



CENTRO FEDERAL DE EDUCAÇÃO
TECNOLÓGICA DE MINAS GERAIS

Diretoria de Pesquisa e Pós-Graduação

Programa de Pós-Graduação em Modelagem
Matemática e Computacional

ALGORITMO GVNS APLICADO AO PROBLEMA DAS p -MEDIANAS CAPACITADO: ABORDAGENS DETERMINÍSTICA E ROBUSTA

Tese de Doutorado apresentado ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, como parte dos requisitos necessários para o título de Doutor em Modelagem Matemática e Computacional.

Aluno : Anderson Moreira de Vasconcelos

Orientador : Prof. Dr. Marcone Jamilson Freitas Souza (UFOP)

Coorientador : Prof. Dr. Sérgio Ricardo de Souza (CEFET-MG)

Coorientadora : Profa. Dra. Elisangela Martins de Sá (CEFET-MG)

Belo Horizonte - MG
25 de maio de 2023

V331a Vasconcelos, Anderson Moreira de
Algoritmo GVNS aplicado ao problema das p -medianas capacitado: abordagens determinística e robusta / Anderson Moreira de Vasconcelos. – 2023.
93 f.

Tese de doutorado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.

Orientador: Marcone Jamilson Freitas Souza.

Coorientador: Sérgio Ricardo de Souza.

Coorientadora: Elisangela Martins de Sá.

Tese (doutorado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Otimização robusta – Teses. 2. Metaheurísticas – Teses. 3. Algoritmos – Teses. 4. Análise estatística – Teses. 5. Computação – Matemática – Teses. I. Souza, Marcone Jamilson Freitas. II. Souza, Sérgio Ricardo de. III. Sá, Elisangela Martins de. IV. Centro Federal de Educação Tecnológica de Minas Gerais. V. Título.

CDD 519.6



SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

**“ALGORITMO GVNS APLICADO AO PROBLEMA DAS P-MEDIANAS
CAPACITADO: ABORDAGENS DETERMINÍSTICA E ROBUSTA”.**

Tese de Doutorado apresentada por **Anderson Moreira de Vasconcelos**, em 20 de dezembro de 2022, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Marcone Jamilson Freitas Souza
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Sérgio Ricardo de Souza (Coorientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Elisângela Martins de Sá (Coorientadora)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Maria Amélia Lopes Silva
Universidade Federal de Viçosa

Prof. Dr. Matheus Nohra Haddad
Universidade Federal de Viçosa

Prof. Dr. Elizabeth Fialho Wanner
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Gustavo Campos Menezes
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Prof. Dr. Elizabeth Fialho Wanner
Presidenta do Colegiado do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

Dedico este trabalho às pessoas mais importantes para mim,
Dárlan, Victor, Caio e Artur.

Agradecimentos

Ao término deste trabalho, desejo muito agradecer a todas as pessoas que contribuíram para mais essa realização. Não seria possível expressar em palavras minha gratidão por elas, mas ainda sim, penso que é necessário registrar os meus sinceros agradecimentos.

Em primeiro lugar, agradeço a Deus, pois nos meus momentos de dificuldade e cansaço, encontrei Nele, toda a força e esperança de que tudo iria dar certo.

À minha esposa, Dárlan, que soube, desde o início, me entender, apoiar e incentivar nos momentos mais cansativos, trabalhosos e difíceis. Agradeço pela companhia, paciência e apoio durante todo o tempo em que me dediquei a este trabalho.

Agradeço aos meus filhos Victor, Caio e Artur por compreenderem, acompanharem e assumirem diversas responsabilidades que me permitiu dedicar “quase que integralmente” a este trabalho.

Ao meu orientador, Prof. Dr. Marcene Jamilson Freitas Souza, por ter me dado a oportunidade de vivenciar essa experiência e acreditar neste trabalho. Obrigado por todas as oportunidades de aprendizado que me foram dadas. Muito obrigado, desejo tudo de bom e espero que você possa voltar em breve à missão que tanto ama.

Ao meu co-orientador, Prof. Dr. Sérgio Ricardo de Souza, que acreditou, incentivou, acompanhou e me proporcionou oportunidades de crescimento profissional desde o mestrado, por toda a sua dedicação, atenção e paciência, não só às questões profissionais, mas também, pessoais e que muito contribuíram para o desenvolvimento deste trabalho. Respeito muito seu trabalho e a forma como o realiza. Muito obrigado Sérgio.

Sou muito grato também a Profa. Dra. Elisangela Martins de Sá, minha co-orientadora, que desde o início trouxe enorme habilidade e sabedoria para meus estudos ao longo desse curso de doutorado. Eu realmente aprecio toda a sua cooperação, dedicação e profissionalismo em todos os trabalhos que realizamos juntos.

Aos meus pais, Maria da Conceição e José Pedro, que me educaram durante grande parte da minha vida e por terem me apoiado e incentivado sempre, por tudo que fizeram por mim, e por sempre terem buscado o melhor para mim e para minha família. Deixo aqui minha eterna gratidão por tudo.

Aos professores do Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, pelos ensinamentos, apreciações e contribuições para este trabalho.

Obrigado a todos os meus colegas de doutorado pelo apoio e por serem grandes companheiros ao longo do caminho. Em especial meu amigo César Óliver, companheiro de viagens, que me ajudou desde o início, principalmente em questões tecnológicas, me deu a maior atenção e apoio no processo de realização deste trabalho.

Muito obrigado.

A todos os funcionários do CEFET-MG, que sempre foram cordiais e me deram a estrutura necessária para o desenvolvimento desse trabalho.

A todas as demais pessoas que contribuíram direta e indiretamente para a realização deste trabalho. Meu muito obrigado.

Resumo

O Problema das p -Medianas Capacitado (PPMC) consiste em localizar p instalações em uma rede composta por n vértices e decidir qual mediana atenderá cada vértice, a fim de minimizar a soma das distâncias dos vértices à mediana a qual ele foi alocado e atender às restrições de capacidade de cada mediana. Considerando que as demandas dos clientes são incertas, é proposta a aplicação de uma abordagem de otimização robusta para lidar com a incerteza nestes parâmetros dando origem ao problema das p -Medianas Robusto Capacitado (PPMRC). Esta tese propõe algoritmos para resolução do PPMC e o PPMRC. Para resolver o PPMC, foram propostas quatro variantes da metaheurística *General Variable Neighborhood Search* (GVNS), que se diferem no método usado para construir uma solução inicial e nos métodos usados para realizar a busca local. Nas duas primeiras variantes, denominadas G-VND e G-RVND, a solução inicial é gerada de forma aleatória e os métodos de busca local são o *Variable Neighborhood Descent* (VND) e o *Random Variable Neighborhood Descent* (RVND), respectivamente. Nas duas últimas variantes, denominadas GG-VND e GG-RVND, a solução inicial é gerada através da fase de construção da metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) e os métodos de busca local são o VND e o RVND, respectivamente. Os resultados de experimentos computacionais usando instâncias da literatura mostraram que o algoritmo GG-RVND apresentou melhor desempenho que os outros três algoritmos. Além disso, os resultados obtidos pelo algoritmo GG-RVND foram melhores do que aqueles da literatura, tanto em termos da qualidade das soluções geradas quanto em relação ao tempo de execução. Para o PPMRC, foi proposto um algoritmo baseado na metaheurística GVNS, com a solução inicial gerada através da fase construtiva do método GRASP e uma busca local realizada via VND. Os experimentos computacionais comparam os resultados do GVNS e as soluções do *solver* CPLEX. Para realizar esses experimentos computacionais, foram utilizadas instâncias da literatura, variando de 318 nós e cinco medianas a 4461 nós e 1000 medianas. Este estudo considera diversos cenários, valores variados dos parâmetros de robustez e de incerteza na demanda. Os resultados mostram que o Algoritmo GVNS tem um bom desempenho em encontrar boas soluções e supera o *solver* CPLEX, encontrando soluções melhores mesmo quando aplicado para resolver o conjunto de dados de menor dimensão.

Palavras-chaves: Problema de localização, Problema de p -Mediana Capacitada, Otimização Robusta, Metaheurísticas, GVNS, VND, RVND, GRASP.

Abstract

This thesis also proposes algorithms for solving PPMC and PPMRC. To solve PPMC, four variants of the General Variable Neighborhood Search (GVNS) metaheuristic were proposed, which differ in the method used to construct an initial solution and the method used to perform the local search. In the first two variants, i.e., G-VND and G-RVND, the initial solution is generated randomly, and the local search methods are the Variable Neighborhood Descent (VND) and the Random Variable Neighborhood Descent (RVND), respectively. In the last two variants, GG-VND and GG-RVND, the initial solution is generated through the constructive phase of the Greedy Randomized Adaptive Search Procedure (GRASP) metaheuristic, and the local search methods are the VND and RVND procedures, respectively. The results of computational experiments using instances from the literature showed that the GG-RVND algorithm performed better than the other three algorithms. Furthermore, the results obtained by the GG-RVND algorithm were better than those in the literature regarding the quality of the solutions generated and the execution time. For PPMRC, an algorithm based on the GVNS metaheuristic is proposed, with the initial solution generated through the constructive phase of the GRASP method and a local search performed via VND. The computational experiments compare GVNS results and the solver CPLEX solutions. To perform these computational experiments, this work used instances from the literature, ranging from 318 nodes and five medians to 4461 nodes and 1000 medians. This study considers several scenarios, varying values of parameters of robustness, and uncertainty in demand. The results show that the GVNS Algorithm performs well in finding good-quality solutions and outperforms the CPLEX solver, finding better solutions even when applied to solve the smallest dataset.

Keywords: Robust Optimization, Capacitated p -Median Problem, GVNS, VND, RVND, GRASP.

Conteúdo

Lista de Tabelas	xii
Lista de Figuras	xiii
Lista de Algoritmos	xiv
1 Introdução	1
1.1 Apresentação Geral	1
1.2 Objetivos Gerais e Específicos	5
1.2.1 Objetivo Geral	5
1.2.2 Objetivos Específicos	5
1.3 Contribuições	6
1.4 Organização da Tese	6
2 Caracterização do Problema	8
2.1 Problema das p -medianas	8
2.2 Problema das p -Medianas não Capacitado	10
2.3 Problema das p -Medianas Capacitado	12
2.4 Problema das p -Medianas Robusto Capacitado	14
2.4.1 Linearização da Formulação Robusta	16
2.4.2 Exemplo do PPMRC	18
2.5 Considerações Parciais	21
3 Revisão da Literatura	23
3.1 Heurísticas	23
3.2 Metaheurísticas	25
3.2.1 Metaheurística <i>Variable Neighborhood Search</i> (VNS)	26
3.2.2 Metaheurística GRASP	29
3.3 Heurísticas e metaheurísticas híbridas	30
3.4 Revisão da Literatura para o PPMC	31
3.5 Revisão da Literatura para o PPMRC	34
3.6 Considerações Parciais	39
4 Solução para o Problema das p-Medianas Capacitado (PPMC)	41
4.1 Representação e Avaliação da Solução	41
4.2 Construção de uma solução inicial	42
4.2.1 Construção randômica da solução inicial	42
4.2.2 Solução inicial por meio da fase de construção GRASP	43
4.3 Estruturas de Vizinhaça	45
4.4 Busca Local	46

4.5	Procedimento de Perturbação	47
4.6	Algoritmo GVNS	48
4.7	Descrição dos Experimentos Computacionais	49
4.8	Resultados para o Grupo de instâncias 1	50
4.8.1	Comparação entre os quatro algoritmos propostos	51
4.8.2	Grupo de instâncias 1: Comparação com os resultados da literatura	52
4.8.3	Grupo de instâncias 1: Comparação em relação aos tempos de execução	53
4.9	Resultados para o Grupo de instâncias 2	55
4.9.1	Grupo de instâncias 2: Comparação com os resultados da literatura	55
4.9.2	Grupo de instâncias 2: Comparação em relação aos tempos de execução	57
4.10	Considerações Parciais	58
5	Solução para o Problema das p-Medianas Robusto Capacitado (PPMRC)	61
5.1	Representação e avaliação da solução	61
5.2	Construção da solução inicial	62
5.3	Estruturas de Vizinhança	64
5.4	Procedimento de busca local	65
5.5	Procedimento de perturbação	65
5.6	Algoritmo GVNS Robusto	66
5.7	Descrição dos Experimentos computacionais	67
5.8	Experimentos computacionais para o Grupo de Instâncias Stefanelo1	68
5.9	Experimentos computacionais para o Grupo de Instâncias Stefanelo2	72
5.10	Experimentos computacionais usando o Grupo de Instâncias Lorena	77
5.11	Considerações Parciais	77
6	Conclusões Gerais e Trabalhos Futuros	79
6.1	Trabalhos Futuros	80
6.2	Trabalhos Publicados e Submetidos	80
	Referências Bibliográficas	82

Lista de Tabelas

1.1	Resumo dos métodos aplicados na resolução do PPMC	4
2.1	Matriz de distância da instância do exemplo ilustrativo	11
2.2	Cálculo da distância total para o Exemplo da Figura 2.1(b).	12
2.3	Valores das demandas dos cliente j	14
2.4	Cálculo da distância total para o Exemplo da Figura 2.3.	14
2.5	Valores das demandas centrais do cliente e os maiores desvios de 10% das demandas centrais.	18
2.6	Valores de demanda central e os maiores valores de desvio considerando $\Gamma = 2$ e $\Delta r = 10\%$	19
2.7	Soma das demandas centrais atendidas por cada mediana com os maiores desvios Γ	20
2.8	Resultados do exemplo ilustrativo para diferentes valores de Γ e desvio.	21
3.1	Tabela resumo - Heurísticas híbridas aplicadas ao PMPC	34
3.2	Trabalhos abordando otimização robusta em problemas de localização de facilidades.	39
4.1	Faixa de valores testados e retornados pela ferramenta IRACE para os parâmetros dos algoritmos nos Grupos de instâncias 1 e 2	50
4.2	Grupo de instâncias 1 - Valores médios para 30 execuções	51
4.3	Grupo de instâncias 1 – Teste de normalidade dos valores de desvio	51
4.4	Grupo de instâncias 1: p -valores referentes a aplicação do Teste F para avaliar a diferença entre os valores de desvio percentual Δ	51
4.5	Grupo de instâncias 1: comparação do valor de função objetivo e dos desvios relativos médios obtidos pelo Algoritmo GG-VND com a literatura	53
4.6	Grupo de instâncias 1: Teste de normalidade dos valores de desvio \times literatura. Intervalo de confiança: $\rho = 95\%$	53
4.7	Grupo de instâncias 1: Teste F dos valores de desvio relativo percentual $\Delta \times$ valores da literatura. Intervalo de confiança: $\rho = 95\%$	54
4.8	Grupo de instâncias 1: comparação entre CPUs	54
4.9	Grupo de instâncias 1: comparação com os trabalhos da literatura quanto ao tempo de execução	55
4.10	Grupo de instâncias 2: Comparação com os resultados da literatura quanto ao valor médio da função objetivo e desvio (%) em relação aos melhores valores conhecidos	56

4.11	Grupo 2: teste de normalidade dos valores de desvio percentual \times valores da literatura, considerando valores médios do Algoritmo GG-VND. Intervalo de confiança: $\rho = 95\%$	57
4.12	Grupo de instâncias 2: teste de significância estatística dos valores de desvio percentual \times valores de literatura	57
4.13	Grupo de instâncias 2: análise de tempo de execução dos métodos	58
5.1	Resultados do ajuste de parâmetros com IRACE.	68
5.2	Comparação do <i>solver</i> CPLEX e do Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1	69
5.3	Teste de normalidade de Shapiro-Wilk para a diferença entre os <i>gaps</i> encontradas pelo <i>solver</i> CPLEX e o Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1	70
5.4	Teste pareado de Wilcoxon-Mann-Whitney para a diferença entre os <i>gaps</i> encontradas pelo <i>solver</i> CPLEX e o Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1	71
5.5	Teste de normalidade Shapiro-Wilk para a diferença entre os tempos de execução encontrados pelo <i>solver</i> CPLEX e o algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1	72
5.6	Teste pareado de Wilcoxon-Mann-Whitney para a diferença entre os tempos de execução encontrados pelo <i>solver</i> CPLEX e o algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1	72
5.7	Comparação do <i>solver</i> CPLEX e do algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo2	74
5.8	Valores p do Teste de Wilcoxon-Mann-Whitney pareado para os <i>gaps</i> fornecidas pelo <i>solver</i> CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias ali535 e u724.	75
5.9	p -valor do Teste de Wilcoxon-Mann-Whitney pareado para os tempos de execução fornecidos pelo <i>solver</i> CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias ali535 e u724.	75
5.10	Comparação das soluções do <i>solver</i> CPLEX e do Algoritmo GVNS robusto para o Grupo de Instâncias Lorena	77

Lista de Figuras

2.1	Exemplo de uma solução.	11
2.2	Exemplo de um grafo \mathcal{G} com 10 vértices e três medianas.	12
2.3	Exemplo de grafo \mathcal{G} com 10 vértices.	14
2.4	Exemplo de grafo \mathcal{G} com 10 vértices.	19
2.5	Desvio percentual da distância total para a solução robusta em relação à distância total relacionada à solução do problema nominal para $\Gamma = 1$ e 2.	21
4.1	Exemplo de representação de uma solução.	42
4.2	Gráfico <i>box-plot</i> mostrando o desvio relativo médio $\Delta\%$ entre os algoritmos propostos no Grupo de instâncias 1.	52
4.3	Gráfico <i>box-plot</i> mostrando o desvio relativo médio Δ entre o algoritmo GG-VND e os da literatura no Grupo de instâncias 1.	54
5.1	Representação de uma solução dada por $s = (vm, vc)$	62
5.2	Gráfico de <i>boxplot</i> para os <i>gaps</i> encontradas pelo <i>solver</i> CPLEX e pelo Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1.	71
5.3	Gráfico de <i>boxplot</i> para os tempos de execução encontrados pelo <i>solver</i> CPLEX e pelo GVNS robusto para o Grupo de Instâncias Stefanelo1.	72
5.4	Gráfico de <i>boxplot</i> para os valores de <i>gap</i> fornecidos pelo <i>solver</i> CPLEX e pelo Algoritmo GVNS robusto ao resolver as instâncias ali535.	75
5.5	Gráfico de <i>boxplot</i> para os valores de <i>gap</i> fornecidos pelo <i>solver</i> CPLEX e pelo Algoritmo GVNS robusto ao resolver as instâncias u724.	75
5.6	Gráfico de <i>boxplot</i> para os tempos de execução fornecidos pelo <i>solver</i> CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias ali535	76
5.7	Gráfico de <i>boxplot</i> para os tempos de execução fornecidos pelo <i>solver</i> CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias u724.	76

Lista de Algoritmos

1	VNS.	28
2	GRASP.	29
3	Solução Inicial construída de forma randômica.	44
4	Solução Inicial do PPMC gerada via fase de construção GRASP.	45
5	VND aplicado à solução do PPMC	46
6	RVND aplicado à solução do PPMC	46
7	Perturbação.	48
8	GVNS	48
9	Geração da solução inicial do PPMRC via fase de construção GRASP	63
10	Procedimento de perturbação	66

Capítulo 1

Introdução

1.1 Apresentação Geral

Nas últimas décadas, a demanda por soluções de problemas de otimização em diferentes áreas tem crescido consideravelmente. Com isso, surge a necessidade de se aplicar métodos para ajudar na tomada de melhores decisões e de se utilizar técnicas de modelagem matemática e computacional para analisar e solucionar situações mais complexas.

Entretanto, o nível de complexidade de muitos problemas de otimização torna a tarefa de resolvê-los muito difícil. Um exemplo de um problema de otimização muito conhecido é o problema que busca determinar a localização de um conjunto de armazéns a partir de um conjunto de locais potenciais que minimize o custo total de instalação. Outro exemplo de problema de otimização muito conhecido é o problema do Caixeiro Viajante, que busca determinar uma rota que se inicia em uma cidade, visita todas as outras cidades e retorna à cidade original percorrendo a menor distância, assumindo que as distâncias entre cada par de cidades são conhecidas. Problemas envolvendo o planejamento de companhias aéreas, como os Problemas de Alocação e Agendamento ([Sanchez et al., 2020](#); [Marla et al., 2018](#); [Katsigiannis e Zografos, 2021](#)), também correspondem a aplicações práticas da otimização. Essas companhias certificam, acompanham, coordenam e utilizam sistemas de redes inteligentes que auxiliam na obtenção da melhor solução. Essas pesquisas são aplicadas a problemas de otimização que buscam minimizar o custo ou obter o máximo valor de um orçamento, os chamados Problemas de Alocação e Agendamento ([Sanchez et al., 2020](#)), ([Marla et al., 2018](#)) e ([Katsigiannis e Zografos, 2021](#)). Outro exemplo de problemas de otimização são os problemas de otimização de portfólio de investimento, que são projetados para auxiliar o investidor na tomada de decisão de como investir no mercado financeiro ([Xiao e Ning, 2022](#); [Khayamim et al., 2018](#); [Tayali, 2020](#)). Estudos recentes mostram que à medida que as redes de distribuição de energia e gás

vão se adaptando à geração de energia mais distribuída vão surgindo, concomitantemente, soluções mais acessíveis e confiáveis que ajudam a reduzir riscos e custos. Nesse sentido, [Duan et al. \(2022\)](#) e [Wu et al. \(2022\)](#) apresentam como os Problemas de Distribuição de Energia Elétrica e Gás podem ser solucionados através de otimização. Outra aplicação importante da otimização é no planejamento das cadeias de suprimentos, que busca planejar o fluxo de materiais desde a fabricação à distribuição, minimizando os custos operacionais de fabricação, transporte e distribuição ([Suckling et al., 2021](#); [Wietschel et al., 2021](#); [Abidi et al., 2021](#)). Há diversos outros problemas de otimização que podemos citar, como Problemas de Corte e Estoque ([Bangun et al., 2019](#); [Iori et al., 2021](#); [Mellouli et al., 2021](#)), Problemas de Roteamento de Veículos ([Pasha et al., 2020](#); [Ma et al., 2021](#); [Qin et al., 2019](#)), dentre diversos outros.

Uma classe de problema de otimização muito importante é a classe de Problemas de Localização. Problemas desta classe consistem em selecionar um conjunto de instalações, dentre um conjunto de instalações candidatas, de forma a otimizar um determinado objetivo e atender um conjunto de restrições. Um aspecto importante dessa classe de problemas é a sua relevância em aplicações reais, tais como localização de escolas ([Ndiaye et al., 2012](#); [Prima e Arymurthy, 2019](#); [Fagueye et al., 2012](#); [Menezes e Pizzolato, 2014](#)), localização de instalações de emergência ([Karatas e Yakici, 2019](#); [Salman e Yücel, 2015](#); [Dzator e Dzator, 2013](#); [Menghao et al., 2018](#); [Ruslim e Ghani, 2006](#)), distribuição de peças ([Ahmadi et al., 2015](#)), aplicações em redes de cargas aéreas ([Lin et al., 2012](#)), localização de abrigo para animais de estimação e com necessidades especiais ([Kocatepea et al., 2014](#)), localização de depósito para produção e distribuição de combustível ([Bargos et al., 2016](#)), localização de agentes de saúde ([Kunkel et al., 2014](#)), localização de centros de distribuição de empresa de alimentos ([Guazzelli e Cunha, 2014](#)), implantação de unidades de atendimento a incidentes de rodovias ([Shanjiang et al., 2012](#)), seleção dos locais ideais para instalações públicas de carregamento de veículos elétricos ([Sylvia et al., 2016](#)), localização de hospitais em região rural ([Carling et al., 2012](#)), localização de estações de bicicleta ([Kaveh e Nasr, 2011](#)), projeto de rede de transporte de ônibus escolares ([Kim e Soh, 2012](#)), gerenciamento de resíduos ([Lin et al., 2011](#)), e diversas outras aplicações.

Em Problemas de Localização, três classes se destacam: (i) os problemas de cobertura; (ii) os problemas de localização de p -centros; e (iii) os problemas de localização de p -medianas ([Daskin, 1997](#)). Um exemplo de um problema de cobertura é o Problema de Máxima Cobertura, que busca localizar um conjunto de instalações de forma a maximizar a cobertura, dentro de um raio de cobertura ([Daskin, 1997](#)). Neste caso, um ponto de demanda é considerado coberto se está no raio de serviço de pelo menos uma facilidade instalada. Os Problemas de Localização de p -Centros (*p -Center Problem* - PCP) em uma rede buscam selecionar p vértices de forma que a maior

distância dos vértices da rede ao vértice selecionado mais próximo seja minimizada (Kariv e Hakimi, 1979). O Problema de localização de p -Medianas (PPM) em um grafo contendo n vértices, que podem se referir a pontos de demanda, consiste em localizar p medianas de forma a minimizar a soma das distâncias de cada vértice até a mediana mais próxima (Avella et al., 2007).

Uma variante de interesse do PPM, é a sua versão capacitada, denominada Problema das p -Medianas Capacitado (PPMC). O PPMC consiste em localizar p facilidades em uma rede composta por n vértices e decidir qual facilidade atende cada vértice, de forma a minimizar a soma de todas as distâncias de cada vértice a cada mediana, respeitando-se as restrições de capacidade máxima de atendimento das medianas. Além disso, é assumido que cada cliente é atendido por um única mediana. Desta forma, o PPMC é um problema muito mais difícil de ser solucionado que o PPM, pois a alocação dos clientes à mediana não é uma alocação trivial. No PPM, por outro lado, os clientes são alocados à mediana mais próxima, mesmo que a localização das medianas sejam conhecidas, o problema associado à alocação de clientes às medianas corresponde ao problema generalizado de atribuição (*Generalized Assignment Problem* – GAP), que é um problema NP-difícil (Fisher et al., 1986).

Tanto o PPM quanto o PPMC são aplicados em diversas áreas, como mencionado anteriormente, como logística e transporte. Atualmente existem inúmeros problemas que podem ser modelados como um PPMC. O PPMC é um problema clássico de otimização que pode ser empregado para a determinação dos locais mais adequados para a instalação de serviços, dentro de uma região pré-estabelecida, que possuem capacidade de atendimento.

Existem várias estratégias que foram aplicadas na resolução do PPMC. Dentre os principais procedimentos propostos, pode-se citar a aplicação de Relaxação Lagrangeana (Senne et al., 2005a; Avella et al., 2012; Aguado e Trandafir, 2012; Rostami et al., 2016), método de Geração de Colunas (Porumbel e Goncalves, 2015; Erdoğan et al., 2016), método *branch-and-price* (Ahmadi et al., 2015; Senne et al., 2005b; Ceselli e Righini, 2005; Ceyhan e Özpeynirci, 2016; Ahmed et al., 2016), método *branch-and-cut* (Boccia et al., 2008; Quadros et al., 2018; Bektaş et al., 2019) e métodos heurísticos (Irawan et al., 2017; Isler et al., 2012; Hansen e Mladenović, 1997; Blum e Roli, 2003; Crainic et al., 2004; Díaz e Fernández, 2006; Fleszar e Hindi, 2008; Lozano e García-Martínez, 2010; Dzator e Dzator, 2013; Isler et al., 2012; Jánošíková et al., 2017; Irawan et al., 2017). É importante destacar que grande parte dessas estratégias de resolução são baseadas em métodos exatos. Entretanto, devido à dificuldade de se encontrar soluções ótimas em um tempo computacional adequado para instâncias relativamente grandes, algoritmos baseados em heurísticas e metaheurísticas têm sido propostos. A Tabela 1.1 resume os principais métodos de resolução aplicados ao

Tabela 1.1: Resumo dos métodos aplicados na resolução do PPMC

Autor	Estratégias de Solução
Senne et al. (2005a)	Relaxação Lagrangeana
Avella et al. (2012)	
Aguado e Trandafir (2012)	
Porumbel e Goncalves (2015)	Geração de Colunas
Erdoğan et al. (2016)	<i>Branch-and-Price</i>
Ahmadi et al. (2015)	
Senne et al. (2005b)	
Ceselli e Righini (2005)	
Ceyhan e Özpeynirci (2016)	<i>Branch-and-Cut</i>
Boccia et al. (2008)	
Quadros et al. (2018)	
Bektaş et al. (2019)	Métodos Heurísticos
Hansen e Mladenović (1997)	
Blum e Roli (2003)	
Crainic et al. (2004)	
Díaz e Fernández (2006)	
Fleszar e Hindi (2008)	
Lozano e García-Martínez (2010)	
Dzator e Dzator (2013)	
Isler et al. (2012)	
Isler et al. (2012)	
Jánošíková et al. (2017)	
Irawan et al. (2017)	

PPMC.

Uma outra abordagem para o Problema de Localização, que é o objeto de estudo desta tese, é o Problema das p -Medianas Robusto Capacitado (PPMRC). A abordagem de otimização robusta é um procedimento de otimização sob condições de incertezas, ou seja, quando parâmetros do problema de otimização são incertos na função objetivo, nas restrições ou em ambas as estruturas do problema. Desta forma, quando há incertezas, o resultado que se obtém ao considerar os dados determinísticos pode ou não ser viável. No PPMC determinístico, os dados de entrada como demanda, distância e capacidade das instalações são considerados conhecidos. Entretanto, esses dados podem ser incertos de acordo com o contexto. Nesta tese, consideramos as demandas dos clientes como dados incertos do problema. Neste caso, a suposição do PPMC que demandas dos clientes são parâmetros determinísticos, quando na realidade eles são incertos, pode fornecer uma solução inviável na prática, em que a demanda total real dos clientes atribuídos a uma determinada instalação pode exceder sua capacidade. Desta forma, é importante uma abordagem de otimização que considere a incerteza nesses parâmetros.

Esta tese propõe uma formulação matemática para modelar o PPMRC e algoritmos para a resolução do PPMC e do PPMRC. Para resolver o PPMC, foram propostas quatro variantes da metaheurística *General Variable Neighborhood Search* (GVNS), que se diferem no método usado para construir uma solução inicial e no método usado para realizar a busca local. Para a geração da solução inicial, foram consideradas construções randômicas e construções utilizando a fase de construção da metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP). Quanto ao procedimento de busca local, foram considerados os métodos *Variable Neighborhood Descent* (VND) e *Random Variable Neighborhood Descent* (RVND). Para a resolução do PPMRC, foi proposto um algoritmo baseado na metaheurística GVNS, com a solução inicial gerada através da fase construtiva do método GRASP e uma busca local realizada via VND. Para avaliar os algoritmos propostos, foram realizados experimentos computacionais usando instâncias da literatura de pequenas, médias e grandes dimensões.

1.2 **Objetivos Gerais e Específicos**

Esta seção descreve os objetivos gerais e específicos desta tese.

1.2.1 **Objetivo Geral**

O presente trabalho tem como objetivo geral a proposição de um modelo matemático para o Problema das p -Medianas Robusto Capacitado (PPMRC) e a proposição de algoritmos para resolver o Problema das p -Medianas Capacitado e o PPMRC.

1.2.2 **Objetivos Específicos**

Para que este objetivo geral seja alcançado, é necessário realizar os seguintes objetivos específicos:

- (i) Conhecer os principais trabalhos da literatura relacionados aos tema desta tese;
- (ii) Propor uma formulação de programação matemática para o PPMRC com incertezas nas demandas;
- (iii) Implementar a formulação proposta para o PPMRC;
- (iv) Desenvolver um algoritmo baseado na metaheurística GVNS para o PPMC;
- (v) Implementar o algoritmo GVNS para o PPMC;

- (vi) Desenvolver um algoritmo baseado na metaheurística GVNS para tratar instâncias de dimensões elevadas do PPMRC;
- (vii) Testar os algoritmos implementados e analisar estatisticamente os resultados obtidos.

1.3 Contribuições

A principal contribuição deste trabalho é a proposição e solução do Problema das p -Medianas Robusto Capacitado (PPMRC), considerando incertezas nas demandas dos clientes, por meio de um algoritmo baseado na metaheurística *General Variable Neighborhood Search* (GVNS).

