V_k|} \right) \quad (4.6)$$

A Figura 4.10 apresenta soluções para uma instância com 10 nós e ilustra como o procedimento proposto aproxima os rótulos dos indivíduos selecionados para participarem do cruzamento, a partir das informações dos centroides. A Figura 4.10(a) ilustra o primeiro indivíduo com três distritos representados pelas diferentes cores, sendo C a posição do centroide de cada distrito. A Figura 4.10(b) apresenta o segundo indivíduo com a definição dos três distritos e os respectivos centroides de cada distrito. Observando as Figuras 4.10(a) e 4.10(b) é possível notar a diferença do posicionamento dos rótulos dos distritos conforme a codificação de cores. A Figura 4.10(c)

apresenta o novo segundo indivíduo resultante, com a reclassificação dos rótulos dos distritos e uma maior similaridade com o primeiro indivíduo, propiciando uma operação de *crossover* mais eficiente. O *crossover* aplicado é o de um ponto conforme apresentado na Figura 4.7.

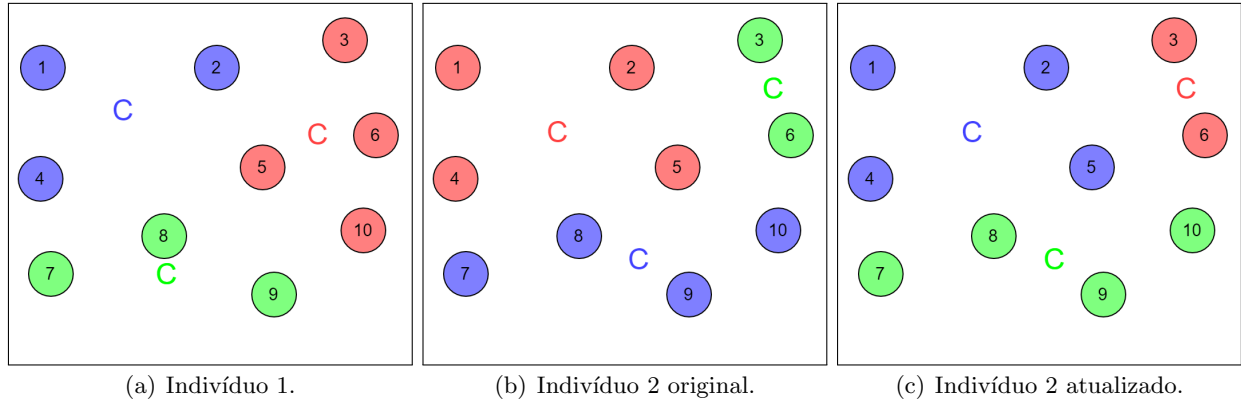


Figura 4.10: Procedimento de atualização dos rótulos dos distritos.

O operador de *crossover* baseado na similaridade geométrica melhorou a performance do AG e a qualidade das soluções. Um ganho considerável deste operador de *crossover* comparado aos dois anteriores (um ponto tradicional e com a distância de Hamming) foi a redução drástica no tempo de execução. Tal fato ocorre devido ao grande custo computacional despendido na factibilização de soluções inviáveis que os outros operadores geram. O tempo de execução médio foi reduzido de 12 minutos para 3,5 minutos, uma redução de quase 71%. Apesar de uma menor incidência, ainda foi possível observar nós que não foram atribuídos a nenhum distrito, na solução obtida após a aplicação deste operador. Para estes nós é executada uma fase de factibilização da solução que consiste em atribuir cada nó ainda não designado ao distrito adjacente com menor soma total de atividade naquele instante (Eq.(4.3)), respeitando as restrições do problema. Após o *crossover* é realizado o cálculo do *fitness*. Apenas dois dos três indivíduos resultantes, os de maior aptidão, permanecem na população.

O Algoritmo 2 apresenta o pseudo-código da operação de *crossover*. I_1 e I_2 são os dois indivíduos selecionados para o cruzamento. A quantidade total de distritos é representada por p , os índices k e j são rótulos de distritos específicos. O método *centroidDistances* recebe dois indivíduos selecionados para participar do *crossover*, encontra os p centróides de cada um e calcula a distância entre eles. O método retorna uma matriz $p \times p$ com as distâncias (linha 1). Com estas informações é possível comparar as distâncias entre os centróides de I_1 com os de I_2 e atualizar os rótulos dos p distritos em I_2 para os valores dos rótulos dos distritos mais próximos em I_1 (linhas 2-8), conforme mostrado na Figura 4.10.

Com os índices dos distritos aproximados nos dois indivíduos é possível iniciar o cruzamento. Primeiro é feito o sorteio de um ponto de corte p_c , que vai de 1 até o tamanho do indivíduo (linha 9).

A partir deste sorteio toda informação do cromossomo de I_1 , da posição zero até p_c , é copiada para o indivíduo filho I_o que está sendo gerado (linhas 10-12). Após o ponto de corte, é copiada a informação genética do segundo indivíduo I_2 até o tamanho final do cromossomo (linhas 13-18). O método *isNotFeasible()* (linha 15) verifica se a informação genética de I_2 violou alguma restrição do problema (Equações (3.4)–(3.9)) no filho I_o e chama o método *repair()* se necessário (linha 16) que inicia o processo de factibilização. A factibilização se faz necessária, pois mesmo com a atualização dos rótulos dos distritos em I_2 , alguns nós podem ficar isolados, por não estarem adjacentes no filho gerado ao mesmo distrito que se encontravam em I_2 . Neste reparo, para cada nó não atribuído são identificados quais os distritos em que estão os respectivos nós adjacentes. Em seguida é realizada a designação de cada nó nos distritos adjacentes, dando prioridade ao distrito com menor valor total de atividade, até que o indivíduo resultante represente uma solução viável para o problema. Finalmente, I_o é retornado.

A complexidade do algoritmo é $O(n)$, considerando $O(p)$ e $O(p_c)$ nas linhas (linhas 2 e 9) com p e p_c referentes a valores baixos e fixos ($O(1)$), restando $O(n)$ do *loop* na linha 13, sendo n o tamanho do indivíduo.

Algorithm 2 Crossover(I_1, I_2)

```

1:  $distances_{p \times p} \leftarrow centroidDistances(I_1, I_2)$ 
2: for  $k \leftarrow 0$  to  $p$  do
3:   for  $j \leftarrow 0$  to  $p$  do
4:     if  $distance_{I_2}^k < distance_{I_1}^j$  then
5:        $I_2^k \leftarrow I_1^j$ 
6:     end if
7:   end for
8: end for
9:  $p_c \leftarrow Random(1, n)$ 
10: for  $i \leftarrow 0$  to  $p_c$  do
11:    $I_o[i] \leftarrow I_1[i]$ 
12: end for
13: for  $i \leftarrow p_c + 1$  to  $n$  do
14:    $I_o[i] \leftarrow I_2[i]$ 
15:   if isNotFeasible( $I_o$ ) then
16:     repair( $I_o$ )
17:   end if
18: end for
19: return  $I_o$ 

```

Para a população estruturada, considerando que esta possui apenas 13 indivíduos, o operador de *crossover* gera um conjunto de 12 filhos. Estes novos indivíduos são comparados aos progenitores subordinados e quando superiores, ocupam seus lugares na população. No entanto, os mesmos indivíduos são descartados quando representam soluções inferiores. A cada geração é necessário

atualizar o ordenamento da população para a manutenção da hierarquia, através do algoritmo *heapsort*, considerando que um indivíduo líder sempre tem melhor *fitness* que seus subordinado.

4.7 Mutação

Com objetivo de introduzir diversidade na população a mutação é realizada nos filhos gerados após a operação de *crossover*. O operador de mutação escolhido para este problema segue o conceito estabelecido em Lopes et al. (2013) que determina que a taxa de mutação m_r é aplicada sobre os genes dos indivíduos com base em um número aleatório entre 0 e 1 e para cada um dos genes compara-se com a taxa de mutação. Caso este valor seja maior que a taxa, realiza-se a mutação.

Essas mudanças reintroduzem diversidade genética na população e auxiliam o algoritmo a fugir de mínimos locais. No AG proposto, quando um indivíduo sofre mutação, um determinado nó pode ser designado a um novo distrito. Para assegurar que o indivíduo permaneça viável após a mutação apenas nós de fronteira entre distritos distintos podem ser mutados.

De acordo com de Assis et al. (2014), um nó i do grafo é um nó de fronteira quando este pertence a um distrito k e é adjacente a um nó j que pertence a um distrito l , sendo $i \neq j$ e $k \neq l$. A Figura 4.11 apresenta um grafo com 15 nós e uma solução para o PDC com 3 distritos. Os nós de fronteira, para tal solução, encontram-se destacados. Para garantir que o distrito de origem do nó também permanecerá conexo após sua remoção, para cada mutação de um gene é executada uma Busca em Largura (BL) neste distrito. Caso a BL alcance todos os demais nós deste distrito, o nó de fronteira pode ser removido sem violar a restrição de territórios conexos (conforme Equação (3.5)).

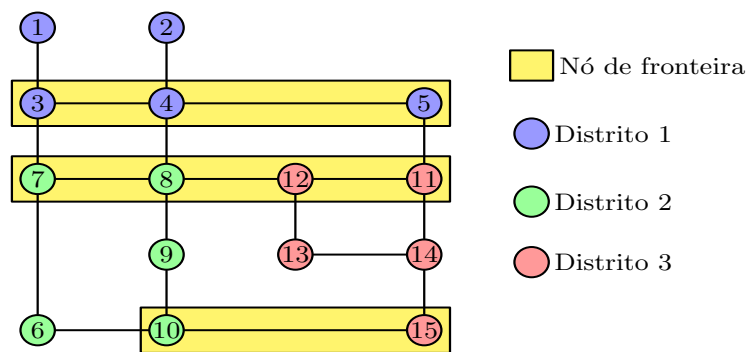


Figura 4.11: Exemplo de solução com os nós de fronteira destacados.

O pseudo-código da operação de mutação é apresentado no Algoritmo 3. I_o é o filho resultante do *crossover*, m_r é a probabilidade de mutação e n é o tamanho do cromossomo. O processo inicia com uma estrutura de repetição que percorre o indivíduo filho (linha 1). Para cada posição do cromossomo, é verificada a probabilidade do respectivo alelo ser mutado (linha 2). Se esse gene do cromossomo for selecionado para sofrer a mutação, duas premissas devem ser verdadeiras: (i)

o nó selecionado deve ser um nó de fronteira na solução atual; e (ii) a mudança de distrito para esse nó, não deve violar a restrição de distritos conexos e manter a contiguidade, o que é verificado segundo resultado da busca em largura (linha 4). Assim, se ambas premissas forem verdadeiras, muda-se o nó do distrito k para o distrito l (linha 5), e os nós de fronteira são atualizados (linha 6). Finalmente, o filho é retornado após a conclusão de todo o processo de mutação.

A complexidade do algoritmo é $O(n)$, considerando o *loop* (linha 1) que percorre todas as posições dos indivíduo (n).

Algorithm 3 Mutation(I_o, m_r, n)

```

1: for  $i \leftarrow 1$  to  $n$  do
2:    $R \leftarrow \text{Random}(0, 1)$ 
3:   if  $R > m_r$  then
4:     if  $\text{isFrontier}(I_o^i)$  and  $BL(I_o^{k_1})$  then
5:        $\text{changeDistrict}(I_o, i, k, l)$ 
6:        $\text{updateFrontiers}(I_o)$ 
7:     end if
8:   end if
9: end for
10: return  $I_o$ 

```

4.8 Algoritmo Genético

O Algoritmo 4 resume as etapas descritas nas seções anteriores proporcionando uma visão geral do AG proposto. Os parâmetros de entrada do AG são $m_r, s_r, p_{size}, g_{max}$ que controlam respectivamente a taxa de mutação, taxa de seleção e tamanho da população (somente na população não estruturada) e o número máximo de gerações. Primeiramente, uma população inicial é gerada através da heurística construtiva apresentada na Seção 4.4 (linha 1). Em seguida, o *fitness* de cada indivíduo é calculado, os indivíduos da população são ordenados pelo valor do *fitness* e o melhor indivíduo é armazenado (linhas 3-4). Para cada geração do AG (linhas 5-25), s_r filhos serão gerados pelo operador de seleção (Seção 4.5) para participar do *crossover* (Seção 4.6) (linha 8). Com probabilidade m_r , cada posição do filho gerado pode sofrer mutação em seu código genético (linha 9). O *fitness* do filho é calculado e os dois indivíduos de maior aptidão permanecem na população na próxima geração (linhas 11-19). Ao final de cada geração, a população é ordenada e o melhor indivíduo é atualizado. Finalmente, após todas as gerações, o indivíduo de maior *fitness* é retornado.

A complexidade do algoritmo é $O(n \log n)$, considerando que a população inicial (linha 1) tem complexidade $O(n)$, e que os *loops* das linhas 5 e 6 tem complexidade $O(1)$ por g_{max} e s_r serem referentes a valores baixos e fixos, resta o método *sort* (linha 19) que executa o algoritmo *quicksort* e tem complexidade de $O(n \log n)$, sendo a operação dominante.

Algorithm 4 - AlgoritmoGenetico($m_r, s_r, psize, gmax$)

```

1: population  $\leftarrow$  initialPopulation( $G(V, E)$ )
2: calculateFitness(population)
3: sort(population)
4: bestIndividual  $\leftarrow$  population[0]
5: for (gen  $\leftarrow$  0 to  $gmax$ ) do
6:   for (sel  $\leftarrow$  0 to  $s_r$ ) do
7:     ( $I_1, I_2$ )  $\leftarrow$  selection(Population)
8:      $I_o \leftarrow$  crossover( $I_1, I_2$ )
9:      $I_o \leftarrow$  mutation( $I_o$ )
10:    calculateFitness( $I_o$ )
11:    if (fitness( $I_1$ ) > fitness( $I_2$ )) then
12:      if (fitness( $I_o$ ) > fitness( $I_2$ )) then
13:         $I_2 \leftarrow I_o$ 
14:      end if
15:    else if (fitness( $I_2$ ) >= fitness( $I_1$ )) then
16:      if (fitness( $I_o$ ) > fitness( $I_1$ )) then
17:         $I_1 \leftarrow I_o$ 
18:      end if
19:    end if
20:    sort(population)
21:    if (fitness(bestIndividual) < fitness(population[0])) then
22:      bestIndividual  $\leftarrow$  population[0]
23:    end if
24:  end for
25: end for
26: return bestIndividual

```

4.8.1 Reset

Uma alteração no Algoritmo 4 foi realizada incluindo um mecanismo de reinício (*reset*). O objetivo é inserir mais uma técnica de diversidade a fim de evitar que o AG fique preso em mínimos locais. Ambas as versões, com e sem o *reset*, são comparadas nos experimentos computacionais realizados na Seção 5.1, com o objetivo de analisar o impacto do mesmo no AG. O mecanismo de *reset* leva em consideração o número consecutivo de gerações onde não foi registado melhoria de *fitness*. Caso esta melhoria não seja verificada em um número predeterminado de vezes então é gerada uma nova população (através da heurística descrita na Seção 4.4) que mantém apenas o melhor indivíduo da população.

Capítulo 5

Resultados

Este Capítulo apresenta os experimentos realizados e respectivos resultados obtidos com a abordagem proposta. Dado que os algoritmos desenvolvidos precisam que sejam definidos um conjunto de hiperparâmetros, foi realizado um estudo de otimização para a definição dos mesmos (Seção 5.1). Esta análise exploratória abrange a população tradicional e a hierarquicamente estruturada. Uma vez definidos os melhores valores dos hiperparâmetros, torna-se possível apresentar os resultados dos experimentos (Seção 5.2). A análise dos resultados também inclui a aplicação da técnica de *reset* e o respectivo impacto. Os experimentos utilizam instâncias geradas aleatoriamente com diferentes tamanhos e características, além de uma instância com características de uma rede real. As especificidades destas instâncias serão apresentadas na Seção 5.2.

5.1 Otimização de Hiperparâmetros

O AG é dependente e sensível a um conjunto de hiperparâmetros (introduzidos no Capítulo 4), os quais afetam diretamente a execução do algoritmo e a qualidade dos resultados. Com o objetivo de obter soluções de qualidade, foi realizada uma análise estatística sobre diferentes combinações de valores dos hiperparâmetros. O espaço de busca foi determinado a partir de valores obtidos em experimentos iniciais e em problemas similares. Como os AGs são métodos inerentemente não determinísticos foi necessário executar cada combinação várias vezes.

Para o processo de otimização dos hiperparâmetros foi utilizada uma instância totalmente conectada (ver Seção 5.2) que possui 512 nós 976 arestas. Esta rede foi utilizada apenas para o processo de calibração de forma a evitar qualquer viés. Cada combinação de valores foi executada 30 vezes a fim de obter maior confiança estatística. Seguidamente, a mediana é calculada para cada combinação. Para determinar qual a configuração de valores é a mais adequada, são selecionadas as 5 melhores combinações de hiperparâmetros, por não serem estatisticamente diferentes entre si.

Optou-se então escolher a combinação com menor variação, o que pode ser feito através da distância interquartil. Estes resultados serão apresentados nas próximas seções. Durante o estudo foi observado que o fator estocástico da heurística construtiva fornece uma leitura pouco precisa do

impacto de cada hiperparâmetro, mesmo após a execução de diversos experimentos. Foi então desenvolvida uma ferramenta para geração de uma mesma população inicial. Esta alteração permitiu uma observação mais precisa do impacto individual de cada hiperparâmetro.

É importante ressaltar que como o problema foi trabalho no contexto de otimização mono-objetivo, conforme a Equação (4.4), em todos os experimentos computacionais foi utilizado o valor $\lambda = 0,5$ para o parâmetro de ponderação das funções, indicando que ambas possuem o mesmo peso, i.e. igual importância.

5.1.1 Algoritmo Genético com População não Estruturada - AGPNE

O procedimento descrito acima foi executado para o AG com a população não estruturada. A Tabela 5.1 apresenta o conjunto de valores que define o espaço de busca dos hiperparâmetros para o estudo com a população não estruturada (AGPNE). O espaço de busca foi definido de acordo com valores encontrados na literatura e testes preliminares.

Tabela 5.1: Conjunto de hiperparâmetros e espaço de busca para o AGPNE.

Hiperparâmetro	Descrição	Espaço de busca
m_r	Taxa de mutação	{0,05; 0,1; 0,2; 0,3; 0,4}
s_r	Taxa de seleção	{0,2; 0,3; 0,4; 0,5}
pop_s	Tamanho da população	{200, 300, 400, 500}
g_{max}	Número máximo de gerações	{300, 400, 500, 600}

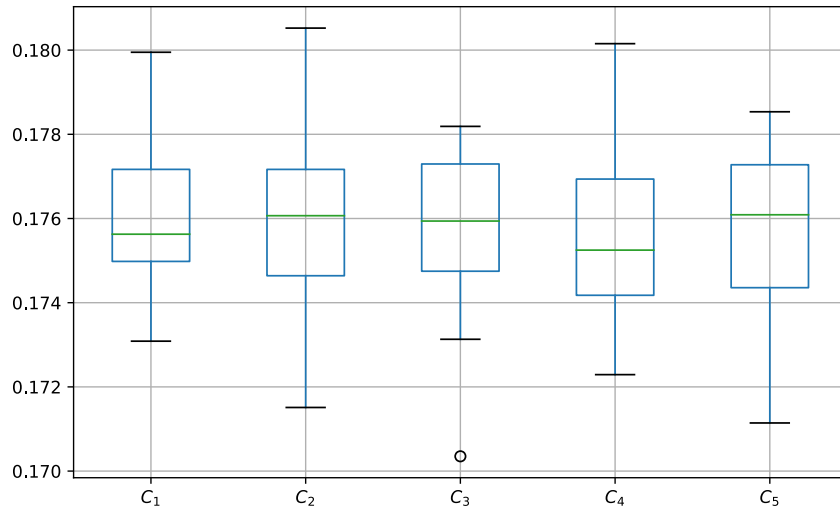
Foram executadas todas as combinações para os valores apresentados, resultando em um espaço de busca total de tamanho $5 \times 4 \times 4 \times 4 = 320$. Adicionalmente, e como referido anteriormente, cada combinação foi executada 30 vezes. A quantidade total de execuções do algoritmo nesta etapa foi de $320 \times 30 = 9600$. O tempo de execução médio foi de 7 minutos para cada execução.

Após a execução destes testes foram selecionadas as 5 combinações das 320 com menor valor de mediana referente ao valor da Função Objetivo (FO). As soluções obtidas são ordenadas, em ordem crescente, pela distância interquartil (DI). A Tabela 5.2 lista as 5 melhores combinações, denominadas de C_i com $i = \{1, \dots, 5\}$, e seus respectivos resultados. A tabela apresenta os valores de mediana da função de compacidade F , da função de homogeneidade G , da FO e o valor da distância interquartil DI , além do tempo médio de execução em segundos.

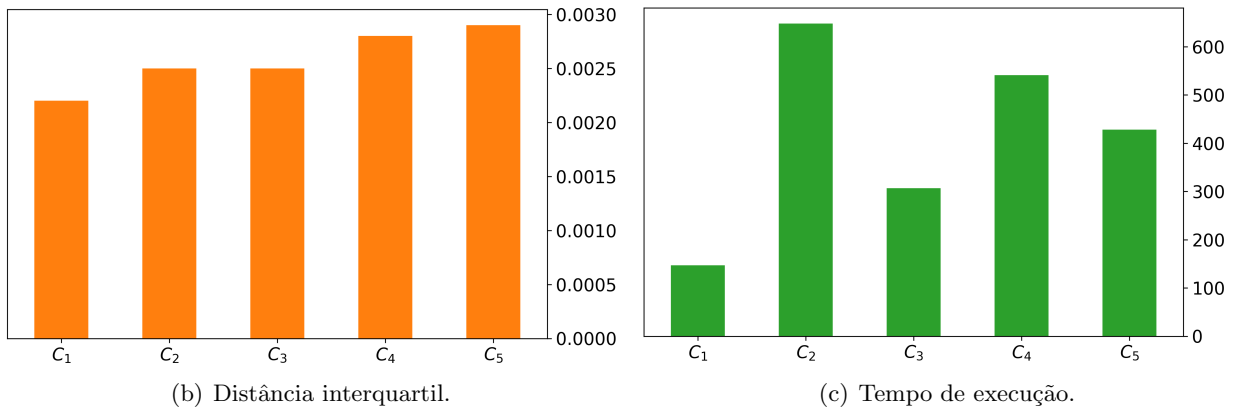
Tabela 5.2: Cinco melhores combinações de hiperparâmetros e resultados para o AGPNE.

C	m_r	s_r	pop_s	g_{max}	F	G	FO	DI	$tempo(s)$
C_1	0,05	0,2	300	500	0,3390	0,0115	0,1756	0,0022	147
C_2	0,05	0,4	500	600	0,3398	0,0119	0,1761	0,0025	648
C_3	0,05	0,2	500	600	0,3395	0,0112	0,1759	0,0025	307
C_4	0,05	0,4	500	500	0,3389	0,0113	0,1752	0,0028	541
C_5	0,05	0,4	400	500	0,3393	0,0116	0,1761	0,0029	428

Para visualização da distribuição das soluções obtidas e apresentadas na Tabela 5.2 foram gerados três gráficos, que estão agrupados na Figura 5.1. O gráfico apresentado na Figura 5.1(a) mostra o *box plot* dos resultados de cada combinação, cujo eixo das ordenadas representa os valores da função objetivo para as soluções obtidas. É possível observar apenas uma ocorrência de *outlier* no gráfico (combinação C_3). A Figura 5.1(b) mostra os valores da distância interquartil de cada combinação e a Figura 5.1(c) apresenta o tempo de execução em segundos para cada combinação.



(a) *Box plot*.



(b) Distância interquartil.

(c) Tempo de execução.

Figura 5.1: Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto C .

Analisando os valores da função objetivo, coluna FO (Tabela 5.2), foi possível observar que todas as cinco combinações C_i possuem valores muito semelhantes variando entre 0,1761 obtido pelas combinações C_2 e C_5 até 0,1752 obtido por C_4 , como também representado no *box-plot* apresentado na Figura 5.1(a). Neste gráfico é possível notar também a similaridade na extensão da dispersão das soluções encontradas. Os resultados apresentados na Figura 5.1(b), exibem a distância interquartil, critério adotado na escolha da melhor combinação de valores. Para os resultados

apresentados na Tabela 5.2 observa-se que a combinação C_1 possui menor valor de $DI = 0,0022$. Reforçando C_1 como combinação ideal para esta etapa dos experimentos, além de menor distância interquartil é possível observar na Figura 5.1(c) um ganho considerável em tempo de execução de C_1 comparado às demais combinações, com total de 147 segundos.

Observando os valores dos hiperparâmetros é possível afirmar que a diferença de tempo entre C_1 e C_2 , por exemplo, é devida aos menores valores de seleção, população e número máximo de gerações empregue por C_1 . Os dados da tabela também revelam que a taxa de mutação em 5% se demonstrou ideal para inserir um pouco de diversidade na população, taxas maiores tiveram melhor performance em testes realizados com o AG sem o *crossover* de similaridade geométrica, porém o fator estocástico excessivo nos outros dois operadores de *crossover* gera uma quantidade grande de filhos infactíveis e apenas atrasa a convergência.

Sobre os valores da taxa de seleção houve uma competição entre as combinações com taxas de 0,2 e de 0,4. A melhor combinação apresenta taxa igual a 0,2, o que possui impacto positivo no tempo de execução e ainda produz um número significativo de novos indivíduos a cada geração. É importante ressaltar que o operador de *crossover* baseado na similaridade geométrica proposto viabilizou taxas mais altas de seleção. Esta situação contrasta com o método original que apresentava melhores resultados com taxas menores. No entanto, os valores obtidos ainda eram piores a nível de FO e o operador padrão de um ponto requeria maior tempo de execução devido a longas fases de factibilização das soluções geradas.

Os tamanhos populacionais ideais para o problema situam-se entre 300 e 500 indivíduos. As combinações com populações maiores normalmente exploram melhor o espaço de soluções e possibilitam um número maior de indivíduos com maior qualidade, que são encontrados inicialmente pela heurística construtiva. Entretanto, quanto maior a população, maior é o impacto no tempo de execução. O valor de 300 indivíduos apresentou um bom compromisso entre a qualidade da solução e o tempo computacional despendido.

A Figura 5.2 apresenta um gráfico de linha da média dos valores de função objetivo por geração das C_i combinações e tem como objetivo determinar se ocorreu convergência em cada combinação. É possível observar o rápido decréscimo inicial e após aproximadamente 300 gerações um desaceleramento. No entanto, não é possível afirmar que em alguma combinação tenha sido alcançado convergência. Considerando o critério adotado, a combinação C_1 apresentou os melhores resultados na primeira etapa dos testes e, conseqüentemente, esta foi a combinação selecionada para a população não estruturada.

Um segundo teste foi realizado aplicando a técnica de *reset* (apresentada na Subseção 4.8.1) considerando o domínio em $\{0, 30, 40, 50, 60, 70, 80\}$. Esses valores representam a quantidade de gerações consecutivas sem melhorias no melhor indivíduo para que um procedimento de reinício

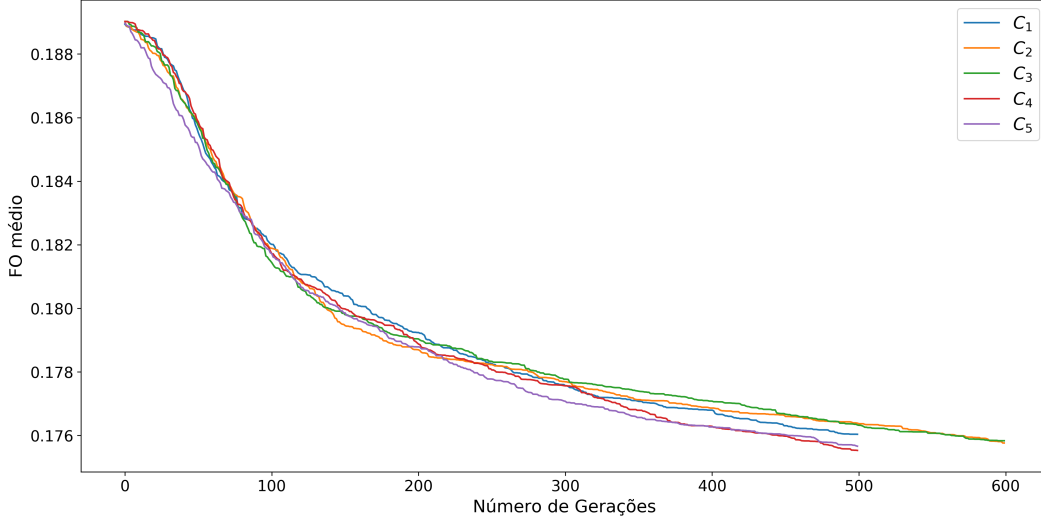


Figura 5.2: Valor médio de FO das combinações de hiperparâmetros C - AGPNE.