A implementação do algoritmo metaheurístico GVNS proposto para resolver o PPMRC utiliza a fase construtiva do método GRASP e a heurística de busca local VND. Até a presente data, não encontramos na literatura um estudo que resolva o Problema Robusto de p -Mediana Capacitado com demanda incerta e que tenha sido resolvido por meio de metaheurísticas. Portanto, este é o primeiro estudo que abordou o PPMRC com incertezas de demanda e solução metaheurística. Podemos, ainda, destacar as principais contribuições deste estudo:

- (i) A proposição de um algoritmo eficiente para resolver o PPMC;
- (ii) A descrição do Problema Robusto Capacitado das p -Medianas sob Demanda Incerta;
- (iii) A proposição de um modelo matemático linear para o PPMRC com incertezas de demanda e sua solução através do solver CPLEX, com exemplo de soluções e análises comparativa dos resultados;
- (iv) A proposição de aplicação do algoritmo GVNS ao PPMRC utilizando a fase de solução inicial do método GRASP e busca local VND;
- (v) A realização de análise estatística comparativa da resolução do problema utilizando um *solver* e utilizando o algoritmo proposto.

1.4 Organização da Tese

O restante desta tese está organizada da seguinte forma. O Capítulo 2 apresenta a caracterização do problema de p -Medianas não capacitado e sua versão capacitada, assim como o Problema das p -Medianas Robusto Capacitado. Apresenta também

como é feita a linearização da formulação robusta e mostra um exemplo para o Problema das p -Medianas Robusto Capacitado. O Capítulo 3 apresenta uma revisão de heurísticas, metaheurísticas e metaheurísticas híbridas. Apresenta também uma revisão da metaheurística *Variable Neighborhood Search* (VNS) (Hansen et al., 2010, 2017) e da metaheurística *Greedy Randomized Adaptive Search Procedure– GRASP* (Feo e Resende, 1995). Além disso, apresenta uma revisão da literatura para o Problema das p -Medianas Capacitado (PPMC) e de aplicações de otimização robusta a problemas de localização. O Capítulo 4 mostra o algoritmo proposto para o solucionar o Problema das p -Medianas Capacitado, bem como a representação e avaliação das soluções, randômica e via fase de construção da heurística GRASP. Também descreve as estruturas de vizinhanças de realocação, troca e substituição. Os procedimentos de busca local utilizados pelo algoritmo são *Variable Neighborhood Descent* (VND) e *Random Variable Neighborhood Descent* (RVND). Descreve também como realizar um processo de perturbação que diversifica a busca por outros pontos do espaço de solução do problema. Por fim, são apresentados experimentos computacionais com diversos exemplos da literatura. O Capítulo 5 apresenta o algoritmo híbrido baseado na metaheurística GVNS e o método para gerar a solução inicial através de uma adaptação da fase de construção da metaheurística GRASP. Este capítulo também realiza uma série de experimentos computacionais em 7 grupos de instâncias PPMC na literatura, com um grande número de nós e medianas (grupos de 35 instâncias, variando de 318 nós e 5 medianas a 4461 nós e 1000 medianas), e compara os resultados gerados pelo *solver* CPLEX e o algoritmo GVNS. Por fim, no Capítulo 6, estão as conclusões gerais e proposições para trabalhos futuros.

Capítulo 2

Caracterização do Problema

Este Capítulo apresenta a caracterização do Problema das p -Medianas Capacitado (PPMC) e do Problema das p -Medianas Robusto Capacitado (PPMRC). Na Seção 2.2, é apresentada a formulação do Problema das p -Medianas não capacitado. A Seção 2.3 apresenta a descrição e formulação para o PPMC, cujo modelo é caracterizado pelas restrições de capacidade. Já a Seção 2.4 apresenta uma formulação para o PPMRC.

2.1 Problema das p -medianas

Problemas de Localização têm recebido significativa atenção nas últimas décadas. Isto se deve à grande importância assumida por problemas de decisão na forma de determinação da localização de pontos de demanda de serviços, também conhecidos como “facilidades”, em uma região de interesse. Esses pontos são considerados os mais adequados para se instalar esses serviços, nos termos postos em [Mladenović et al. \(2007\)](#).

Localizações desta natureza são importantes, seja para o atendimento de clientes de uma empresa de forma mais eficaz, ou também para a gestão de resíduos sólidos ([Sahin e Süral, 2007](#)), em sistemas de distribuição ([Klose e Drexler, 2005](#)), sistemas de Ensino ([Pizzolato et al., 2004](#)), problema de corte guilhotinado ([Müller e Gómez, 2007](#)), sistemas de informações geográficas ([Lorena et al., 2001](#)), alocação de antenas de transmissão ([Arroyo et al., 2006](#)) ou localização de unidades de atendimento ([Abreu et al., 2008](#)), monitoramento da poluição do ar ([Lancia et al., 2018](#)), localização de instalações de serviço médico de emergência ([Deng et al., 2021](#)), localização de instalações em serviços de impressão e entrega de encomendas ([Avgerinos et al., 2022](#)), dentre outros.

Diversos fatores podem ser considerados na decisão de localização. Por exemplo,

disponibilidade de matéria-prima para instalação de uma fábrica ou instalações que estão sujeitas à matéria-prima. Há casos em que as decisões de localização são baseadas na disponibilidade de mão-de-obra abundante ou de mão-de-obra intensiva em uma certa região. Além disso, decisões de localização podem avaliar se as instalações devem ser localizadas próximas a áreas sujeitas ao maior volume de demandas, como lojas de varejo e bancos. Existem localizações que podem depender de meios de transportes, como, por exemplo, proximidade a portos, aeroportos, rodovias ou terminais de cargas. Outro fator relevante a ser considerado é no tocante a decisões de localização que dependem de subsídios governamentais, que podem ser oferecidos para localizar certos tipos de indústrias em regiões específicas de um país. Portanto, conclui-se que existem diversos fatores que contribuem para que o local de uma determinada instalação seja escolhido.

Devido a isso, a classe de problemas de localização é extremamente ampla e diversas revisões de literatura mostram essa gama de temas que podem ser abordados por essa classe de problemas. Dentre elas, podemos citar, (Melo et al., 2009), (Terouhid et al., 2012), (Hosseini-Nasab et al., 2017), (Junzo et al., 2019) e (Liu et al., 2021), que realizam uma extensa revisão sobre o problema de localização.

O Problema das p -medianas (PPM) é um dos problemas mais significativos dessa classe geral de problemas. Esta categoria de problema é descrita pela primeira vez em (Hakimi, 1964). O objetivo deste problema é localizar p nós, denominados medianas, em um grafo contendo um conjunto de nós, de forma a obter a menor soma das distâncias de cada nó até a mediana mais próxima. Uma vez localizadas as diversas medianas, são constituídos agrupamentos (*clusters*), nos quais cada mediana é alocada a um certo número de nós de demanda.

Segundo Garey e Johnson (1979), o PPM é um problema NP-difícil. Além disso, é um problema de Otimização Combinatória que apresenta, em geral, um elevado número de variáveis e restrições. Esses fatos justificam a utilização de técnicas heurísticas na busca por uma solução, pois tais métodos podem, mesmo sem garantir otimalidade, gerar bons resultados em tempos computacionais aceitáveis para a tomada de decisão, quando comparados à resolução através de métodos exatos de otimização.

Duas versões do PPM são tratadas na literatura:

- (i) Problema das p -Medianas não-Capacitado (PPMnC), que não considera a capacidade de atendimento de uma determinada mediana;
- (ii) Problema das p -Medianas Capacitado (PPMC), em que as restrição de capacidade de atendimento de cada mediana são levadas em consideração.

Essas duas versões serão descritas a seguir.

2.2 Problema das p -Medianas não Capacitado

Esta seção apresenta uma formulação matemática para o Problema de p -Medianas não capacitado (PPMnC), baseada em Lorena et al. (2001), Senne e Lorena (2003) e Crainic et al. (2004). Seja $\mathcal{G} = (\mathcal{V}, \mathcal{A})$ um grafo não direcionado, em que \mathcal{V} é o conjunto dos nós e \mathcal{A} é o conjunto das arestas do grafo. O número de nós do grafo é denotado por $n = |\mathcal{V}|$. O parâmetro p , tal que $p \leq n$, representa o número de instalações que serão selecionados para serem medianas. A distância (ou custo de transporte) entre os nós $i \in \mathcal{V}$ e $j \in \mathcal{V}$ será denotada por d_{ij} , sendo $d_{ij} > 0$ e $d_{jj} = 0$.

O PPMnC consiste em determinar um conjunto $\mathcal{V}_p \subseteq \mathcal{V}$, que representa o conjunto de p -medianas escolhidas dentre o conjunto de instalações \mathcal{V} , de modo a minimizar a distância total de cada nó à mediana mais próxima. Neste caso, cada nó é alocado à mediana mais próxima.

Seja x_{ij} a variável de decisão binária que assume valor $x_{ij} = 1$ se o nó j é o alocado à mediana i e $x_{ij} = 0$, caso contrário. Além disso, $x_{ii} = 1$ se o nó i for definido como mediana e $x_{ii} = 0$, caso contrário.

Assim sendo, a formulação matemática para o PPMnC pode ser dada por:

$$\min \sum_{i=1}^n \sum_{j=1}^n d_{ij} x_{ij} \quad (2.1)$$

$$\text{sujeito a: } \sum_{i=1}^n x_{ij} = 1 \quad \forall j \in N \quad (2.2)$$

$$\sum_{i=1}^n x_{ii} = p \quad (2.3)$$

$$x_{ij} \leq x_{ii} \quad \forall i, j \in N \quad (2.4)$$

$$x_{ij} \in \{0,1\} \quad \forall i, j \in N \quad (2.5)$$

No modelo, a expressão (2.1) corresponde à função objetivo do problema. As restrições (2.2) e (2.4) garantem que cada nó j seja alocado a apenas um nó i , que deve corresponder a uma mediana. A restrição (2.2) determina o número exato de medianas a serem selecionadas e as restrições (2.5) corresponde às condições de integralidade das variáveis de decisão envolvidas.

A Figura 2.1(a) apresenta um exemplo do PPMnC. Os dados correspondem a uma instância real, mostrada por um grafo \mathcal{G} com 10 vértices. O grafo apresentado é completo, não direcionado e totalmente conectado, conforme descrito na definição do PPM. Deseja-se encontrar três medianas entre os 10 vértices pertencentes a \mathcal{G} e, para isso, os seguintes elementos do problema são considerados:

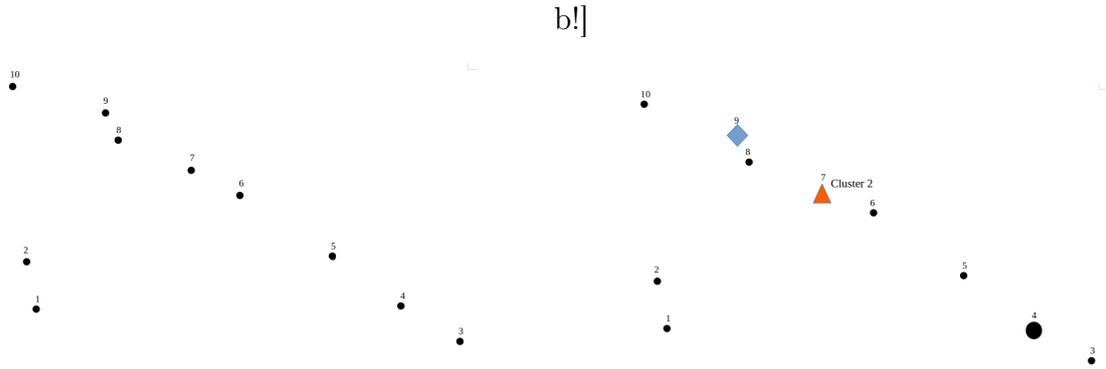
(a) Exemplo de grafo \mathcal{G} com 10 vértices.(b) Exemplo de *cluster* e suas medianas.

Figura 2.1: Exemplo de uma solução.

Tabela 2.1: Matriz de distância da instância do exemplo ilustrativo

	1	2	3	4	5	6	7	8	9	10
1	0,00	155,03	163,69	106,47	187,84	373,69	456,22	550,52	643,31	728,03
2	155,03	0,00	291,62	181,40	89,81	222,80	302,83	395,92	488,54	573,01
3	163,69	291,62	0,00	119,22	277,48	481,27	567,39	665,41	758,03	846,05
4	106,47	181,40	119,22	0,00	158,27	362,06	448,17	546,19	638,82	726,84
5	187,84	89,81	277,48	158,27	0,00	203,80	289,91	387,99	480,55	568,66
6	373,69	222,80	481,27	362,06	203,80	0,00	86,15	184,35	276,76	365,02
7	456,22	302,83	567,39	448,17	289,91	86,15	0,00	98,27	190,65	278,89
8	550,52	395,92	665,41	546,19	387,99	184,35	98,27	0,00	93,09	180,68
9	643,31	488,54	758,03	638,82	480,55	276,76	190,65	93,09	0,00	89,19
10	728,03	573,01	846,05	726,84	568,66	365,02	278,89	180,68	89,19	0,00

- (i) d : matriz das distâncias entre os vértices do grafo \mathcal{G} , mostrada na Tabela 2.1;
- (ii) w : vetor de pesos associados ao vértices de demanda. Para esse exemplo, considera-se que $w=1$;
- (iii) σ : vetor dos valores de transmissão de cada vértice v_i .

Conforme a Figura 2.1(a), tem-se, então, que:

$$\sigma = dw \quad (2.6)$$

Para o grafo \mathcal{G} apresentado, as medianas $\{4, 7, 9\}$ foram selecionadas e cada uma será agregada a um determinado número de vértices. Neste caso, são instituídos os agrupamentos (*cluster*), de modo que cada *cluster* apresentará apenas uma única mediana.

A solução ótima, apresentada na Figura 2.2, mostra que os clientes $\{1, 2, 3, 5\}$ foram alocados à mediana 1; o cliente $\{6\}$ alocado à mediana 7; e, por fim, os clientes $\{8, 10\}$ foram alocados à mediana 9. A Tabela 2.2, apresenta os custos de cada *Cluster*, tendo como referência as distâncias da Tabela 2.1. O valor da menor distância para esta solução é 833,79.

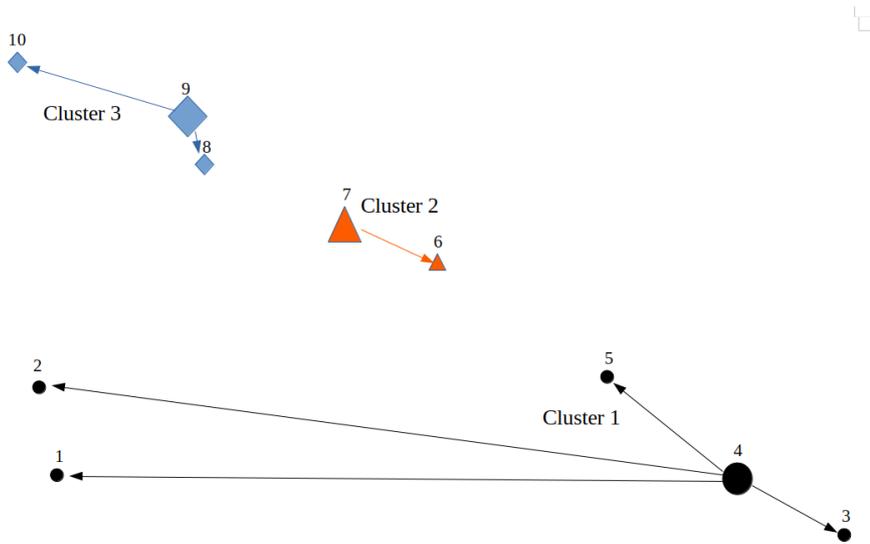


Figura 2.2: Exemplo de um grafo \mathcal{G} com 10 vértices e três medianas.

Tabela 2.2: Cálculo da distância total para o Exemplo da Figura 2.1(b).

<i>Cluster</i>	<i>Distância</i>
1	565,36
2	86,15
3	182,28
Func. Obj.	833,79

2.3 Problema das p -Medianas Capacitado

Esta seção descreve o Problema de p -Medianas Capacitado (PPMC). De acordo com [Yaghini et al. \(2013\)](#), o PPMC é definido a partir de um grafo \mathcal{G} não direcionado $\mathcal{G} = (\mathcal{V}, \mathcal{A})$, em que \mathcal{V} é o conjunto dos nós e \mathcal{A} é o conjunto das arestas do grafo. Seja então \mathcal{N} o conjunto de instalações e \mathcal{M} o conjunto de clientes, de modo que $\mathcal{N} \subseteq \mathcal{V}$ e $\mathcal{M} \subseteq \mathcal{V}$, sendo $n = |\mathcal{N}|$ o número máximo de possíveis instalações e $m = |\mathcal{M}|$ o número de clientes a serem atendidos. É interessante salientar que as formulações em que $\mathcal{N} \subseteq \mathcal{M}$ ou, no limite, $\mathcal{N} = \mathcal{M}$, são situações particulares da formulação adotada. A cada cliente $j \in \mathcal{M}$ está associada uma demanda q_j e a cada instalação $i \in \mathcal{N}$ está associada uma capacidade máxima de atendimento Q_i . O custo do transporte de uma unidade de um produto de uma determinada instalação $i \in \mathcal{N}$ para um cliente $j \in \mathcal{M}$ é proporcional à distância d_{ij} entre as localizações i e j , sendo $d_{ij} > 0$ e $d_{jj} = 0$. O valor $p \leq n$ define o número de instalações que serão denominadas como medianas.

O PPMC consiste, assim, em determinar um conjunto $\mathcal{N}_p \subseteq \mathcal{N}$, que representa

o conjunto de p -Medianas escolhidas dentre o conjunto de instalações \mathcal{N} , de modo a minimizar a distância total entre os clientes e suas respectivas instalações, asseguradas as capacidades das instalações. Seja y_i a variável de decisão binária que representa a decisão de seleção da instalação i como mediana, de modo que, se $y_i = 1$, então a instalação i foi definida como mediana; e $y_i = 0$, caso contrário. Seja x_{ij} a variável de decisão binária que assume valor $x_{ij} = 1$ se o cliente j é atendido pela instalação i ; e o valor $x_{ij} = 0$, caso contrário.

Então, a formulação matemática do PPMC é dada por:

$$(PPMC) \quad \min \quad \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} \quad (2.7)$$

$$\text{suj. a} \quad \sum_{i \in \mathcal{N}} x_{ij} = 1, \quad \forall j \in \mathcal{M} \quad (2.8)$$

$$\sum_{i \in \mathcal{N}} y_i = p \quad (2.9)$$

$$\sum_{j \in \mathcal{M}} q_j x_{ij} \leq Q_i y_i, \quad \forall i \in \mathcal{N} \quad (2.10)$$

$$x_{ij} \in \{0,1\}, \quad \forall i \in \mathcal{N}, \quad \forall j \in \mathcal{M} \quad (2.11)$$

$$y_i \in \{0,1\}, \quad \forall i \in \mathcal{N} \quad (2.12)$$

Uma solução ótima neste problema corresponde a um *cluster* viável, dado pelas instalações em \mathcal{N}_p com menor distância total entre todas as instalações e seus respectivos clientes. A função objetivo (2.7), a ser minimizada, representa a soma dos custos de todos os clientes j a todas as instalações i . As restrições (2.8) garantem que cada cliente é atendido por apenas uma instalação, enquanto que a restrição (2.9) garante que a instalação i só pode ser escolhida se ela estiver ativa. As restrições (2.10) impõem que a capacidade total das instalações deve ser respeitada. As restrições (2.11) e (2.12) definem o domínio das variáveis de decisão.

A Figura 2.3 apresenta a solução ótima para o mesmo exemplo da Seção 2.2, considerando agora, a capacidade de atendimento de cada instalação i . A Tabela 2.3 apresenta as demandas q_j de cada cliente. Tendo em vista a matriz de distancias da Tabela 2.1 e a capacidade de atendimento $Q_i = 132$, temos agora as instalações $\{4, 7, 8\}$ e seus respectivos clientes $\{1, 2, 3, 5\}$, $\{9\}$ e $\{6, 10\}$. A Tabela 2.3 apresenta as demandas atendidas em cada *Cluster* e o valor ótimo da menor distância. Observa-se que, ao considerar uma demanda associada a cada instalação i , o valor ótimo da função objetivo aumenta. De fato, ao considerarmos as demandas associadas aos clientes j e suas capacidades de atendimento, não avalia-se somente a menor distância, como no problema das PPMnC, apresentado na Seção 2.2, mas, também, considera-se as restrições de capacidade de atendimento das instalações i às demandas dos clientes

j . Isso demonstra que a restrição de capacidade é uma restrição forte para o PPMC e que influencia diretamente no valor da função objetivo.

Tabela 2.3: Valores das demandas dos cliente j .

Cliente j	1	2	3	4	5	6	7	8	9	10
Demanda q_j	50	4	33	15	1	5	87	91	45	1

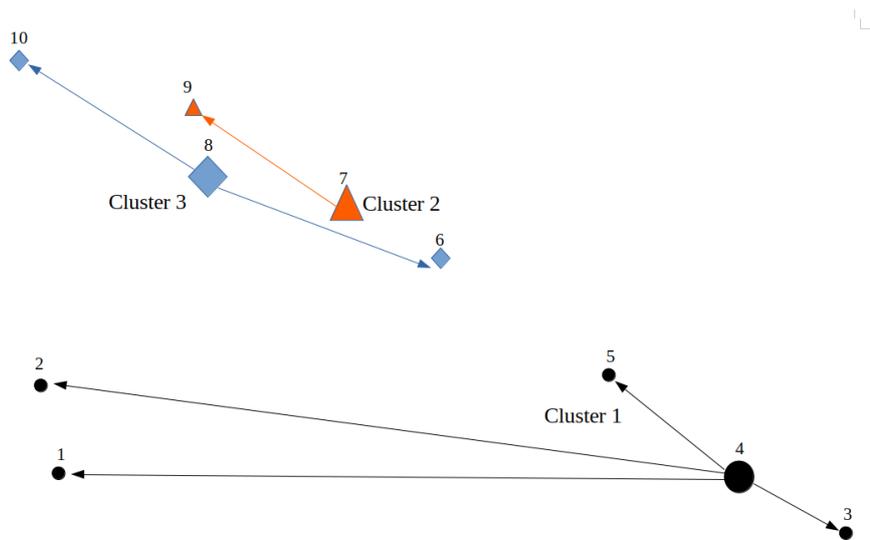
Figura 2.3: Exemplo de grafo \mathcal{G} com 10 vértices.

Tabela 2.4: Cálculo da distância total para o Exemplo da Figura 2.3.

Cluster	Distância	Demanda
1	565,36	103
2	190,65	132
3	365,03	97
Func. Obj.	1121,04	

A Figura 2.3 apresenta a solução ótima para o PPMC, na qual foram selecionados, respectivamente, os clientes $\{1, 2, 3, 5\}$, $\{9\}$ e $\{6, 10\}$. A Tabela 2.4 apresenta os custos de cada *Cluster*, considerando as demandas associadas a cada cliente j . O valor da menor distância para esta solução é 1121,04.

2.4 Problema das p -Medianas Robusto Capacitado

O Problema das p -Medianas Robusto Capacitado (PPMRC) pode ser definido como segue. Considere que a demanda q_j do cliente $j \in \mathcal{M}$ é um parâmetro incerto,

pertencente ao intervalo $[q_j^-, q_j^+]$. Para obter uma solução robusta quando as demandas dos clientes são incertas, é necessário modificar apenas o conjunto de restrições de capacidade (2.10), que representa desigualdades que podem não ser satisfeitas para os valores reais de demanda.

Seja \hat{q}_j o valor central do intervalo que a incerteza na demanda e r_j o maior desvio possível deste valor central. Considerando o intervalo $[q_j^-, q_j^+]$, pode-se estabelecer:

$$\hat{q}_j = (q_j^- + q_j^+)/2 \quad (2.13)$$

e

$$r_j = q_j^+ - \hat{q}_j. \quad (2.14)$$

Portanto, a demanda q_j do cliente $j \in \mathcal{M}$ é um valor incerto $q_j \in [\hat{q}_j - r_j, \hat{q}_j + r_j]$. O modelo robusto mais conservador para essa configuração de demandas incertas é o modelo robusto que busca uma solução que satisfaça a restrição de capacidade para todos os valores possíveis de demandas dentro desses intervalos.

Para encontrar a solução ótima para este modelo robusto, basta resolver o problema definido pelas Expressões (2.7)-(2.12), assumindo que as demandas dos clientes são as maiores possíveis, ou seja, $q_j = q_j^+, \forall j \in \mathcal{M}$, pois esses são os valores de demanda mais críticos para o atendimento das restrições de capacidade. No entanto, assumindo a situação em que todos os valores de demanda real são os mais altos possíveis, então este problema robusto pode ser considerado muito conservador, uma vez que esta situação dificilmente ocorreria.

Assim, [Bertsimas e Sim \(2003\)](#) propuseram um modelo robusto que assume que o nível de variação de todos os parâmetros incertos é limitado por um parâmetro $\Gamma \in \mathbb{R}_+$, sendo Γ o número de parâmetros incertos. Esta abordagem considera que o conjunto de incertezas na demanda para um determinado valor de Γ , denotado por $\mathcal{U}(\Gamma)$, pode ser representado, segundo [Bertsimas et al. \(2011\)](#), por:

$$\mathcal{U}(\Gamma) = \left\{ q \in \mathbb{R}_+^m : q_j^- \leq q_j \leq q_j^+, \forall j \in \mathcal{M}, \sum_{j \in \mathcal{M}} \frac{|q_j - \hat{q}_j|}{r_j} \leq \Gamma \right\}. \quad (2.15)$$

O parâmetro Γ é responsável por ajustar a robustez do método proposto ([Bertsimas e Sim, 2004](#)). Se $\Gamma = 0$, o problema é equivalente ao problema nominal. Se $\Gamma = m$, o problema robusto mostra o cenário mais conservador possível, assumindo a possibilidade de que todos os parâmetros incertos atinjam o pior caso. Se $0 < \Gamma < m$, os cenários de configuração de demanda mais críticos para atender às restrições de capacidade dentro do conjunto de incertezas são os casos em que o valor das demandas de Γ clientes é assumido como sendo o pior caso, ou seja, têm o maior valor possível,

sendo o valor das demais demandas igual ao seu valor nominal. Assim, o problema robusto, assumindo o atendimento de todos os valores de demandas em um conjunto de incertezas $\mathcal{U}(\Gamma)$ definido por um parâmetro Γ , é dado por:

$$\text{(PPMRC)} \quad \min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} \quad (2.16)$$

$$\text{suj. a} \quad \sum_{i \in \mathcal{N}} x_{ij} = 1, \quad \forall j \in \mathcal{M} \quad (2.17)$$

$$\sum_{i \in \mathcal{N}} y_i = p \quad (2.18)$$

$$\sum_{j \in \mathcal{M}} \hat{q}_j x_{ij} + \max_{\substack{\mathcal{S} \subseteq \mathcal{M} \\ |\mathcal{S}| = \Gamma}} \left\{ \sum_{j \in \mathcal{S}} r_j x_{ij} \right\} \leq Q_i y_i, \quad \forall i \in \mathcal{M} \quad (2.19)$$

$$x_{ij} \in \{0,1\}; \forall i \in \mathcal{N}, j \in \mathcal{M} \quad (2.20)$$

$$y_i \in \{0,1\}; \forall i \in \mathcal{N} \quad (2.21)$$

A opção de escrita aqui feita é mostrar a representação matemática como autocontida, sem que seja necessário buscar outras formulações para que ela seja compreendida. Assim, as restrições (2.17), (2.18), (2.20) e (2.21) equivalem às restrições (2.8), (2.9), (2.11) e (2.12), respectivamente, do modelo (PPMC) mostrado na Seção 2.3. Nesta formulação, a função objetivo (2.16) minimiza a distância total dos clientes às medianas às quais foram alocados. As restrições (2.19) garantem que a solução encontrada será viável para todos os valores de demanda pertencentes ao conjunto de incerteza $\mathcal{U}(\Gamma)$, definido pela expressão (2.15), considerando os cenários mais críticos para atender às restrições de capacidade, que é o caso em que as demandas Γ têm o maior valor possível $q_j^+ = \hat{q}_j + r_j$, enquanto assume-se que as demais demandas são iguais ao valor central \hat{q}_j . Entretanto, o problema robusto (2.16)-(2.21) é não linear, devido à maximização interna nestas restrições, o que impede que sua solução seja alcançada por meio de um *solver* de programação linear inteira mista.

Diante disso, uma alternativa para a determinação de solução é a linearização do problema robusto (2.16)-(2.19). Isso é apresentado a seguir.

2.4.1 Linearização da Formulação Robusta

A linearização do problema robusto (2.16)-(2.19) é realizada da seguinte forma (Bertsimas e Sim, 2003). Para cada instalação $i \in \mathcal{N}$ e considerando cada atribuição de solução $x = (x_{ij})$, a maximização interna na restrição (2.19) pode ser representada

como:

$$v_i(x) = \max_{\substack{\mathcal{S} \subseteq \mathcal{M} \\ |\mathcal{S}| = \Gamma}} \left\{ \sum_{j \in \mathcal{S}} r_j x_{ij} \right\}. \quad (2.22)$$

Seja z_j uma variável binária que indica se o parâmetro incerto q_j atinge o limite superior ($z_j = 1$) ou não ($z_j = 0$). Então, as variáveis z_j podem ser usadas para representar o conjunto $\mathcal{S} \subseteq \mathcal{M}$ com uma cardinalidade Γ em $v_i(x)$. Desta forma, a função $v_i(x)$ pode ser representada pelo seguinte problema linear, para cada $i \in \mathcal{N}$:

$$v_i(x) = \max \sum_{j \in \mathcal{M}} r_j x_{ij} z_j \quad (2.23)$$

$$\text{suj. a } \sum_{j \in \mathcal{M}} z_j \leq \Gamma \quad (2.24)$$

$$0 \leq z_j \leq 1, \quad \forall j \in \mathcal{M}. \quad (2.25)$$

Observe que, mesmo que a variável z_j seja definida como uma variável contínua, sempre haverá uma solução ótima para este problema na qual todas as variáveis são binárias. Dados os valores da variável de atribuição $x = (x_{ij})$, a função objetivo (2.23) maximiza a soma do maior desvio referente aos clientes que estão alocados à mediana i . As restrições (2.24) garantem que a quantidade de parâmetros incertos que atingem o maior desvio não exceda seu limite máximo definido por Γ .

Associando as variáveis duais β_i e ρ_{ij} , respectivamente, às restrições (2.24) e (2.25) do subproblema referente à mediana $i \in \mathcal{N}$, temos o seguinte problema dual $d_i(x)$, para cada $i \in \mathcal{N}$:

$$d_i(x) = \min \Gamma \beta_i + \sum_{j \in \mathcal{M}} \rho_{ij} \quad (2.26)$$

$$\text{suj. a } \beta_i + \rho_{ij} \geq r_j x_{ij}, \quad \forall j \in \mathcal{M} \quad (2.27)$$

$$\rho_{ij} \geq 0, \quad \forall j \in \mathcal{M} \quad (2.28)$$

$$\beta_i \geq 0 \quad (2.29)$$

Pela Teorema da Dualidade Forte (Bazaraa et al., 2010), uma vez que o problema (2.23)-(2.25) é viável e limitado para todo $\Gamma \in [0, m]$, o problema dual (2.26)-(2.29) também é viável e limitado e seus valores de função objetivo coincidem, ou seja, $d_i(x) = v_i(x)$. Portanto, a solução ótima do problema robusto (2.16)-(2.19) pode

ser encontrada por meio do seguinte problema de otimização linear inteira mista:

$$\min \sum_{i \in \mathcal{N}} \sum_{j \in \mathcal{M}} d_{ij} x_{ij} \quad (2.30)$$

$$\text{suj. a } \sum_{i \in \mathcal{N}} x_{ij} = 1; \forall j \in \mathcal{M} \quad (2.31)$$

$$\sum_{i \in \mathcal{N}} y_i = p \quad (2.32)$$

$$\sum_{j \in \mathcal{M}} \hat{q}_j x_{ij} + \Gamma \beta_i + \sum_{j \in \mathcal{M}} \rho_{ij} \leq Q_i y_i, \quad \forall i \in \mathcal{M} \quad (2.33)$$

$$\beta_i + \rho_{ij} \geq r_j x_{ij}, \quad \forall i \in \mathcal{N}, j \in \mathcal{M} \quad (2.34)$$

$$\rho_{ij} \geq 0, \quad \forall j \in \mathcal{M} \quad (2.35)$$

$$\beta_i \geq 0, \forall i \in \mathcal{N}, \quad (2.36)$$

$$x_{ij} \in \{0,1\}; \forall i \in \mathcal{N}, j \in \mathcal{M} \quad (2.37)$$

$$y_i \in \{0,1\}; \forall i \in \mathcal{N} \quad (2.38)$$

Note que as restrições (2.33) são resultantes da substituição da maximização interna das restrições (2.19) pela expressão da função objetivo (2.26). Além disso, todas as restrições do problema dual (2.26)-(2.29) estão incluídas no problema robusto linearizado.