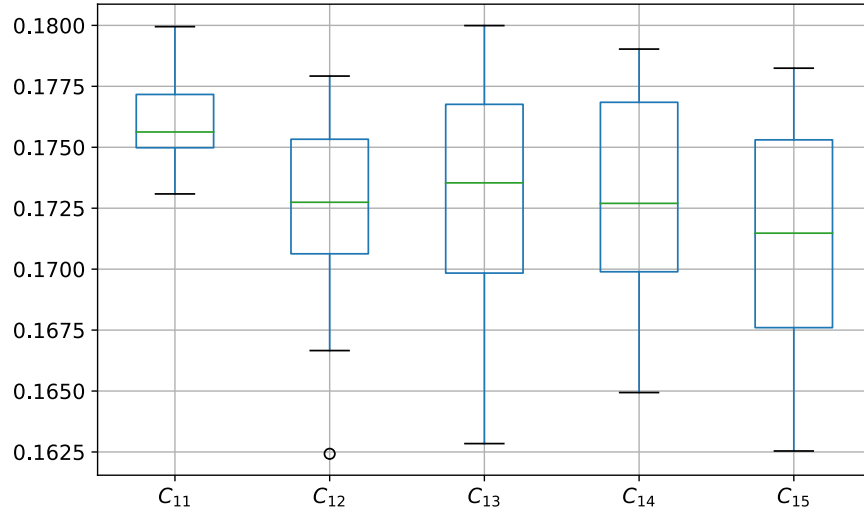
do processo evolutivo seja executado. Nesta etapa o algoritmo foi executado $30 \times 6 = 180$ vezes. O mesmo procedimento de execução e análise foi novamente realizado. Para estes testes as combinações foram denominadas de C_{1i} e os resultados das 5 melhores combinações, $i = \{1, \dots, 5\}$, foram selecionados e podem ser visualizados na Tabela 5.3, novamente ordenados pela distância interquartil.

Tabela 5.3: Cinco melhores combinações de hiperparâmetros e resultados - *reset* - AGPNE.

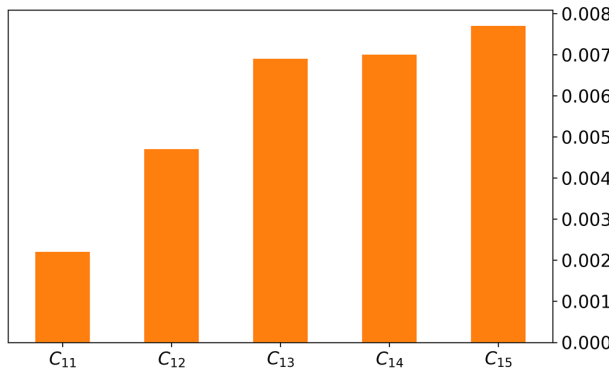
C_1	m_r	s_r	pop_s	g_{max}	$reset$	F	G	FO	DI	$tempo(s)$
C_{11}	0,05	0,2	300	500	0	0,3390	0,0115	0,1756	0,0022	147
C_{12}	0,05	0,2	300	500	60	0,3293	0,0149	0,1727	0,0047	324
C_{13}	0,05	0,2	300	500	50	0,3304	0,0141	0,1735	0,0069	448
C_{14}	0,05	0,2	300	500	40	0,3301	0,0141	0,1727	0,0070	591
C_{15}	0,05	0,2	300	500	30	0,3268	0,0149	0,1715	0,0077	884

A Figura 5.3 apresenta gráficos dos resultados mostrados na Tabela 5.3. A Figura 5.3(a) apresenta o *box plot* dessas 5 combinações e mostra a distribuição das soluções para cada combinação. As Figuras 5.3(b) e 5.3(c) apresentam a DI e o tempo de execução para cada combinação, respectivamente.

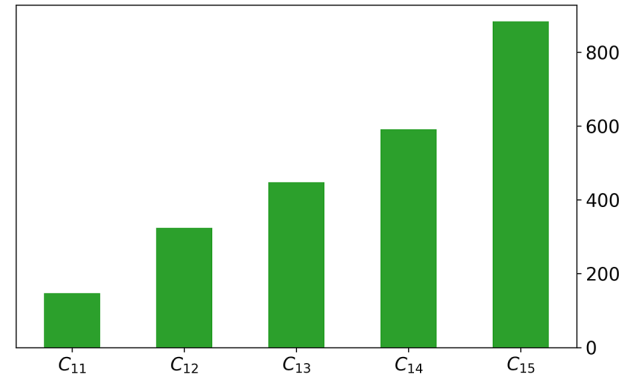
Analisando a Figura 5.3 é possível observar que as cinco melhores combinações são muito similares no valor da mediana da FO . A combinação C_{11} ($reset = 0$) produz o melhor resultado no que diz respeito a valor de DI e a tempo de execução, como pode ser visualizado na Tabela 5.3 e na Figura 5.3(c). Os dados recolhidos também ilustram que quanto maior o valor de $reset$, mais demorada a execução e pior a DI . O aumento temporal deve-se ao tempo de reexecução da heurística construtiva que é computacionalmente custosa.



(a) Box plot.



(b) Distância interquartil.



(c) Tempo de execução.

Figura 5.3: Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto C_1 - *reset*.

Considerando os resultados encontrados nesse experimento, a técnica de *reset* não apresentou ganho significativo, além de ter causado grande impacto no tempo de execução e maior variação (distância interquartil). A melhor combinação C_{11} (equivalente a C_1), sem *reset*, apresentou valor de *FO* igual a 0,1756 e uma *DI* de 0,0022 e um total de 2,45 minutos de execução, enquanto que a segunda melhor combinação C_{12} apresentou *FO* em 0,1727, *DI* = 0,0047 e com tempo de execução em 5,4 minutos. A partir desta análise foi possível determinar que C_{11} foi a melhor combinação de valores para os hiperparâmetros considerando o espaço de busca e o AG apresentado com a população não estruturada.

5.1.2 Algoritmo Genético com População Estruturada - AGPE

Pela definição da população estruturada utilizada nesta dissertação (ver Seção 4.5.2) existem apenas 13 indivíduos em uma mesma população. Como o tamanho da população é fixo, isso implica que não é necessário analisar a variação deste parâmetro. O AGPE utiliza o crossover determinístico tornando desnecessário a avaliação da taxa de seleção. A Tabela 5.4 mostra o espaço de busca para os valores dos hiperparâmetros do AG com população estruturada.

Tabela 5.4: Conjunto de hiperparâmetros e espaço de busca para o AGPE.

Hiperparâmetro	Descrição	Espaço de busca
m_r	Taxa de mutação	$\{0,05; 0,1; 0,2; 0,3; 0,4\}$
g_{max}	Número máximo de gerações	$\{300, 400, 500, 600, 700, 800, 900, 1000\}$
$reset$	Reinícios por gerações sem melhorias	$\{0, 30, 40, 50, 60, 70, 80\}$

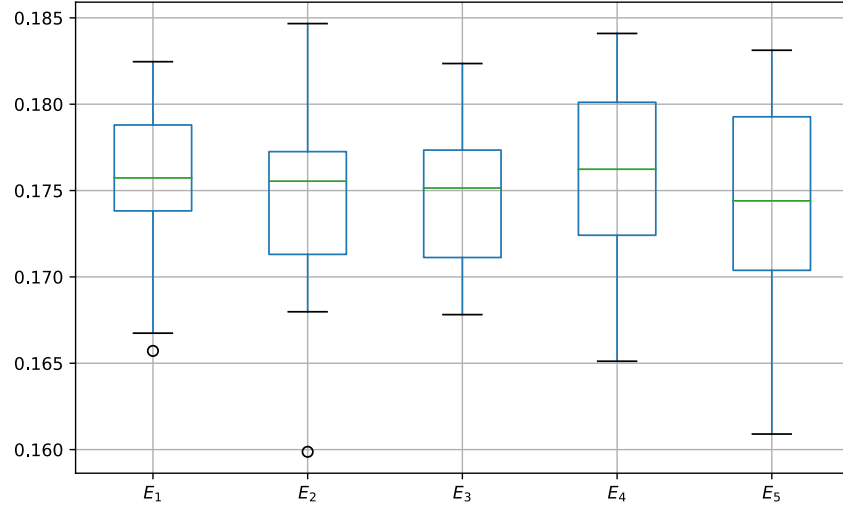
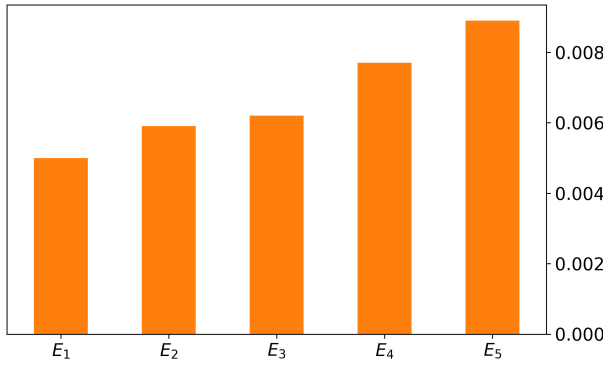
Para o AGPE também foram executadas todas as combinações de hiperparâmetros resultando em um espaço de busca de tamanho $5 \times 8 \times 7 = 280$. Como realizado anteriormente, cada combinação foi executada 30 vezes totalizando em 8400 execuções do algoritmo. Cada execução teve duração média de 2,5 minutos. A Tabela 5.5 mostra as 5 melhores combinações encontradas neste estudo, denominadas de E_i com $i = \{1, \dots, 5\}$ e novamente ordenadas pela DI.

Tabela 5.5: Cinco melhores combinações de hiperparâmetros e resultados para o AGPE.

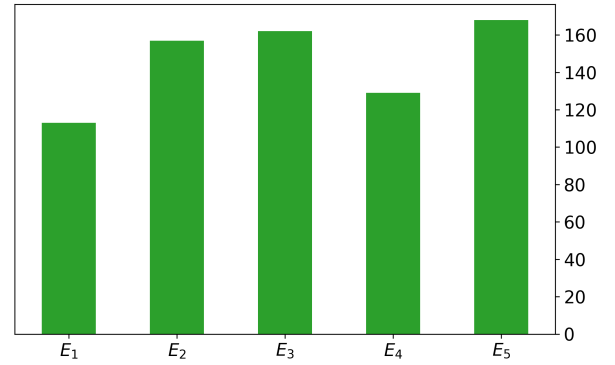
E	m_r	g_{max}	$reset$	F	G	FO	DI	$tempo(s)$
E_1	0,05	1000	50	0,3315	0,0169	0,1757	0,0050	113
E_2	0,05	1000	30	0,3283	0,0188	0,1755	0,0059	157
E_3	0,01	1000	30	0,3324	0,0166	0,1751	0,0062	162
E_4	0,01	800	30	0,3328	0,0191	0,1762	0,0077	129
E_5	0,02	1000	30	0,3316	0,0179	0,1744	0,0089	168

As melhores combinações encontradas E_1 e E_2 apresentam taxa de mutação igual a 0,05 (mesmo valor do AGPNE), 1000 gerações e diferem apenas no valor de $reset$ de 50 e 30. Nas demais combinações é possível observar um aumento gradual da mutação até 0,2 e a predominância de $reset = 30$ e de 1000 gerações. A única exceção foi a combinação E_4 que executou por 800 gerações. A Figura 5.4 apresenta o *box plot* onde pode-se observar a distribuição das soluções para as 5 melhores combinações e os gráficos com a distância interquartil e tempo de execução para cada combinação.

Apesar dos 5 melhores resultados apresentarem baixa variação no valor da mediana da FO , conforme mostrado na Figura 5.4(a), as soluções apresentam uma diferença mais significativa na dispersão das soluções encontradas, como pode ser observado através do valor da DI e das Figuras 5.4(a) e 5.4(b). No entanto, os experimentos possuem tempos de execução mais semelhantes do que acontecia com a população não estruturada. É possível observar que a combinação E_1 apresenta

(a) *Box plot.*

(b) Distância interquartil.



(c) Tempo de execução.

Figura 5.4: Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto E .

um valor de FO muito semelhante às outras combinações, porém com o menor tempo de execução e com menor variação (DI). A Figura 5.5 exibe um gráfico com a evolução média das soluções por geração para cada uma das 5 melhores combinações (Tabela 5.5). É importante ressaltar que a combinação E_4 foi executada por apenas 800 gerações e por este motivo sua execução termina antes das demais.

É importante salientar que a combinação E_1 utiliza o limite superior de número de gerações. Por esse motivo foram realizados experimentos adicionais variando o número de gerações em $g_{max} = \{1000, 1100, 1200, 1300, 1400, 1500\}$ e fixando os demais parâmetros em E_1 . Os mesmos critérios de execução e classificação foram utilizados. Os resultados das 5 melhores combinações para este experimento são denominados de E_{1i} com $i = \{1, \dots, 5\}$ e apresentados na Tabela 5.6. Esta expansão do número máximo de gerações não alterou a melhor combinação. Apesar dos resultados das combinações E_{11} até E_{14} não se diferenciarem muito em relação à dispersão interquartil, o

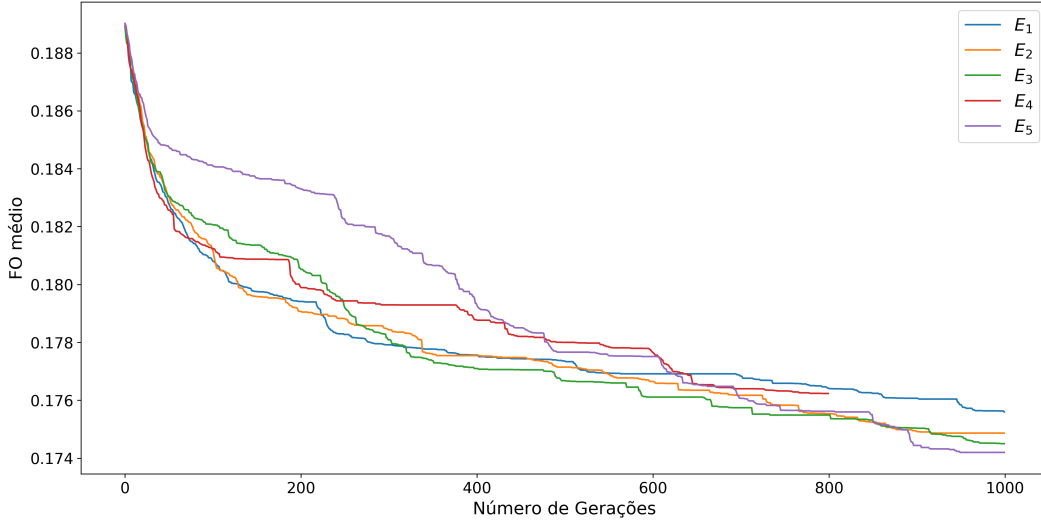


Figura 5.5: Valor médio de FO das combinação de hiperparâmetros E - AGPE.

que pode ser observado através das Figuras 5.6(a) e 5.6(b) pode-se notar que a combinação que apresenta menor variação é a E_{11} que é a melhor combinação já obtida previamente. A Figura 5.6(c) ilustra ainda a diferença no tempo de execução.

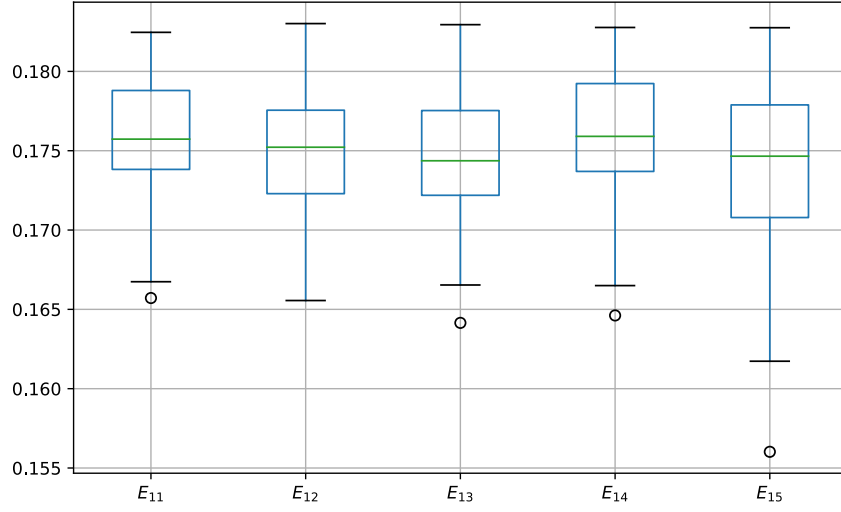
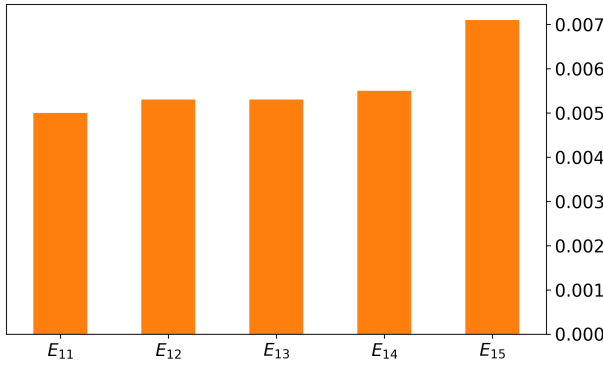
Tabela 5.6: Cinco melhores combinações de hiperparâmetros e seus resultados.

E_1	m_r	g_{max}	$reset$	F	G	FO	DI	$tempo(s)$
E_{11}	0,05	1000	50	0,3315	0,0169	0,1757	0,0050	113
E_{12}	0,05	1300	50	0,3347	0,0159	0,1752	0,0053	169
E_{13}	0,05	1200	50	0,3338	0,0166	0,1744	0,0053	139
E_{14}	0,05	1500	50	0,3349	0,0175	0,1759	0,0055	201
E_{15}	0,05	1400	50	0,3337	0,0146	0,1747	0,0071	179

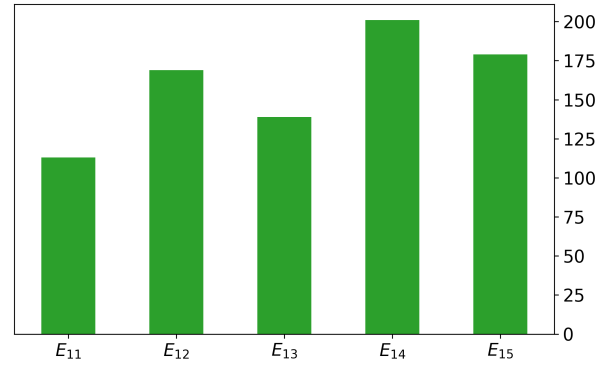
Por meio da análise dos resultados apresentados, é possível observar a proximidade nos valores obtidos de DI que variaram entre 0,0050 e 0,0071, e o aumento no tempo de execução conforme o número de gerações cresce. Como, pelos critérios adotados, E_1 teve o menor valor de DI entre as combinações avaliadas, logo tais valores serão então utilizados para a execução do AG com a população estruturada.

5.2 Experimentos Computacionais

Esta seção apresenta os experimentos computacionais realizados para resolver o PDC abordado. A descrição das instâncias utilizadas nos testes, a evolução de cada algoritmo a partir da solução inicial e a comparação de ambos algoritmos também são apresentados e analisados. Os experimentos foram realizados em computador com processador *Intel Core i7 7700HQ* com 16 GB de memória RAM, os algoritmos foram implementados na linguagem C++ com o Visual Studio 2019

(a) *Box plot.*

(b) Distância interquartil.



(c) Tempo de execução.

Figura 5.6: Resultados gráficos das 5 melhores combinações de hiperparâmetros do conjunto E_1 - extensão do g_{max} .

versão 16.3.5 no sistema operacional Windows 10.

5.2.1 Descrição das Instâncias

Nesta dissertação são apresentados resultados para 24 instâncias geradas aleatoriamente e uma instância real, conforme utilizadas em de Assis et al. (2014). As instâncias estão disponíveis em repositório público¹ e se encontram divididas em 24 pastas, ou categorias. As características presentes nas instâncias de cada categoria, são classificadas seguindo a nomenclatura subsequente.

- **Grupo:** (A) com 512 nós e $p = 10$ distritos, (B) com 1024 nós e $p = 20$ distritos;
- **Família:** Atividade 1 (distribuição do tempo de leitura) (1) concentrada, valores aleatórios no intervalo $[16, 24]$, (2) esparsa, valores aleatórios no intervalo $[12, 28]$;

¹Disponível em <https://github.com/lauraassis/CDP>.

- **Classe:** Atividade 2 (distribuição do número de clientes) (1) concentrada, valores aleatórios no intervalo $[160, 240]$, (2) esparsa, valores aleatórios no intervalo $[120, 280]$;
- **Esparsidade:** Relativa a esparsidade da rede: totalmente conectada (00), parcialmente esparsa (75) e minimamente conectada (50).

Dessa forma, uma instância, denominada $A1100a$, representa a primeira de dez instâncias que possuem 512 nós, 976 arestas, tempo de leitura no intervalo $[16, 24]$, número de medidores no intervalo $[160, 240]$ e é uma rede totalmente conectada. Como são 10 instâncias de cada categoria, a nomenclatura termina com a variação $\{a \cdots j\}$. Adicionalmente, uma instância denominada RedeSP foi extraída de uma região da cidade de São Paulo, representando uma rede que possui um milhão de clientes residenciais (de Assis et al., 2014). Esta rede possui 1659 nós, 2408 arestas e $p = 20$.

A Tabela 5.7 apresenta o tamanho das redes usadas nos estudos, com o tamanho original da instância (G) e a respectiva dimensão do grafo transformado (G'), conforme o processo de transformação apresentado na Seção 3.2.

Tabela 5.7: Tamanho das instâncias.

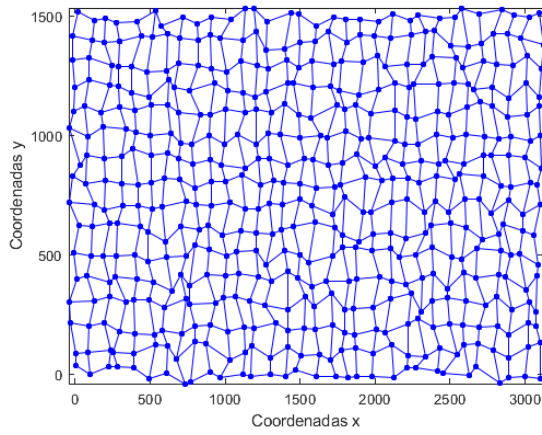
Instância	Nós		Arestas	
	G	G'	G	G'
A1100	512	976	976	2788
A1175	512	732	732	1569
A1150	512	511	511	685
B1100	1024	1984	1984	5764
B1175	1024	1488	1488	3244
B1150	1024	1023	1023	1396
RedeSP	1659	2408	2408	5324

Para uma visualização das características topológicas das instâncias, as Figuras 5.7 e 5.8 exemplificam a representação por grafo para redes de 512 e 1024 nós com as três esparsidades e para a Rede São Paulo, respectivamente.

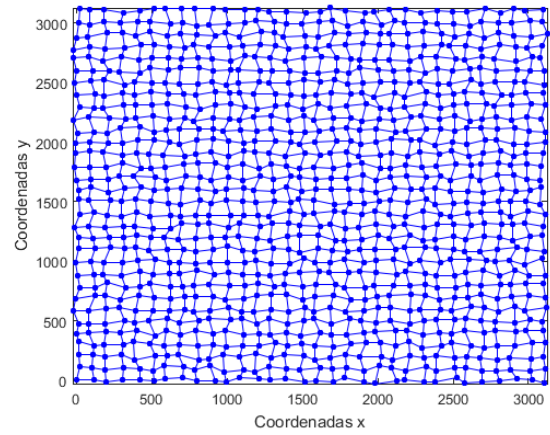
5.2.2 Estudos de Caso

Esta seção apresenta os resultados dos estudos de caso utilizando as redes supracitadas. O mesmo procedimento de validação estatística utilizado na otimização dos hiperparâmetros foi realizado para cada estudo desta seção. Cada instância foi executada 30 vezes. O valor médio dos resultados encontrados é exibido em cada estudo realizado. O objetivo é comparar e analisar a qualidade das soluções, bem como, o tempo de execução.

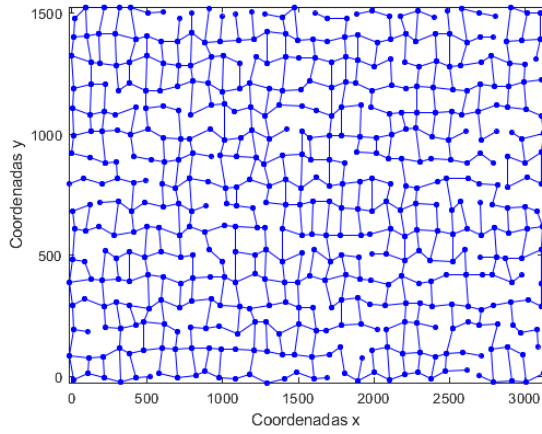
A Tabela 5.8 mostra os valores dos hiperparâmetros que foram utilizados para a execução dos resultados. A combinação C_{11} foi utilizada para o AG com população não estruturada (AGPNE)



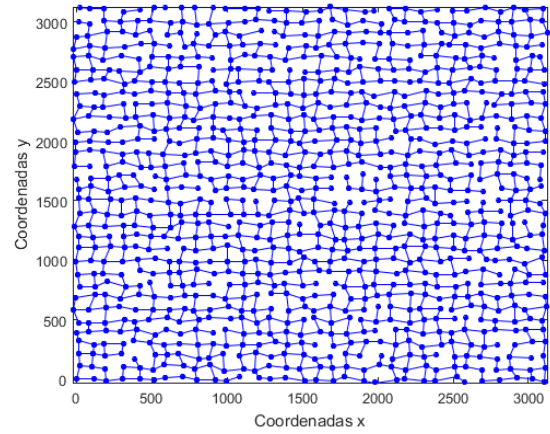
(a) Instância A1100 de 512 nós totalmente conectada.



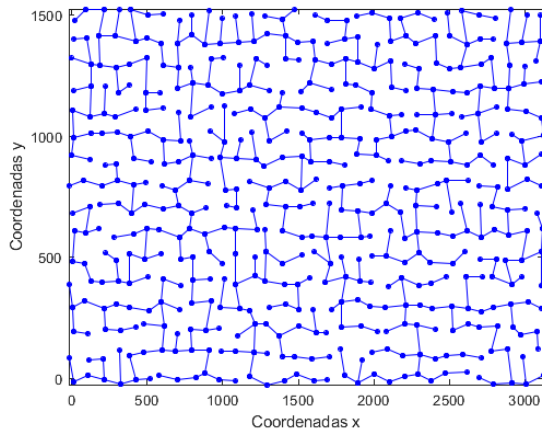
(b) Instância B1100 de 1024 nós totalmente conectada.



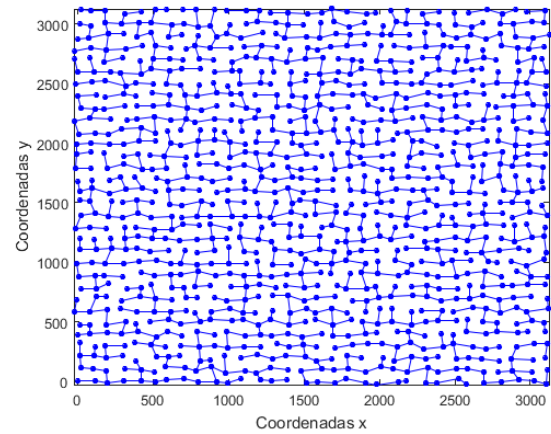
(c) Instância A1175 de 512 nós parcialmente esparsa.



(d) Instância B1175 de 1024 nós parcialmente esparsa.



(e) Instância A1150 de 512 nós minimamente conectada.



(f) Instância B1150 de 1024 nós minimamente conectada.

Figura 5.7: Exemplos da topologia de instâncias com diferentes esparsidades.

e a combinação E_{11} para o AG com população estruturada (AGPE).

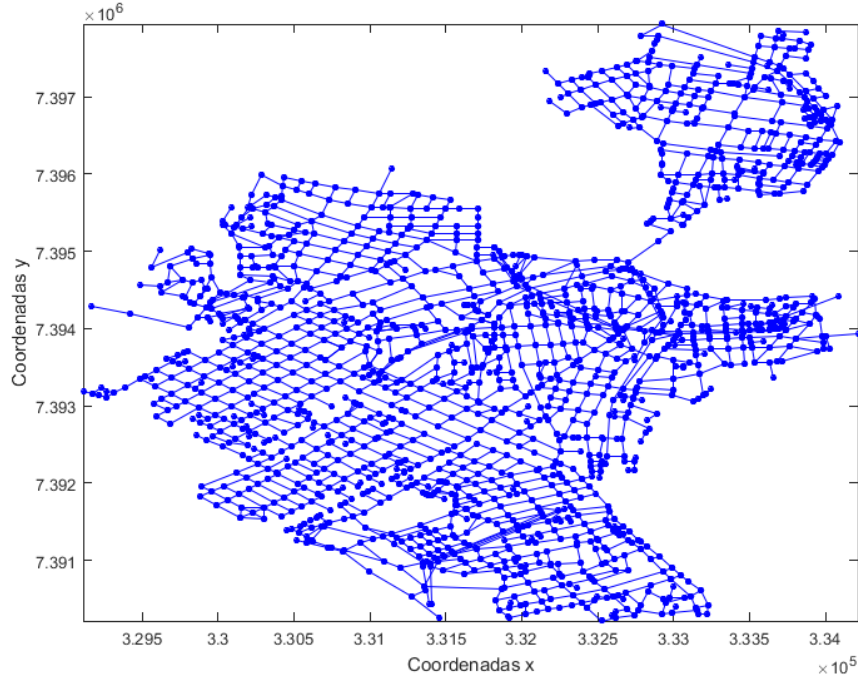


Figura 5.8: Instância RedeSP de 1659 nós.