2.4.2 Exemplo do PPMRC

Esta seção se dedica a apresentar um exemplo do Problema de p -Medianas Robusto Capacitado para uma instância particular desse problema. A instância de teste tem um conjunto de 10 nós, que representam clientes e medianas em potencial. A Tabela 2.5 mostra os parâmetros de entrada referentes a demandas incertas, apresentando, para cada cliente j , o valor de sua demanda central \hat{q}_j e o valor r_j do maior desvio para a demanda central, que se supõe igual a 10% desse valor de demanda central. A Tabela 2.1 apresenta a matriz de distâncias. A capacidade de cada mediana potencial é dada por $Q_i = 132$ e devem ser instaladas $p = 3$ medianas.

Tabela 2.5: Valores das demandas centrais do cliente e os maiores desvios de 10% das demandas centrais.

j	1	2	3	4	5	6	7	8	9	10
\hat{q}_j	50	4	33	15	1	5	87	91	45	1
r_j	5,50	0,40	3,30	1,50	0,10	0,50	8,70	9,10	4,50	0,10

Considere a solução ótima para $\Gamma = 2$ e o maior desvio de 10% da demanda central. Os nós 2, 3 e 8 foram selecionados como medianas e os conjuntos de nós

$\{1, 2, 4, 5, 9\}$, $\{3, 7\}$ e $\{6, 8, 10\}$ foram atribuídos às medianas 2, 3 e 8, respectivamente. Considerando esta solução e os parâmetros de entrada apresentados na Tabela 2.1 e na Tabela 2.5, os resultados para o problema robusto são apresentados nas Tabelas 2.6 e 2.7 para $\Gamma = 2$. A Tabela 2.6 apresenta os maiores desvios de demanda para cada cliente atendido por cada mediana, bem como a demanda central total atendida e o maior desvio Γ alcançado. A Tabela 2.7 apresenta, para cada mediana, a soma das demandas centrais atendidas com os Γ maiores desvios de demanda. Estas tabelas mostram que esta instância é viável.

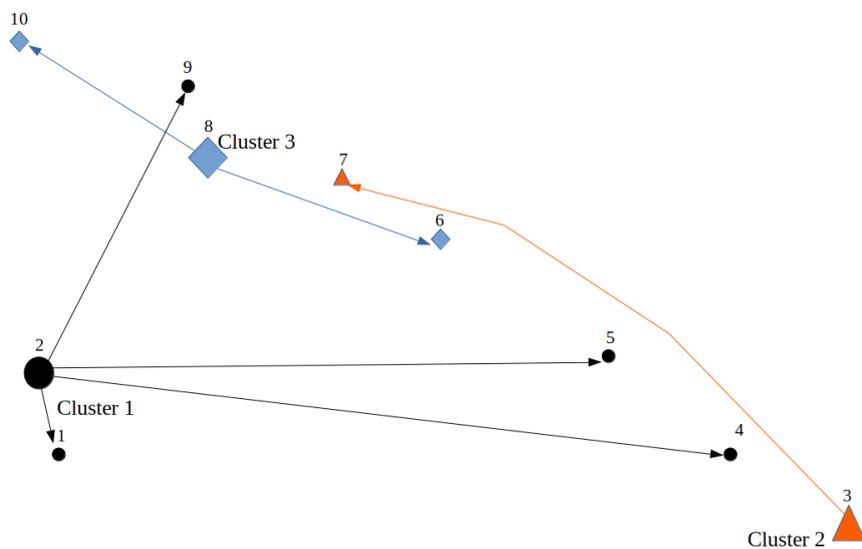


Figura 2.4: Exemplo de grafo \mathcal{G} com 10 vértices.

Tabela 2.6: Valores de demanda central e os maiores valores de desvio considerando $\Gamma = 2$ e $\Delta r = 10\%$.

Desvios			Demanda central			Γ Maiores desvios		
2	3	8	2	3	8	2	3	8
5,50	3,30	0,50	115,00	120,00	97,00	5,50	3,30	9,10
0,40	8,70	9,10				4,50	8,70	0,50
1,50		0,10						
0,10								
4,50								

A Tabela 2.8 apresenta as soluções ótimas para a instância de teste, considerando diferentes valores de maiores desvios e diferentes valores para o parâmetro de robustez Γ . O objetivo destes experimentos é mostrar o comportamento do modelo PPMRC para diferentes valores desses parâmetros. A primeira coluna desta tabela

Tabela 2.7: Soma das demandas centrais atendidas por cada mediana com os maiores desvios Γ .

2	3	8
125,00	132,00	106,60

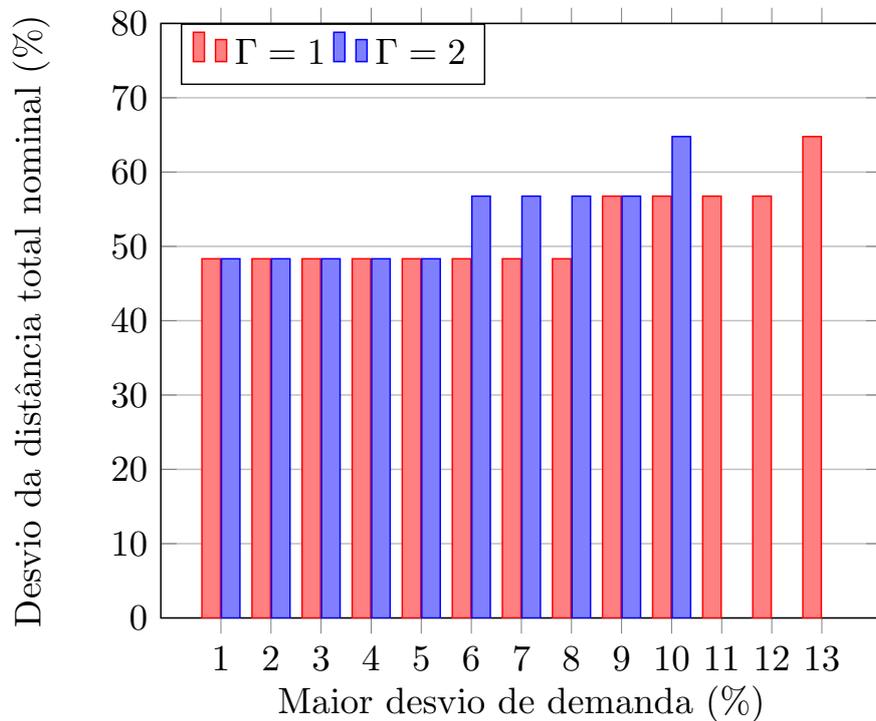
mostra o valor de Γ . A segunda coluna exhibe os valores do maior desvio percentual (Δr) em (%). Os valores para Δr e Γ que resultaram na mesma solução foram colocados na mesma linha. As colunas 3-5 apresentam as medianas instaladas. As colunas 6-15 mostram qual mediana foi atribuída a cada cliente. Por fim, a última coluna apresenta os valores da função objetivo.

De acordo com a Tabela 2.8, a instalação 2 foi selecionada em todos os casos, exceto o problema nominal, ou seja, para o caso $\Gamma = 0$. Também é importante observar que diferentes níveis de robustez e maior desvio resultam em diferentes seleções de medianas e, conseqüentemente, em diferentes atribuições de clientes. Observe que, por exemplo, para $\Gamma = 1$, ao aumentar o desvio de 8% para 13%, o nó 3 foi escolhido como mediana em vez do nó 6. A mediana 3 é atribuída a todos os clientes atendidos pela mediana 6, exceto o cliente 6, que está alocado na mediana 8. Além disso, observa-se também que, para incertezas maiores ($\Delta r = 10$), as medianas 2, 3 e 8 são sempre selecionadas, exceto quando $\Gamma = 4$.

A Figura 2.5 apresenta o aumento percentual na distância total para a solução robusta para atender as demandas dos clientes em relação à distância total para a solução ótima do problema nominal, para $\Gamma \in \{1,2\}$ e maiores desvios de demanda correspondente a $\Delta r \in \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$. A distância total para as soluções ótimas robustas é 48,33%, 56,77% ou 64,78% maior que a da solução ótima para o problema nominal, de acordo com o parâmetro de robustez e variação de demanda. Como esperado, maiores valores de parâmetros de robustez estão associados a distâncias totais ótimas iguais ou maiores que para parâmetros de menor robustez, e maiores variações na demanda correspondem a distâncias totais ótimas maiores ou iguais a menores variações na demanda. Para variações de até 5% das demandas, as distâncias totais para as soluções robustas ótimas para robustez correspondentes a $\Gamma = 1$ e $\Gamma = 2$ são iguais. Ao considerar variações de até 6%, a distância total ótima é aumentada para $\Gamma = 2$, enquanto para $\Gamma = 1$ a distância total é igual à distância total correspondente a até 5% de variação. Além disso, para $\Gamma = 2$, o modelo robusto não encontra solução viável para variações de demanda maiores que 10% da demanda nominal, enquanto para $\Gamma = 1$ o modelo robusto poderia encontrar uma solução para variações de 13% com uma distância total de 64,78% maior que a distância total para a solução nominal ótima.

Tabela 2.8: Resultados do exemplo ilustrativo para diferentes valores de Γ e desvio.

Γ	Δr (%)	Medianas	Designações	f_o
0	0	4 7 8	4 4 4 4 4 8 7 8 7 8	1121,03
1	1-8	2 6 8	2 2 6 2 2 6 6 8 2 8	1662,87
	9-12	2 5 8	2 2 5 2 5 8 5 8 2 8	1757,39
	13	2 3 8	2 2 3 2 2 8 3 8 2 8	1847,19
2-10	1-5	2 6 8	2 2 6 2 2 6 6 8 2 8	1662,87
	6-9	2 5 8	2 2 5 2 5 8 5 8 2 8	1757,39
	10	2 3 8	2 2 3 2 2 8 3 8 2 8	1847,19

Figura 2.5: Desvio percentual da distância total para a solução robusta em relação à distância total relacionada à solução do problema nominal para $\Gamma = 1$ e 2.

2.5 Considerações Parciais

Este Capítulo apresentou a caracterização do Problema das p -Medianas (PPM), uma formulação para o Problema de p -Medianas não capacitado (PPMnC) e a descrição e uma formulação para o Problema das p -Medianas Capacitado (PPMC), cujo modelo é caracterizado pela inclusão das restrições de capacidade. Como principal contribuição, este Capítulo apresentou uma formulação para o Problema das p -Medianas Robusto Capacitado (PPMRC).

É importante observar que o PPMRC apresenta características bem particulares em relação às demais formulações do PPMC, visto que as soluções para esse pro-

blema considera a existência de demandas incertas. Desta forma, busca-se soluções que sejam viáveis considerando diferentes cenários de incertezas, mesmo que estejam distantes da solução ótima para o PPMC.

Entretanto, o PPMRC é um problema de otimização com características importantes e únicas, que o tornam mais desafiador em relação ao problema determinístico. Isso, de certa forma, limita a utilização de métodos exatos de otimização para a resolução de problemas com instâncias de dimensões mais elevadas. Estas limitações se devem ao fato de que, segundo [Senne e Lorena \(2003\)](#), o tempo de resolução aumenta exponencialmente à medida que se aumenta a dimensão das instâncias.

O Capítulo 3 apresenta a revisão da literatura para o Problema das p -Medianas Capacitado e sua versão robusta.

Capítulo 3

Revisão da Literatura

Este Capítulo apresenta uma revisão da literatura referente ao Problema das p -Medianas Capacitado (PPMC) e de problemas de localização envolvendo considerações de robustez em parâmetros (PPMRC), bem como um referencial teórico sobre os procedimentos utilizados para solucionar os problemas abordados nesta tese. O Capítulo se inicia com uma breve descrição sobre heurísticas, metaheurísticas e heurísticas híbridas, apresentadas, respectivamente, nas Seções 3.1, 3.2 e 3.3. Em particular, a Seção 3.2.1 apresenta o algoritmo *Variable Neighborhood Search* (VNS) (Hansen e Mladenović, 1997) e a Seção 3.2.2 mostra a metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) (Feo e Resende, 1995). As Seções 3.4 e 3.5 fazem uma revisão da literatura para o PPMC e para o PPMRC, respectivamente, finalizando o Capítulo.

3.1 Heurísticas

Nesta seção são discutidos conceitos importantes para o estudo das heurísticas e metaheurísticas usadas nas seções seguintes. Além disso, é apresentada uma revisão dos principais métodos heurísticos e meta-heurísticos utilizados nesta tese.

Um método heurístico pode ser definido como um procedimento desenvolvido a partir de regras observadas pelos desenvolvedores. Conforme Ziviane (2004), “*uma heurística é um algoritmo que pode produzir um bom resultado, ou até mesmo obter a solução ótima, mas pode também não produzir solução alguma ou uma solução que está distante da solução ótima*”. Seu objetivo é, portanto, o de obter boas soluções em tempo computacional inferior àquele necessário pelos métodos exatos.

A busca por técnicas para a solução de problemas complexos, não abordados por metodologias clássicas, motivou a exploração de técnicas da inteligência computacional. Os métodos heurísticos são exemplos destas técnicas utilizadas na resolução

de problemas de otimização. Eles oferecem uma solução eficiente, sendo conhecidos como algoritmos aproximados.

Segundo [Martins \(2009\)](#), a ideia por trás dos algoritmos heurísticos é encontrar soluções de boa qualidade para sistemas de médio e grande porte. [Martins \(2009\)](#) afirma, ainda, que a vantagem dos algoritmos heurísticos é que são de fácil compreensão, robustos e muito rápidos. Diferentemente dos algoritmos exatos, as heurísticas não garantem a obtenção da solução ótima do problema objeto de interesse, mas permitem, por outro lado, a determinação de soluções de boa qualidade em um tempo computacional reduzido. Este fato é particularmente vantajoso em problemas NP-difíceis, nos quais a aplicação de métodos completos pode, no pior caso, necessitar de tempo computacional exponencial para obter a solução ótima.

As heurísticas podem ser classificadas em duas classes: heurísticas construtivas e heurísticas de refinamento. As heurísticas construtivas são aquelas em que uma solução é construída elemento a elemento. Estes métodos podem ser do tipo guloso, em que o elemento escolhido para ser inserido na solução a cada passo é o melhor elemento de acordo com um critério guloso, ou podem ser do tipo randômico, em que os elementos são selecionados de forma aleatória. Heurísticas construtivas, normalmente, são empregadas como método para a geração de uma solução inicial para uma metaheurística e ou para uma busca local.

As heurísticas de refinamento partem de uma solução inicial previamente construída e tentam melhorá-la. As heurísticas de refinamento baseadas em busca local utilizam o conceito de vizinhança. Uma vizinhança está associado ao conceito de movimento, que é uma determinada modificação na solução atual que dá origem a outra solução. Segundo [Milano e Roli \(2004\)](#), uma solução vizinha, nos métodos de busca local, é determinada a partir de uma função de vizinhança (estrutura de vizinhança). Uma função de vizinhança para uma estrutura de vizinhança \mathcal{N} é definida como um operador que recebe uma solução s_1 e a transforma em outra solução s_2 , pertencente à vizinhança $\mathcal{N}(s_1)$ de s_1 . Assim, estes métodos buscam soluções através da troca sistemática da solução corrente por uma melhor solução, dentro da vizinhança da solução corrente.

As técnicas de refinamento podem ser subdivididas basicamente em três estratégias, caracterizando-se pelo critério de parada ao percorrer o espaço de soluções. São elas:

- (i) **Best improvement:** A estratégia de *best-improvement* (ou *greedy hill-climbing*) ou estratégia do melhor vizinho percorre todos os vizinhos de uma determinada solução e retorna o melhor deles. Esta estratégia tem, como pontos positivos, o retorno da melhor solução possível, dada uma determinada vizinhança. Por

outro lado, caso a quantidade de vizinhos seja muito alta, o método pode acabar perdendo desempenho;

- (ii) ***First improvement***: A estratégia *First improvement* ou estratégia da primeira melhora percorre os vizinhos de uma determinada solução até encontrar uma solução que seja melhor que a solução corrente, situação em que a busca é interrompida. Esta estratégia tem como principal ponto positivo evitar uma busca exaustiva, pois apenas no pior caso toda a vizinhança será explorada;
- (iii) ***Random Descent***: Na estratégia *Random Descent*, cada vizinho é escolhido aleatoriamente. Se ele representar uma melhora em relação à solução corrente, ele é aceito e passa a ser a nova solução. Caso contrário, outro vizinho é testado. Se, ao final de um número predeterminado de iterações, nenhum vizinho for aceito, o método é interrompido e retorna a melhor solução encontrada durante a busca.

Um ótimo local, em relação à vizinhança \mathcal{N} , é uma solução s que é melhor do que todas as soluções pertencentes a $\mathcal{N}(s)$. Da mesma forma, um ótimo global é uma solução s que é melhor do que todas as soluções pertencentes ao espaço de busca. As heurísticas de refinamento ficam presas no primeiro ótimo local encontrado, o que restringe a busca no espaço de soluções e limita a qualidade da solução final.

3.2 Metaheurísticas

O termo metaheurística foi introduzido inicialmente por Glover (1986) e deriva da junção de duas palavras gregas: “*heuriskein*”, que significa “buscar”, e “*meta*”, que significa “em um nível superior”.

Segundo Glover (1986), metaheurísticas são procedimentos heurísticos de alto nível que possuem estratégias para escapar de ótimos locais. Assim, as chances de obtenção de melhores soluções são maiores, quando comparadas às heurísticas. Esses métodos exploram melhor o espaço de busca, procurando soluções em outras regiões. Glover e Sörensen (2015) definiram metaheurística como uma estrutura algorítmica avançada e independente que fornece diretrizes ou estratégias para o desenvolvimento de algoritmos de otimização.

Em seu processo de busca, as metaheurísticas utilizam, usualmente, métodos de busca local. As propriedades fundamentais associadas às metaheurísticas, de acordo com Blum e Roli (2003), são:

- (i) as metaheurísticas devem incorporar mecanismos que evitem que a busca fique presa em áreas limitadas do espaço de busca;

- (ii) o objetivo de uma metaheurística é explorar eficientemente o espaço de busca, para encontrar soluções próximas do ótimo;
- (iii) metaheurísticas não são voltadas para a solução de problemas específicos;
- (iv) técnicas que constituem algoritmos de metaheurísticas variam de um simples procedimento de busca local a processos complexos de aprendizagem;
- (v) metaheurísticas podem utilizar experiências de busca (na forma de algum tipo de memória, por exemplo) para guiar sua busca.

Assim, de forma objetiva, a principal finalidade das metaheurísticas é guiar a exploração do espaço de busca utilizando diversas estratégias. Podem ser utilizadas ainda estratégias de aprendizagem para a estrutura das informações, com o objetivo de melhorar a busca pela solução ótima.

A principal diferença entre métodos heurísticos e as metaheurísticas está na capacidade das metaheurísticas em escapar de ótimos locais. Para isto, as metaheurísticas utilizam diferentes estratégias que permitem uma melhor exploração do espaço, a partir de movimentos sobre as diferentes vizinhanças do espaço de busca e procurando melhores ótimos locais (Souza et al., 2002).

Conforme Blum e Roli (2003), várias são as estratégias que permitem escapar de ótimos locais, como, por exemplo, movimentos de piora da solução, a geração de novas soluções iniciais, o uso de memória, ou, até mesmo, decisões probabilísticas. A identificação de elementos comuns nos algoritmos das metaheurísticas permite um melhor conhecimento das mesmas. Neste sentido, a classificação das metaheurísticas existentes possibilita a comparação destas e, assim, a identificação de suas similaridades e diferenças. Blum e Roli (2003) apresentam uma revisão detalhada dos conceitos básicos envolvidos em metaheurísticas e suas principais classificações.

As próximas subseções descrevem as metaheurísticas utilizadas nesta tese, quais sejam, VNS (*Variable Neighborhood Search*) e GRASP (*Greedy Randomized Adaptive Search Procedure*).

3.2.1 Metaheurística *Variable Neighborhood Search* (VNS)

A metaheurística *Variable Neighborhood Search* (VNS), proposta por Hansen e Mladenović (1997), é um algoritmo baseado em mudanças sistemáticas na estrutura de vizinhança. Essas mudanças podem ocorrer na fase de busca local ou em uma fase de perturbação utilizada para diversificar as soluções.

Segundo Hansen et al. (2010), o algoritmo VNS é amparado nas seguintes observações:

- um mínimo local de uma estrutura de vizinhança não é necessariamente um mínimo local de outra estrutura de vizinhança;
- um mínimo global é um mínimo local em relação a todas as vizinhanças possíveis;
- para muitos problemas, os mínimos locais são relativamente próximos uns dos outros em uma ou mais vizinhanças.

[Hansen et al. \(2010\)](#) reforçam, ainda, que a última observação é empírica. Isso significa que os ótimos locais geralmente fornecem informações sobre os ótimos globais. Portanto, antes de encontrar uma vizinhança melhor, é necessário um estudo organizado dessa vizinhança localmente ótima. Desta forma, além de fornecer uma solução muito boa, o algoritmo VNS também pode fornecer informações sobre os motivos desse desempenho, o que pode gerar uma implementação mais eficiente.

A partir de uma solução inicial, a metaheurística VNS define um raio de vizinhança e busca encontrar, naquele raio, uma solução melhor que a solução anterior. Caso não encontre, o algoritmo passa para a próxima estrutura de vizinhança e repete esse procedimento até que uma melhor solução seja encontrada. Quando uma solução melhor do que a incumbente é encontrada, a busca recomeça com a primeira vizinhança. Ao final da execução, um ótimo local com relação a todas as estruturas de vizinhança é encontrado.

Se a solução atual não for de boa qualidade, [Hansen et al. \(2010\)](#) mostram cinco propostas de melhoria:

- (i) ***First vs. best improvement***: comparar experimentalmente as primeiras e melhores estratégias de buscas locais. Caso a solução inicial tenha sido escolhida aleatoriamente, eles sugerem usar a estratégia *first improvement*. Caso se esteja usando algum outro método de heurística construtiva, a sugestão é usar a estratégia *best improvement*;
- (ii) **Redução de vizinhança**: a razão para o mal comportamento de qualquer busca local pode ser as visitas desnecessárias a todas as soluções da vizinhança. A sugestão é tentar identificar subconjuntos “promissores” da vizinhança e explorar apenas estes; o ideal é encontrar uma regra que remova automaticamente soluções da vizinhança de soluções cujos valores da função objetivos não sejam melhores que os atuais;
- (iii) **Intensificação da perturbação**: ao desenvolver uma implementação da metaheurística VNS mais eficiente, gasta-se algum tempo examinando a sensibilidade da função objetivo a pequenas mudanças na solução. Para algumas

instâncias de problemas, “saltos” completamente aleatórios nas vizinhanças são muito diversos. Neste caso, alguns procedimentos de perturbação intensiva são necessários. Um caso particular é a chamada busca em grandes vizinhanças, onde as propriedades de k atributos são escolhidos aleatoriamente, são destruídos e imediatamente reconstruídos de forma ótima;

- (iv) **Busca Local VND:** analisar várias estruturas de vizinhanças possíveis, mantendo as estruturas mais eficientes. De posse dessas estruturas selecionadas, a sugestão é usá-las para se realizar a busca local através do procedimento *Variable Neighborhood Descent* (VND), que realiza a busca local através de trocas sistemáticas de estruturas de vizinhança, em substituição à rotina de busca local baseada em uma única estrutura de vizinhança. Desta forma, obtém-se um procedimento *General Variable Neighborhood Search* (GVNS), proposto por Hansen e Mladenović (1997), que utiliza o VND para realizar a busca local e o VNS para realizar perturbações na solução;
- (v) **Configurações de Parâmetros:** o único parâmetro central do VNS é o *IterMaxSM*, que deve ser ajustado experimentalmente.

Hansen et al. (2010) afirmam que os problemas de localização também atraíram muita atenção para o uso da metaheurística VNS. Entre os modelos discretos, o problema das p -Medianas é o mais estudado e desempenha um papel importante no melhor desenvolvimento do próprio método.

Algoritmo 1: VNS.

```

1  $s \leftarrow s_0$  (solução inicial),  $s' \leftarrow s$ ,  $k \leftarrow 1$ 
2 while ( $Iter < IterMaxSM$ ) do
3    $s' \leftarrow Shake(s, k)$  /* (Gere uma solução aleatória da  $k$ -ésima vizinhança) */;
4    $s'' \leftarrow BuscaLocal(s')$ 
5   if  $f(s'') < f(s)$  then
6      $s \leftarrow s''$ ,  $k \leftarrow 1$ ,  $iter \leftarrow 0$ 
7   else
8      $iter \leftarrow iter + 1$ ,  $k \leftarrow k + 1$  /* (Troca vizinhança) */
9   end
10 end
11 return  $s$ 

```

O Algoritmo 1 mostra o pseudocódigo geral do algoritmo VNS. Na (linha 1) é gerada a solução inicial e o algoritmo aplica sobre ela um método de busca local usando o procedimento VND, conforme Algoritmo 5, para gerar a solução atual s e inicializa o contador k do número de perturbações a serem aplicadas sobre a solução atual. Esse contador, diversifica a busca para outras regiões do espaço de soluções do problema. Em seguida, na linha 2, tendo como critério de parada o número máximo de iterações

sem melhora ($iterMaxSM$), o algoritmo entra em um laço iterativo com aplicações sucessivas dos procedimentos de perturbação da solução atual, usando o Algoritmo 10, como mostrado na (linha 3). Em seguida o VNS aplica a busca local na solução perturbada (linha 4), para explorar as soluções do espaço de soluções do problema. Caso ocorra melhora na solução atual, a solução s é atualizada, a estrutura de vizinhança retorna à estrutura inicial ($k = 1$) e o procedimento continua, como mostrado na (linha 6). Caso contrário, a estrutura de vizinhança é incrementada (linha 8). A melhor solução encontrada é retornada ao fim do processo de busca (linha 11).

3.2.2 Metaheurística GRASP

A metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*) (Feo e Resende, 1995) é um método de busca adaptativa gulosa e aleatória, que combina heurísticas construtiva e de busca local.

Este método é composto por dois elementos: um procedimento de construção de solução gulosa aleatória e um procedimento de busca local.

No procedimento de construção, a cada iteração, uma lista restrita de candidatos é gerada com os melhores elementos. Em seguida, cada elemento a ser inserido na solução é escolhido aleatoriamente, a partir da lista restrita de candidatos. Conforme Blum e Roli (2003), esta heurística construtiva é chamada dinâmica já que, a cada iteração, os valores são atualizados, alterando-se a probabilidade de inserção dos elementos durante a construção.

A fase de busca local é iniciada logo após a construção da solução. Esta fase tem o objetivo de melhorar a solução encontrada na fase de construção. Este dois passos se repetem iterativamente.

Algoritmo 2: GRASP.

```

1 while (Critério de Parada GRASP não Satisfeito) do
2   | Constrói uma solução  $s$  com um procedimento guloso aleatório;
3   |  $s' \leftarrow \text{BuscaLocal}(s)$ ;
4   |  $s^* \leftarrow \text{MelhorSolução}(s, s')$ ;
5 end
6 return  $s^*$ 

```

O Algoritmo 2 mostra o pseudocódigo geral do algoritmo GRASP. Enquanto houver um cliente j não alocado a uma instalação i , o algoritmo constrói iterativamente e de forma aleatória, uma solução viável, elemento a elemento. O algoritmo aplica sobre essa solução inicial um método de busca local e atualiza a melhor solução s^* encontrada até o momento.

3.3 Heurísticas e metaheurísticas híbridas

Heurísticas híbridas em problemas de otimização combinatória consistem na combinação dos melhores elementos encontrados nas heurísticas utilizadas na implementação particular realizada (Blum et al., 2011). Segundo Blum et al. (2011), essa combinação é feita principalmente para aproveitar os “pontos fortes” de cada heurística. Adicionalmente, segundo este mesmo artigo, grande parte dos melhores resultados encontrados através de metaheurísticas para diversos problemas foram obtidos a partir de hibridismo. O hibridismo é a combinação de uma ou mais metaheurísticas e/ou estratégias com o objetivo de melhorar o desempenho da busca. Por exemplo, a utilização de procedimentos de busca local por métodos como otimização colônia de formigas e algoritmos evolucionários é uma forma de hibridismo.

A partir das afirmações acima, podemos afirmar que heurísticas podem ser combinadas para criar um novo algoritmo heurístico híbrido. Os algoritmos híbridos se beneficiam da cooperação das diferentes estratégias e característica que é particular de cada um. Além disso, hibridização explora recursos que contém estratégias apropriada para alcançar o desempenho ideal aos problemas a serem solucionados. Entretanto, desenvolver uma abordagem heurística híbrida eficaz é, em muitos casos, uma tarefa difícil e requer conhecimento de diferentes áreas.

Para Lozano e García-Martínez (2010), um ponto forte para hibridização heurística é contar com a intensificação e diversificação. Os algoritmos de busca devem encontrar um equilíbrio entre esses dois objetivos, às vezes conflitantes. A diversificação geralmente se refere à capacidade de acessar diferentes regiões do espaço de busca, enquanto a intensificação se refere à obtenção de soluções de alta qualidade.

De fato, analisando de forma geral, vemos que fornecer um equilíbrio adequado entre os componentes de intensificação e diversificação em uma heurística, torna-se uma tarefa não trivial do ponto de vista computacional. Também sabemos que existem heurísticas que possuem uma maior tendência para a intensificação e outras para a diversificação. Entretanto, heurísticas híbridas que combinam esses dois elementos, são susceptíveis a melhores resultados.

Revisões literárias a respeito de heurísticas híbridas podem ser encontradas em Blum et al. (2008), Jourdan et al. (2009), Blum et al. (2011), Lozano e García-Martínez (2010) e Rodriguez et al. (2012), dentre outros trabalhos.

3.4 Revisão da Literatura para o PPMC

Diversos métodos têm sido propostos para a solução do Problema das p -Medianas Capacitado (PPMC). Esta seção se concentra em uma revisão da literatura relacionada à utilização de métodos para solucioná-lo. Para isso, são descritos a seguir alguns trabalhos considerados relevantes para presente tese.

Baldacci et al. (2002) apresentam um algoritmo exato para resolver o PPMC com base em uma formulação de particionamento do problema. Um limite inferior válido para o custo ótimo da solução é obtido pela combinação de dois métodos heurísticos diferentes para resolver o duplo relaxamento de otimização linear da formulação exata. Os autores propõem dois métodos para resolver o PPMC. O primeiro, chamado EHP, consiste em reduzir o número de variáveis do subproblema de otimização associado, para que o problema resultante possa ser resolvido por um solucionador de otimização inteira. O segundo é o método *Branch-and-Bound* (B&B), que foi implementado para comparar o desempenho computacional do algoritmo EHP. Este procedimento B&B utiliza o limite inferior (λ) obtido por um procedimento que gera, para uma determinada mediana j , um conjunto de *clusters* e, em seguida, relaxa as restrições de particionamento utilizando relaxação Lagrangeana. Os autores utilizam a pesquisa em árvore binária de dois níveis, na qual o primeiro nível da árvore é formado pela fixação das variáveis que definem as medianas da solução. Sempre que um nó folha desta árvore de nível superior é atingido, ele pode ser tratado como a raiz de uma nova subárvore que é explorada pela fixação de variáveis, que correspondem à atribuição dos clientes às medianas definidas no nível superior da árvore. Em cada nó da árvore, um limite inferior é calculado por meio do procedimento de subgradiente aplicado à relaxação proposta. Os valores dos multiplicadores são inicializados com as penalidades associadas ao limite inferior encontrado no nó predecessor. Cinquenta iterações do subgradiente são realizadas em cada nó da árvore, com exceção do nó raiz, no qual são executadas 300 iterações do subgradiente. Para lidar com as restrições do mundo real, novos requisitos foram impostos aos *clusters*, como limites na cardinalidade dos *clusters*, incompatibilidade entre clientes, restrições de distância, etc. As restrições adicionais podem ser facilmente incorporadas no procedimento EHP, rejeitando qualquer *clusters* inviável, e no método B&B, alterando a estratégia de ramificação do segundo nível da árvore, rejeitando os ramos que levam a soluções inviáveis. No entanto, segundo Baldacci et al. (2002), essas modificações simples por si só podem levar a um desempenho severamente degradado se nenhuma alteração correspondente for introduzida no limite. Conforme Baldacci et al. (2002), as soluções alcançadas podem não ser uma solução ótima para PPMC; no entanto, o novo método utilizados por eles permite estimar sua distância máxima da otimização.