Tabela 5.8: Conjunto de Hiperparâmetros C_{11} e E_{11} .

Combinação	m_r (%)	s_r (%)	pop_s	g_{max}	$reset$
C_{11}	0,05	0,2	300	500	0
E_{11}	0,05	1	13	1000	50

Estudo de Caso 1: Evolução do AGPNE

Os resultados do AGPNE são apresentados na Tabela 5.9. A tabela mostra a referência para instância de teste, os valores para compacidade (coluna F), homogeneidade (coluna G), função objetivo (coluna FO) e tempo de execução em segundos. Tais resultados mostram a evolução da metaheurística a partir da média dos valores obtidos pela heurística de geração da população inicial em relação à média dos valores obtidos pelo AGPNE ao final de sua execução.

A heurística de geração da população inicial, comum nos dois algoritmos, concebe indivíduos de boa compacidade, por realizar uma distribuição equilibrada dos nós centrais (primeiro nó atribuído em cada distrito) que estão localizados a uma distância mínima pré-determinada. As soluções iniciais também têm boa homogeneidade porque os distritos são criados simultaneamente e nós são adicionados conforme uma tolerância de capacidade das atividades. A conjunção destes fatos significa que existem indivíduos iniciais com qualidade, o que torna mais difícil que sejam obtidas grandes melhorias. Este fato se reflete nestes resultados que apresentaram em média, uma melhoria de 4,91% no valor da FO , com um destaque para a melhoria na homogeneidade G das soluções em média de 11,6% e uma melhora de apenas 1,48% na compacidade F . Um destaque foi a instância

Tabela 5.9: Evolução do AGPNE.

Instância	Heurística Construtiva			AGPNE			Evolução		
	F	G	FO	F	G	FO	$\Delta F\%$	$\Delta G\%$	$\Delta FO\%$
A1100	0,3425	0,0343	0,1884	0,3346	0,0170	0,1758	2,30	50,56	6,69
A1175	0,3657	0,0647	0,2152	0,3485	0,0206	0,1846	4,72	68,09	14,25
A1150	0,3500	0,2828	0,3164	0,3469	0,2425	0,2947	0,89	14,27	6,87
A1200	0,3424	0,0370	0,1897	0,3309	0,0225	0,1767	3,35	39,20	6,85
A1275	0,3677	0,0669	0,2173	0,3530	0,0241	0,1885	4,00	64,01	13,24
A1250	0,3505	0,2841	0,3173	0,3519	0,2489	0,3004	0,39	12,40	5,33
A2100	0,3428	0,0384	0,1906	0,3381	0,0194	0,1788	1,36	49,42	6,20
A2175	0,3632	0,0671	0,2152	0,3474	0,0277	0,1875	4,35	58,78	12,84
A2150	0,3505	0,2831	0,3168	0,3480	0,2492	0,2986	0,71	11,98	5,75
A2200	0,3355	0,0461	0,1908	0,3295	0,0278	0,1787	1,79	39,80	6,39
A2275	0,3638	0,0698	0,2168	0,3523	0,0299	0,1911	3,15	57,22	11,86
A2250	0,3543	0,2804	0,3173	0,3522	0,2337	0,2930	0,58	16,64	7,68
B1100	0,2678	0,0548	0,1613	0,2648	0,0569	0,1608	1,12	3,80	0,28
B1175	0,2887	0,0874	0,1881	0,2884	0,0853	0,1868	0,12	2,41	0,65
B1150	0,2946	0,4368	0,3657	0,2942	0,4207	0,3574	0,12	3,70	2,26
B1200	0,2677	0,0595	0,1636	0,2679	0,0574	0,1627	0,09	3,52	0,57
B1275	0,2869	0,0947	0,1908	0,2850	0,0930	0,1890	0,67	1,80	0,95
B1250	0,2970	0,4455	0,3712	0,2932	0,4308	0,3620	1,27	3,29	2,48
B2100	0,2692	0,0583	0,1637	0,2688	0,0569	0,1628	0,14	2,42	0,55
B2175	0,2891	0,0872	0,1881	0,2868	0,0861	0,1864	0,80	1,24	0,90
B2150	0,2987	0,4387	0,3687	0,2968	0,4261	0,3615	0,63	2,87	1,96
B2200	0,2667	0,0654	0,1661	0,2669	0,0632	0,1651	0,06	3,38	0,62
B2275	0,2891	0,0926	0,1908	0,2863	0,0928	0,1895	0,99	0,27	0,68
B2250	0,2957	0,4475	0,3716	0,2923	0,4279	0,3601	1,14	4,40	3,10
RedeSP	0,2320	0,1142	0,1731	0,2306	0,1090	0,1698	0,59	4,57	1,90
Média total	0,3149	0,1615	0,2382	0,3102	0,1428	0,2265	1,48	11,60	4,91

A1175 que teve 14,25% de melhora na FO .

O AG com população não estruturada obteve performance pior nas instâncias maiores e encontrou especial dificuldade em evoluir para a RedeSP, obtendo um ganho de apenas 1,9% no valor da FO comparando as soluções iniciais com as obtidas após o processo evolutivo. É importante ressaltar que por se tratar de valores médios, estes resultados não retratam a amplitude máxima de melhoria do AG, que alcança 24,9% para a população não estruturada. Adicionalmente, para a rede com características reais e de grande porte também encontrou dificuldade semelhante às redes de 1024 nós.

Estudo de Caso 2: Evolução do AGPE

A Tabela 5.10 apresenta os resultados obtidos para o AGPE. A tabela exibe a performance da metaheurística considerando a média dos resultados iniciais obtidos pela heurística de geração da população inicial em relação à média dos valores obtidos por 30 execuções do AGPE.

Tabela 5.10: Evolução do AGPE.

Instância	Heurística Construtiva			AGPE			Evolução		
	F	G	FO	F	G	FO	$\Delta F\%$	$\Delta G\%$	$\Delta FO\%$
A1100	0,3510	0,0387	0,1949	0,3363	0,0201	0,1782	4,17	47,99	8,53
A1175	0,3770	0,1178	0,2474	0,3589	0,0353	0,1971	4,81	70,00	20,33
A1150	0,3565	0,4673	0,4119	0,3532	0,2436	0,2984	0,92	47,87	27,55
A1200	0,3698	0,0514	0,2106	0,3327	0,0312	0,1819	10,03	39,37	13,62
A1275	0,4370	0,0887	0,2628	0,3574	0,0382	0,1978	18,22	56,90	24,74
A1250	0,3516	0,4595	0,4055	0,3506	0,2436	0,2971	0,30	46,98	26,75
A2100	0,3584	0,0684	0,2134	0,3365	0,0274	0,1820	6,10	59,90	14,72
A2175	0,3836	0,1494	0,2665	0,3604	0,0407	0,2005	6,04	72,78	24,75
A2150	0,3571	0,4734	0,4152	0,3533	0,2521	0,3027	1,05	46,75	27,10
A2200	0,3457	0,0499	0,1978	0,3338	0,0346	0,1842	3,45	30,65	6,88
A2275	0,3800	0,0899	0,2350	0,3556	0,0420	0,1988	6,43	53,29	15,40
A2250	0,3580	0,4671	0,4125	0,3543	0,2640	0,3091	1,04	43,48	25,07
B1100	0,2865	0,0928	0,1897	0,2652	0,0524	0,1588	7,45	43,60	16,30
B1175	0,3061	0,0988	0,2025	0,2884	0,0852	0,1868	5,81	13,79	7,76
B1150	0,2941	0,5617	0,4279	0,2940	0,4177	0,3559	0,04	25,64	16,84
B1200	0,2777	0,0826	0,1802	0,2670	0,0574	0,1622	3,87	30,56	9,99
B1275	0,3015	0,1333	0,2174	0,2850	0,0909	0,1880	5,46	31,80	13,53
B1250	0,2928	0,5671	0,4299	0,2986	0,4194	0,3590	2,01	26,04	16,49
B2100	0,2777	0,0863	0,1820	0,2694	0,0548	0,1621	2,99	36,46	10,93
B2175	0,2786	0,1269	0,2028	0,2878	0,0881	0,1879	3,30	30,62	7,32
B2150	0,2955	0,5220	0,4088	0,2965	0,4285	0,3625	0,33	17,91	11,31
B2200	0,2852	0,0920	0,1886	0,2680	0,0591	0,1635	6,05	35,76	13,29
B2275	0,2961	0,1553	0,2257	0,2868	0,0901	0,1885	3,13	41,97	16,49
B2250	0,3010	0,4931	0,3970	0,2941	0,4356	0,3648	2,29	11,66	8,11
RedeSP	0,2340	0,1398	0,1869	0,2355	0,0878	0,1616	0,61	37,20	13,53
Média total	0,3261	0,2269	0,2765	0,3128	0,1456	0,2292	4,09	35,84	17,12

O AGPE, possui uma população com apenas 13 indivíduos. Esta razão pode ser o motivo do algoritmo não conseguir explorar bem, inicialmente, o espaço de soluções. Porém, foi possível constatar que com o evoluir do número das gerações o método obteve um ganho médio de 17,12% nos valores da FO , permitindo alcançar resultados com qualidade próximas ao AGPNE, mesmo com um número tão reduzido de indivíduos. Destacam-se as instâncias de 512 nós com esparsidade 50 que apresentaram ganhos em torno de 20% a 30% no valor da FO . Outro destaque positivo foi a RedeSP que obteve uma evolução de 13,53% no valor médio de FO .

Estudo de Caso 3: Comparação dos Resultados do AGPNE $\times$ AGPE

Esta seção realiza a comparação dos resultados obtidos pelo AGPNE e AGPE para as redes em estudo. A Tabela 5.11 apresenta os resultados do AG para ambas estruturas populacionais. Os melhores valores de FO no comparativo entre os dois algoritmos estão em negrito.

Os resultados apresentados revelam que ambas as estruturas populacionais alcançaram soluções

Tabela 5.11: Resultados: AGPE $\times$ AGPNE.

Instância	AGPE				AGPNE			
	F	G	FO	tempo(s)	F	G	FO	tempo(s)
A1100	0,3363	0,0201	0,1782	117	0,3346	0,0170	0,1758	223
A1150	0,3532	0,2436	0,2984	37	0,3469	0,2425	0,2947	72
A1175	0,3589	0,0353	0,1971	58	0,3485	0,0206	0,1846	116
A1200	0,3327	0,0312	0,1819	117	0,3309	0,0225	0,1767	216
A1250	0,3506	0,2436	0,2971	37	0,3519	0,2489	0,3004	72
A1275	0,3574	0,0382	0,1978	55	0,3530	0,0241	0,1885	119
A2100	0,3365	0,0274	0,1820	110	0,3381	0,0194	0,1788	214
A2150	0,3533	0,2521	0,3027	36	0,3480	0,2492	0,2986	73
A2175	0,3604	0,0407	0,2005	57	0,3474	0,0277	0,1875	114
A2200	0,3338	0,0346	0,1842	117	0,3295	0,0278	0,1787	212
A2250	0,3543	0,2640	0,3091	36	0,3522	0,2337	0,2930	73
A2275	0,3556	0,0420	0,1988	58	0,3523	0,0299	0,1911	117
Média A	0,3486	0,1061	0,2273	69	0,3444	0,0969	0,2207	135
B1100	0,2652	0,0524	0,1588	494	0,2648	0,0569	0,1608	859
B1150	0,2940	0,4177	0,3559	135	0,2942	0,4207	0,3574	265
B1175	0,2884	0,0852	0,1868	230	0,2884	0,0853	0,1868	429
B1200	0,2670	0,0574	0,1622	495	0,2679	0,0574	0,1627	856
B1250	0,2986	0,4194	0,3590	135	0,2932	0,4308	0,3620	267
B1275	0,2850	0,0909	0,1880	228	0,2850	0,0930	0,1890	420
B2100	0,2694	0,0548	0,1621	508	0,2688	0,0569	0,1628	843
B2150	0,2965	0,4285	0,3625	137	0,2968	0,4261	0,3615	259
B2175	0,2878	0,0881	0,1879	230	0,2868	0,0861	0,1864	417
B2200	0,2680	0,0591	0,1635	505	0,2669	0,0632	0,1651	862
B2250	0,2941	0,4356	0,3648	137	0,2923	0,4279	0,3601	252
B2275	0,2868	0,0901	0,1885	228	0,2863	0,0928	0,1895	457
Média B	0,2834	0,1899	0,2367	288	0,2826	0,1914	0,2370	515
RedeSP	0,2355	0,0878	0,1616	898	0,2306	0,1090	0,1698	1458
Média total	0,3128	0,1456	0,2292	208	0,3102	0,1428	0,2265	371

de qualidade com baixo tempo de execução para população estruturada e não estruturada (3,5 e 6,18 minutos em média). Porém existem algumas diferenças nos resultados obtidos que merecem ser ressaltados. Para as instâncias menores (512 nós) o AGPNE alcançou soluções de melhor qualidade em 11 das 12 instâncias do estudo. Esses resultados foram em média 3% superiores considerando o valor da FO , ainda que em um tempo computacional 48,57% maior, considerando uma média de 1,17 minutos do AGPE contra 2,25 minutos do AGPNE.

O comportamento superior do AGPNE não se manteve nas instâncias de tamanho maior (1024 e 1659 nós) onde o AGPE teve uma performance um pouco superior, em 10 das 13 instâncias, ainda que de apenas 0,41% em média. Em relação ao tempo de execução ocorreu um ganho significativo de 4,2 minutos em média ou 42,94% (5,6 minutos contra 9,8 minutos) para o AGPE. Um destaque do AG com população estruturada foi no resultado obtido para a maior instância tratada no escopo

deste trabalho, a RedeSP, que apresentou comparativamente um ganho de 4,83% na FO e uma redução média de 9,3 minutos (ganho de 38,38% no tempo).

Considerando todas as instâncias utilizadas neste experimento, o AG com a população não estruturada apresentou resultados de melhor qualidade em 14 das 25 instâncias, embora o ganho tenha sido de apenas 1,34% em média no valor da FO e com um tempo de execução médio superior de 6,2 minutos contra 3,5 minutos gasto pelo AG com a população estruturada (redução de 43,75%).

5.2.3 Estudo de Caso 4: Análise da Evolução do AG

A Tabela 5.12 tem como objetivo apresentar a extensão da evolução do AG. A tabela apresenta a amplitude máxima do AG considerando a pior solução encontrada pela heurística de geração da população inicial e o melhor resultado encontrado tanto para AGPNE e AGPE.

Tabela 5.12: Amplitude máxima de evolução dos Algoritmos.

Instância	AGPE			AGPNE		
	Heurística Construtiva	AG	$\Delta FO(S)\%$	Heurística Construtiva	AG	$\Delta FO(S)\%$
A1100	0,2077	0,1677	23,84	0,1996	0,1610	23,97
A1150	0,4300	0,2658	61,78	0,3512	0,2610	34,57
A1175	0,2579	0,1788	44,24	0,2298	0,1707	34,60
A1200	0,2249	0,1730	30,00	0,2035	0,1649	23,41
A1250	0,4157	0,2403	72,96	0,3417	0,2708	26,19
A1275	0,2672	0,1836	45,56	0,2377	0,1750	35,86
A2100	0,2279	0,1724	32,22	0,2046	0,1685	21,40
A2150	0,4245	0,2788	52,24	0,3456	0,2400	44,01
A2175	0,2831	0,1878	50,77	0,2287	0,1626	40,64
A2200	0,2037	0,1761	15,72	0,1984	0,1675	18,45
A2250	0,4384	0,2824	55,25	0,3551	0,2594	36,86
A2275	0,2668	0,1742	53,14	0,2287	0,1737	31,63
B1100	0,1918	0,1488	28,91	0,1754	0,1479	18,62
B1150	0,4279	0,3164	35,25	0,3941	0,3049	29,23
B1175	0,2142	0,1731	23,76	0,2010	0,1738	15,62
B1200	0,1843	0,1503	22,62	0,1721	0,1483	16,06
B1250	0,4393	0,3215	36,67	0,3959	0,3435	15,24
B1275	0,2240	0,1739	28,81	0,2120	0,1700	24,67
B2100	0,1845	0,1531	20,49	0,1729	0,1540	12,26
B2150	0,4136	0,3333	24,11	0,3946	0,3277	20,41
B2175	0,2178	0,1770	23,07	0,1997	0,1759	13,55
B2200	0,1910	0,1522	25,53	0,1740	0,1510	15,20
B2250	0,4000	0,3203	24,88	0,3935	0,3286	19,74
B2275	0,2304	0,1744	32,09	0,2025	0,1746	15,93
RedeSP	0,1932	0,1432	34,88	0,1983	0,1550	27,93
Média Total	0,2864	0,2089	37,11	0,2564	0,2053	24,90

Analisando os resultados na Tabela 5.12, é possível observar os valores das melhores soluções para cada instância e sua amplitude máxima de evolução em cada algoritmo. O AGPE teve em média um valor máximo da FO pela heurística construtiva de 0,2864, e o valor médio mínimo da otimização de 0,2087, uma diferença de 37,11%. Já o AG com a população não estruturada

teve em uma média dos valores máximo de FO pela heurística construtiva de 0,2564 e alcançou uma média mínima de 0,2052, uma melhoria de 24,9%. É possível constatar que, apesar de ambas as estruturas populacionais partirem de valores um pouco distantes pela heurística construtiva de 0,2864 e 0,2564, os dois métodos alcançaram valores médios próximos com 0,2087 e 0,2052.

Para avaliar a distribuição empírica dos dados foram desenvolvidos gráficos de *box plot* com os resultados apresentados. As Figuras foram divididas conforme a esparsidade das instâncias e ambos algoritmos (AGPNE e AGPE) são apresentados lado a lado. Para as instâncias de 512 nós os *box plots* são apresentados nas Figuras: 5.9, 5.10, 5.11. Para as instâncias de 1024 nós os *box-plots* são exibidos nas Figuras: 5.12, 5.13 e 5.14. O *box plot* para a RedeSP é apresentado na Figura 5.15. É possível observar nestas Figuras a baixa dispersão das soluções e um número relativamente baixo de *outliers*, o que representa a confiabilidade dos resultados obtidos.

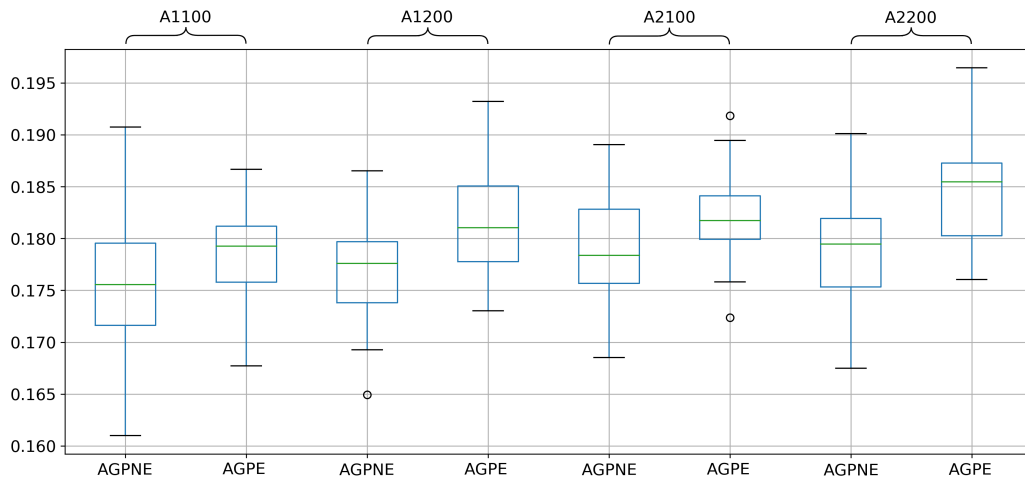


Figura 5.9: Instâncias de 512 nós totalmente conectadas.

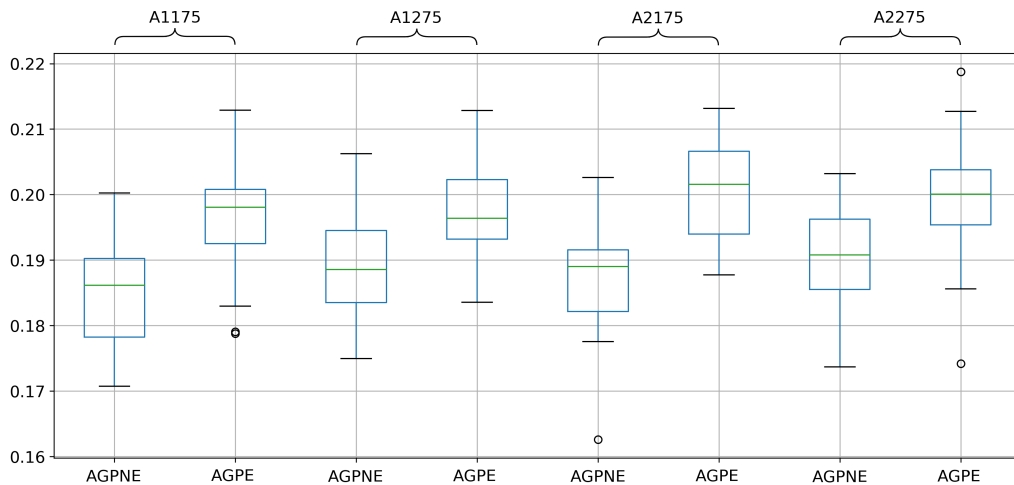


Figura 5.10: Instâncias de 512 nós parcialmente esparsas.

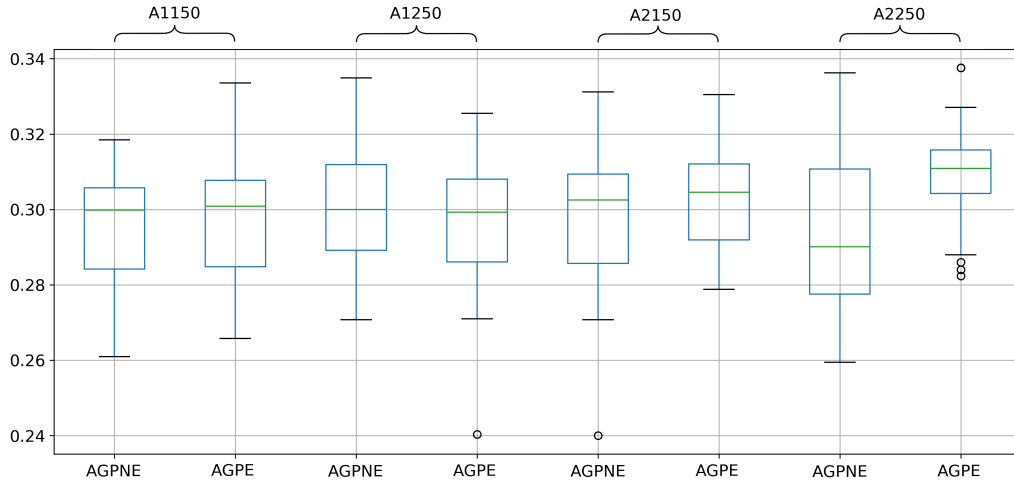


Figura 5.11: Instâncias de 512 nós minimamente conectadas.

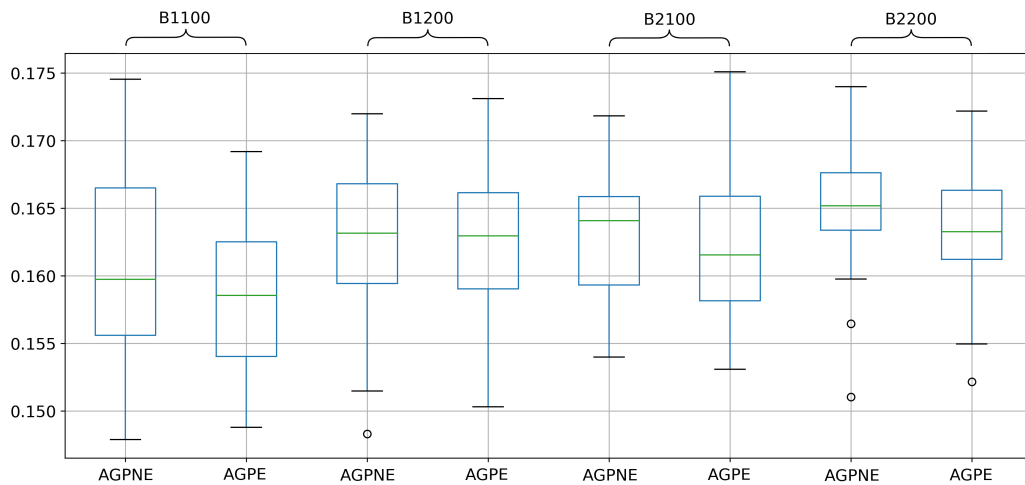


Figura 5.12: Instâncias de 1024 nós totalmente conectadas.

5.2.4 Visualização de Soluções

Esta seção oferece uma visualização gráfica dos resultados do distritamento com a determinação dos distritos. Devido ao grande volume de gráficos e a semelhança dos mesmos foram selecionadas sete instâncias das quais, três com 512 nós, três com 1024 nós e a RedeSP. Cada instância do grupo A e B são de esparsidades distintas. Foram selecionados os melhores resultados independentemente da estrutura populacional aplicada para chegar na solução.

A Figura 5.16 mostra o resultado para seis instâncias pareadas conforme sua esparsidade. As três Figuras 5.16(a), 5.16(c) e 5.16(e), são relativas às instâncias de 512 nós: A1100, A1175 e A1150. O AGPNE tem um desempenho a nível de FO de 0,1630, 0,2610 e 0,1707, respectivamente, para estas instâncias. As Figuras 5.16(b), 5.16(d), 5.16(f) mostram os resultados para as instâncias de 1024 nós B1100, B1175, B1150. A Figura 5.17 apresenta o resultado para a RedeSP que tem 1659

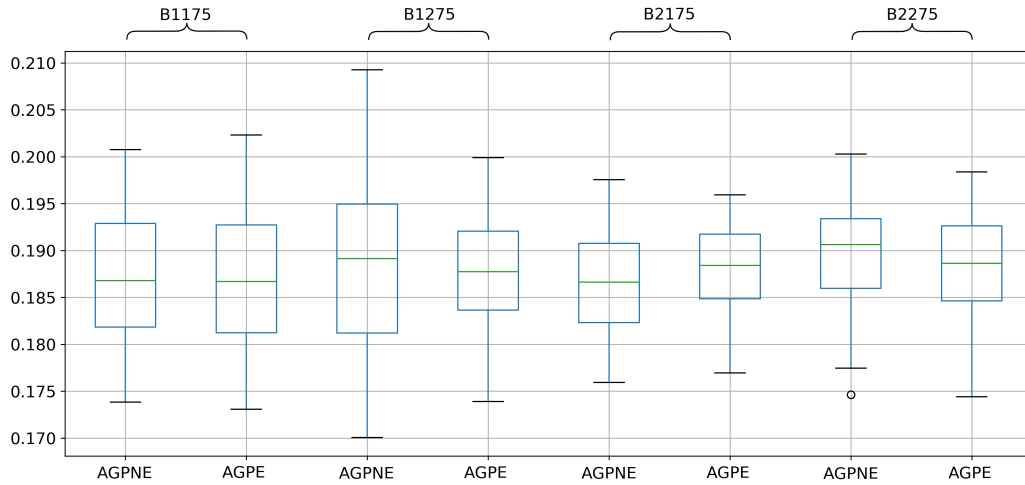


Figura 5.13: Instâncias de 1024 nós parcialmente esparsas.