Lorena e Senne (2004) apresentam abordagens de geração de colunas para a solução do PPMC. Essas abordagens integram a geração tradicional de colunas ao contexto de relaxação Lagrangeana/*surrogate*, identificando novas colunas produtivas e acelerando o processo computacional. O problema principal otimiza a cobertura de *clusters* de uma única mediana que satisfazem às restrições de capacidade. Novas colunas são geradas considerando os subproblemas da mochila. Para os autores, a geração de colunas é uma ferramenta poderosa para resolver problemas de otimização linear para grandes instâncias. A otimização linear pode surgir quando as colunas no problema não são conhecidas antecipadamente e uma enumeração completa de todas as colunas não é uma opção ou o problema é reescrito usando a decomposição. Em muitos casos, uma aplicação direta da geração de colunas pode resultar em convergência lenta. Sendo assim, os autores utilizam o método de relaxação lagrangeana como processo de aceleração para a geração de colunas. O conjunto de instâncias utilizada por Lorena e Senne (2004) é formada por dados reais e foram coletados usando o Sistema de Informações Geográficas ArcView. Estas instâncias descrevem a área central da cidade de São José dos Campos, SP, Brasil. Cada ponto está localizado em um bloco, que apresenta um nó de demanda e também é um local possível para localizar medianas. A demanda foi estimada considerando o número de casas em cada bloco. Um bloco vazio recebeu o valor um. Além dessas instâncias, Lorena e Senne (2004) utilizaram o conjunto de instância Pcb3038 da biblioteca TSPLIB. Os resultados computacionais mostraram que o subproblema Lagrangeano/*surrogate* gera um pequeno número de colunas produtivas. Os autores concluem que os limites inferiores Lagrangeano/*surrogate* podem ser úteis para o método *Branch-and-Price*.

Díaz e Fernández (2006) propuseram dois algoritmos heurísticos para a solução do PPMC. No primeiro, as soluções são obtidas com o método *Scatter Search* (SSH), e o segundo algoritmo utiliza a fase de construção inicial da metaheurística GRASP associada a um procedimento *Path Relinking* (PRH). A segunda fase é vista como uma etapa de intensificação, uma vez que o conjunto de medianas potenciais se restringe àquelas utilizadas em algumas das soluções combinadas. O algoritmo consiste em aplicar, iterativamente, as duas etapas. A primeira é aplicada quando as duas soluções têm uma mediana em comum. Seu objetivo é ter a mesma tarefa na mediana comum nas duas soluções. O segundo passo é fazer com que as duas soluções tenham uma mediana em comum. Uma terceira abordagem que combina o *Path Relinking* com o *Scatter Search* também é analisada. A heurística GRASP, utilizada na construção inicial, consiste em duas fases que são aplicadas iterativamente. A primeira baseia-se na randomização parcial de um critério guloso, enquanto o segundo é uma fase de busca local, que explora várias vizinhanças. Duas delas estão relacionados à atribuição de clientes dentro do conjunto de medianas selecionadas. Elas são vi-

zinhanças clássicas de *Shift* e *Swap*. O *Shift* contém todas as soluções que diferem da atual na designação de exatamente um cliente e o *Swap* contém todas as soluções que diferem da atual na designação de exatamente dois clientes que trocam sua designação atual. A heurística *Scatter Search* (SSH) implementada por eles é um método iterativo em que, a cada iteração, é criado um conjunto de soluções que são combinadas entre si para gerar novas soluções. Essas novas soluções possivelmente substituirão alguma solução no conjunto de soluções. Existem casos em que a solução gerada é uma solução inviável em que o número de medianas usadas possivelmente excederá a quantidade exigida ou algumas restrições de capacidade serão violadas. Para recuperar a viabilidade, é aplicado primeiro uma fase de redistribuição, em que busca-se uma solução que não viole as restrições de capacidade, seguida por uma segunda fase, que fecha iterativamente as medianas menos promissoras até que apenas p medianas sejam usadas. A cada iteração da fase de fechamento é necessário redistribuir os clientes atribuídos anteriormente à mediana que está fechada. Isso é feito aplicando novamente a fase de redistribuição até que sejam satisfeitas todas as restrições de capacidade. Uma série de experimentos computacionais em diferentes conjuntos de instâncias foram utilizadas para testar os dois algoritmos. Segundo [Díaz e Fernández \(2006\)](#), os resultados foram satisfatórios em termos de qualidade das soluções encontradas e dos tempos computacionais. Eles concluem que, em geral, a heurística *Scatter Search* supera os resultados do algoritmo *GRASP-Path Relinking*. Por outro lado, a combinação de *Path Relinking* e *Scatter Search* apresentou os melhores resultados, pois gerou soluções com uma qualidade semelhante ao algoritmo *Scatter Search*, mas com tempos computacionais menores.

[Fleszar e Hindi \(2008\)](#) desenvolveram uma heurística *Variable Neighborhood Search* (VNS) para a solução do PPMC. Em certo sentido, o algoritmo baseia-se no ponto de vista adotado pelas primeiras tentativas de criar uma solução heurística, que é considerar o problema geral como se fosse o de resolver sequencialmente um problema de seleção de mediana e resolver o problema de designar clientes para as medianas selecionadas.

[Landa-Torres et al. \(2012\)](#) apresentam um Algoritmo Genético de Agrupamento (GGA) e uma abordagem de Agrupamento de Busca de Harmonia (GHS). Ambos os algoritmos compartilham o mesmo procedimento de codificação; no entanto, são baseados em conceitos muito diferentes no que diz respeito aos seus procedimentos operacionais. Segundo os autores, o Algoritmo Genético de Agrupamento (GGA) é considerado um algoritmo evolutivo especialmente projetado para lidar com problemas de agrupamento, isto é, problemas nos quais um número de itens deve ser atribuído a um conjunto de grupos pré-definidos. O algoritmo (GHS) é um algoritmo de otimização iterativo baseado na imitação da coordenação dos músicos em

uma orquestra quando buscam conjuntamente a melhor harmonia sob uma métrica estética.

Yaghini et al. (2013) apresentam um algoritmo metaheurístico híbrido combinando uma estrutura de vizinhança de plano de corte e uma metaheurística Busca Tabu para solução do PPMC. O algoritmo proposto é testado em vários conjuntos de instâncias e a análise estatística mostra a eficiência e eficácia do algoritmo híbrido. Herda (2015) combina as melhores características entre dois Algoritmos Genéticos para obter um terceiro AG que obtém melhores soluções.

Jánošíková et al. (2017) propuseram duas maneiras de combinar um algoritmo genético com otimização inteira mista (MIP) para melhorar a qualidade da solução do PPMC. De acordo com os autores, os resolvedores de otimização inteira atuais são muito sofisticados e eficientes, o que justifica combiná-los com métodos metaheurísticos.

A Tabela 3.1 apresenta um resumo dos trabalhos apresentados. Em todos os trabalhos citados foram utilizados heurísticas híbridas. Entretanto, nenhum deles aborda a metaheurística *General Variable Neighborhood Search* – GVNS (Hansen et al., 2017), com solução inicial definida pela fase de construção GRASP.

Esta tese, por outro lado, apresenta quatro implementações da metaheurística GVNS. Duas delas utilizam a solução inicial definida pelo método clássico, ou seja, de forma aleatória. As outras duas metaheurísticas GVNS utilizam, na solução inicial, a fase de construção do método *Greedy Randomized Adaptive Search Procedure* – GRASP (Feo e Resende, 1995). Além disso, dois métodos de buscas locais são utilizados para as quatro metaheurísticas. O primeiro é o algoritmo *Variable Neighborhood Descent* – VND (Hansen et al., 2008; Mladenović et al., 2007) e o segundo, o algoritmo *Random Variable Neighborhood Descent* RVND (Souza et al., 2010; Subramanian et al., 2010).

Tabela 3.1: Tabela resumo - Heurísticas híbridas aplicadas ao PMPC

Artigo	Métodos Híbridos Aplicados ao PMPC	Método
Baldacci et al. (2002)	<i>Branch and Bound</i> /Lagrangeano	MIP
Lorena e Senne (2004)	Geração de colunaw/relaxamento Lagrangeano/ <i>surrogate</i>	MIP
Díaz e Fernández (2006)	<i>Hybrid scatter search</i> e <i>Path relinking</i>	Híbrido
Fleszar e Hindi (2008)	VNS híbrido	Heurística híbrida
Landa-Torres et al. (2012)	Algoritmo Genético híbrido	Heurística híbrida
Jánošíková et al. (2017)	Algoritmo Genético/MIP	Heurística/MIP

3.5 Revisão da Literatura para o PPMRC

Entre as principais abordagens para otimização sob incertezas estão a otimização estocástica e a otimização robusta (Govindan et al., 2017). A otimização estocástica

assume que os parâmetros incertos são variáveis aleatórias com distribuição de probabilidade conhecida ou parcialmente conhecida (Shapiro, 2008). A otimização robusta, por outro lado, não pressupõe a existência de qualquer distribuição de probabilidade para os parâmetros incertos, apenas assumindo que os parâmetros incertos pertencem a um conjunto de incertezas (Bertsimas et al., 2011). Considerando um determinado orçamento de incertezas, que considera o *trade-off* entre conservadorismo e riscos, os modelos robustos de otimização são, em geral, baseados na análise do pior caso ou minimização do arrependimento (Bertsimas et al., 2011; Govindan et al., 2017). De acordo com Govindan et al. (2017), a otimização robusta tem sido usada para buscar a garantia de viabilidade e tem a vantagem de gerar problemas tratáveis. Portanto, a tratabilidade dos modelos de otimização robustos torna essa abordagem muito atraente para lidar com incertezas para adicionar proteção contra violações de restrições.

Soyster (1973) desenvolveu um trabalho pioneiro no estudo sobre otimização robusta, em que foi proposta uma abordagem para lidar com uma matriz de restrições sujeitas a variações incertas pertencentes a hiperesferas. Eles mostram que a solução ótima robusta no caso proposto corresponde a variação máxima permitida para o modelo. Além disso, Soyster (1973) observou que isto traz um aumento excessivo no valor da função objetivo, considerando sua abordagem como bastante conservadora.

Ben-Tal e Nemirovski (1999) consideram que não se pode ignorar as incertezas inerentes de cada classe de problemas, pois, ao ignorá-las, corre-se o risco de encontrar valores ótimos, porém, insignificantes do ponto de vista de situações reais. Estes autores afirmam que, em muitos casos, a viabilidade da solução ótima de um problema de otimização linear pode ser fortemente afetada por pequenas perturbações. Isso significa que, do ponto de vista prático, existe uma forma sistemática e racional de se obter soluções viáveis e fortemente confiáveis que resistam a incerteza dos dados. Testes computacionais foram efetuados em problemas NETLIB com 0,1% de perturbações nos dados e comprovou-se que as perdas não ultrapassaram 1%.

Bertsimas e Sim (2004) propuseram uma abordagem de otimização robusta em que o grau de conservadorismo da solução é controlado e tratado, tanto do ponto de vista prático quanto teórico. Este controle é feito pela definição do conjunto de incertezas considerado na otimização.

Estudos envolvendo o problema de localização de instalações sob incertezas, em que o número de instalações não é predefinido, como no caso do PPM, são destacados a seguir. Carrizosa e Nickel (2003) abordaram o problema de encontrar a instalação mais robusta considerando, como medida de robustez de uma instalação, o aumento mínimo na demanda referente a uma estimativa de demanda que atendesse a um limite do maior custo de transporte permitido. Considerando que o planejamento de

serviços médicos de emergência sob incerteza é uma situação típica em problemas do mundo real, [Beraldi et al. \(2004\)](#) propuseram um modelo estocástico robusto de localização de instalações para serviços médicos de emergência com restrições de chance. [Baron et al. \(2011\)](#) abordaram o problema de localizar instalações em uma rede enfrentando demanda incerta por vários períodos a partir de uma abordagem de otimização robusta. Os autores apresentam dois modelos de incerteza de demanda: (i) demanda dentro de uma caixa multidimensional limitada e simétrica e (ii) demanda dentro de um elipsóide multidimensional. Além disso, os autores consideraram o objetivo de maximizar o lucro, estabelecendo um *trade-off* necessário entre capacidade suficiente e a receita futura incerta. [Gülpinar et al. \(2013\)](#) usaram otimização robusta para resolver o problema de localização com demandas incertas para um único produto, que pode ser atendido simultaneamente por várias instalações. Os autores também consideraram a probabilidade de falta de estoque.

[Ji e Qi \(2020\)](#) abordaram um problema robusto de localização de instalações que otimiza o desempenho do pior caso em um plano de atribuição. O objetivo do problema proposto é melhorar a eficiência da descontaminação em aldeias afetadas por vazamento de radiação em um acidente nuclear. [Egri et al. \(2021\)](#) estudaram um problema robusto de localização de instalações para o transporte de materiais de madeira de seus armazéns de origem para as instalações de processamento. [Basciftci et al. \(2021\)](#) propuseram um problema robusto de localização de instalações em dois estágios considerando a incerteza na demanda, que usa a média e a variância da demanda. Os autores apresentam uma formulação de Programação Inteira Mista (MIP) para modelar o problema e desigualdades válidas e para melhorar o tempo computacional para resolver grandes instâncias. [Saif e Delage \(2021\)](#) analisou uma versão robusta de dois estágios do problema de localização de instalações capacitadas em uma pequena instância, abordando incertezas nas demandas. [Liu et al. \(2023\)](#) aplicou o problema de localização de instalações para identificar instalações para armazenamento de suprimentos de emergência e locais de teste ou vacinação para controlar a propagação do vírus que causa o COVID-19. Os autores propuseram uma estrutura de otimização de duas fases para localizar as instalações. Na primeira fase, as estratégias iniciais de pré-posicionamento são determinadas para atender aos requisitos de taxa de preenchimento predeterminados usando a aproximação média da amostra. Na segunda fase, a capacidade de lidar com a variação da demanda é ajustada dinamicamente para superar a imprecisão da previsão de longo prazo. [Shehadeh e Sanci \(2021\)](#) apresentam um problema robusto de localização de instalações onde a demanda do cliente é bimodal. O problema proposto determina um subconjunto de localizações candidatas a instalações abertas e a atribuição da demanda do cliente a instalações abertas. [Ryu e Park \(2022\)](#) também consideraram um problema robusto de

localização de instalações com incerteza na demanda. [Baldacci et al. \(2022\)](#) avaliou as características de soluções robustas para o Problema de Localização de Instalações Capacitadas com demanda incerta.

Na sequência, são apresentados alguns estudos abordando o problema de medianas p -hub de alocação única e múltipla sob incertezas. [Meraklı e Yaman \(2016\)](#) estudou o problema médio robusto de alocação múltipla não capacitada p -hub sob incerteza de demanda poliédrica, considerando, primeiro, que a única informação disponível da incerteza de demanda é o limite superior do fluxo total adjacente em cada nó, e, depois, impondo limites inferior e superior em cada demanda de pares. Para resolver o problema proposto, os autores utilizaram critérios minimax e dois algoritmos de decomposição baseados em Benders. [Talbi e Todosijević \(2017\)](#) estudaram o problema de medianas de alocação múltipla não capacitada p -hub (UMApHMP), abordando a incerteza no fluxo entre pares de nós. Esta incerteza é assumida como uma variável aleatória limitada com distribuição desconhecida que assume valores em um intervalo. Os autores utilizaram a abordagem proposta por [Bertsimas e Sim \(2003\)](#) para lidar com o UMApHMP robusto. Uma nova medida robusta é introduzida, considerando todas as possibilidades de realização do número de mudanças que podem ocorrer a partir do modelo nominal. Além disso, um algoritmo baseado em VNS foi proposto para resolver grandes instâncias do problema tratado.

[Ghaffarinasab et al. \(2019\)](#) consideraram o problema das p -medianas robusto de alocação única de p -hubs não capacitado sob incertezas de demanda híbrida. Os autores formularam o problema como um problema de programação linear inteira mista e propuseram uma solução matheurística usando a metaheurística *Tabu Search*. [Ghaffarinasab \(2022\)](#) abordou o problema mediano robusto de múltiplas alocações não capacitadas p -hub, no qual o tráfego de demanda entre pares origem-destino está sujeito a incertezas. O autor introduziu uma formulação linear de programação inteira mista para o problema e dois algoritmos de solução baseados em decomposição de Benders para resolver o problema abordado.

Finalmente, são descritos artigos que tratam do problema p -medianas sob incertezas. [Berman e Drezner \(2008\)](#) estudaram o problema das p -medianas não capacitada considerando a incerteza no número de instalações a serem instaladas. O problema é formulado em um grafo através de um modelo de programação inteira e resolvido por algoritmos heurísticos utilizando CPLEX. [Nikoofal e Sadjadi \(2010\)](#) apresentaram o problema robusto de p -medianas não capacitado. No problema considerado, os pesos dos nós são determinísticos e os comprimentos das arestas são incertos e pertencem a intervalos conhecidos. O modelo proposto permite ajustar a robustez da solução aos vários níveis de conservadorismo ou diferentes graus de incerteza. [An et al. \(2014\)](#) consideraram um conjunto de modelos robustos de otimização de dois

estágios para resolver redes confiáveis de localização de instalações p -medianas sujeitas a interrupções. Os autores usaram o conceito de incerteza para capturar as interrupções aleatórias e o método de otimização robusto para estudar problemas de localização de instalações confiáveis. [López-Monzalvo e Brizuela \(2014\)](#) analisaram o problema de p -medianas não capacitada sob incerteza no número de instalações a serem instaladas e no número de nós a serem atendidos. O problema robusto é desenhado como um problema de otimização bi-objetivo, usando o NSGA-II para resolvê-lo. [Peško e Majer \(2015\)](#) abordaram um problema robusto de p -medianas não capacitado com distâncias incertas. [Baroughi et al. \(2019\)](#) avaliaram o problema de p -mediana não capacitada com pesos de vértices incertos e distâncias incertas com valor de cauda na função objetivo de risco. Para resolver este problema, os autores propuseram um algoritmo de otimização de enxame de partículas binárias modificadas híbridas (HMBPSO). [Soltanpour et al. \(2020\)](#) abordaram o problema clássico de localização p -mediana não capacitada em uma rede aleatória incerta. Alguns pesos de vértices e as distâncias entre vértices são parâmetros incertos, enquanto outros são valores aleatórios. Para resolver este problema incerto de p -medianas não capacitado, os autores propuseram um modelo de otimização baseado na teoria da chance e, após, um algoritmo é apresentado para encontrar a p -Mediana. [Lin et al. \(2020\)](#) usaram um modelo robusto de localização de instalações para melhorar a eficácia e eficiência do sistema de coleta de resíduos e minimizar o custo total. Os autores assumem incerteza na variedade de locais de origem da geração de resíduos na construção civil. Por fim, [Liu e Li \(2021\)](#) propuseram um problema de planejamento da localização de uma instalação de reciclagem com incerteza no volume e composição da futura matéria-prima.

A Tabela 3.2 apresenta um resumo das referências citadas nesta seção. Nesta tabela, a coluna “Problema” mostra o tipo de problema de localização da instalação abordado. A coluna “Procedimento de solução” apresenta o método de solução proposto em cada trabalho, enquanto o MIP refere-se à estratégia do procedimento de solução utilizando um solver. A coluna “Incerteza” mostra qual parâmetro do problema foi considerado incerto. Desta coluna, em particular, concluímos que poucos artigos abordam incertezas além da demanda. A conclusão mais importante desta tabela é que poucos trabalhos abordaram o problema p -mediana. Surpreendentemente, no entanto, nenhum trabalho aborda diretamente o problema capacitado de p -mediana sob incertezas. Portanto, a descrição do problema robusto capacitado p -mediana sob demanda incerta e a proposição de um algoritmo para resolver este problema são as principais contribuições do presente trabalho.

Tabela 3.2: Trabalhos abordando otimização robusta em problemas de localização de facilidades.

Referencia	Problema	Procedimento de solução	Incerteza
Carrizosa e Nickel (2003)	Localização de facilidade	Método Lagrangeano	Demanda Incerta
Beraldi et al. (2004)	Serviço médico de emergência robusto	MIP	Demanda Incerta
Berman e Drezner (2008)	Problema das p -Medianas	MIP	Número incerto de instalações
Nikoofal e Sadjadi (2010)	Problema das p -Medianas	Comprimentos de borda incertos	
Baron et al. (2011)	Localização de facilidade	MIP	Demanda Incerta
Gülpinar et al. (2013)	Localização de facilidade	GAMS + CONOPT	Demanda Incerta
An et al. (2014)	Problema de localização de instalação p -mediana confiável	Método de geração de colunas e restrição	A capacidade da instalação e a demanda mudam devido à interrupção do local
López-Monzalvo e Brizuela (2014)	Problema das p -Medianas	Problema de otimização bi-objetivo + abordagem de algoritmo evolutivo	Incerteza no número de instalações e de nós a serem atendidos
Peško e Majer (2015)	Problema das p -medianas	MILP	Distância incerta
Merakli e Yaman (2016)	Problema de localização do hub intermodal	MILP + Decomposição de Benders	Incerteza de demanda poliédrica
Talbi e Todosijević (2017)	Problema de alocação múltipla mediana p -hub não capacitada	MILP e VNS	Incerteza de fluxos
Baroughi et al. (2019)	Problema de localização de p -medianas	Algoritmo de otimização de enxame de partículas binárias modificadas híbridas (HMBPSO)	Pesos de vértices incertos e distâncias incertas
Ghaffarinasab et al. (2019)	Problema de medianas p -hub de alocação única	MILP + Matheurística baseada em busca tabu	Incertezas de demanda híbrida
Soltanpour et al. (2020)	Problema de localização de p -medianas	Modelo de otimização baseado na teoria do acaso	Alguns dos pesos dos vértices e as distâncias entre os vértices são incertos e outros são aleatórios
Egri et al. (2021)	Localização de instalações em logística reversa	Pesquisa Algébrica Geral MIP e Busca Tabu	Demanda incerta
Liu e Li (2021)	Problema de localização de instalações de reciclagem com agrupamento	MIP	Volume incerto e composição da matéria-prima
Basciftci et al. (2021)	Problema de localização de instalações sob demanda estocástica dependente de decisão	MIP	Demanda estocástica dependente de decisão
Cheng et al. (2021)	Problema de localização de facilidades	MIP	Demanda incerta e interrupções nas instalações
Saif e Delage (2021)	Problema de localização de instalações capacitadas	MIP	Demanda incerta
Liu et al. (2023)	Problema de localização de facilidades	Relaxação Lagrangeana	Planejamento dinâmico de capacidade para pandemias com incerteza de demanda
Shehadeh e Sanci (2021)	Problema de localização de facilidades	MIP	Demanda aleatória bimodal
Ji e Qi (2020)	Planejamento de descontaminação da localização da instalação da zona de planejamento de emergência e plano de atribuição	MIP	Demanda incerta
Lin et al. (2020)	Modelo de localização de instalações para estações de transferência de resíduos de construção civil e demolição	MIP	Custo e ambiente incertos
Ryu e Park (2022)	Problema de localização de instalação capacitada de fonte única	Algoritmo <i>Branch-and-price</i>	Demanda incerta
Ghaffarinasab (2022)	Problema de medianas p -hub de alocação múltipla não capacitada	MILP + Decomposição de Benders	Incerteza no tráfego de demanda entre pares origem-destino
Baldacci et al. (2022)	Localização de facilidades capacitada	MIP	Demanda incerta

3.6 Considerações Parciais

Este capítulo apresentou a revisão da literatura associada ao Problema de p -Medianas Capacitado e ao Problema de p -Medianas Robusto Capacitado, bem como a uma revisão da metodologia utilizada para solucionar estes problemas.

Observou-se, também, que uma metaheurística é um tipo de algoritmo de otimização que possui estratégias para escapar de ótimos locais, aumentando assim, as chances de obtenção de melhores soluções, e que, vários algoritmos exatos e técnicas metaheurísticas têm sido usadas para resolver o problema das p -medianas ao longo dos anos. Também foi observado o uso crescente de heurísticas/metaheurísticas híbridas em problemas diversos de localização de facilidades. Além disso, destacou-se

que as metaheurísticas híbridas combinam as melhores características de vários algoritmos de otimização diferentes para um determinado problema. A ideia é encontrar um algoritmo que funcione bem em algumas partes do problema e depois usá-lo como parte de uma abordagem híbrida que aproveita outras boas abordagens conhecidas. Sendo assim, podemos destacar que a principal razão para a crescente utilização desta técnica são os bons resultados obtidos no processo. Muitos dos melhores resultados na literatura para vários problemas de otimização, empregam metaheurísticas híbridas, combinando heurísticas com heurísticas, heurísticas com metaheurísticas e metaheurísticas com outros métodos, como por exemplo, métodos exatos. Há uma variedade de maneiras de usar metaheurísticas híbridas em problemas de otimização.

Além disso, foram listados os diversos métodos para a solução do PPMC bem como as principais abordagens para otimização sob incertezas. Observou-se também que a otimização estocástica assume que os parâmetros incertos são variáveis aleatórias com distribuição de probabilidade conhecida ou parcialmente conhecida, e a otimização robusta não pressupõe a existência de qualquer distribuição de probabilidade para os parâmetros incertos, apenas assume que os parâmetros incertos pertencem a um conjunto de incertezas. Além disso, a revisão da literatura mostrou que nenhum trabalho apresentou uma abordagem robusta para o PPMC.

No Capítulo 4 são apresentados os algoritmos utilizados nesta tese, que também são métodos hibridizados.

Capítulo 4

Solução para o Problema das p -Medianas Capacitado (PPMC)

Este Capítulo apresenta métodos de solução para o Problema das p -Medianas Capacitado (PPMC). Para este fim, foram propostos algoritmos heurísticos baseados na metaheurística *General Variable Neighborhood Search* (GVNS). Esses algoritmos diferem entre si com relação ao método usado para construir a solução inicial e com relação ao método usado para a heurística de busca local. Este Capítulo está organizado como segue. A Seção 4.1 descreve como uma solução do PPMC é representada e avaliada. Os dois métodos de construção de uma solução são apresentados na Seção 4.2. As estruturas de vizinhanças utilizadas para explorar o espaço de soluções do problema são apresentadas na Seção 4.3. As heurísticas de busca local e de perturbação são apresentadas nas Seções 4.4 e 4.5, respectivamente. A Seção 4.6 resume o algoritmo GVNS completo para a solução do PPMC. A Seção 4.7 descreve os experimentos realizados usando o algoritmo GVNS para a solução do PPMC. A Seção 4.8 mostra os resultados encontrados para o Grupo de Instância 1, enquanto a Seção 4.9 mostra os resultados do Grupo de Instância 2. A Seção 4.10 finaliza o capítulo, apresentando conclusões parciais a respeito do PPMC e a solução aqui proposta.

4.1 Representação e Avaliação da Solução

Uma solução s para o Problema das p -Medianas Capacitado (PPMC) é representada por dois vetores. O primeiro é um vetor com p componentes, denotado por vm , que representa o conjunto de facilidades selecionadas como medianas, cujos elementos, portanto, estão contidos em \mathcal{N}_p . O segundo vetor, denotado por vc , é um vetor com m elementos, no qual uma posição $j = 1, 2, \dots, m$ indica a qual mediana foi atribuído o j -ésimo cliente. Assim, uma solução para o problema pode ser represen-

tada por um par de vetores como $s = (vm, vc)$. Esta estrutura de representação foi proposta em [Chaves e Lorena \(2010\)](#).

Um exemplo de representação é apresentado na Figura 4.1. A Figura 4.1(a) mostra o vetor de instalações selecionadas como medianas, de modo que $\mathcal{N}_p = \{4, 7, 9\}$. A Figura 4.1(b) mostra a alocação dos $m = 10$ clientes às $p = 3$ instalações-medianas. Note que, à instalação $i = 4$, estão alocados os clientes 3, 4 e 5; à instalação $i = 7$, estão alocados os clientes 2, 7 e 10; e, à instalação $i = 9$, estão alocados os clientes 1, 6, 8 e 9.

$$vm \quad \boxed{4 \quad 7 \quad 9}$$

(a) Vetor de medianas.

$$vc \quad \boxed{9 \quad 7 \quad 4 \quad 4 \quad 4 \quad 9 \quad 7 \quad 9 \quad 9 \quad 7}$$

(b) Vetor de alocação de clientes a instalações.

Figura 4.1: Exemplo de representação de uma solução.

A função de avaliação é a própria função objetivo do problema, dada pela Expressão (2.7), que representa a soma das distâncias entre as instalações i e seus clientes j .

4.2 Construção de uma solução inicial

Este trabalho considera a construção da solução inicial de duas formas distintas. No primeiro método de construção, detalhado na Subseção 4.2.1, a geração de uma solução inicial para o PPMC é feita de forma randômica. No segundo método, detalhado na Subseção 4.2.2, a solução inicial é gerada pela fase de construção da metaheurística GRASP.

4.2.1 Construção randômica da solução inicial

O Algoritmo 3 mostra o pseudocódigo do procedimento de construção de uma solução inicial $s^0 = (vm^0, vc^0)$ de forma randômica. Os dados de entrada para este algoritmo são os conjuntos de clientes \mathcal{M} , de instalações-candidatas \mathcal{N} , o vetor Q contendo a capacidade de cada instalação-candidata, o vetor q de demanda de cada cliente, além do número p de medianas desejadas, do número m de clientes, do número n de instalações-candidatas e do número máximo de tentativas para encontrar uma solução viável, denotado por $MaxTentativas$. Esse último parâmetro foi fixado em p tentativas. O procedimento se inicia com a solução inicial vazia na linha 3; número de tentativas para encontrar uma solução viável igual a zero na linha 4; uma cópia \mathcal{M}' do conjunto de clientes \mathcal{M} na linha 5; uma cópia \mathcal{N}' do conjunto de instalações-

candidatas na linha 6; o conjunto vazio de medianas, isto é, das instalações \mathcal{N}_p a serem abertas na linha 7; e o contador do número de medianas com valor zero, na linha 8.

Em seguida, o algoritmo entra em um laço formado pelas linhas 9-15 para escolher as medianas no conjunto de instalações-candidatas. Inicialmente, o contador do número de medianas é acrescido em uma unidade na linha 10. A seguir, na linha 11, uma instalação-candidata $r \in \mathcal{N}'$ é aleatoriamente escolhida como mediana. Nas linhas 12-14, essa mediana r é inserida na δ -ésima posição do vetor de medianas vm^0 , incluída no conjunto \mathcal{N}_p de medianas e excluída do conjunto \mathcal{N}' de instalações, respectivamente. Escolhidas as medianas dentre as instalações-candidatas, inicia-se um laço, entre as linhas 16-30, para atribuir cada cliente a uma mediana. Para tanto, na linha 17, um cliente é aleatoriamente escolhido. Em seguida, verifica-se se o conjunto $\mathcal{N}_p^v \subseteq \mathcal{N}_p$ de medianas com capacidade para atender a demanda q_j do cliente j é não-vazio. Havendo medianas disponíveis para alocação, uma delas, denotada por $r \in \mathcal{N}_p^v$, é aleatoriamente escolhida para atender ao cliente j . Em seguida, nas linhas 20 e 21, respectivamente, são feitas a inserção dessa mediana na j -ésima posição do vetor de clientes vc^0 e atualização do conjunto de clientes. Não havendo uma mediana disponível para atender ao cliente j , o procedimento de construção é reinicializado na linha 24 e o número de tentativas de encontrar uma solução viável é incrementado na linha 25. Caso o número máximo de tentativas de construção tenha sido atingido, o procedimento construtivo retorna uma solução inicial vazia na linha 27. Sendo capaz de encontrar uma solução inicial viável, o algoritmo retorna a solução s^0 na linha 31.