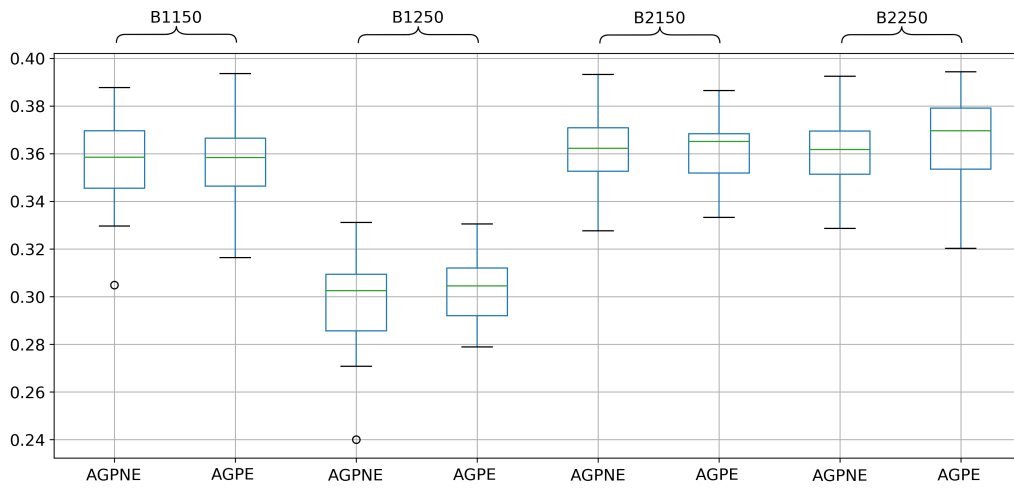


Figura 5.14: Instâncias de 1024 nós minimamente conectadas.

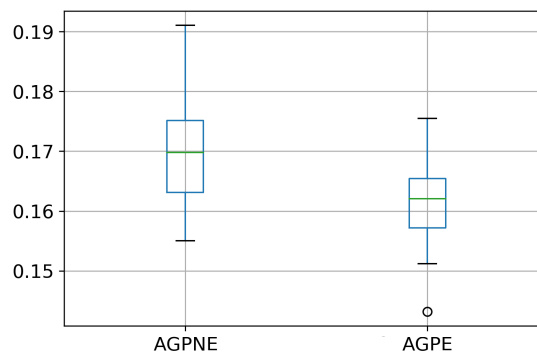
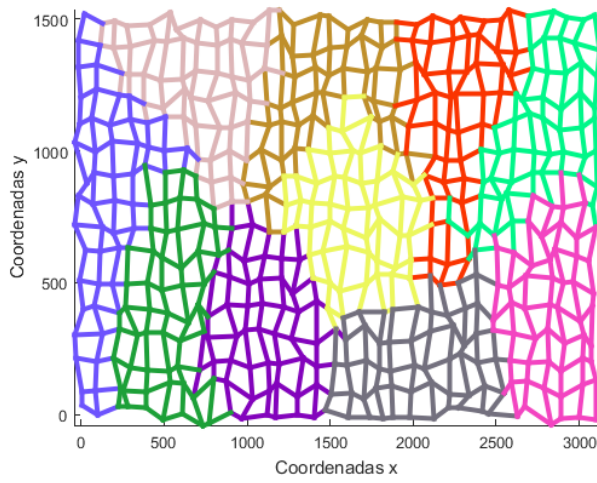


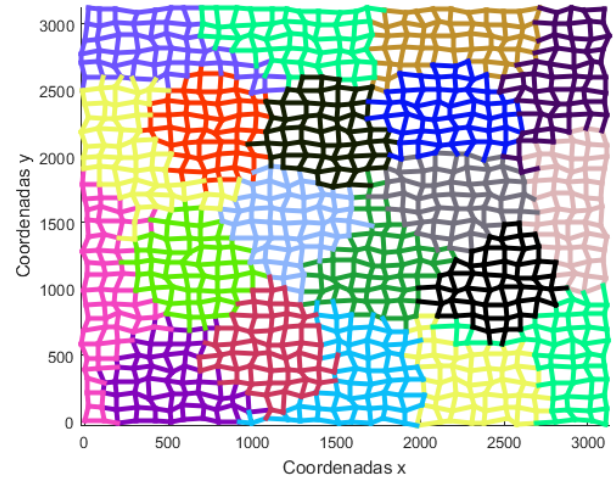
Figura 5.15: Resultado para a RedeSP.

nós. No caso das instâncias B1100 e B1150 o AGPNE teve o valor da FO de 0,1479 e 0,3049. No que diz respeito às instâncias B1175 e RedeSP, o AGPE obteve o valor da FO de 0,1731 e 0,1432

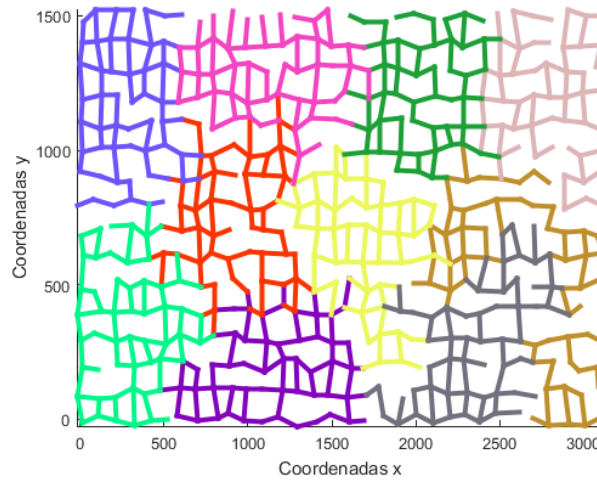
respectivamente. Também é possível observar a qualidade das soluções com relação à compacidade, principalmente na RedeSP que obteve $F(S) = 0,2251$, $G(S) = 0,0612$, $FO(S) = 0,1432$ e esta solução foi encontrada em 890 segundos. É também possível observar os diferentes graus de esparsidade e uma certa qualidade das soluções quanto à compacidade.



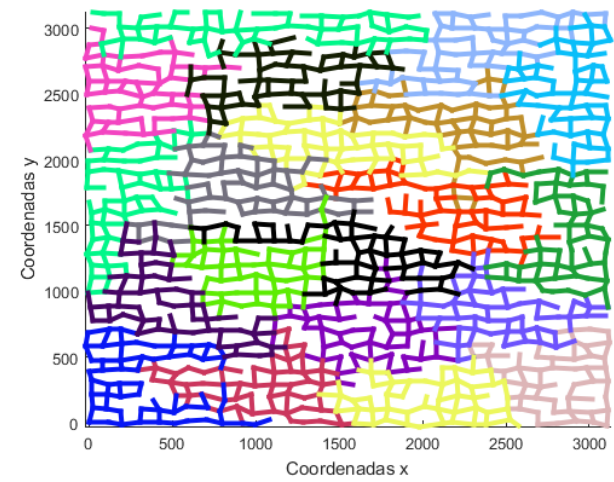
(a) Instância A1100 de 512 nós totalmente conectada.



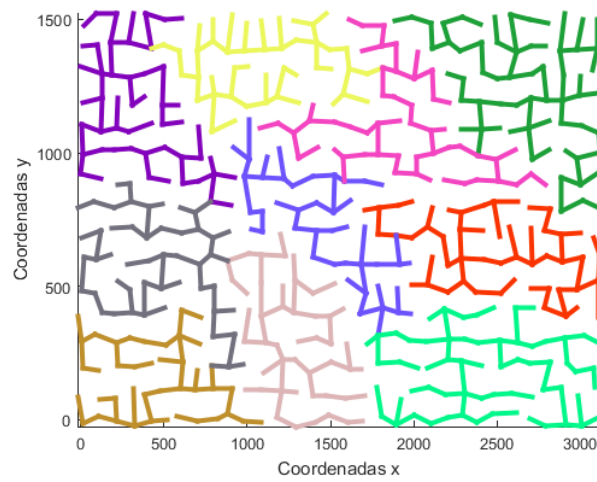
(b) Instância B1100 de 1024 nós totalmente conectada.



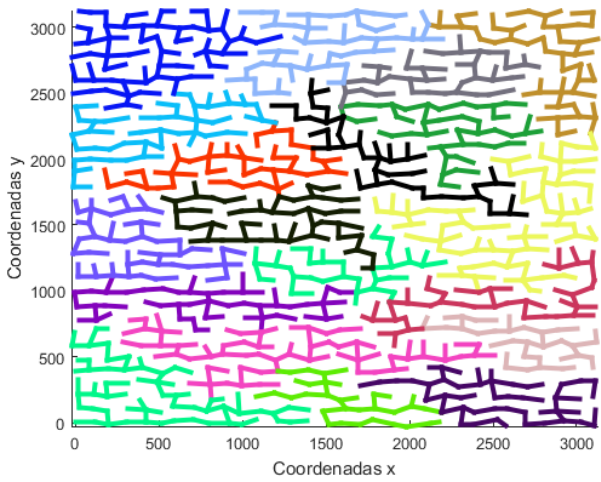
(c) Instância A1175 de 512 nós parcialmente esparsa.



(d) Instância B1175 de 1024 nós parcialmente esparsa.



(e) Instância A1150 de 512 nós minimamente conectada.



(f) Instância B1150 de 1024 nós minimamente conectada.

Figura 5.16: Resultados para as instâncias com diferentes esparsidades.

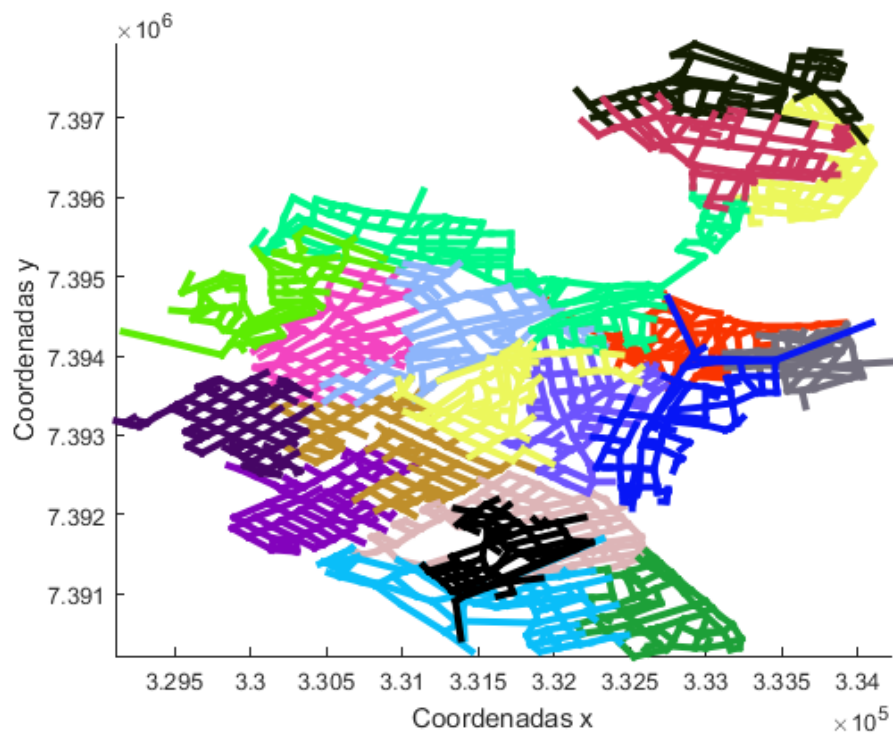


Figura 5.17: Resultado da instância RedeSP.

Capítulo 6

Considerações Finais

Este trabalho apresenta um estudo sobre o problema de distritamento capacitado. O principal objetivo alcançado foi desenvolver um método de solução para o PDC aplicado às redes de serviços com foco no problema de agrupamento de distritos de faturamento para distribuidoras de energia elétrica. Para este fim foram desenvolvidos dois algoritmos genéticos com estruturas populacionais distintas. Até o presente momento este é o primeiro trabalho a utilizar uma população hierarquicamente estruturada para esta classe de problemas.

Partindo da geração da população inicial, a heurística proposta foi adaptada ao problema de forma a reduzir a geração de soluções iniciais ineficazes pela criação simultânea dos distritos. Para homogeneidade da carga de trabalho, a atribuição é feita preferencialmente em distritos distantes da média ideal. Foi também realizada uma distribuição equidistante dos nós centrais dos distritos que visa compacidade.

Para a fase de seleção foram utilizados dois operadores distintos para cada algoritmo. Para o Algoritmo Genético com População não Estruturada (AGPNE), a seleção se deu por um processo híbrido de elitismo e torneio, que oferece um ganho maior na qualidade das soluções e ao mesmo tempo permite diversidade genética. Já para o Algoritmo Genético com População Estruturada (AGPE), a seleção ocorre de forma determinística, na qual todos os pares de indivíduos (líder, subordinado) da população participam do *crossover* em todas as gerações.

O operador de *crossover* foi alvo de diversos experimentos até se alcançar a principal contribuição deste trabalho, o operador que considera a similaridade geométrica. Este novo operador possibilitou um maior aproveitamento da informação genética do segundo progenitor e reduziu a incidência da fase de reparo. Para o operador de mutação, apenas os nós em fronteiras de distritos são considerados candidatos a sofrerem mutação. Adicionalmente, a mutação apenas ocorre quando não há quebra da conectividade do distrito envolvido.

Todos os operadores no AG tem seu funcionamento determinado pelos valores de seus hiperparâmetros. Considerando a elevada dimensão do espaço de busca foi realizada uma etapa de otimização de hiperparâmetros para ambos os algoritmos. Esta etapa apresentou o impacto dos diferentes valores em cada operador e permitiu uma maior previsibilidade dos resultados obtidos.

Ambos os algoritmos propostos foram capazes de alcançar soluções de qualidade com baixo esforço computacional. Segundo os resultados obtidos, o AGPE se mostra promissor para resolver o PDC com instâncias de grande porte (acima de 1024 nós). Como a classe de PD compartilha características semelhantes, com as adaptações necessárias, a proposta de solução apresentada pode ser aplicada nos mais diversos problemas de distritamento. Aplicações ainda mais diretas podem ser realizadas para qualquer tipo de rede de serviço como fornecimento de água, gás, entre outras.

Para trabalhos futuros, uma possível proposta é a adaptação do modelo matemático e da metodologia para uma abordagem multiobjetivo, considerando que os resultados apresentados nesta dissertação tratam ambas as funções com o mesmo peso $\lambda = 0,5$. É possível também considerar que uma etapa de pré-otimização da população inicial pode trazer impactos positivos na qualidade das soluções. Adicionalmente, a geração simultânea das rotas que serão percorridas pelos leitores pode agregar valor ao problema em estudo. A aplicação de um algoritmo memético pode ser analisada, com a adição de uma busca local após o operador de mutação. O uso de computação de alto desempenho com paralelismo pode representar ganhos significativos. Por fim, uma combinação com um método exato pode ser avaliada e oferecer benefícios.