4.2.2 Solução inicial por meio da fase de construção GRASP

O próximo algoritmo desenvolvido utiliza o procedimento de construção da metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) (Feo e Resende, 1995) para gerar uma solução inicial. Nesse procedimento, dado um conjunto \mathcal{N} de instalações-candidatas e um conjunto \mathcal{M} de clientes, cada elemento candidato $i \in \mathcal{N}$ é, inicialmente, avaliado por uma função guia $g(\cdot)$, dada por:

$$g(i) = \sum_{j \in \mathcal{M}} d_{ij} \quad \forall i \in \mathcal{N} \quad (4.1)$$

que avalia a distância de uma instalação-candidata $i \in \mathcal{N}$ a todos os clientes.

A lista restrita de candidatos (LRC) é constituída pelas “melhores” instalações-candidatas avaliadas segundo a função (4.1). A Expressão (4.2) mostra como essa lista é constituída.

$$LRC = \{i \in \mathcal{N} \mid g(i) \leq g_{\min} + \alpha \times [g_{\max} - g_{\min}]\}. \quad (4.2)$$

Algoritmo 3: Solução Inicial construída de forma randômica.

```

1 Entrada:  $\mathcal{M}, \mathcal{N}, Q, m, n, p, q, MaxTentativas$ 
2 Saída: Solução inicial  $s^0 = (vm^0, vc^0)$ , Conjunto  $\mathcal{N}_p$  de medianas
3  $s^0 \leftarrow \emptyset$ 
4  $Tentativas \leftarrow 0$ 
5  $\mathcal{M}' \leftarrow \mathcal{M}$  /* Cópia do conjunto de clientes */;
6  $\mathcal{N}' \leftarrow \mathcal{N}$  /* Cópia do conj. de instalações-candidatas */
7  $\mathcal{N}_p \leftarrow \emptyset$  /* Conjunto de medianas */
8  $\delta \leftarrow 0$  /* Contador do número de medianas criadas */
9 enquanto ( $\delta \leq p$ ) faça
10 |    $\delta \leftarrow \delta + 1$ 
11 |   Selecione aleatoriamente uma instalação  $r \in \mathcal{N}'$  para ser mediana
12 |    $vm_\delta^0 \leftarrow r$  /* Adicione a mediana  $r$  à  $\delta$ -ésima posição do vetor de medianas  $vm^0$  */
13 |    $\mathcal{N}_p \leftarrow \mathcal{N}_p \cup \{r\}$  /* Atualize o conjunto  $\mathcal{N}_p$  */
14 |    $\mathcal{N}' \leftarrow \mathcal{N}' \setminus \{r\}$  /* Atualize o conjunto  $\mathcal{N}'$  */
15 fim
16 enquanto ( $\mathcal{M}' \neq \emptyset$ ) faça
17 |   Selecione aleatoriamente um cliente  $j \in \mathcal{M}'$ 
18 |   se o conjunto  $\mathcal{N}_p^v \subseteq \mathcal{N}_p$  de medianas com capacidade para atender a demanda  $q_j$  do cliente  $j$  é
19 |   |   não-vazio então
20 |   |   |   Selecione aleatoriamente uma mediana  $r \in \mathcal{N}_p^v$ 
21 |   |   |    $vc_j^0 \leftarrow r$  /* Insira a mediana  $r$  na  $j$ -ésima posição do vetor de clientes  $vc^0$  */
22 |   |   |    $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \{j\}$  /* Atualize o conjunto de clientes */
23 |   |   senão
24 |   |   |   se  $Tentativas \leq MaxTentativas$  então
25 |   |   |   |   Reinicialize a construção, retornando à linha 3
26 |   |   |   |    $Tentativas \leftarrow Tentativas + 1$ 
27 |   |   |   senão
28 |   |   |   |   Retorne  $s_0 = \emptyset$ , ou seja, não foi encontrada uma solução viável
29 |   |   fim
30 fim
31 Retorne  $s^0 = (vm^0, vc^0)$ 

```

Na Expressão (4.2), g_{\min} representa o menor valor da função g nesse conjunto de candidatos e g_{\max} o maior valor dessa função. Observe que a LRC é controlada pelo parâmetro real $\alpha \in [0, 1]$, que define o quão gulosa ou aleatória é a construção da solução, de modo que, se $\alpha = 1$, a construção é totalmente aleatória, e se $\alpha = 0$, ela é totalmente gulosa. Formada a LRC com os melhores candidatos, um elemento $r \in LRC$ é escolhido aleatoriamente para compor a solução parcial.

A solução inicial, via o procedimento da fase de construção GRASP, foi implementada conforme o Algoritmo 4. Os dados de entrada para este algoritmo são: conjuntos de clientes \mathcal{M} e de instalações-candidatas \mathcal{N} ; vetor Q com a capacidade de cada instalação-candidata; vetor q de demanda de cada cliente; número p de medianas desejadas; número m de clientes; número n de instalações-candidatas; função $g(\cdot)$; parâmetro α ; e número máximo de tentativas $MaxTentativas$ para encontrar uma solução inicial viável, fixado em p . Como no procedimento anterior, a inicialização é realizada nas linhas 3 a 8.

O algoritmo entra, então, em um laço formado pelas linhas 9-16 para definir as medianas dentre o conjunto de instalações-candidatas. Inicialmente, o contador do número de medianas é acrescido em uma unidade na linha 10. Em seguida, na linha 12, uma instalação $r \in LRC$ é aleatoriamente escolhida como mediana dentre

Algoritmo 4: Solução Inicial do PPMC gerada via fase de construção GRASP.

```

1  Entrada:  $\mathcal{M}, \mathcal{N}, Q, m, n, p, q, g, \alpha, MaxTentativas$ 
2  Saída: Solução inicial  $s^0 = (vm^0, vc^0)$ , Conjunto  $\mathcal{N}_p$  de medianas
3   $s^0 \leftarrow \emptyset$ 
4   $Tentativas \leftarrow 0$ 
5   $\mathcal{M}' \leftarrow \mathcal{M}$ 
6   $\mathcal{N}' \leftarrow \mathcal{N}$ 
7   $\mathcal{N}_p \leftarrow \emptyset$ 
8   $\delta \leftarrow 0$ 
9  enquanto ( $\delta \leq p$ ) faça
10 |    $\delta \leftarrow \delta + 1$ 
11 |    $LRC \leftarrow \{i, i \in \mathcal{N}' \mid g(i) \leq g_{\min} + \alpha \times [g_{\max} - g_{\min}]\}$ 
12 |   Selecione aleatoriamente uma instalação  $r \in LRC$  para ser mediana
13 |    $vm_{\delta}^0 \leftarrow r$ 
14 |    $\mathcal{N}_p \leftarrow \mathcal{N}_p \cup \{r\}$ 
15 |    $\mathcal{N}' \leftarrow \mathcal{N}' \setminus \{r\}$ 
16 | fim
17 enquanto ( $\mathcal{M}' \neq \emptyset$ ) faça
18 |   Selecione aleatoriamente um cliente  $j \in \mathcal{M}'$ 
19 |   se o conjunto  $\mathcal{N}_p^v \subseteq \mathcal{N}_p$  de medianas com capacidade para atender a demanda  $q_j$  do cliente  $j$  é
20 |   não-vazio então
21 |   |   Selecione aleatoriamente uma mediana  $r \in \mathcal{N}_p^v$ 
22 |   |    $vc_j^0 \leftarrow r$ 
23 |   |    $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \{j\}$ 
24 |   |   senão
25 |   |   |   se  $Tentativas \leq MaxTentativas$  então
26 |   |   |   |   Reinicialize a construção, retornando à linha 3
27 |   |   |   |    $Tentativas \leftarrow Tentativas + 1$ 
28 |   |   |   |   senão
29 |   |   |   |   |   Retorne  $s_0 = \emptyset$ , ou seja, não foi encontrada uma solução viável
30 |   |   |   |   |   fim
31 |   |   |   fim
32 Retorne  $s^0 = (vm^0, vc^0), \mathcal{N}_p$ 

```

as instalações-candidatas. Nas linhas 13-15, essa mediana r é inserida na δ -ésima posição do vetor de medianas vm^0 , incluída no conjunto \mathcal{N}_p de medianas e excluída do conjunto \mathcal{N}' de instalações-candidatas, respectivamente. Uma vez escolhidas as medianas dentre as instalações-candidatas, inicia-se um laço, entre as linhas 17-31, para atribuir cada cliente a uma mediana. Essa atribuição é feita da mesma forma que no Algoritmo 3.

4.3 Estruturas de Vizinhança

Para explorar o espaço de soluções do problema, foram utilizados três tipos de movimentos para serem aplicados a uma solução s . Cada movimento dá origem a uma estrutura de vizinhança $\mathcal{L}_k(s)$, $k = 1, 2, 3$, definidas como segue:

- (i) $\mathcal{L}_1(s)$ (Realocação): Consiste em realocar um cliente j de uma mediana $r_1 \in \mathcal{N}_p$ para uma outra mediana $r_2 \in \mathcal{N}_p$;
- (ii) $\mathcal{L}_2(s)$ (Troca): Consiste em trocar um cliente j_1 de uma mediana $r_1 \in \mathcal{N}_p$ com um cliente j_2 de uma outra mediana $r_2 \in \mathcal{N}_p$;

Algoritmo 5: VND aplicado à solução do PPMC

```

1 Entrada: Solução  $s$ , Conjunto  $\mathcal{L}$  de vizinhanças, Número  $k_{\max}$  de vizinhanças
2 Saída: Solução  $s$  refinada
3  $k \leftarrow 1$  /* Inicie pela primeira vizinhança */
4 enquanto  $k \leq k_{\max}$  faça
5   Encontre o melhor vizinho  $s' \in \mathcal{L}'_k(s) \subset \mathcal{L}_k(s)$ 
6   se  $f(s') < f(s)$  então
7      $s \leftarrow s'$  /* Atualize a solução corrente */
8      $k \leftarrow 1$  /* Retorne à primeira vizinhança de  $\mathcal{L}$  */
9   senão
10     $k \leftarrow k + 1$  /* Passe para a próxima vizinhança de  $\mathcal{L}$  */
11  fim
12 fim
13 Retorne  $s$ 

```

(iii) $\mathcal{L}_3(s)$ (Substituição): Consiste em substituir uma mediana $r_1 \in \mathcal{N}_p$ por uma instalação-candidata $r_2 \in \mathcal{N} \setminus \mathcal{N}_p$.

4.4 Busca Local

Uma heurística de busca local é um procedimento de refinamento de uma solução s que utiliza uma ou mais estruturas de vizinhança para encontrar um ótimo local com relação às estruturas de vizinhança utilizadas na exploração do espaço de soluções.

Algoritmo 6: RVND aplicado à solução do PPMC

```

1 Entrada: Solução  $s$ , Conjunto  $\mathcal{L}$  de vizinhanças, Número  $k_{\max}$  de vizinhanças
2 Saída: Solução  $s$  refinada
3  $\mathcal{LR} \leftarrow \mathcal{L}$  em ordem aleatória
4  $k \leftarrow 1$  /* Inicie pela primeira vizinhança de  $\mathcal{LR}$  */
5 enquanto  $k \leq k_{\max}$  faça
6   Encontre o melhor vizinho  $s' \in \mathcal{LR}'_k(s) \subset \mathcal{LR}_k(s)$ 
7   se  $f(s') < f(s)$  então
8      $s \leftarrow s'$  /* Atualize a solução corrente */
9      $k \leftarrow 1$  /* Retorne à primeira vizinhança de  $\mathcal{LR}$  */
10  senão
11    $k \leftarrow k + 1$  /* Passe p/a próxima vizinhança de  $\mathcal{LR}$  */
12  fim
13 fim
14 Retorne  $s$ 

```

Foram utilizadas duas versões diferentes para o procedimento de busca local, ambas baseadas na troca sistemática das estruturas de vizinhança. A primeira delas, apresentada no Algoritmo 5, aplica o método *Variable Neighborhood Descent* (VND) (Hansen et al., 2017; Mjirda et al., 2017), em sua forma clássica. A segunda versão aplica o método *Random Variable Neighborhood Descent* (RVND) (Souza et al., 2010; Subramanian et al., 2010). A heurística RVND é uma variação do algoritmo VND e é apresentado no Algoritmo 6. A diferença entre as duas versões reside na ordem de exploração das vizinhanças. Enquanto no VND clássico a ordem das vizinhanças é pré-definida e não muda durante as chamadas ao método, no RVND essa ordem é aleatória a cada chamada do método. Essa diferença entre essas duas buscas locais

ocorre na linha 3 do Algoritmo 6, que inclui um procedimento para aleatorizar a ordem das vizinhanças. Note que, em ambos os métodos, a função $f(s)$ representa a função de avaliação definida na Subseção 4.1.

No método VND implementado, as estruturas de vizinhanças foram ordenadas como apresentado na Subseção 4.3. Esta ordenação segue o grau de complexidade da exploração das vizinhanças usadas, sendo a primeira vizinhança (\mathcal{L}_3) a de menor complexidade e a última, \mathcal{L}_3 , a de maior complexidade. Esta estratégia é normalmente adotada em buscas locais baseadas em VND. Como a determinação do ótimo local de cada vizinhança é muito custosa computacionalmente em instâncias maiores do problema, em ambos os métodos a análise da vizinhança foi reduzida a apenas uma parte dela. Mais precisamente, foram analisados na vizinhança \mathcal{L}_1 apenas m vizinhos gerados por realocações aleatórias de clientes de uma mediana para outra na solução atual. Igualmente, na vizinhança \mathcal{L}_2 foram analisados somente m vizinhos gerados por trocas aleatórias entre clientes de duas medianas na solução atual. Por fim, na vizinhança \mathcal{L}_3 foram analisados m vizinhos gerados por movimentos aleatórios de substituição de medianas $r_1 \in \mathcal{N}_p$ da solução atual por instalações-candidatas $r_2 \in \mathcal{N} \setminus \mathcal{N}_p$. Em todas essas análises, m indica o número de clientes. Para indicar que apenas uma parcela da vizinhança é analisada, essa operação será denotada por \mathcal{L}'_k .

4.5 Procedimento de Perturbação

Para diversificar a busca para outras regiões do espaço de soluções do PPMC, foi utilizado um procedimento de perturbação da solução corrente. O Algoritmo 7 mostra o seu funcionamento. Na linha 1 são dadas como entradas, a solução corrente s , a estrutura de vizinhança utilizada para a realização das perturbações, no caso, a vizinhança \mathcal{L}_3 , e o número k de perturbações a serem realizadas nessa vizinhança. Na linha 3, é feita uma cópia da solução atual na solução s' , a qual representará a solução perturbada após a aplicação do procedimento. Na linha 4, o contador i do número de perturbações a ser aplicado à solução s' é iniciado com o valor 1. Enquanto esse contador for menor ou igual ao número k de perturbações a serem aplicadas, o procedimento seleciona uma solução vizinha aleatória s'' na vizinhança da solução s' (veja linha 6). Em seguida, são atualizados a solução perturbada e o contador do número de perturbações nas linhas 7 e 8, respectivamente. O procedimento termina retornando a solução perturbada na linha 10.

Algoritmo 7: Perturbação.

```

1 Entrada: Solução atual  $s$ , Vizinhança  $\mathcal{L}_3$ , Número  $k$  de perturbações
2 Saída: Solução perturbada  $s'$ 
3  $s' \leftarrow s$  /* Cópia da solução atual */
4  $i \leftarrow 1$  /* Inicie o contador de perturbações */
5 enquanto ( $i \leq k$ ) faça
6   | Seleccione aleatoriamente uma solução vizinha  $s'' \in \mathcal{L}_3(s')$ 
7   |  $s' \leftarrow s''$  /* Atualize a solução perturbada */
8   |  $i \leftarrow i + 1$  /* Atualize o contador de perturbações */
9 fim
10 Retorne  $s'$ 

```

Algoritmo 8: GVNS

```

1 Entrada:  $\mathcal{L}$ ,  $iterMaxSM$ 
2 Saída:  $s$ 
3  $s_0 \leftarrow GeraSolucaoInicial()$  /* Conforme Algoritmos 3 ou 4 */
4  $s \leftarrow BuscaLocal(s_0, \mathcal{L}, k_{max})$  /* Conforme Algoritmos 5 ou 6 */
5  $iter \leftarrow 0$  /* Contador do número de iterações sem melhora */
6  $k \leftarrow 2$  /* Contador do número de perturbações */
7 enquanto ( $iter < iterMaxSM$ ) faça
8   |  $s' \leftarrow Perturbacao(s, \mathcal{L}_3, k)$  /* De acordo com o Algoritmo 5 */
9   |  $s'' \leftarrow BuscaLocal(s', \mathcal{L}, k_{max})$  /* Cfe. Algoritmos 5 ou 6 */
10  | se  $f(s'') < f(s)$  então
11  |   |  $s \leftarrow s''$  /* Atualize a solução atual */
12  |   |  $k \leftarrow 1$  /* Retorne à primeira vizinhança */
13  |   |  $iter \leftarrow 0$  /* Reinicialize o contador de iterações sem melhora */
14  | senão
15  |   |  $iter \leftarrow iter + 1$  /* Atualize o contador de iterações sem melhora */
16  |   |  $k \leftarrow k + 1$  /* Passe para a próxima vizinhança */
17  | fim
18 fim
19 Retorne  $s$ 

```

4.6 Algoritmo GVNS

O Algoritmo 8 mostra o pseudocódigo do algoritmo GVNS implementado. Os parâmetros de entrada são o conjunto de estruturas de vizinhanças \mathcal{L} e o número máximo permitido de iterações sem melhora $iterMaxSM$. A solução inicial s_0 (linha 1) é gerada ou aleatoriamente, como mostrado pelo Algoritmo 3, ou por meio da fase de construção do método GRASP, como mostrado pelo Algoritmo 4, ambos mostrados na Subseção 4.2.

Gerada a solução inicial, na linha 4, o algoritmo aplica sobre ela um método de busca local usando ou o procedimento VND, conforme Algoritmo 5, ou o procedimento RVND, conforme Algoritmo 6, para gerar a solução atual s . Em seguida, o algoritmo inicializa o contador do número de iterações sem melhora da solução atual e o contador k do número de perturbações a serem aplicadas sobre a solução atual para diversificar a busca para outras regiões do espaço de soluções do problema. Esse contador k é inicializado com o valor 2 porque a vizinhança utilizada pela perturbação é uma das vizinhanças usadas para aplicar a busca local. Assim, se $k = 1$, a solução perturbada seria um vizinho da solução atual e, portanto, uma busca local a partir dela alcançaria a própria solução atual. Em seguida, tendo como critério de

parada o número máximo de iterações sem melhora ($iterMaxSM$), o algoritmo entra em um laço iterativo com aplicações sucessivas dos procedimentos de perturbação da solução atual e busca local da solução perturbada para explorar as soluções do espaço de soluções do problema. A perturbação (linha 8) é aplicada sobre a solução atual usando o Algoritmo 7. Esse Algoritmo realiza k movimentos aleatórios em s , produzindo vizinhos cada vez mais distantes da solução atual quando uma melhoria não é encontrada. Desta forma, esse procedimento diversifica o caminho de busca. Tal como na linha 4, o procedimento de busca local usa ou a heurística VND, conforme Algoritmo 5, ou a heurística RVND, conforme Algoritmo 6 (linha 9). A solução s'' gerada pelo procedimento de busca local será aceita se melhorar a solução atual (linha 10-13). Caso isso ocorra, a solução atual é atualizada, a estrutura de vizinhança retorna à estrutura inicial ($k = 1$) e o procedimento continua. Caso contrário, o contador de estrutura de vizinhança é incrementado (linha 14-17). A melhor solução encontrada é retornada ao fim do processo de busca.

4.7 Descrição dos Experimentos Computacionais

Esta Seção apresenta os resultados dos experimentos computacionais realizados neste trabalho e faz uma análise estatística do desempenho das quatro variantes do algoritmo baseado na metaheurística *General Variable Neighborhood Search* (GVNS) para resolver o PPMC:

- (i) G-RVND: GVNS com a solução inicial gerada de forma randômica e busca local via *Random Variable Neighborhood Descent*;
- (ii) G-VND: GVNS com a solução inicial gerada de forma randômica e busca local via *Variable Neighborhood Descent*;
- (iii) GG-RVND: GVNS com a solução inicial gerada pela fase de construção do método GRASP e busca local via RVND; e
- (iv) GG-VND: GVNS com a solução inicial gerada pela fase de construção do método GRASP e busca local via VND.

Todos os algoritmos foram implementados na linguagem Java e testados em um computador Intel Core 2 duo 2.93 GHz, 32 bits, 3 MB de cache L2, 8 GB de memória RAM e sistema operacional Ubuntu Server.

Dois grupos de instâncias foram utilizados nos testes computacionais. O Grupo de instâncias 1, proposto em Chaves e Lorena (2010), possui 5 instâncias que variam de 3038 nós e 600 medianas a 3038 nós e 1000 medianas. Os resultados dos algoritmos

neste grupo estão reportados na Seção 4.8. O Grupo de instâncias 2, proposto em [Stefanello et al. \(2015\)](#), totaliza 30 instâncias, que variam de 318 nós e 5 medianas a 4461 nós e 1000 medianas. Os resultados para este grupo são mostrados na Seção 4.9. Para os testes computacionais, estes grupos de instâncias foram executados 30 vezes.

Para comparar a qualidade das soluções produzidas pelos algoritmos, calcula-se o desvio relativo médio Δ_i^A :

$$\Delta_i^A = \frac{f_i^A - f_i^*}{f_i^*} \times 100, \quad (4.3)$$

entre o valor f_i^A encontrado pelo algoritmo A para a instância i e a melhor solução encontrada na literatura para a respectiva instância, representada por f_i^* . Este valor, portanto, representa o desvio percentual em relação ao melhor resultado da literatura e, assim, quanto mais próximo de 0, melhor é o resultado do algoritmo A .

Os valores dos parâmetros de todos os algoritmos implementados foram calibrados pela ferramenta IRACE (*Iterated Racing for Automatic Algorithm Configuration*) ([López-Ibáñez et al., 2016](#)). Para o treinamento da ferramenta, foram utilizadas as instâncias `lin318_70`, `ali535_100`, `u724_125`, `r11304_200`, `pr_300` e `p3038_800`. A Tabela 4.1 mostra os resultados desse treinamento. Nessa tabela, a primeira coluna apresenta o parâmetro testado, a segunda descreve seu significado e a terceira, a faixa de valores testados. A última coluna mostra o valor retornado pela ferramenta IRACE.

Tabela 4.1: Faixa de valores testados e retornados pela ferramenta IRACE para os parâmetros dos algoritmos nos Grupos de instâncias 1 e 2

Parâmetro	Descrição	Faixa de valores	Valor retornado
α	Valor utilizado pela LRC	0,1 a 0,9	0,40
<i>iterMaxSM</i> c/busca local VND	Número máximo de iterações sem melhora	{500, 600, 700, 800 }	600
<i>iterMaxSM</i> c/busca local RVND	Número máximo de iterações sem melhora	{500, 600, 700, 800 }	700

4.8 Resultados para o Grupo de instâncias 1

Esta seção apresenta os resultados dos testes computacionais referentes ao Grupo de instâncias 1. Primeiramente, na Subseção 4.8.1, compara-se o desempenho das quatro variantes propostas, de modo a eleger aquela que possui o melhor comportamento. Os resultados desta variante são comparados com os resultados de outros algoritmos encontrados na literatura, tanto no que diz respeito aos valores obtidos, na Subseção 4.8.2, quanto, na Subseção 4.8, em relação ao tempo de execução despendido.

4.8.1 Comparação entre os quatro algoritmos propostos

A Tabela 4.2 apresenta os resultados referentes à aplicação dos algoritmos G-RVND, G-VND, GG-RVND e GG-VND para o Grupo de instâncias 1, no que diz respeito aos desvios em relação aos resultados encontrados em Lorena e Senne (2004). A primeira coluna caracteriza as instâncias, na forma $n-p$, de modo que $n = 3038$ indica o número de clientes e p o número de medianas. A segunda coluna mostra os melhores resultados encontrados em Lorena e Senne (2004). Entre a terceira até a décima colunas são apresentados os valores dos desvios relativos médios $\Delta\%$, conforme a Expressão (4.3), e o tempo de execução associado ao uso de cada um dos quatro algoritmos implementados.

Tabela 4.2: Grupo de instâncias 1 - Valores médios para 30 execuções

Instância teste	Melhor resultado da literatura	G-RVND		G-VND		GG-RVND		GG-VND	
		$\Delta(\%)$	tempo(s)	$\Delta(\%)$	tempo(s)	$\Delta(\%)$	tempo(s)	$\Delta(\%)$	tempo(s)
p3038-600	122020,66	2,42	396,15	0,90	425,59	1,20	359,56	0,38	517,50
p3038-700	108685,59	2,60	467,28	0,53	544,48	1,54	458,07	0,63	660,18
p3038-800	98530,99	1,70	538,80	1,63	625,50	1,22	566,76	0,29	765,17
p3038-900	90239,65	2,11	803,41	1,92	789,47	1,48	709,60	0,28	942,04
p3038-1000	83231,58	2,39	1154,76	0,92	1069,50	0,78	1026,79	0,48	986,82

A Tabela 4.3 mostra o p -valor obtido pelo teste de normalidade dos valores dos desvios para cada um dos conjuntos de dados associados à implementação dos algoritmos no Grupo de instâncias 1. O Teste utilizado foi o de Shapiro-Wilk, que tem como objetivo avaliar se uma distribuição é semelhante a uma distribuição normal. Para dizer que uma distribuição é normal, o p -valor precisa ser maior do que 0,05, desta forma, podemos dizer que esses dados apresentam normalidade estatística. Sendo assim, utilizou-se o Teste F (Montgomery e Runge, 2016) para verificar se há diferença estatística significativa entre os algoritmos. A Figura 4.2 mostra o gráfico *box-plot* dos algoritmos propostos quando aplicados ao Grupo de instâncias 1 em relação ao desvio relativo médio Δ .

Tabela 4.3: Grupo de instâncias 1 – Teste de normalidade dos valores de desvio

	G-RVND	G-VND	GG-RVND	GG-VND
p -valor	0,8167	0,9286	0,6197	0,4513

Tabela 4.4: Grupo de instâncias 1: p -valores referentes a aplicação do Teste F para avaliar a diferença entre os valores de desvio percentual Δ

Algoritmo	GG-VND	GG-RVND	G-VND	G-RVND
G-RVND	0,0022	0,0636	0,01406	–
G-VND	6,2052E-06	0,4371	–	–
GG-RVND	3,3022E-05	–	–	–

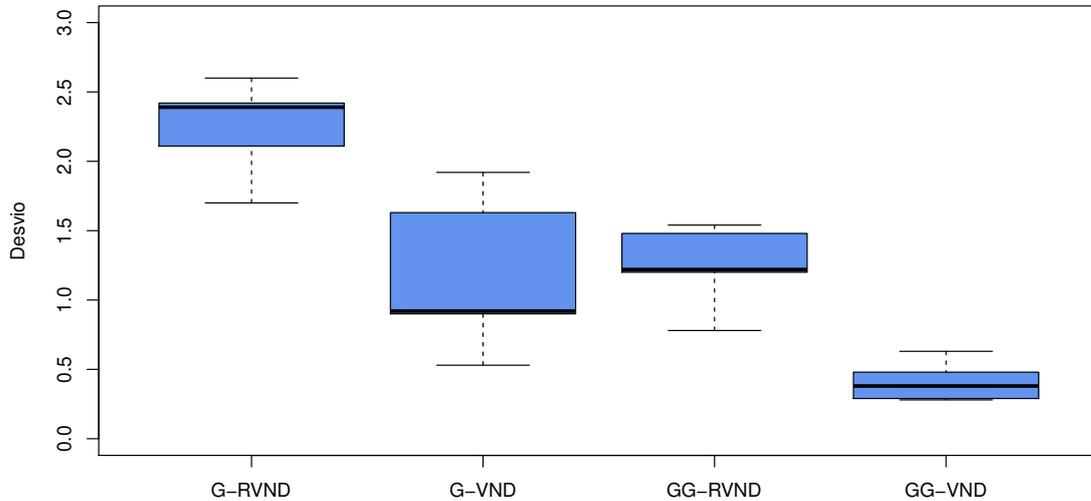


Figura 4.2: Gráfico *box-plot* mostrando o desvio relativo médio $\Delta\%$ entre os algoritmos propostos no Grupo de instâncias 1.

Analisando o gráfico da Figura 4.2 e os p -valores da Tabela 4.4, observa-se que há diferença estatística significativa entre a variante GG-VND e as demais, pois o p -valor apresentado é inferior a 0,05. Portanto, comprova-se a superioridade desta variante em relação às demais, uma vez que, pela Tabela 4.2, os valores dos desvios relativos do GG-VND foram os menores dentre todos os algoritmos testados.

4.8.2 Grupo de instâncias 1: Comparação com os resultados da literatura

Os testes realizados na Subseção 4.8.1 mostraram que, estatisticamente, o Algoritmo GG-VND é superior aos algoritmos G-RVND, G-GND e GG-RVND ao solucionar o Grupo de instâncias 1. Portanto, nas seções subsequentes, apenas o Algoritmo GG-VND será comparado em relação aos melhores resultados da literatura para este grupo de instâncias, apresentados em Lorena e Senne (2004), bem como aos resultados apresentados em Stefanello et al. (2015), Chaves e Lorena (2010) e Jánošíková et al. (2017).

A Tabela 4.5 mostra esta comparação. Nesta tabela, a primeira coluna descreve o conjunto de instâncias abordado. Entre a terceira e a sexta colunas são exibidos os melhores resultados para a função objetivo encontrados em Stefanello et al. (2015), Chaves e Lorena (2010), Jánošíková et al. (2017) e o referente ao algoritmo GG-VND. Em seguida, entre a sétima e a décima colunas são mostrados os resultados dos desvios relativos médios, calculados como determinado em (4.3), entre os melhores

resultados da literatura, apresentados em [Lorena e Senne \(2004\)](#), os resultados dos algoritmos da literatura e os do Algoritmo GG-VND.

Tabela 4.5: Grupo de instâncias 1: comparação do valor de função objetivo e dos desvios relativos médios obtidos pelo Algoritmo GG-VND com a literatura

Instâncias de teste	Melhor Resultado da literatura	Valor de função objetivo				Δ (%)			
		Stefanello	Chaves	Jánošíková	GG-VND	Stefanello	Chaves	Jánošíková	GG-VND
p3038-600	122020,66	122724,79	129194,11	125638,33	122752,19	0,57	5,879	2,96	0,38
p3038-700	108685,59	109695,61	117295,47	114977,77	109381,22	0,92	7,92	5,78	0,63
p3038-800	98530,99	100084,41	109532,61	105483,82	99054,98	1,57	11,16	7,05	0,29
p3038-900	90239,65	92317,78	102458,93	100372,64	90969,96	2,30	13,54	11,22	0,28
p3038-1000	83231,58	85856,05	97777,67	96290,2	83693,38	3,15	17,47	5,68	0,48

Notação para as referências bibliográficas citadas:

- (1) Melhor Resultado da literatura: [Lorena e Senne \(2004\)](#);
- (2) Chaves: [Chaves e Lorena \(2010\)](#);
- (3) Jánošíková: [Jánošíková et al. \(2017\)](#);
- (4) Stefanello: [Stefanello et al. \(2015\)](#).

De acordo com o p -valor, maior que 0,05, obtido ao realizar o Teste de Shapiro, apresentado na Tabela 4.6, conclui-se que os dados apresentam padrão de normalidade. Sendo assim, foi aplicado o Teste F para verificar se existe diferença significativa entre o desvio relativo obtido pelo Algoritmo GG-VND e os desvios referentes aos demais trabalhos da literatura. O p -valor do teste F é apresentado na Tabela 4.7. Tendo em vista o p -valor, menor que 0,05, apresentado na Tabela 4.7 e o gráfico de *box-plot* da Figura 4.3, conclui-se que há diferença significativa dos resultados do Algoritmo GG-VND em relação aos dos algoritmos apresentados em [Chaves e Lorena \(2010\)](#), [Jánošíková et al. \(2017\)](#) e [Stefanello et al. \(2015\)](#). Por fim, analisando-se a Tabela 4.5, conclui-se que o Algoritmo GG-VND possui melhor desempenho que estes algoritmos da literatura.

Tabela 4.6: Grupo de instâncias 1: Teste de normalidade dos valores de desvio \times literatura. Intervalo de confiança: $\rho = 95\%$

	Stefanello	Chaves	Jánošíková	GG-VND
p -Value	0,8045	0,08549	0,548	0,4513

Notação para as referências bibliográficas citadas:

- (1) Chaves: [Chaves e Lorena \(2010\)](#);
- (2) Jánošíková: [Jánošíková et al. \(2017\)](#);
- (3) Stefanello: [Stefanello et al. \(2015\)](#).

4.8.3 Grupo de instâncias 1: Comparação em relação aos tempos de execução

Esta seção apresenta a comparação dos tempos de execução computacional do algoritmo GG-VND frente aos encontrados para os algoritmos de [Chaves e Lorena \(2010\)](#), [Stefanello et al. \(2015\)](#) e [Jánošíková et al. \(2017\)](#) na solução do Grupo de

Tabela 4.7: Grupo de instâncias 1: Teste F dos valores de desvio relativo percentual $\Delta \times$ valores da literatura. Intervalo de confiança: $\rho = 95\%$

	Stefanello	Chaves	Janosikova
<i>p</i> -valor GG-VND	0,0022	6,2052E-06	3,3022E-05

Notação para as referências bibliográficas citadas:

- (1) Chaves: Chaves e Lorena (2010);
- (2) Jánošíková: Jánošíková et al. (2017);
- (3) Stefanello: Stefanello et al. (2015).

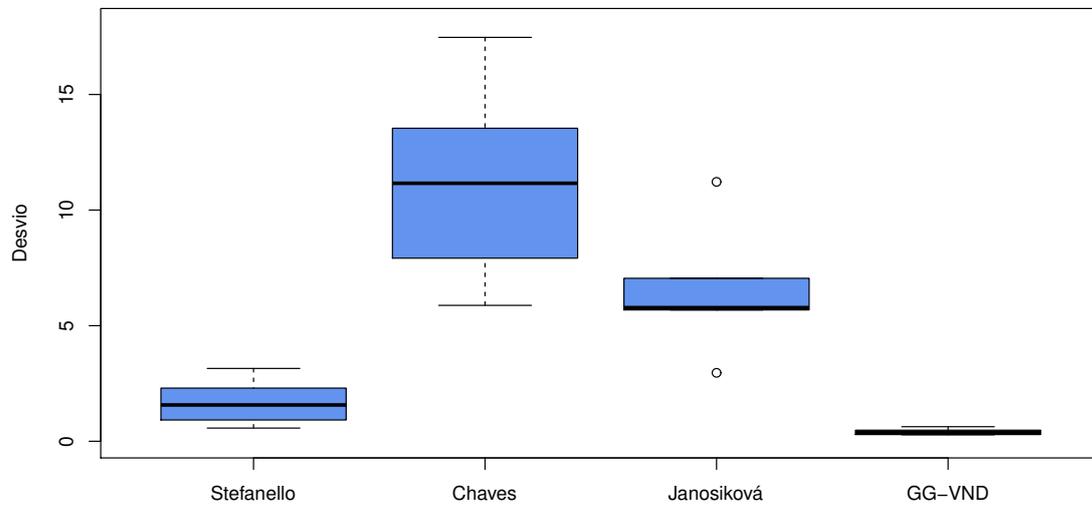


Figura 4.3: Gráfico *box-plot* mostrando o desvio relativo médio Δ entre o algoritmo GG-VND e os da literatura no Grupo de instâncias 1.

instâncias 1. É importante destacar que não é uma tarefa simples comparar os tempos computacionais entre algoritmos executados em diferentes CPUs, linguagens de programação, *solvers* e sistemas operacionais. Para melhor comparação, usou-se, como referência, a Tabela 4.8, que apresenta fatores de ajuste, obtidos em <https://www.cpubenchmark.net/compare/>, com acesso em 17 de novembro de 2019, que permitem auxiliar nessa tarefa.

A Tabela 4.9 mostra a comparação entre o tempo de execução do Algoritmo GG-VND e o tempo de execução alcançado pelos algoritmos de Chaves e Lorena (2010),

Tabela 4.8: Grupo de instâncias 1: comparação entre CPUs

	Fator de desempenho CPU				Fator de correção (%)		
	GG-VND	Stefanello	Chaves	Jánošíková	Stefanello	Chaves	Jánošíková
<i>single-thread</i>	1143	1575	793	2125	1,3779	0,6937	1,8591

Notação para as referências bibliográficas citadas:

- (1) Chaves: Chaves e Lorena (2010);
- (2) Jánošíková: Jánošíková et al. (2017);
- (3) Stefanello: Stefanello et al. (2015).

Stefanello et al. (2015) e Jánošíková et al. (2017). Nesta tabela, a sexta, a sétima e a oitava colunas apresentam os valores de tempos de execução corrigidos a partir dos fatores da Tabela 4.8 e, portanto, conclui-se que o Algoritmo GG-VND requer menor tempo de processamento e, em vista dos resultados apresentados na Tabela 4.5, possui melhor desempenho que os demais algoritmos analisados.

Tabela 4.9: Grupo de instâncias 1: comparação com os trabalhos da literatura quanto ao tempo de execução

Instâncias de teste	Resultados da literatura (s)			Tempo de execução ajustado (s)			Tempo (s) GG-VND
	Chaves	Jánošíková	Stefanello	Chaves	Jánošíková	Stefanello	
p3038-600	2685,38	39379,33	3600	3729,69	27171,73	6792,45	517,50
p3038-700	2239,84	47286,32	3600	3110,88	32627,56	6792,45	660,18
p3038-800	2819,26	54584,69	3600	3915,63	37663,43	6792,45	765,17
p3038-900	1578,17	63040,95	3600	2191,90	43498,25	6792,45	942,04
p3038-1000	1874,08	70413,38	3600	2602,88	48585,23	6792,45	986,82

Notação para as referências bibliográficas citadas:

- (1) Chaves: Chaves e Lorena (2010);
- (2) Jánošíková: Jánošíková et al. (2017);
- (3) Stefanello: Stefanello et al. (2015).

4.9 Resultados para o Grupo de instâncias 2

Esta seção apresenta os resultados referentes à aplicação do Algoritmo GG-VND para a solução do Grupo de instâncias 2, proposto em Stefanello et al. (2015). Este grupo de instâncias é dividido em subgrupos, em que o primeiro número indica o número de clientes (n) e o segundo o número de medianas (p) adotado. Esta Seção está organizada da seguinte forma. A Subseção 4.9.1 apresenta a comparação entre os resultados alcançados pelo Algoritmo GG-VND e os resultados encontrados em Stefanello et al. (2015) e Steglich (2019) para este mesmo Grupo. A Subseção 4.9 analisa os tempos de execução computacional dos algoritmos na solução deste Grupo de instâncias.

4.9.1 Grupo de instâncias 2: Comparação com os resultados da literatura

A Tabela 4.10 mostra os resultados médios obtidos pelo Algoritmo GG-VND e a comparação em relação aos melhores valores conhecidos da literatura (MVCL), encontrados em <http://www-usr.inf.ufsm.br/~stefanello/instances/cmp/>, com os alcançados em Stefanello et al. (2015) e Steglich (2019). Note que Steglich (2019) não apresenta resultados para as instâncias das classes lin318- p e ali535- p .

Observa-se que os desvios em relação ao MVCL obtidos pelo Algoritmo GG-VND são menores que os das demais referências em 18 das 30 instâncias desse Grupo. Em

Tabela 4.10: Grupo de instâncias 2: Comparação com os resultados da literatura quanto ao valor médio da função objetivo e desvio (%) em relação aos melhores valores conhecidos

Instância de teste	MVCL	Valor médio da Função Objetivo			Desvio (%) em relação aos MVCL		
		Stefanello	Steglich	GG-VND	Stefanello	Steglich	GG-VND
lin318-005	180281,2	180281,2	–	180284,20	0,00	–	0,03
lin318-015	88901,56	88901,56	–	88905,20	0,00	–	0,02
lin318-040	47988,38	48003,88	–	47995,69	0,03	–	0,07
lin318-070	32198,64	32290,39	–	32200,53	0,28	–	0,03
lin318-100	22942,69	22942,69	–	23168,12	0,00	–	1,02
ali535-005	9956,77	10210,58	–	9956,60	2,54	–	0,07
ali535-025	3695,15	3701,88	–	3695,22	0,18	–	0,06
ali535-050	2460,93	2478,04	–	2510,21	0,69	–	2,21
ali535-100	1437,62	1448	–	1451,92	0,72	–	1,05
ali535-150	1032,28	1037,7	–	1032,45	0,52	–	0,03
u724-010	181580,86	182611,19	182382	181586,01	0,56	0,44	0,08
u724-030	95034,01	95159,96	95824	95043,90	0,13	0,83	0,02
u724-075	54735,05	54735,05	54867	54737,77	0,00	0,24	0,04
u724-125	38976,76	38976,76	39808	38976,60	0,00	2,13	0,06
u724-200	28079,97	28082,72	28881	28592,14	0,00	2,85	1,91
rl1304-010	2146484,1	2166552	2149366	2158039,60	0,93	0,13	0,61
rl1304-050	799694,32	806425,28	804254	799718,88	0,84	0,57	0,10
rl1304-100	495925,93	498411,69	502275	495945,74	0,50	1,28	0,03
rl1304-200	276977,60	276983,73	280879	277477,09	0,00	1,40	0,17
rl1304-300	191224,85	191258,69	195690	192995,41	0,01	2,33	0,87
pr2392-020	2235376,7	2250292,40	2235790	2235448,34	0,66	0,01	0,06
pr2392-075	1092294	1098560	1092917	1092362,69	0,57	0,05	0,05
pr2392-150	711115,25	711315,15	715719	711157,29	0,02	0,64	0,03
pr2392-300	458145,29	458221,63	462726	458170,55	0,01	0,99	0,07
pr2392-500	316042,97	316092,46	323348	316056,12	0,01	2,31	0,06
fnl4461-0020	1283536,7	1292621,6	1284037	1296295,22	0,70	0,03	1,07
fnl4461-0100	548909,01	550758,21	549126	548917,23	0,33	0,03	0,05
fnl4461-0250	335888,87	336006,96	337205	335911,49	0,03	0,39	0,06
fnl4461-0500	224662,49	224684,37	226283	228165,55	0,00	0,72	1,61
fnl4461-1000	145862,47	145870,78	148454	147703,03	0,00	1,77	1,37

Notação para as referências bibliográficas citadas:

(1) Stefanello: [Stefanello et al. \(2015\)](#);

(2) Steglich: [Steglich \(2019\)](#).

10 instâncias, os resultados de [Stefanello et al. \(2015\)](#) são superiores aos demais. Nas 2 instâncias restantes, os resultados de [Steglich \(2019\)](#) são superiores. É interessante ressaltar que o Algoritmo GG-VND não apresenta diferenças na qualidade dos resultados com instâncias de pequena, média ou grande dimensão, seja quanto ao número de clientes, seja quanto ao número de medianas, se adaptando completamente a estas diferentes situações. A avaliação estatística desse resultado é mostrada a seguir.

A Tabela 4.11 mostra os resultados do teste de normalidade dos valores de desvio percentual para o Grupo de instâncias 2. Como a normalidade não foi comprovada, p -valor menor do que 0,05, para todas as instâncias, foi realizado o Teste não-paramétrico de Kruskal-Wallis para comparar os resultados do Algoritmo GG-VND com os resultados da literatura. Os resultados deste teste, apresentados na Tabela 4.12, mostram que não há diferença estatística entre os algoritmos avaliados para todas as instâncias de teste, p -valor maior que o índice de significância 0,05. No entanto, deve-se enfatizar que os valores de desvio relativo médio percentual mos-

Tabela 4.11: Grupo 2: teste de normalidade dos valores de desvio percentual \times valores da literatura, considerando valores médios do Algoritmo GG-VND. Intervalo de confiança: $\rho = 95\%$.

Subgrupo de Instâncias	Stefanello	Steglich	GG-VND
lin318-p	0,01361	–	0,0001
ali535-p	0,03903	–	0,0469
u724-p	0,00973	0,3420	0,0001
rl1304-p	0,2339	0,8802	0,2570
pr2392-p	0,02252	0,2829	0,4925
fnl4461-p	0,06954	0,1665	0,1882

Notação para as referências bibliográficas:

(¹) Stefanello: [Stefanello et al. \(2015\)](#);

(²) Steglich: [Steglich \(2019\)](#).

Tabela 4.12: Grupo de instâncias 2: teste de significância estatística dos valores de desvio percentual \times valores de literatura

Subgrupo de Instâncias	Stefanello	Steglich
lin318-p	0,0454	–
ali535-p	0,406	–
u724-p	0,6547	0,03623
rl1304-p	0,9283	0,9842
pr2392-p		
textcolorred	3,542e-04	2,5137e-05
fnl4461-p	0,2547	0,9995

Notação para as referências bibliográficas:

(¹) Stefanello: [Stefanello et al. \(2015\)](#);

(²) Steglich: [Steglich \(2019\)](#).

trados na Tabela 4.10 indicam um melhor desempenho do Algoritmo GG-VND em relação aos algoritmos usados para comparação.

4.9.2 Grupo de instâncias 2: Comparação em relação aos tempos de execução

A Tabela 4.13 mostra a comparação entre o Algoritmo GG-VND e os algoritmos encontrados em [Stefanello et al. \(2015\)](#) e [Steglich \(2019\)](#), com relação ao tempo de execução computacional, em segundos. Novamente, a superioridade do Algoritmo GG-VND proposto é realçada, uma vez que, tendo em vista os tempos computacionais corrigidos, o Algoritmo GG-VND possui tempo de execução menor em 28 das 30 instâncias analisadas. Nas 2 instâncias restantes, [Steglich \(2019\)](#) apresenta tempo computacional menor.

Novamente, é interessante ressaltar que o Algoritmo GG-VND se adaptou às di-

Tabela 4.13: Grupo de instâncias 2: análise de tempo de execução dos métodos

Instâncias de teste	Tempo de execução da literatura (s)		Tempo de execução ajustado (s)		Tempo de execução (s) GG-VND
	Stefanello	Steglich	Stefanello	Steglich	
lin318-005	9,15	–	13,43	–	2,10
lin318-015	26,35	–	38,67	–	2,61
lin318-040	319,46	–	468,92	–	2,95
lin318-070	127,45	–	187,07	–	4,32
lin318-100	364,87	–	535,57	–	5,91
ali535-005	45,42	–	66,66	–	6,08
ali535-025	544,26	–	798,89	–	7,60
ali535-050	726,3	–	1066,10	–	11,42
ali535-100	637,64	–	935,96	–	18,12
ali535-150	761,31	–	1117,49	–	28,95
u724-010	59,65	0,98	87,55	4,28	5,36
u724-030	300,72	6,63	441,41	28,99	23,55
u724-075	546,39	14,68	802,01	64,20	26,39
u724-125	643,31	25,99	944,28	113,67	35,06
u724-200	706,29	34,53	1036,72	151,02	69,75
rl1304-010	181,66	1,63	266,65	7,12	49,39
rl1304-050	1200	14,7	1761,39	64,29	90,17
rl1304-100	1634,2	35,04	2398,73	153,25	79,73
rl1304-200	1227,8	71,41	1802,19	312,32	100,98
rl1304-300	951,75	117,7	1397,02	514,70	170,23
pr2392-020	551,82	6,07	809,99	26,54	32,81
pr2392-075	825,85	51,11	1212,22	223,54	38,09
pr2392-150	2019,20	119	2963,93	520,38	65,25
pr2392-300	2382,4	171,1	3496,99	748,52	101,32
pr2392-500	2402,6	233	3526,66	1019,03	200,47
fnl4461-0020	538,97	21,24	791,12	92,89	79,25
fnl4461-0100	3880,4	92,58	5695,80	404,92	75,01
fnl4461-0250	4592,7	178,9	6741,34	782,28	167,23
fnl4461-0500	3912,4	286,9	5742,76	1254,73	323,97
fnl4461-1000	3433,3	57	5039,51	2519,18	599,95

Notação para as referências bibliográficas citadas:

(1) Stefanello: [Stefanello et al. \(2015\)](#);

(2) Steglich: [Steglich \(2019\)](#).

ferentes situações quanto ao número de clientes, número de medianas e dimensão das instâncias. Em especial, deve-se ressaltar que, para a instância de maior dimensão (fnl4461- p), em todos os casos o tempo computacional do algoritmo proposto é menor que o encontrado nas referências avaliadas.

4.10 Considerações Parciais

Este Capítulo reviu o algoritmo *General Variable Neighborhood Search* – GVNS ([Hansen et al., 2017](#)), e apresentou sua aplicação para a solução do Problema de p -Medianas Capacitado (PPMC). Foi apresentado também como é feita a representação e avaliação de uma solução. Além disso, foram descritas as construções da solução inicial randômica e da solução inicial por meio da fase de construção da metaheurística *Greedy Randomized Adaptive Search Procedure* – GRASP ([Feo e Resende, 1995](#)). Para

explorar o espaço de soluções do problema, foram utilizados três tipos de movimentos nas estruturas de vizinhança. São elas: (i) Realocação, que consiste em realocar um cliente de uma mediana para uma outra mediana; (ii) Troca, que consiste em trocar um cliente de uma mediana com um cliente de uma outra mediana; e (iii) Substituição, que consiste em substituir uma mediana por uma instalação-candidata. Na Seção 4.4, são apresentadas as duas diferentes versões utilizadas para o procedimento de busca local, ambas baseadas na troca sistemática das estruturas de vizinhança. A primeira delas, apresentada no Algoritmo 5, aplica o método *Variable Neighborhood Descent* (VND) (Hansen et al., 2017; Mjirda et al., 2017), em sua forma clássica. A segunda versão aplica o método *Random Variable Neighborhood Descent* (RVND) (Souza et al., 2010; Subramanian et al., 2010). Na Seção 4.5 é apresentado o Algoritmo 7, que mostra, com detalhes, como é feita a perturbação em uma determinada solução, a fim de diversificar a busca para outras regiões do espaço de soluções. Na Seção 4.6 é apresentado o algoritmo proposto GVNS. Por fim, das Seções 4.7 até 4.9, são apresentados os resultados computacionais.

Quatro variantes da metaheurística GVNS foram estudadas para a solução do PPMC, nomeadas nesta tese de G-VND, G-RVND, GG-RVND e GG-VND. As variantes do GVNS foram testadas em dois conjuntos de instâncias da literatura com número elevado de depósitos e clientes, nomeadas como Grupo de instâncias 1 e Grupo de instâncias 2. Primeiramente, o desempenho destas quatro variantes foi avaliado no Grupo 1. A partir de análises estatísticas dos resultados, comprovou-se a superioridade da variante GG-VND sobre as demais. Isto se explica pelo fato de a variante GG-VND utilizar, em sua solução inicial, a fase de construção do método GRASP, que gera uma solução inicial de boa qualidade, ao mesmo tempo que o método VND, como busca local, promove buscas exaustivas nas vizinhanças exploradas.

Em seguida, comparou-se o desempenho da variante GG-VND com os principais algoritmos da literatura utilizados para a solução do PPMC. Para o Grupo 1, comprovou-se, após a análise estatística associada, a superioridade desta variante em relação aos algoritmos de Chaves e Lorena (2010), Jánošíková et al. (2017) e Stefanello et al. (2015), no que diz respeito aos valores de desvio médio. Quanto ao tempo de execução computacional, usando-se fatores de correção para tornar justa a comparação entre as diferentes CPUs utilizadas, mostra-se que a variante GG-VND também possui melhor desempenho que os demais algoritmos analisados. Para o Grupo 2, quanto ao desvio médio relativo, a variante produziu resultados melhores em 17 das 30 instâncias analisadas. No que diz respeito ao tempo de execução, novamente utilizando fatores de correção, a variante GG-VND demandou menor tempo de execução em 28 das 30 instâncias. Portanto, conclui-se que esta variante mostrou-se fortemente competitiva frente aos algoritmos existentes na literatura para a solução do PPMC.

No Capítulo 5 é apresentado o algoritmo proposto para o Problema das p -Medianas Robusto Capacitado (PPMRC).

Capítulo 5

Solução para o Problema das p -Medianas Robusto Capacitado (PPMRC)

Este Capítulo introduz uma proposta de solução para o Problema das p -Medianas Robusto Capacitado (PPMRC). Essa proposta se baseia na metaheurística *General Variable Neighborhood Search* (GVNS). Está organizado da seguinte forma. A Seção 5.1 apresenta a representação da solução e como uma solução é avaliada. A Seção 5.2 descreve o algoritmo construtivo utilizado para encontrar uma solução inicial viável para o PPMRC. As estruturas de vizinhança usadas para explorar o espaço de solução do problema são apresentadas na Seção 5.3. Os procedimentos de busca local e de perturbação são apresentados nas Seções 5.4 e 5.5, respectivamente. Finalmente, a Seção 5.6 apresenta o algoritmo GVNS completo para resolver o PPMRC.

5.1 Representação e avaliação da solução

Uma solução s para o PPMRC é representada por dois vetores. O primeiro é um vetor com p componentes, denotado por vm , que representa o conjunto de facilidades selecionadas como medianas, cujos elementos, portanto, estão contidos em \mathcal{N}_p . O segundo vetor, denotado por vc , é um vetor com m elementos, no qual uma posição $j = 1, 2, \dots, m$ representa a qual mediana foi atribuído o j -ésimo cliente. Assim, uma solução para o problema pode ser representada por um par de vetores como $s = (vm, vc)$. Esta estrutura de representação foi proposta em [Chaves e Lorena \(2010\)](#).

Um exemplo desta representação é apresentado na Figura 5.1. A Figura 5.1(a) mostra o vetor com as $p = 3$ medianas selecionadas, em que $\mathcal{N}_p = \{2, 3, 8\}$. A

Figura 5.1(b) mostra a alocação de $m = 10$ clientes para as medianas. Observe que os clientes 1, 2, 4, 5 e 9 são atribuídos à mediana 2; os clientes 3 e 7 são atribuídos à mediana 3; e os clientes 6, 8 e 10 são atribuídos à mediana 8.

$$vm = \boxed{2 \mid 3 \mid 8}$$

(a) Vetor Mediana.

$$vc = \boxed{2 \mid 2 \mid 3 \mid 2 \mid 2 \mid 8 \mid 3 \mid 8 \mid 2 \mid 8}$$

(b) Vetor de atribuições do cliente.

Figura 5.1: Representação de uma solução dada por $s = (vm, vc)$.

As heurísticas propostas permitem apenas soluções viáveis para o problema. Assim, a função de avaliação para a solução s , denotada por $f(s)$, é baseada na função objetivo do problema, dada pela Expressão (2.7), que representa a soma das distâncias entre as instalações i e seus clientes j . A função de avaliação, considerando a representação da solução $s = (vm, vc)$, e assumindo que $\mathcal{M} = \{1, 2, \dots, m\}$ e $\mathcal{N} = \{1, 2, \dots, n\}$, é dada por:

$$f(s) = \sum_{j=1}^m d_{vc_j j} \quad (5.1)$$

No entanto, durante o procedimento de busca local, essa avaliação é realizada de forma eficiente, ou seja, apenas as alterações que foram realizadas na solução são consideradas no cálculo.

5.2 Construção da solução inicial

A solução inicial é gerada pela fase construtiva da metaheurística *Greedy Randomized Adaptive Search Procedure* (GRASP) (Feo e Resende, 1995), descrita de maneira geral na Subseção 3.2.2. A implementação realizada está representada no Algoritmo 9. Esta abordagem foi escolhida devido ao desafio de encontrar uma boa solução que atenda à restrição de capacidade da mediana, o que pode exigir muitas tentativas para construir uma solução viável. Além disso, como apresentado na Seção 4.7, a versão do GVNS que utiliza a fase construtiva do GRASP apresentou melhores resultados na resolução do PPMC.

A solução inicial, através do procedimento da fase de construção GRASP, é gerada usando o Algoritmo 9. Os dados de entrada para este algoritmo são:

- (i) o parâmetro α ; e
- (ii) o número máximo de tentativas, denotado por $MaxTrials$, para encontrar uma solução inicial viável.

Algoritmo 9: Geração da solução inicial do PPMRC via fase de construção GRASP

```

1  Entrada:  $\alpha$ ,  $MaxTentativas$ 
2  Saída: Solução inicial  $s^0 = (vm^0, vc^0)$ 
3   $s^0 \leftarrow \emptyset$ 
4   $\mathcal{M}' \leftarrow \mathcal{M}$  /* Cópia do conjunto do cliente */
5   $\mathcal{N}' \leftarrow \mathcal{N}$  /* Cópia do conj. de instalações-candidatas */
6   $S_i \leftarrow \emptyset$  /* Conjunto de índices dos  $\Gamma$  maiores desvios máximos para a mediana  $i$  */
7   $\delta_i \leftarrow 0$  /* Soma dos  $\Gamma$  maiores desvios máximos para os cliente atribuídos à mediana  $i$  */
8   $\mathcal{D}_i \leftarrow 0$  /* Soma das demandas centrais do cliente atribuído à mediana  $i$ . */
9  for  $l \leftarrow 1$  to  $p$  do
10 |    $LRC \leftarrow \{i \in \mathcal{N}' \mid g(i) \leq g_{\min} + \alpha [g_{\max} - g_{\min}], \text{ and } i \neq vm_l, \forall l = 1, 2, \dots, p\}$ 
11 |    $vm_l^0 \leftarrow$  Local selecionado aleatoriamente de  $LRC$  para ser uma mediana
12 |    $\mathcal{N}' \leftarrow \mathcal{N}' \setminus \{vm_l^0\}$  /* Atualize o conjunto  $\mathcal{N}'$  */
13 end
14 while (Houver um cliente não alocado a uma mediana) do
15 |    $j \leftarrow$  Cliente selecionado aleatoriamente em  $\mathcal{M}'$ 
16 |   Tentativas  $\leftarrow 0$ 
17 |    $i \leftarrow$  Mediana selecionada aleatoriamente em  $vm^0$ 
18 |   if ( $|\mathcal{S}_i| < \Gamma$ ) and ( $q_j + r_j + \mathcal{D}_i + \delta_i \leq Q_i$ ) then
19 |      $vc_j^0 \leftarrow i$ 
20 |      $S_i \leftarrow S_i \cup \{j\}$ 
21 |      $\delta_i \leftarrow \delta_i + r_j$ 
22 |      $\mathcal{D}_i \leftarrow \mathcal{D}_i + q_j$ 
23 |      $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \{j\}$  /* Atualize  $\mathcal{M}'$  */
24 |   else
25 |      $k \leftarrow$  índice do menor desvio in  $S_i$ 
26 |     if ( $r_j > r_k$ ) e ( $q_j + r_j + \mathcal{D}_i + \delta_i - r_k \leq Q_i$ ) then
27 |        $S_i \leftarrow S_i \cup \{j\} \setminus \{k\}$  /* Remove  $k$  e adicione  $j$  in  $S_i$  */
28 |        $vc_j^0 \leftarrow i$  /* Atribua o cliente  $j$  à mediana  $i$  */
29 |        $\delta_i \leftarrow \delta_i + r_j - r_k$ 
30 |        $\mathcal{D}_i \leftarrow \mathcal{D}_i + q_j$  /* Insere  $q_j$  no conjunto  $\mathcal{D}_i$  */
31 |        $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \{j\}$  /* Atualiza  $\mathcal{M}'$  */
32 |     else
33 |       if ( $q_j + \mathcal{D}_i + \delta_i \leq Q_i$ ) then
34 |          $vc_j^0 \leftarrow i$  /* Aloque o cliente  $j$  à mediana  $i$  */
35 |          $\mathcal{D}_i \leftarrow \mathcal{D}_i + q_j$  /* Insere  $q_j$  no conjunto  $\mathcal{D}_i$  */
36 |          $\mathcal{M}' \leftarrow \mathcal{M}' \setminus \{j\}$  /* Atualiza  $\mathcal{M}'$  */
37 |       else
38 |         if  $Trials < MaxTrials$  then
39 |            $Trials \leftarrow Trials + 1$ 
40 |           Ir para a linha 17
41 |         else
42 |           Reinicie a construção, retornando à linha 3.
43 |         end
44 |       end
45 |     end
46 |   end
47 end
48 if Solução viável then
49 |   Retorne  $s^0$ 
50 else
51 end

```

Neste algoritmo, a inicialização é realizada nas linhas 3-8. O algoritmo, então, inicia em um *loop* formado pelas linhas 9-13 para definir as medianas entre o conjunto de candidatos. Inicialmente (linha 11), uma candidata a mediana na *LRC* é escolhida aleatoriamente como mediana e inserida na l -ésima posição do vetor de medianas vm^0 . Uma vez que as medianas p tenham sido escolhidas entre as instalações candidatas, um novo *loop* é iniciado entre as linhas 14-47 para atribuir cada cliente

a uma mediana. Nesse *loop*, um cliente ainda não atribuído a nenhuma mediana é selecionado aleatoriamente, bem como uma mediana. Caso a mediana selecionada não tenha capacidade para atender este cliente, são feitas mais tentativas, selecionando aleatoriamente uma nova mediana, mas considerando um limite de tentativas definido pelo parâmetro *MaxTrials*. Se essas tentativas não forem bem sucedidas, o algoritmo retorna para a linha 3 e novas medianas são então selecionadas.

Para avaliar a viabilidade de atendimento de um cliente por uma mediana, o algoritmo mantém, para cada mediana i , um conjunto denotado por S_i , que contem os índices dos Γ clientes correspondentes aos desvios máximos possíveis do valor central de demanda, considerando os clientes alocados a esta mediana. Inicialmente, este conjunto é vazio. Caso uma mediana tenha até Γ clientes alocados a ela, então S_i conterá todos estes clientes. Isto significa que a mediana pode atender a demanda máxima destes clientes, em que a demanda máxima de um cliente j é igual a sua demanda central (q_j) mais o desvio máximo de demanda (r_j). Caso o conjunto S_i contenha menos de Γ clientes, ao adicionar um novo cliente a esta mediana, ela deverá ser capaz de atender a sua demanda máxima (linha 18). Neste caso, este cliente será adicionado ao conjunto S_i . Caso o conjunto S_i já contenha Γ clientes, ao adicionar um novo cliente à mediana i , se o desvio de demanda deste cliente for maior que o menor desvio dos clientes em S_i , então o novo cliente irá substituir o cliente k correspondente ao menor desvio. Neste caso, a mediana i deve ser capaz de atender à demanda máxima ($q_j + r_j$) do novo cliente j (linha 26), considerando que ela não precisa mais atender o desvio de demanda do cliente k (mas apenas a sua demanda central). Por fim, caso o conjunto S_i já contenha Γ clientes e o desvio de demanda do novo cliente for menor que o menor desvio dos clientes em S_i , então a mediana i deve ser capaz apenas de atender à demanda central do novo cliente (linha 33). Desta forma, ao avaliar a viabilidade da capacidade de uma mediana na inserção de um novo cliente, é verificado se aquela mediana é capaz de atender a demanda de todos os clientes alocados a esta mediana juntamente com a demanda do novo cliente e atender as Γ maiores variações da demanda para garantir a robustez da solução.

5.3 Estruturas de Vizinhança

Para explorar o espaço de solução do problema, foram utilizados três tipos de movimentos para serem aplicados a uma solução s . Cada movimento dá origem a uma estrutura de vizinhança \mathcal{L}_k , $k = 1, 2, 3$, tal que $\mathcal{L}_k(s)$ se refere ao conjunto de soluções na k -ésima vizinhança de s . Essas vizinhanças são definidas como:

- (i) $\mathcal{L}_1(s)$ (Realocação): conjunto de soluções obtidas ao reatribuir um cliente de

uma mediana i_1 para outra mediana i_2 ;

- (ii) $\mathcal{L}_2(s)$ (Troca): consiste em trocar a atribuição do cliente j_1 de uma mediana i_1 por um cliente j_2 de outra mediana i_2 ;
- (iii) $\mathcal{L}_3(s)$ (Substituição): Consiste em substituir uma mediana $i_1 \in \mathcal{N}_p$ por uma instalação candidata $i_2 \in \mathcal{N} \setminus \mathcal{N}_p$.