Bibliografia

- Ahmad, I., Bard, J. and Zan, J. [2010]. Validating vehicle routing zone construction using monte carlo simulation, *European Journal of Operational Research* **206**: 73–85.
- Apollonio, N., Becker, R., Lari, I., Ricca, F. and Simeone, B. [2009]. Bicolored graph partitioning, or: gerrymandering at its worst, *Discrete Applied Mathematics* **157**: 3601–3614.
- Assis, L. S. [2009]. *Problema de reagrupamento capacitado*, Master’s thesis, Universidade Estadual de Campinas.
- Bacao, F., Lobo, V. and Painho, M. [2005]. Applying genetic algorithms to zone design, *Soft Computing* **9**: 28–35.
- Bodin, L. D. [1973]. A districting experiment with a clustering algorithm, *Annals of the New York Academy of Sciences* **219**(1): 209–214.
- Bondy, A. and Murty, U. S. R. [2008]. *Graph Theory*, Springer.
- Bondy, J. and Murty, U. [1982]. *Graph Theory With Applications*.
- Bourjolly, j.-m., Laporte, G. and Rousseau, J.-M. [1981]. Découpage electoral automatisé: Application a l’île de montréal, *INFOR* **19**: 113–124.
- Bozkaya, B., Erkut, E. and Laporte, G. [2003]. A tabu search heuristic and adaptive memory procedure for political districting, *European Journal of Operational Research* **144**: 12–26.
- Camacho-Collados, M., Liberatore, F. and Angulo, J. [2015]. A multi-criteria police districting problem for the efficient and effective design of patrol sector, *European Journal of Operational Research* **246**.
- Castelli, M., Henriques, R. and Vanneschi, L. [2015]. A geometric semantic genetic programming system for the electoral redistricting problem, *Neurocomputing* **154**: 200–207.
- Chen, H., Cheng, T. and Ye, X. [2019]. Designing efficient and balanced police patrol districts on an urban street network, *International Journal of Geographical Information Science* **33**(2): 269–290.

- Costa, P. B., Silva, I. F. S., Vieira, P. H. C., Santos, R. D. A., Silva, M. G., Lima, C. S. C., Júnior, D. L. G., Garros, E. M., Silva, I. F. S. and Pinheiro, L. P. A. [2018]. Rmeans: A recursive capacitated means for districting problem, *Journal on Advances in Theoretical and Applied Informatics* **4**: 28.
- de Assis, L. S. [2014]. *Otimização de Alocação de Chaves em Redes de Distribuição de Energia Elétrica*, PhD thesis, Universidade Estadual de Campinas.
- de Assis, L. S., Franca, P. M. and Usberti, F. L. [2014]. A redistricting problem applied to meter reading in power distribution networks, *Computers & Operations Research* **41**: 65 – 75.
- de Assis, L. S., González, J. F. V., Usberti, F. L., Lyra, C., Cavellucci, C. and Zuben, F. J. V. [2015]. Switch allocation problems in power distribution systems, *IEEE Transactions on Power Systems* **30**(1): 246–253.
- Deakin, R., Bird, S. and Grenfell, R. [2002]. The centroid? where would you like it to be be?, *Cartography* **31**: 153–167.
- Eiben, A. and Smith, J. [2015]. *Introduction to Evolutionary Computing (Natural Computing Series)*.
- Farughi, H., Mostafayi, S. and Arkat, J. [2019]. Healthcare districting optimization using gray wolf optimizer and ant lion optimizer algorithms (case study: South khorasan healthcare system in iran), *Journal of Optimization in Industrial Engineering* **12**(1): 119–131.
- Fréminville, P., Desaulniers, G., Rousseau, L.-M. and Perron, S. [2015]. A column generation heuristic for districting the price of a financial product, *Journal of the Operational Research Society* **66**: 965–978.
- G. Elizondo-Amaya, M., Ríos-Mercado, R. and Díaz, J. [2014]. A dual bounding scheme for a territory design problem, *Computers & Operations Research* **44**: 193–205.
- García-Ayala, G., González-Velarde, J., Ríos-Mercado, R. and Fernández, E. [2016]. A novel model for arc territory design: Promoting eulerian districts, *International Transactions in Operational Research* **23**(3): 433–458.
- Garey, M. R. and Johnson, D. S. [1979]. *Computers and Intractability: A Guide to the Theory of Np-Completeness*, W H Freeman & Co, Gordonsville, Virginia.
- Garfinkel, R. and Nemhauser, G. [1970]. Optimal political districting by implicit enumeration techniques, *Management Science* **16**: 495–495.

- Goldberg, D. E. [1989]. *Genetic Algorithms in Search, Optimization, and Machine Learning*, Addison-Wesley, New York.
- Gong, Y.-J., Chen, W.-N., Zhan, Z.-H., Zhang, J., Li, Y., Zhang, Q. and Li, J.-J. [2015]. Distributed evolutionary algorithms and their models: A survey of the state-of-the-art, *Applied Soft Computing* **34**: 286 – 300.
- González-Ramírez, R., Smith, N., Askin, R., Camacho-Vallejo, J.-F. and González-Velarde, J. [2017]. A grasp-tabu heuristic approach to territory design for pickup and delivery operations for large-scale instances, *Mathematical Problems in Engineering* **2017**.
- Grosan, C., Grosan, C., Abraham, A. and Ishibuchi, H. [2007]. *Hybrid Evolutionary Algorithms*, 1st edn, Springer Publishing Company, Incorporated.
- Gutiérrez-Gutiérrez, E. and Vidal, C. [2015]. A home health care districting problem in a rapid-growing city [un problema de zonificación en servicios de hospitalización domiciliaria en una ciudad de rápido crecimiento], *Ingeniería y Universidad* **19**(1): 87–113.
- Hess, S. W., Weaver, J. B., Siegfeldt, H. J., Whelan, J. N. and Zitlau, P. A. [1965]. Nonpartisan political redistricting by computer, *Operations Research* **13**(6): 998–1006.
- Hirose, A., Scarpin, C. and Pecora Jr, J. [2020]. Goal programming approach for political districting in santa catarina state: Brazil, *Annals of Operations Research* **287**.
- Hojati, M. [1996]. Optimal political districting, *Computers Operations Research* **23**(12): 1147 – 1161.
- Holland, J. H. [1975]. *Adaptation in Natural and Artificial Systems*, University of Michigan Press, Ann Arbor, MI. second edition, 1992.
- Hu, F., Yang, S. and Xu, W. [2014]. A non-dominated sorting genetic algorithm for the location and districting planning of earthquake shelters, *International Journal of Geographical Information Science* **28**(7): 1482–1501.
- Joshi, D., Soh, L.-K. and Samal, A. [2012]. Redistricting using constrained polygonal clustering, *IEEE Transactions on Knowledge and Data Engineering* **24**(11): 2065–2079.
- Kalcsics, J. [2015]. *Districting Problems*, pp. 595–622.
- Kalcsics, J., Nickel, S. and Schröder, M. [2005]. Towards a unified territorial design approach — applications, algorithms and gis integration, *TOP: An Official Journal of the Spanish Society of Statistics and Operations Research* **13**: 1–56.

- Kazazi, S. and Bashiri, M. [2018]. A multi-district asset protection problem with time windows for disaster management, *International Journal of Engineering, Transactions B: Applications* **31**: 1929–1934.
- Kirkpatrick, S., Gelatt, C. and Vecchi, M. [1983]. Optimization by simulated annealing, *Science (New York, N.Y.)* **220**: 671–80.
- Lara-Caballero, A., De-Los-Cobos-Silva, S., Mora-Gutiérrez, R., Rincón-García, E., Gutiérrez-Andrade, M. and Lara-Velázquez, P. [2019]. Multiobjective genetic algorithms for reinforcing equal population in congressional districts, *Mathematical Problems in Engineering* **2019**.
- Lari, I., Maravalle, M. and Simeone, B. [2009]. Computing sharp bounds for hard clustering problems on trees, *Discrete Applied Mathematics* **157**(5): 991–1008.
- Levin, H. A. and Friedler, S. A. [2019]. Automated congressional redistricting, *J. Exp. Algorithmics* **24**(1): 1.10:1–1.10:24.
- Liberatore, F. and Camacho-Collados, M. [2016]. A comparison of local search methods for the multicriteria police districting problem on graph, *Mathematical Problems in Engineering* **2016**.
- Lin, M., Chin, K.-S., Fu, C. and Tsui, K.-L. [2017]. An effective greedy method for the meals-on-wheels service districting problem, *Computers & Industrial Engineering* **106**.
- Lin, M., Chin, K. S., Ma, L. and Tsui, K. L. [2018]. A comprehensive multi-objective mixed integer nonlinear programming model for an integrated elderly care service districting problem, *Annals of Operations Research* .
- Linden, R. [2006]. *Algoritmos genéticos: uma importante ferramenta da inteligência computacional*, Brasport.
- Lopes, H., Rodrigues, L. and Teresinha, M. [2013]. *Meta-Heurísticas em Pesquisa Operacional*, Omnipax.
- MacKay, D. J. C. [2002]. *Information Theory, Inference & Learning Algorithms*, Cambridge University Press, New York, NY, USA.
- Marianov, V. and Fresard, F. [2005]. A procedure for the strategic planning of locations, capacities and districting of jails: Application to chile, *Journal of The Operational Research Society - J OPER RES SOC* **56**: 244–251.
- Michalewicz, Z. [1996]. *Genetic Algorithm+Data Structures=Evolution Programs*.

- Mohammadi, M., Dashti, Z. and Mirzazadeh, A. [2014]. Mclp and sqm models for the emergency vehicle districting and location problem, *Decision Science Letters* **3**(4): 479–490.
- Moraglio, A., Krawiec, K. and Johnson, C. G. [2012]. Geometric semantic genetic programming, in C. A. C. Coello, V. Cutello, K. Deb, S. Forrest, G. Nicosia and M. Pavone (eds), *Parallel Problem Solving from Nature - PPSN XII*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 21–31.
- Moscato, P. [1989]. On evolution, search, optimization, gas and martial arts: Toward memetic algorithms, *On Evolution, Search, Optimization, GAs and Martial Arts: Toward Memetic Algorithms* .
- Moscato, P. and Cotta, C. [2003]. An introduction to memetic algorithms, *Inteligencia Artificial, revista Iberoamericana De Inteligencia Artificial - AEPIA* **7**.
- Moscato, P., Plata, L., Plata, L. and Norman, M. G. [1992]. A "memetic" approach for the traveling salesman problem implementation of a computational ecology for combinatorial optimization on message-passing systems, *In Proceedings of the International Conference on Parallel Computing and Transputer Applications*, IOS Press, pp. 177–186.
- Mulvey, J. M. and Beck, M. P. [1984]. Solving capacitated clustering problems, *European Journal of Operational Research* **18**(3): 339–348.
- Pereira, F., Figueira, J., Mousseau, V. and Roy, B. [2007]. Multiple criteria districting problems: The public transportation network pricing system of the Paris region, *Annals of Operations Research* **154**(1): 69–92.
- Regis-Hernández, F., Lanzarone, E., Bélanger, V. and Ruiz, A. [2018]. A data-driven districting to improve emergency medical service systems, *IFAC-PapersOnLine* **51**(11): 998–1003.
- Ricca, F., Scozzari, A. and Simeone, B. [2007]. Weighted voronoi region algorithms for political districting, *Mathematical and Computer Modelling* **48**: 1468–1477.
- Ricca, F., Scozzari, A. and Simeone, B. [2013]. Political districting: From classical models to recent approaches, *Annals of Operations Research* **204**: 271–299.
- Ricca, F. and Simeone, B. [2008]. Local search algorithms for political districting, *European Journal of Operational Research* **189**: 1409–1426.
- Rincón-García, E., Gutiérrez-Andrade, M., De-Los-Cobos-Silva, S., Lara-Velázquez, P., Ponsich, A. and Mora-Gutiérrez, R. [2013]. A multiobjective algorithm for redistricting, *Journal of Applied Research and Technology* **11**(3): 324–330.

- Ríos-Mercado, R. [2020]. *Optimal Districting and Territory Design*.
- Ríos-Mercado, R. and Bard, J. [2018]. An exact algorithm for designing optimal districts in the collection of waste electric and electronic equipment through an improved reformulation, *European Journal of Operational Research* **276**.
- Ríos-Mercado, R. and López, F. [2013]. Commercial territory design planning with realignment and disjoint assignment requirements, *Omega* **41**: 525–535.
- Ríos-Mercado, R. Z. and Escalante, H. J. [2016]. Grasp with path relinking for commercial districting, *Expert Syst. Appl.* **44**(C): 102–113.
- Ríos-Mercado, R. Z. and Fernández, E. [2009]. A reactive grasp for a commercial territory design problem with multiple balancing requirements, *Computers & Operations Research* **36**(3): 755 – 776.
- Ruvalcaba, L., Correa, G. and Zanella, V. [2011]. Multiobjective evolutionary algorithm for redesigning sales territories, *Proceedings of the Second International Conference on Computational Logistics*, ICCL’11, Springer-Verlag, Berlin, Heidelberg, pp. 183–193.
- Saracco, A. and Saracco, G. [2019]. A discrete districting plan, *Networks and Heterogeneous Media* **14**(4): 771–788.
- Shanker, R. J., Turner, R. E. and Zoltners, A. A. [1975]. Sales territory design: An integrated approach, *Manage. Sci.* **22**(3): 309–320.
- Speranza, M. and Archetti, C. [2014]. A survey on matheuristics for routing problems, *EURO Journal on Computational Optimization* **2**.
- Steiner, M. T. A., Datta, D., Neto, P. S., Scarpin, C. and Figueira, J. [2014]. Multi-objective optimization in partitioning the healthcare system of parana state in brazil, *Omega* **52**: 53–64.
- Toledo, C., Oliveira, L. and França, P. [2014]. Global optimization using a genetic algorithm with hierarchically structured population, *Journal of Computational and Applied Mathematics* **261**: 341 – 351.
- Vanneschi, L., Henriques, R. and Castelli, M. [2017]. Multi-objective genetic algorithm with variable neighbourhood search for the electoral redistricting problem, *Swarm and Evolutionary Computation* **36**: 37 – 51.
- Vickrey, V. [1961]. On the prevention of gerrymandering, *Political Science Quarterly* **76**.

- Wang, D., Zhao, J., D'Ariano, A. and Peng, Q. [2020]. Simultaneous node and link districting in transportation networks: Model, algorithms and railway application, *European Journal of Operational Research* .
- Whitley, D. [1994]. A genetic algorithm tutorial, *Statistics and Computing* **4**: 65–85.
- Yang, B., Viswanathan, K., Lertworawanich, P. and Kumar, S. [2004]. Fire station districting using simulation: Case study in centre region, pennsylvania, *Journal of Urban Planning and Development-asce - J URBAN PLAN DEV-ASCE* **130**.
- Yanik, S., Kalcsics, J., Nickel, S. and Bozkaya, B. [2019]. A multi-period multi-criteria districting problem applied to primary care scheme with gradual assignment, *International Transactions in Operational Research* **26**(5): 1676–1697.