5.4 Procedimento de busca local

Uma heurística de busca local é um procedimento de refinamento de uma solução s que utiliza uma ou mais estruturas de vizinhança para encontrar um ótimo local referente às estruturas de vizinhança utilizadas na busca. Nesta tese, foi utilizado o procedimento de busca local *Variable Neighborhood Descent* (VND) (Hansen et al., 2017; Mjirda et al., 2017), apresentado no Algoritmo 5. Este método consiste em realizar trocas sistemáticas nas estruturas de vizinhança, seguindo uma ordem pré-definida de vizinhanças. Em seguida, o método utiliza a função de avaliação $f(s)$, definida pela Expressão (5.1), para aceitar ou não a nova solução gerada.

No método VND implementado, as estruturas de vizinhança foram ordenadas conforme mostrado em Subseção 5.3. Essa ordenação segue o grau de complexidade dessas estruturas de vizinhança, em que a primeira estrutura de vizinhança (\mathcal{L}_1) é a menos complexa e a última (\mathcal{L}_3) é a estrutura de vizinhança mais complexa. Esta estratégia é normalmente adotada em buscas locais baseadas em VND. Foram considerados todos os vizinhos gerados por realocações aleatórias de clientes de uma mediana para outra na solução atual na vizinhança \mathcal{L}_1 . Da mesma forma, na vizinhança \mathcal{L}_2 , foram explorados todos os vizinhos gerados trocando-se clientes de duas medianas na solução atual. Finalmente, foram explorados todos os vizinhos gerados aleatoriamente a partir da vizinhança \mathcal{L}_3 .

5.5 Procedimento de perturbação

Para diversificar a busca local, foi utilizado um procedimento de perturbação, que é apresentado pelo Algoritmo 10. A linha 1 apresenta as entradas do algoritmo, que são a solução atual s , a estrutura de vizinhança usada para realizar a perturbação (neste caso, \mathcal{L}_3), e a número k de *shakes* a serem realizados nesta vizinhança.

Na linha 3, a solução s' , que representará a solução perturbada quando o procedimento terminar, obtém uma cópia da solução atual. Na linha 4, o contador i do número de *shakes* a serem aplicados na solução s' começa com o valor 1. Desde que

Algoritmo 10: Procedimento de perturbação

```

1 Entrada: Solução corrente  $s$ , vizinho  $\mathcal{L}_3$ , número de perturbações  $k$ 
2 Saída: Solução perturbada  $s'$ 
3  $s' \leftarrow s$  /* Cópia da solução corrente */
4  $i \leftarrow 1$  /* Inicialize o contador de perturbações */
5 while ( $i \leq k$ ) do
6   Seleccione aleatoriamente uma solução vizinha  $s'' \in \mathcal{L}_3(s')$ 
7   Destruição de  $s'$  com base em  $s''$  /* Destrua a solução  $s'$  */
8   Reconstruir  $s'$  /* Reconstrói a nova solução  $s'$  */
9    $i \leftarrow i + 1$  /* Atualiza o contador de perturbações */
10 end
11 Retorne  $s'$ 

```

este contador seja menor ou igual ao número k de *shakes* a serem aplicados, o procedimento seleciona uma solução vizinha aleatória s'' na vizinhança da solução s' (veja linha 6). Então, a solução s' é destruída e imediatamente reconstruída com a nova instalação selecionada (veja as linhas 7 e 8), respectivamente. A destruição consiste em retirar a mediana selecionada para ser retirada e retirar a atribuição de todos os clientes, incluindo os clientes não atribuídos à mediana removida. A reconstrução consiste em adicionar a nova mediana ao conjunto de medianas e reatribuir todos os clientes a uma mediana, semelhante ao procedimento de construção dado pelo Algoritmo 9. Na linha 9, o contador do número de *shakes* é atualizado, e o procedimento finaliza, retornando a solução perturbada na linha 11.

5.6 Algoritmo GVNS Robusto

O Algoritmo 8 apresenta o pseudocódigo do algoritmo GVNS robusto implementado. Os parâmetros de entrada são o conjunto de estruturas de vizinhança \mathcal{L} , o número máximo permitido de iterações sem melhoria *iterMaxSM*, e os parâmetros do procedimento construtivo α e *MaxTrials*. A solução inicial s_0 (linha 1) é gerada através da fase de construção do método GRASP, conforme mostrado pelo Algoritmo 9 na Seção 5.3.

Após gerar a solução inicial, o algoritmo aplica uma busca local (linha 4) utilizando o procedimento VND, conforme Algoritmo 5. Em seguida, o algoritmo inicializa o contador do número de iterações sem melhorar a solução atual e o contador k do número de *shakes*. O procedimento de perturbação diversifica a busca para outras regiões do espaço de solução. O contador k é inicializado com o valor $k = 2$ porque a vizinhança utilizada pela perturbação é uma das vizinhanças utilizadas para aplicar a busca local.

Então, tendo como critério de parada o número máximo de iterações sem melhora (*iterMaxSM*), o algoritmo entra em um *loop* iterativo com sucessivas aplicações dos procedimentos de perturbação da solução atual e busca local da solução perturbada.

A perturbação (linha 8) é aplicada sobre a solução atual usando o Algoritmo 10, e o procedimento de busca local é executado na solução perturbada (linha 9). A solução s'' gerada pelo procedimento de busca local será aceita se for melhor que a solução atual (linha 10-13). Caso ocorra essa aceitação, a solução atual é atualizada, a estrutura de vizinhança retorna à estrutura inicial ($k = 2$), e o procedimento continua. Caso contrário, a estrutura de vizinhança é incrementada (linha 14-17). A melhor solução encontrada é retornada ao final do processo de busca.

5.7 Descrição dos Experimentos computacionais

Esta seção descreve os experimentos computacionais realizados para o estudo da solução do Problema de p -Medianas Robusto Capacitado (PPMRC). Os experimentos consistem na análise estatística do desempenho do algoritmo implementado com base na metaheurística *General Variable Neighborhood Search* (GVNS) e no *solver* CPLEX, versão 12.9.0.0. O algoritmo GVNS foi implementado na linguagem Java, e o modelo resolvido pelo *solver* CPLEX foi implementado na linguagem OPL. Ambos foram testados em um computador com processador Intel Core i7-3770 CPU 3,40 GHz, 32 GB de RAM, utilizando o sistema operacional Ubuntu 20.04 LTS. Em ambos os experimentos, o tempo limite de execução foi fixado em 86.400 segundos, ou seja, 24 horas.

Para os testes, foram utilizados seis conjuntos de instâncias introduzidos por [Stefanello et al. \(2015\)](#), totalizando 30 instâncias, variando de 318 nós e 5 medianas a 4461 nós e 1000 medianas, além do conjunto de instâncias introduzido por [Lorena e Senne \(2004\)](#), com 5 instâncias com 3038 nós e um número de medianas variando de 600 a 1000. Os resultados para esses conjuntos de dados são mostrados nas Subseções 5.8-5.9.

Para comparar a qualidade das soluções produzidas pelo algoritmo A na instância i , usamos o valor de desvio (gap), denotado por gap_i^A , entre o valor objetivo f_i^A para a melhor solução encontrada pelo método A e o limite inferior fornecido pelo otimizador CPLEX nessa instância, representado por LB_i . O gap é calculada como:

$$gap_i^A = \frac{f_i^A - LB_i}{LB_i} \times 100. \quad (5.2)$$

Portanto, quanto mais próximo o valor de gap_i^A estiver de 0, melhor será o resultado do algoritmo A .

O parâmetro *MaxTentativas*, utilizado para construir a solução viável inicial, foi fixado em p . Os valores dos demais parâmetros do algoritmo foram ajustados pela ferramenta IRACE (*Iterated Racing for Automatic Algorithm Configuration*) ([López-](#)

(Ibáñez et al., 2016). Para a fase de treinamento desta ferramenta, foram utilizadas as instâncias lin318_040, ali535_050, u724_010, rl1304_300, pr2392_020 e fnl_0100, escolhidas aleatoriamente. A Tabela 5.1 mostra, na primeira coluna, os dois parâmetros que foram ajustados, a descrição do parâmetro na segunda coluna, o intervalo de valores fornecido para os testes na terceira coluna e o valor para cada parâmetro retornado pelo IRACE na última coluna.

Tabela 5.1: Resultados do ajuste de parâmetros com IRACE.

Parâmetro	Descrição	Faixa de valores	Valor retornado
$iterMaxSM$	Número máximo de iterações sem melhoria	{300, 400, 500}	500
α	Parâmetro da heurística construtiva baseada em GRASP	{0,4; 0,5; 0,6}	0,4

O restante deste Capítulo está organizado da seguinte forma. A Seção 5.8 apresenta os resultados dos experimentos computacionais para instâncias do conjunto de instâncias lin318, que é o conjunto de dados de Stefanello et al. (2015) com o menor número de nós. Para facilidade de notação, este conjunto de instâncias será nomeado como Grupo de Instâncias Stefanelo1. A Seção 5.9 apresenta os resultados para os demais conjuntos de instâncias de Stefanello et al. (2015), ou seja, ali535, u724, rl1304, pr2392 e fnl4461. Este conjunto de instâncias será nomeado como Grupo de Instâncias Stefanelo2. Por fim, a Seção 5.10 mostra os resultados alcançados para o conjunto de dados proposto por Lorena e Senne (2004), que será nomeado como Grupo de Instâncias Lorena.

5.8 Experimentos computacionais para o Grupo de Instâncias Stefanelo1

A Tabela 5.2 apresenta os resultados dos experimentos computacionais realizados usando o *solver* CPLEX e o Algoritmo GVNS robusto para resolver as instâncias do Grupo de Instâncias Stefanelo1, dado pelo conjunto de dados lin318. Esta tabela e todas as demais tabelas apresentadas nas seções seguintes estão organizadas da seguinte forma. A primeira coluna mostra o nome do conjunto de dados da instância. A segunda coluna mostra o número de medianas (p), enquanto a terceira e quarta colunas mostram os valores do parâmetro de robustez Γ e o parâmetro que representa o desvio percentual de demanda Δr , respectivamente. As colunas LB e UB apresentam, respectivamente, o limite inferior e o limite superior fornecidos pelo *solver* CPLEX. A coluna “tempo(s)” apresenta o tempo de execução em segundos, enquanto a coluna *gap* apresenta o *gap* entre a melhor solução encontrada e o LB , conforme a Expressão (5.2). A coluna “F.O.” apresenta os valores para a função objetivo da melhor solução fornecida pelo Algoritmo GVNS robusto.

Tabela 5.2: Comparação do *solver* CPLEX e do Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1

Instância		Incerteza		CPLEX				GVNS robusto		
Nome	p	Γ	Δr	LB	UB	tempo(s)	gap (%)	F.O.	tempo(s)	gap (%)
lin318	005	0	0	180281,20	180281,20	1475	0,00	180281,20	358	0,00
		01	5	180282,37	180282,37	6143	0,00	180359,47	479	0,04
			10	180282,37	180282,37	7765	0,00	180422,52	469	0,08
			15	180383,08	180383,08	10354	0,00	180422,52	460	0,02
		06	5	180400,58	180400,58	8524	0,00	180950,80	461	0,30
			10	180535,57	180535,57	7514	0,00	180954,45	508	0,22
			15	180539,21	180539,21	7982	0,00	181008,61	415	0,25
		16	5	180768,35	180768,35	9845	0,00	181346,80	477	0,31
			10	180768,35	181313,54	8742	0,30*	181402,84	461	0,35
			15	180768,35	181489,78	11840	0,23*	181420,92	422	0,36
		26	5	181069,19	181069,19	8158	0,00	182246,13	428	0,65
			10	181070,20	183291,86	9094	1,22*	182301,47	437	0,68
	15		181077,20	182658,34	10325	0,87*	182362,84	453	0,70	
	36	5	181082,19	181082,19	8265	0,00	182331,65	463	0,69	
		10	181083,19	183126,05	9184	1,12*	182441,31	447	0,75	
		15	181083,19	184193,04	9201	1,60*	182441,31	429	0,75	
	46	5	181278,27	184193,04	11346	1,60*	182819,13	472	0,85	
		10	181287,35	188050,31	9412	3,74*	182828,29	451	0,85	
		15	181287,35	188807,58	9354	4,14*	182828,29	443	0,85	
	56	5	180734,01	182055,87	7189	0,73*	183653,22	474	1,25	
		10	182342,71	278001,12	4088	34,43*	185865,61	458	1,93	
		15	-	-	-	-	186667,61	475	-	
	015	0	0	88901,56	88901,56	322,11	0,00	89026,28	574	0,14
		12	5	88815,64	89806,18	4093	1,10*	89635,81	1471	0,92
			10	89226,57	93822,75	6535	4,90*	91089,76	1747	2,09
			15	90120,47	91314,70	11848	1,31*	92563,98	1663	2,71
		22	5	89630,12	89630,12	13084	0,00	90262,54	1951	0,71
			10	89667,07	95265,08	4883	5,88*	91232,37	1768	1,75
	15		90701,37	93554,46	8615	3,05*	92376,86	2115	1,84	
	040	0	0	47982,75	47982,75	3877	0,00	47982,75	853	0,00
		5	5	47849,62	48443,70	13303	1,23*	48208,97	1169	0,75
			10	48241,32	49899,75	10834	3,32*	49115,52	1854	1,81
			15	48652,39	52160,86	9193	6,73*	50178,08	1860	3,14
		8	5	47846,56	48390,03	12223	1,12*	48287,01	1794	0,92
			10	48684,60	50232,73	13042	3,08*	50837,22	2106	4,42
	15		49286,38	55500,77	9738	11,20*	51858,64	1824	5,21	
	070	0	0	32198,63	32198,63	61	0,00	32201,85	527	0,01
		1	5	32686,01	32686,01	3342	0,00	32699,08	1285	0,03
			10	32729,61	32729,61	2977	0,00	32755,79	1715	0,08
			15	32910,62	33162,28	6625	0,76*	33259,47	1974	1,06
		5	5	32927,82	33322,94	17388	1,19*	32954,16	1621	0,08
			10	34114,49	35165,77	8083	2,99*	34636,44	1865	1,52
	15		35436,48	37697,18	8816	6,00*	36765,35	1749	3,75	
	100	0	0	22942,69	22942,69	371	0,00	22971,72	752	0,12
		1	5	23321,38	23525,61	5234	0,87*	23492,57	1585	0,73
			10	24197,77	24558,77	11742	1,47*	24446,65	1963	1,02
			15	24878,86	25108,66	5420	0,92*	25192,21	1982	1,25
		5	5	24082,52	26374,20	14222	8,69*	24385,25	1622	1,25
10			25796,56	26288,20	5652	3,45*	26097,98	1657	1,16	
15	27990,61		29996,58	7624	6,69*	28716,74	1521	2,59		

(1) O símbolo “*” significa *Sem Memória*.

(2) O símbolo “-” significa que nenhum resultado foi encontrado.

Uma vez que as instâncias do conjunto de dados lin318 são aquelas com o menor número de nós usados neste estudo e, portanto, poderiam representar menor desafio para o *solver* CPLEX que as instâncias de maior dimensão, um conjunto intensivo de experimentos foi realizado, considerando diferentes valores para o parâmetro robusto Γ e para o parâmetro de maior desvio de demanda permitido Δr .

Assim, conforme mostra a Tabela 5.2, o tempo de execução necessário para re-

Tabela 5.3: Teste de normalidade de Shapiro-Wilk para a diferença entre os *gaps* encontradas pelo *solver* CPLEX e o Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1

Instâncias	<i>p</i> -valor
lin318	7,24e-10

solver as instâncias do problema sem robustez (valor nominal da instância, ou seja, $\Gamma = 0$) foi menor que aquele necessário para resolver todas as instâncias com $\Gamma > 0$ para todos os valores de p . Para $\Gamma = 0$, o *solver* conseguiu encontrar a solução ótima para todas as instâncias, enquanto o Algoritmo GVNS robusto encontrou a solução ótima para a maioria das instâncias e, quando não conseguiu encontrar uma solução ótima, o *gap* associado à melhor solução foi inferior a 0,14%. Em 17 das 50 instâncias avaliadas nesse Grupo de Instâncias Stefanelo1, o *solver* não encontrou a solução ótima. Observe que, para todas as instâncias em que o *solver* comprova a otimalidade, o *gap* do Algoritmo GVNS robusto não excedeu 0,71%. Além disso, a qualidade da melhor solução fornecida pelo *solver* CPLEX para a maioria das instâncias foi pior que a da melhor solução encontrada pelo Algoritmo GVNS robusto. Observe também que o CPLEX não conseguiu encontrar uma solução viável para a instância na qual $p = 5$, $\Gamma = 56$ e $\Delta r = 15$.

Para realizar a análise estatística dos *gaps* referentes à melhor solução viável encontrada pelo *solver* CPLEX e pelo algoritmo GVNS robusto, inicialmente foi realizado o Teste de normalidade Shapiro-Wilk (Montgomery e Runge, 2016). A Tabela 5.3 mostra o *p*-valor para o teste de normalidade para a diferença entre o *gap* do *solver* CPLEX e o *gap* do Algoritmo GVNS robusto. Como esses dados não apresentam normalidade estatística para um nível de significância estatística de 0,05, isto é, o *p*-valor foi menor que 0,05, foi utilizado o Teste pareado de Wilcoxon-Mann-Whitney (Montgomery e Runge, 2016) para verificar se há diferença estatisticamente significativa entre os *gaps* alcançados por esses algoritmos. Considerando um nível de significância estatística de 0,05, os resultados da Tabela 5.4 mostram que não há diferença significativa entre os *gaps* referentes à melhor solução encontrada pelo *solver* e os *gaps* referentes à melhor solução encontrada pelo Algoritmo GVNS robusto. Esta comprovação se dá pelo fato de o *p*-valor ser menor que 0,05.

A Figura 5.2 apresenta um gráfico *boxplot* de valores de *gap* para cada procedimento. De acordo com esta figura e com a Tabela 5.2, os valores dos *gaps* relativos do Algoritmo GVNS robusto são menores que aqueles referentes ao *solver* CPLEX, principalmente à medida que o parâmetro de robustez Γ aumenta, ou seja, que aumenta o número de clientes para o qual se considera incerteza no valor da demanda q_j , nos termos da representação de incertezas posta no conjunto (2.15).

Tabela 5.4: Teste pareado de Wilcoxon-Mann-Whitney para a diferença entre os *gaps* encontradas pelo *solver* CPLEX e o Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1

instâncias	p -valor
lin318	0,01583

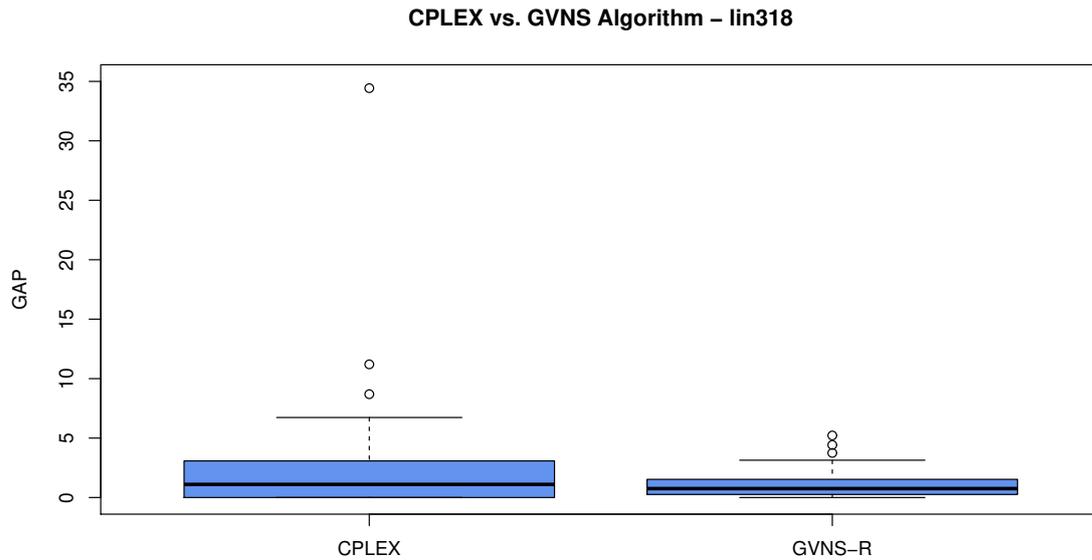


Figura 5.2: Gráfico de *boxplot* para os *gaps* encontradas pelo *solver* CPLEX e pelo Algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1.

Além de comparar os valores de *gaps*, também foram analisados o tempo de execução associado ao *solver* CPLEX e o tempo de execução do Algoritmo GVNS robusto para resolver o Grupo de Instâncias Stefanelo1. A Tabela 5.5 mostra o p -valor para o Teste de normalidade referente à diferença entre os tempos de execução encontrados para o *solver* CPLEX e o Algoritmo GVNS robusto. De acordo com os resultados, não podemos rejeitar a hipótese de normalidade para um nível de significância estatística de 0,05. Em seguida, aplicamos o teste pareado de Wilcoxon-Mann-Whitney, um teste não paramétrico, e o Teste t pareado, um teste paramétrico, para verificar se há diferença estatisticamente significativa entre os tempos de execução de ambos os procedimentos. De acordo com os p -valores apresentados na Tabela 5.6, para um nível de significância estatística de 0,05, há uma diferença significativa entre esses tempos de execução.

A Figura 5.3 mostra um gráfico *boxplot* com os tempos de execução para cada abordagem, e, a partir dela e da análise estatística realizada, conclui-se que o tempo de execução do Algoritmo GVNS robusto é, em geral, menor quando comparado com o tempo de execução demandado pelo *solver* CPLEX.

Tabela 5.5: Teste de normalidade Shapiro-Wilk para a diferença entre os tempos de execução encontrados pelo *solver* CPLEX e o algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1

Instâncias	p -valor
lin318	0,3484

Tabela 5.6: Teste pareado de Wilcoxon-Mann-Whitney para a diferença entre os tempos de execução encontrados pelo *solver* CPLEX e o algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo1

Teste	p -valor
Wilcoxon-Mann-Whitney	4,97E-11
t-teste	2,2E-16

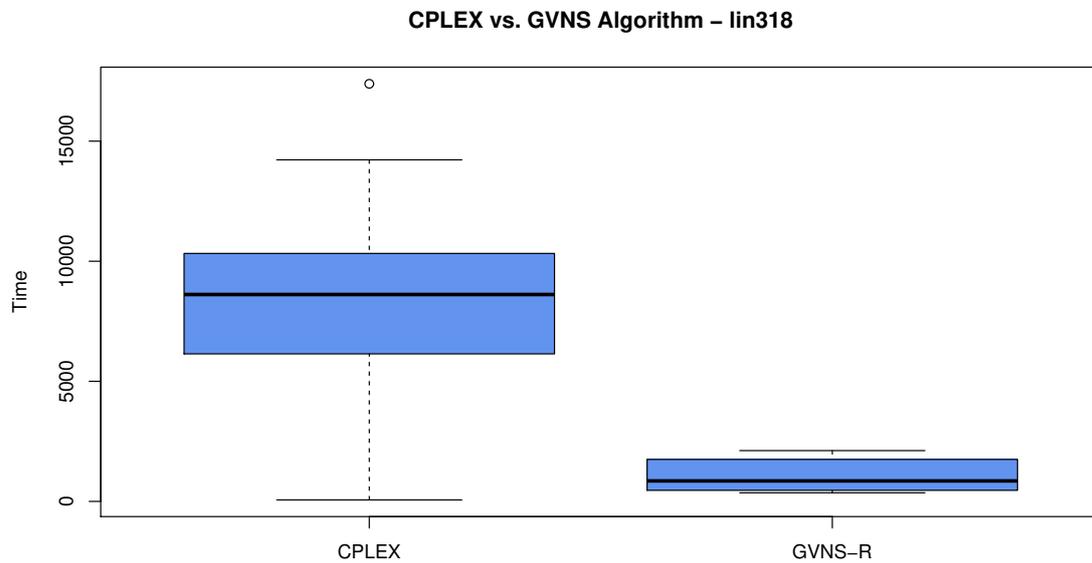


Figura 5.3: Gráfico de *boxplot* para os tempos de execução encontrados pelo *solver* CPLEX e pelo GVNS robusto para o Grupo de Instâncias Stefanelo1.

5.9 Experimentos computacionais para o Grupo de Instâncias Stefanelo2

A Tabela 5.7 apresenta os resultados dos experimentos computacionais para o Grupo de Instâncias Stefanelo2, formado pelos conjuntos de dados ali535, u724, rl1304, pr2392 e fnl4461, propostos por Stefanello et al. (2015). Nos experimentos computacionais realizados envolvendo esse Grupo de Instâncias, consideramos as situações em que $\Gamma = 0$ (valor nominal da instância) e a situação em que $\Gamma = n/p$. Os valores de desvio percentual de demanda foram definidos como 10%, ou seja, $\Delta r = 0,1$.

De acordo com a Tabela 5.7, dentre as 50 instâncias do Grupo de Instâncias Ste-

fanelo2, o *solver* CPLEX retornou “Falta memória” ou não encontrou uma solução viável em 24 instâncias analisadas, o que mostra a dificuldade de solução computacional do PPMRC. Nenhuma solução viável foi encontrada para a instância do conjunto de dados ali535 com $p = 150$, $\Gamma = 4$ e $\Delta r = 10$. Para o conjunto de dados u724, para $p = 125$, $\Gamma = 6$, $\Delta r = 10$ e para $p = 200$, $\Gamma = 4$ e $\Delta r = 10$, o *solver* não encontrou uma solução. Para o conjunto de dados pr2392, pode-se observar que o *solver* CPLEX fornece resultados apenas para $p = 20$, $\Gamma = 0$, $\Delta r = 0$ e para $p = 75$, $\Gamma = 0$, $\Delta r = 0$, ambos com $gap = 100\%$. Para o conjunto de dados pr4461, observa-se que o *solver* não conseguiu encontrar soluções viáveis para nenhuma instância, enquanto o algoritmo GVNS implementado não encontrou uma solução viável apenas para a instância com $p = 250$, $\Gamma = 18$ e $\Delta r = 10$. Assim, em especial, o *solver* CPLEX não encontrou solução viável para os conjuntos de dados de maior dimensão.

Assim, das 50 instâncias avaliadas, envolvendo 25 instâncias nominais e dois valores de Γ para cada uma), o CPLEX apresentou resultados em apenas 24 delas (representando 50% do total de instâncias), dos quais apenas 6 foram efetivamente resolvidas (ou 12% do total). Por outro lado, o Algoritmo GVNS robusto apresentou resultados e efetivamente resolveu 49 instâncias (98% do total) e não apresentou solução em apenas duas instâncias (4% do total), qual seja, a instância ali535, para $p = 150$, $\Gamma = 4$ e $\Delta r = 10\%$, e a instância fnl4461, $p = 150$, para $\Gamma = 18$ e $\Delta r = 10\%$. Portanto, o Algoritmo GVNS robusto superou o *solver* CPLEX por uma grande margem quando aplicado para resolver instâncias de médio e grande porte do PPMRC, adaptando-se totalmente às dificuldades de solução impostas ao considerar incertezas com desvio médio do caso nominal.

Para a análise estatística destes resultados, como há valores para o *solver* CPLEX em apenas algumas instâncias, então um teste não paramétrico, ou seja, o Teste pareado de Wilcoxon-Mann-Whitney (Montgomery e Runge, 2016) foi usado para verificar se há uma diferença significativa nos *gaps* fornecidas pelo *solver* CPLEX e pelo Algoritmo GVNS robusto. Os p -valores para cada conjunto de dados são mostrados na Tabela 5.8. Para um nível de significância de 0,05, os resultados mostram que não há diferença significativa entre os *gaps*. Os *gaps* também podem ser analisados pelos gráficos *boxplot* mostrados nas Figuras 5.4-5.5, que confirmam que não há diferença significativa entre os *gaps* obtidos por ambos os procedimentos.

Uma comparação estatística a respeito dos tempos de execução encontrados para o *solver* CPLEX e o Algoritmo GVNS robusto ao resolver os conjuntos de dados ali535 e u724 também é realizada. Uma vez que o *solver* CPLEX fornece poucos resultados ao resolver esses conjuntos de dados, utilizamos o Teste pareado de Wilcoxon-Mann-Whitney (Montgomery e Runge, 2016) para verificar se há uma diferença significativa nos tempos de execução obtidos com o *solver* CPLEX e com o Algoritmo GVNS

Tabela 5.7: Comparação do *solver* CPLEX e do algoritmo GVNS robusto para o Grupo de Instâncias Stefanelo2

Instância		Incerteza		CPLEX				GVNS robusto		
Nome	p	Γ	Δr	LB	UB	tempo(s)	gap (%)	F.O.	tempo(s)	gap (%)
ali535	005	0	0	9899,61	9899,61	13498	0,00	9933,43	1824	0,34
		107	10	10066,47	10207,02	86400	1,38	10077,45	1873	0,11
	025	0	0	3671,63	3673,34	26980	0,05*	3688,51	2100	0,46
		22	10	3731,12	3791,44	67736	1,59*	3750,63	1877	0,52
	050	0	0	2446,72	2431,86	10560	0,61*	2458,70	1970	0,49
		11	10	2487,81	2799,13	21308	11,12*	2573,13	1978	3,43
	100	0	0	1414,18	1435,07	12855	1,46*	1433,60	1903	1,37
		6	10	1490,80	1906,17	12775	21,79*	1558,92	1853	4,57
	150	0	0	1013,49	1023,06	14111	0,94*	1025,37	2070	1,17
		4	10	–	–	–	–	–	–	–
u724	010	0	0	181766,97	181787,59	47079	0,00	181823,17	857	0,03
		73	10	180133,66	223238,40	68698	19,24*	188394,35	2924	4,58
	030	0	0	94973,06	95053,86	31486	0,09*	95222,02	1525	0,26
		25	10	94132,24	106371,27	14408	11,51*	99892,82	3154	6,11
	075	0	0	54748,37	54748,37	159	0,00	54913,01	925	0,30
		10	10	55727,39	56532,55	20344	21,66	59503,95	2758	6,77
	125	0	0	38936,76	38980,58	8542	0,11*	39196,73	1124	0,66
		6	10	–	–	–	–	43445,41	4524	–
	200	0	0	28081,93	28081,93	4724	0,00	28340,10	2114	0,91
		4	10	–	–	–	–	32919,23	5254	–
rl1304	010	0	0	2216000,10	2216000,10	86400	0,00	2221470,50	5385	0,25
		130	10	–	–	–	–	2263824,79	4190	–
	050	0	0	801455,18	803286,74	57501	0,23*	803150,33	3715	0,21
		27	10	–	–	–	–	815242,45	4737	–
	100	0	0	496421,85	498530,04	17385	0,42*	497796,79	3198	0,28
		13	10	–	–	–	–	507106,86	5095	–
200	0	0	275787,25	277172,36	8106	0,50*	276310,23	4179	0,19	
	7	10	–	–	–	–	279819,07	4173	–	
300	0	0	190764,02	191233,72	10837	0,25*	191544,95	5411	0,41	
	5	10	–	–	–	–	195266,29	3231	–	
pr2392	020	0	0	0,00	4464282,96	432	100*	2332221,94	3796	–
		120	10	–	–	–	–	2421727,01	4514	–
	075	0	0	0,00	4309402,14	459	100*	1132362,51	3981	–
		32	10	–	–	–	–	1161815,85	4789	–
	150	0	0	–	–	–	–	733273,63	4438	–
		16	10	–	–	–	–	769083,32	4602	–
300	0	0	–	–	–	–	475276,82	3635	–	
	8	10	–	–	–	–	496853,71	4270	–	
500	0	0	–	–	–	–	321174,72	4717	–	
	5	10	–	–	–	–	329358,08	4922	–	
fml4461	0020	0	0	–	–	–	–	1311798,15	8522	–
		224	10	–	–	–	–	1396572,88	8769	–
	0100	0	0	–	–	–	–	560030,68	9115	–
		45	10	–	–	–	–	599220,93	9258	–
	0250	0	0	–	–	–	–	339654,89	9411	–
		18	10	–	–	–	–	–	–	–
0500	0	0	–	–	–	–	232173,69	9781	–	
	9	10	–	–	–	–	248307,90	10241	–	
1000	0	0	–	–	–	–	150366,44	9251	–	
	5	10	–	–	–	–	162774,76	10579	–	

(1) O símbolo “*” significa *Sem Memória*.

(2) O símbolo “–” significa que nenhum resultado foi encontrado.

robusto. A Tabela 5.9 apresenta os respectivos p -valores. A partir dos resultados da Tabela 5.9 e dos gráficos *boxplot* das Figuras 5.6-5.7, concluímos que existe uma diferença estatística entre os tempos de execução desses procedimentos, p -valor menor que 0,05 e, conseqüentemente, o tempo de execução do Algoritmo GVNS robusto é menor que o tempo de execução do *solver* CPLEX.

Tabela 5.8: Valores p do Teste de Wilcoxon-Mann-Whitney pareado para os *gaps* fornecidas pelo *solver* CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias ali535 e u724.

Instância	p -valor
ali535	0,1484
u724	0,7422

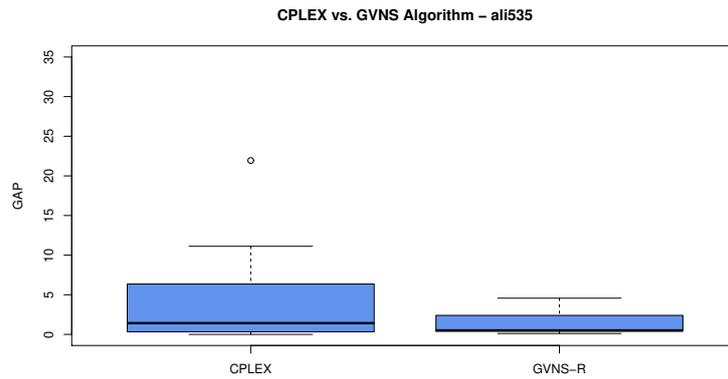


Figura 5.4: Gráfico de *boxplot* para os valores de *gap* fornecidos pelo *solver* CPLEX e pelo Algoritmo GVNS robusto ao resolver as instâncias ali535.

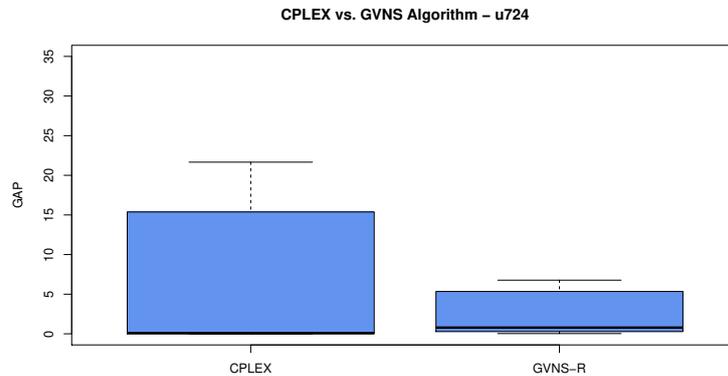


Figura 5.5: Gráfico de *boxplot* para os valores de *gap* fornecidos pelo *solver* CPLEX e pelo Algoritmo GVNS robusto ao resolver as instâncias u724.

Tabela 5.9: p -valor do Teste de Wilcoxon-Mann-Whitney pareado para os tempos de execução fornecidos pelo *solver* CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias ali535 e u724.

Instância	p -valor
ali535	0,0003
u724	0,0051

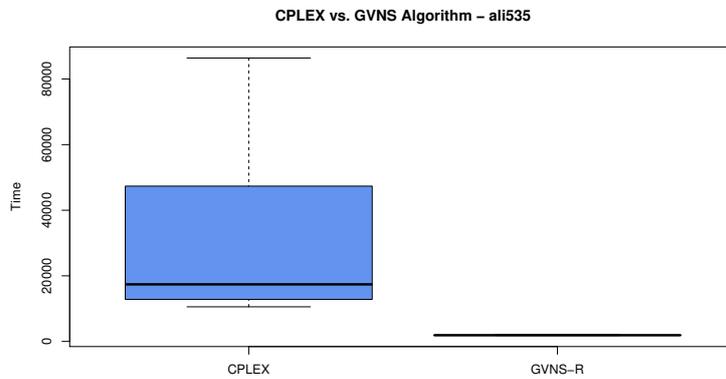


Figura 5.6: Gráfico de *boxplot* para os tempos de execução fornecidos pelo *solver* CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias ali535

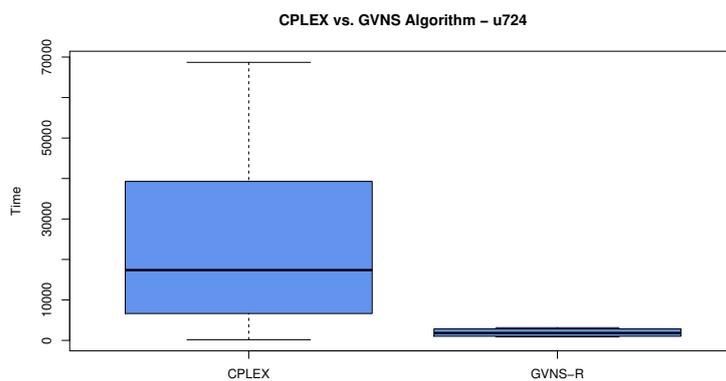


Figura 5.7: Gráfico de *boxplot* para os tempos de execução fornecidos pelo *solver* CPLEX e o Algoritmo GVNS robusto ao resolver as instâncias u724.

5.10 Experimentos computacionais usando o Grupo de Instâncias Lorena

A Tabela 5.10 apresenta os resultados encontrados ao aplicar o Algoritmo GVNS robusto para resolver o Grupo de Instâncias Lorena, formado pelo conjunto de dados p3038, proposto por Lorena e Senne (2004). Este algoritmo encontrou soluções para todas as instâncias do Grupo. Por outro lado, o *solver* CPLEX não conseguiu determinar soluções para nenhuma das instâncias desse Grupo.

Tabela 5.10: Comparação das soluções do *solver* CPLEX e do Algoritmo GVNS robusto para o Grupo de Instâncias Lorena

Instância		Incerteza		CPLEX				GVNS robusto		
Nome	p	Γ	Δr	LB	UB	tempo(s)	gap (%)	F.O.	tempo(s)	gap (%)
p3038	600	0	0	–	–	–	–	124822,19	4547	
		6	10	–	–	–	–	135554,65	6289	
	700	0	0	–	–	–	–	110908,85	4316	
		5	10	–	–	–	–	126754,58	6414	
	800	0	0	–	–	–	–	100548,07	5283	
		4	10	–	–	–	–	106808,35	7248	
	900	0	0	–	–	–	–	91944,43	5384	
		4	10	–	–	–	–	97154,85	7632	
	1000	0	0	–	–	–	–	84845,35	5688	
		4	10	–	–	–	–	90515,18	8479	

⁽¹⁾ O símbolo “–” significa que nenhum resultado foi encontrado.

É interessante observar que, para este Grupo de Instâncias, a inserção de incertezas na demanda de clientes implica no aumento dos valores de função objetivo e nos valores de tempo de execução computacional em relação à situação nominal de cada instância avaliada.

5.11 Considerações Parciais

Para resolver o PPMRC, esta tese propôs um algoritmo híbrido baseado na metaheurística *General Variable Neighborhood Search* (GVNS) envolvendo um método para gerar uma solução inicial através de uma adaptação da fase de construção da metaheurística *Greedy Randomized Adaptive Procedure* (GRASP) (Algoritmo 9 da Subseção 5.2) e um algoritmo de busca local baseado no procedimento *Variable Neighborhood Descent* (Algoritmo 5). Para fins de comparação, esta tese também resolveu o modelo matemático do problema abordado por meio do *solver* de uso geral CPLEX.

Para a descrição completa do PPMRC, este Capítulo relatou uma série de experimentos computacionais em sete conjuntos de instâncias da literatura propostas para o PPMC, com um número elevado de nós e medianas (grupos de 35 instâncias variando de 318 nós e 5 medianas a 4461 nós e 1000 medianas). O *solver* do CPLEX não resolveu várias instâncias desse grupo de forma otimizada.

O Algoritmo GVNS robusto resolveu todas as instâncias para o caso nominal e permitiu variações nas demandas dos clientes para quase todas elas. A partir da análise estatística dos resultados, confirmou-se a superioridade da variante GVNS sobre o *solver* em termos de valores de *gap*. Com relação ao tempo de execução computacional, mostra-se que o Algoritmo GVNS robusto também apresenta melhor desempenho que o *solver* CPLEX. Portanto, o Algoritmo GVNS robusto mostrou-se um procedimento competitivo para a solução do PPMRC.

Capítulo 6

Conclusões Gerais e Trabalhos Futuros

Esta tese teve seu foco no Problema das p -Medianas Capacitado (PPMC), em sua abordagem determinística, e no Problema das p -Medianas Robusto Capacitado (PPMRC), considerando incertezas nas demandas dos clientes.

Para a solução do primeiro problema foi usada uma formulação clássica de programação linear inteira da literatura. Para o segundo problema, foi proposta uma formulação de programação matemática para tratar o caso robusto. Ambas as formulações foram resolvidas pela aplicação do *solver* CPLEX. Além disso, esta tese propõe um algoritmo heurístico baseado na metaheurística *General Variable Neighborhood Search* – GVNS (Hansen et al., 2017) para tratar instâncias maiores dos dois problemas, dado que os dois problemas objetos de investigação são da classe NP-difícil. No algoritmo GVNS, a solução inicial é feita via a fase de construção do método *Greedy Randomized Adaptive Search Procedure* – GRASP (Feo e Resende, 1995) e a busca local é feita pelo método *Variable Neighborhood Descent* – VND (Hansen e Mladenović, 1997).

Ao se definir o método de otimização adequado para a resolução do Problema de p -Medianas Robusto Capacitado, um fator determinante é a dimensão do conjunto de dados de entrada. Instâncias com pequeno número de pontos de demanda podem ser resolvidos na otimalidade pelo *solver* CPLEX. Entretanto, com o aumento na dimensão do conjunto de dados, tais métodos podem vir a exigir tempos computacionais extremamente elevados. Nestes casos, é apropriado o uso de métodos alternativos, como as metaheurísticas, como as que foram aqui empregadas, pois elas podem gerar soluções viáveis em tempo computacional aceitável para a tomada de decisão.

Como contribuições deste trabalho, pode-se citar a proposição de uma formulação de programação matemática para o PPMRC com incertezas de demanda e de um

algoritmo heurístico eficiente para tratar instâncias de grande porte tanto do PPMRC quanto do PPMC.

Acredita-se que o método heurístico proposto tem potencial para ser útil em problemas reais de localização, especialmente para Problema das p -Medianas Capacitado com incerteza nas demandas dos clientes.

6.1 Trabalhos Futuros

Fica claro, a partir deste trabalho, que uma das maiores dificuldades da abordagem GVNS robusta foi gerar uma solução inicial factível. Ao selecionar um cliente, ele somente será inserido em uma instalação se a sua demanda somada às demandas já atendidas por aquela instalação não exceder a capacidade de atendimento. Com base nisto, uma proposta para trabalhos futuros, é a criação uma solução inicial a partir das seguintes três etapas. Primeiro, a seleção das instalações, em seguida a seleção dos clientes e a atribuição desses clientes às instalações (isso pode ser feito de forma aleatória ou usando algum recurso heurístico). A partir daí, verifica-se a factibilidade da solução e, posteriormente, a realocação dos clientes cuja demanda supere a capacidade de atendimento da instalação.

Uma outra abordagem interessante para o problema das p -Medianas Robusto é considerar as incertezas nos coeficientes da Matriz de distância. Neste sentido, ficam três sugestões de continuidade do trabalho:

- Proposição de formulação de programação linear inteira para o Problema das p -Medianas Robusto com incerteza nos coeficientes da matriz de distância;
- Proposição de formulação de programação linear inteira para o Problema das p -Medianas Robusto com incerteza nos coeficientes da matriz de distância e nas restrições de capacidade;
- Proposição de um algoritmo GVNS para o tratar o Problema das p -Medianas Robusto com incerteza nos coeficientes da matriz de distância e nas restrições de capacidade.

Recomenda-se também testar outras metaheurísticas para tratar os problemas objetos de investigação nesta tese.

6.2 Trabalhos Publicados e Submetidos

Durante a realização deste trabalho, dois artigos foram publicados e um submetido. São eles:

Trabalhos publicados:

- Vasconcelos, A. M.; de Souza, S. R. e Souza, M. J. F. Resolução do problema das p -medianas utilizando algoritmos meméticos. Proceedings of the XXXVIII Iberian Latin-American Congress on Computational Methods in Engineering - CILAMCE, Florianópolis, v. 38, 2017.
- Vasconcelos, A. M.; Souza, M. J. F. e de Souza, S. R. Algoritmo GVNS Híbrido Aplicado ao Problema das p -Medianas Capacitado. Trends in Computational and Applied Mathematics, v. 22, p. 453-473, 2021.

Trabalho submetido:

- Vasconcelos, A. M.; de Sá, E. M.; Souza, M. J. F. e de Souza, S. R. A GVNS-based algorithm applied to the Robust Capacitated p -Median Problem. Applied Soft Computing, 2022.

Bibliografia

Abidi, M. H.; Alkhalefah, H.; Umer, U. e Mohammed, M. K. (2021). Blockchain-based secure information sharing for supply chain management: Optimization assisted data sanitization process. *International Journal of Intelligent Systems*, v. 36, n. 1, p. 260–290. doi: <https://doi.org/10.1002/int.22299>.

Abreu, L. F.; Hörner, D.; do Rosário, R. R. L.; Gonçalves, M. B. e Coelho, A. S. (2008). Localização de unidades de atendimento ao cidadão: Comparação e proposta para cidade de manaus, utilizando o algoritmo de taitz e bart e algoritmo genético. *Anais do XXVIII Encontro Nacional de Engenharia de Produção*, Rio de Janeiro, RJ, Brasil.

Aguado, J. S. e Trandafir, P. C. (2012). Some heuristic methods for solving p -median problems with a coverage constraint. *European Journal of Operational Research*, v. 220, n. 2, p. 320–327. ISSN 0377-2217.

Ahmadi, T.; Karimi, H.; Davoudpour, H. e Hosseinijou, S. A. (2015). A robust decision-making approach for p -hub median location problems based on two-stage stochastic programming and mean-variance theory: a real case study. *Int Journal Adv ManufTechnol*, v. 77, p. 1943–1953.

Ahmed, M. M.; Centeno, E. M. e Üster, H. (2016). A branch-and-price algorithm for solving the hamiltonian p -median problem. *Journal on Computing*, v. 28, p. 674–689. doi: <https://doi.org/10.1287/ijoc.2016.0704>.

An, Yu; Zeng, Bo; Zhang, Yu e Zhao, Long. (2014). Reliable p -median facility location problem: two-stage robust models and algorithms. *Transportation Research Part B: Methodological*, v. 64, p. 54–72. doi: <https://doi.org/10.1016/j.trb.2014.02.005>. URL <https://www.sciencedirect.com/science/article/pii/S0191261514000319>.

Arroyo, J. E. C.; Marques, T. B. e Cortes, J. M. R. (2006). Um algoritmo genético para o problema de alocação de antenas de transmissão. *Anais do XXVI Encontro Nacional de Engenharia de Produção*, Fortaleza, CE.

Avella, P.; Boccia, M.; Salerno, S. e Vasilyev, I. (2012). An aggregation heuristic for large scale p -median problem. *Computers & Operations Research*, v. 39, n. 7, p. 1625–1632. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2011.09.016>. URL <https://www.sciencedirect.com/science/article/pii/S0305054811002772>.

Avella, Pasquale; Sassano, Antonio e Vasil'ev, Igor. (2007). Computational study of large-scale p -median problems. *Mathematical Programming*, v. 109, n. 1, p. 89–114.

- Avgerinos, I.; Mourtos, I. e Zois, G. (2022). Multi-type facility location in printing and parcel delivery services. *Annals of Operations Research*, v. 309, n. 7, p. 365–393. doi: 10.1007/s10479-021-04469-3.
- Baldacci, R; Hadjiconstantinou, E; Maniezzo, V e Mingozzi, A. (2002). A new method for solving capacitated location problems based on a set partitioning approach. *Computers & Operations Research*, v. 29, p. 365–386.
- Baldacci, Roberto; Caserta, Marco; Traversi, Emiliano e Calvo, Roberto Wolfler. (2022). Robustness of solutions to the capacitated facility location problem with uncertain demand. *Optimization Letters*, v. . doi: 10.1007/s11590-021-01848-4.
- Bangun, Putra BJ; Octarina, Sisca e Pertama, Ari Putra. jul(2019). Implementation of branch and cut method on n-sheet model in solving two dimensional cutting stock problem. *Journal of Physics: Conference Series*, v. 1282, n. 1, p. 012012. doi: 10.1088/1742-6596/1282/1/012012.
- Bargos, F. F.; Lamas, W. Q.; Bargos, D. C.; Neto, M. B. e Pardal, P. C. P. M. (2016). Location problem method applied to sugar and ethanol mills location optimization. *Renewable and Sustainable Energy Reviews*, v. 65, p. 274–282.
- Baron, Opher; Milner, Joseph e Naseraldin, Hussein. (2011). Facility location: A robust optimization approach. *Production and Operations Management*, v. 20, n. 5, p. 772–785. doi: 10.3401/poms.1080.01194.
- Baroughi, F.; Soltanpour, A. e Alizadeh, B. (2019). A novel hybrid modified binary particle swarm optimization algorithm for the uncertain p -median location problem. *Iranian Journal of Operations Research*, v. 10, n. 2, p. 78–93.
- Basciftci, B; Ahmed, S e Shen, S. (2021). Distributionally robust facility location problem under decision-dependent stochastic demand. *European Journal of Operational Research*, v. 292, n. 2, p. 548–561. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2020.11.002>. URL <https://www.sciencedirect.com/science/article/pii/S0377221720309449>.
- Bazaraa, Mokhtar S.; Jarvis, John J. e Sherali, Hanif D. (2010). *Linear Programming and Network Flows*. Wiley, 4a. edição.
- Bektaş, Tolga; Gouveia, Luís e Santos, Daniel. (2019). Revisiting the hamiltonian p -median problem: A new formulation on directed graphs and a branch-and-cut algorithm. *European Journal of Operational Research*, v. 276, n. 1, p. 40–64. doi: <https://doi.org/10.1016/j.ejor.2018.12.041>.
- Ben-Tal, A. e Nemirovski, A. (1999). Robust solutions of uncertain linear programs. *Operations Research Letters*, v. 25, n. 1, p. 1–13. ISSN 0167-6377.
- Beraldi, P.; Bruni, M. E. e Conforti, D. (2004). Designing robust emergency medical service via stochastic programming. *European Journal of Operational Research*, v. 158, n. 1, p. 183–193. ISSN 0377-2217. doi: [https://doi.org/10.1016/S0377-2217\(03\)00351-5](https://doi.org/10.1016/S0377-2217(03)00351-5). URL <https://www.sciencedirect.com/science/article/pii/S0377221703003515>.

- Berman, Oded e Drezner, Zvi. (2008). The p -median problem under uncertainty. *Networks and Spatial Economics*, v. 189, p. 19–30. doi: 10.1016/j.ejor.2007.05.045.
- Bertsimas, D. e Sim, M. (2003). Robust discrete optimization and network flows. *Mathematical Programming*, v. 98, n. 1-3, p. 49–71.
- Bertsimas, D e Sim, M. (2004). The price of robustness. *Operations Research*, v. 52, n. 1, p. 35–53.
- Bertsimas, Dimitris; Brown, David B e Caramanis, Constantine. (2011). Theory and applications of robust optimization. *SIAM review*, v. 53, n. 3, p. 464–501.
- Blum, C.; Puchinger, J.; Raidl, G. R. e Roli, A. (2011). Hybrid metaheuristics in combinatorial optimization: A survey. *Applied Soft Computing*, v. 11, n. 6, p. 4135–4151. ISSN 1568-4946. doi: <https://doi.org/10.1016/j.asoc.2011.02.032>.
- Blum, Christian; Aguilera, Maria José Blesa; Roli, Andrea e Sampels, Michael. (2008). *Hybrid metaheuristics: an emerging approach to optimization*, volume 114 of *Studies in Computational Intelligence*. Springer.
- Blum, Christian e Roli, Andrea. (2003). Metaheuristics in combinatorial optimization: Overview and conceptual comparison. *ACM Computing Surveys*, v. 35, p. 268–308.
- Boccia, M; Sforza, A.; Sterle, C. e Vasilyev, I. (2008). A cut and branch approach for the capacitated p -median problem based on Fenchel cutting planes. *Journal of Mathematical Modelling and Algorithms*, v. 7, p. 43–58.
- Carling, K.; Han, M. e Håkansson, J. Dec(2012). Does euclidean distance work well when the p -median model is applied in rural areas. *Annals of Operations Research*, v. 201, n. 1, p. 83–97.
- Carrizosa, Emilio e Nickel, Stefan. (2003). Robust facility location. *Mathematical Methods of Operations Research*, v. 58, n. 2, p. 331–349. doi: 10.1007/s001860300294.
- Ceselli, A e Righini, G. (2005). A branch-and-price algorithm for the capacitated p -median problem. *Networks*, v. 45, n. 3, p. 125–142.
- Ceyhan, Gökhan e Özpeynirci, Özgür. (2016). A branch and price algorithm for the pharmacy duty scheduling problem. *Computers & Operations Research*, v. 72, p. 175–182. doi: <https://doi.org/10.1016/j.cor.2016.02.007>.
- Chaves, A A e Lorena, L A N. (2010). Clustering search algorithm for the capacitated centered clustering problem. *Computers & Operations Research*, v. 37, n. 3, p. 552–558. ISSN 0305-0548. Hybrid Metaheuristics.
- Cheng, Chun; Adulyasak, Yossiri e Rousseau, Louis-Martin. (2021). Robust facility location under demand uncertainty and facility disruptions. *Omega (Oxford)*, v. 103, p. 102429. ISSN 0305-0483.
- Crainic, T. G.; Gendreau, M.; Hansen, P. e Mladenović, N. (2004). Cooperative parallel variable neighborhood search for the p -median. *Journal of Heuristics*, v. 10, p. 293–314.

- Daskin, Mark. (1997). Network and discrete location: models, algorithms and applications. *Journal of the Operational Research Society*, v. 48, n. 7, p. 763–764.
- Deng, Y.; Zhang, Y. e Pan, J. (2021). Optimization for locating emergency medical service facilities: A case study for health planning from china. *Risk management and healthcare policy*, v. 14, p. 1791–1802. ISSN 1179-1594.
- Díaz, J. A. e Fernández, E. (2006). Hybrid scatter search and path relinking for the capacitated p -median problem. *European Journal of Operational Research*, v. 169, n. 2, p. 570–585.
- Duan, J.; Yang, Y. e Liu, F. (2022). Distributed optimization of integrated electricity-natural gas distribution networks considering wind power uncertainties. *International journal of electrical power & energy systems*, v. 135, p. 107460. ISSN 0142-0615.
- Dzator, M. e Dzator, J. (2013). An effective heuristic for the p -median problem with application to ambulance location. *Computers & Operations Research*, v. 50, n. 1, p. 60–74.
- Egri, Péter; Balász, Dávid; Kis, Tamás e Krész, Miklós. (2021). Robust facility location in reverse logistics. *Annals of Operations Research*, v. , p. 1–26. doi: 10.1007/s10479-021-04405-5.
- Erdoğan, Güneş; Laporte, Gilbert e Chía, Antonio M Rodríguez. (2016). Exact and heuristic algorithms for the hamiltonian p -median problem. *European Journal of Operational Research*, v. 253, n. 2, p. 280–289. doi: <https://doi.org/10.1016/j.ejor.2016.02.012>.
- Fagueye, N.; Babacar, M. N. e Ndiaye, L. (2012). Application of the p -median problem in school allocation. *American Journal of Operations Research*, v. 2, p. 253–259.
- Feo, T A e Resende, M G C. (1995). Greedy randomized adaptative search procedures. *Journal of Global Optimization*, v. 6, p. 109–133.
- Fisher, Marshall L; Jaikumar, Ramchandran e Van Wassenhove, Luk N. (1986). A multiplier adjustment method for the generalized assignment problem. *Management science*, v. 32, n. 9, p. 1095–1103.
- Fleszar, K. e Hindi, K.S. (2008). An effective VNS for the capacitated p -median problem. *European Journal of Operational Research*, v. 191, n. 3, p. 612 – 622.
- Garey, Michael R. e Johnson, David S. (1979). *Computers and Intractability: A guide to the theory of NP-completeness*. W. H. Freeman and Company.
- Ghaffarinasab, Nader. (2022). Exact algorithms for the robust uncapacitated multiple allocation p -hub median problem. *Optimization Letters*, v. 16, p. 1745–1772. doi: 10.1007/s11590-021-01799-w.

- Ghaffarinasab, Nader; Andaryan, Abdullah Zare e Torkayesh, Ali Ebadi. (2019). Robust single allocation p-hub median problem under hose and hybrid demand uncertainties: models and algorithms. *International Journal of Management Science and Engineering Management*, v. 15, n. 3, p. 184–195. doi: 10.1080/17509653.2019.1683479.
- Glover, F. e Sörensen, K. (2015). Metaheuristics. *Scholarpedia journal*, v. 10, n. 4, p. 6532. ISSN 1941-6016.
- Glover, Fred. (1986). Future paths for integer programming and links to artificial intelligence. *Computers & Operations Research*, v. 13, n. 5, p. 533–549.
- Govindan, Kannan; Fattahi, Mohammad e Keyvanshokoh, Esmaeil. (2017). Supply chain network design under uncertainty: A comprehensive review and future research directions. *European Journal of Operational Research*, v. 263, n. 1, p. 108–141.
- Guazzelli, C. S. e Cunha, C. B. (2014). Modelagem matemática para o problema de localização de centros de distribuição de empresa de alimentos. *Transportes*, v. 22, n. 3, p. 2237–1346.
- Gülpinar, N.; Pachamanova, D. e Çanakoğlu, E. (2013). Robust strategies for facility location under uncertainty. *European Journal of Operational Research*, v. 225, n. 1, p. 21–35.
- Hakimi, S. L. (1964). Optimum locations of switching center and the absolute centers and medians of a graph. *Operations Research*, v. 12, n. 3, p. 450–459. doi: 10.1287/opre.12.3.450.
- Hansen, Pierre e Mladenović, Nenad. (1997). Variable neighborhood search for the p-median. *Location Science*, v. 5, n. 4, p. 207–226.
- Hansen, Pierre; Mladenović, Nenad e Moreno-Pérez, José A. (2008). Variable neighbourhood search: methods and applications. *4OR*, v. 6(4), p. 319–360.
- Hansen, Pierre; Mladenović, Nenad e Moreno Pérez, José A. (2010). Variable neighbourhood search: methods and applications. *Annals of Operations Research*, v. 175, p. 367–407. doi: <https://doi-org.ez107.periodicos.capes.gov.br/10.1007/s10479-009-0657-6>.
- Hansen, Pierre; Mladenović, Nenad; Todosijević, Raca e Hanafi, Saïd. (2017). Variable neighborhood search: basics and variants. *EURO Journal on Computational Optimization*, v. 5, p. 423–454. doi: 10.1007/s13675-016-0075-x.
- Herda, Miloš. (2015). Combined genetic algorithm for capacitated p-median problem. *2015 16th IEEE International Symposium on Computational Intelligence and Informatics (CINTI)*, p. 151–154, (2015).
- Hosseini-Nasab, H.; Fereidouni, S.; Ghomi, F.S. M. T. e Fakhrzad, M. B. (2017). Classification of facility layout problems: a review study. *International Journal of Advanced Manufacturing Technology*, v. 94, n. 1-4, p. 957–977. ISSN 0268-3768.

- Iori, M.; de Lima, V. L.; Martello, S.; Miyazawa, F. K. e Monaci, M. (2021). Exact solution techniques for two-dimensional cutting and packing. *European Journal of Operational Research*, v. 289, n. 2, p. 399–415. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2020.06.050>.
- Irawan, C A; Imran, A e Luis, M. (2017). Solving the bi objective capacitated p median problem with multilevel capacities using compromise programming and vns. *International Transactions in Operational Research*, v. 1, n. 1, p. 1–20.
- Isler, Cassiano Augusto; Bonassa, Antonio Carlos e daCunha, Cláudio Barbieri. (2012). Algoritmo genético para resolução do problema de p -medianas capacitado associado à distribuição de peças automotivas. *Transportes*, v. 20, n. 2, p. 5–14.
- Jánošíková, L'udmila; Herda, Miloš e Haviar, Michal. (2017). Hybrid genetic algorithms with selective crossover for the capacitated p -median problem. *Central European Journal of Operations Research*, v. 25, n. 3, p. 651–664. doi: 10.1007/s10100-017-0471-1.
- Ji, Ya-Min e Qi, Ming-Liang. (2020). A robust optimization approach for decontamination planning of emergency planning zone: Facility location and assignment plan. *Socio-economic planning sciences*, v. 70, p. 100740. ISSN 0038-0121.
- Jourdan, L.; Basseur, M. e Talbi, E.-G. (2009). Hybridizing exact methods and metaheuristics: A taxonomy. *European Journal of Operational Research*, v. 199, n. 3, p. 620–629.
- Junzo, W.; Arunava, R.; Raturaj, K.; Hoang, P. e Bing, X. (2019). Emerging trends, techniques and open issues of containerization: A review. *IEEE access*, v. 7, p. 152443–152472.
- Karatas, M. e Yakici, E. (2019). An analysis of p -median location problem: Effects of backup service level and demand assignment policy. *European Journal of the Operational Research*, v. 272, p. 207–218.
- Kariv, O. e Hakimi, S. L. (1979). An algorithmic approach to network location problems. I: The p -centers. *SIAM Journal on Applied Mathematics*, v. 37, n. 3, p. 513–538. doi: 10.2307/2100911.
- Katsigiannis, Fotios A e Zografos, Konstantinos G. (2021). Optimising airport slot allocation considering flight-scheduling flexibility and total airport capacity constraints. *Transportation research. Part B: methodological*, v. 146, p. 50–87. ISSN 0191-2615.
- Kaveh, A. e Nasr, H. (2011). Solving the conditional and unconditional p -center problem with modified harmony search: A real case study. *Scientia Iranica*, v. 18, n. 4, p. 867–877.
- Khayamim, A.; Mirzazadeh, A. e Naderi, B. (2018). Portfolio rebalancing with respect to market psychology in a fuzzy environment: A case study in tehran stock exchange. *Applied Soft Computing*, v. 64, p. 244–259. ISSN 1568-4946.

- Kim, Jun-Hyuk e Soh, Soonhu. (2012). Designing hub-and-spoke school bus transportation network: A case study of wonkwang university. *Promet - Traffic & Transportation*, v. 24, n. 5, p. 389–394. doi: 10.7307/ptt.v24i5.1174.
- Klose, A. e Drexl, A. (2005). Facility location models for distribution system design. *European Journal of Operational Research*, v. 162, p. 4–29.
- Kocatepea, A.; Ozguvena, E. E.; Hornerb, M. e Ozelc, H. (2014). Pet-and special needs-friendly shelter planning in south florida: A spatial capacitated p-median-based approach. *International Journal of Disaster Risk Reduction*, v. 31, p. 1207–1222.
- Kunkel, A. G.; Elizabeth, S.; Itallie, V. e Wu, D. (2014). Optimal distribution of medical backpacks and health surveillance assistants in malawi. *Health Care Management Science*, v. 17, p. 230–244.
- Lancia, G.; Rinaldi, F. e Serafini, P. (2018). A facility location model for air pollution detection. *Hindawi - Mathematical Problems in Engineering*, v. 2018. doi: 10.1155/2018/1683249.
- Landa-Torres, I.; Ser, J. Del; Salcedo-Sanz, S.; Gil-Lopez, S.; Portilla-Figueras, J.A. e Alonso-Garrido, O. (2012). A comparative study of two hybrid grouping evolutionary techniques for the capacitated p-median problem. *Computers & Operations Research*, v. 39, n. 9, p. 2214–2222. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2011.11.004>.
- Lin, C. C.; Yung, L. L. e Chen, Y. C. (2012). The capacitated p-hub median problem with integral constraints: An application to a chinese air cargo network. *Applied Mathematical Modelling*, v. 6, p. 2777–2787.
- Lin, Y.; Chunming, Y e Chuang, Y F. (2011). Location set covering for waste resource recycling centers in taiwan. *Resources, Conservation and Recycling*, v. 55, p. 979–985.
- Lin, Zhuangqin; Xie, Qiu; Feng, Yingbin; Zhang, Peng e Yao, Ping. (2020). Towards a robust facility location model for construction and demolition waste transfer stations under uncertain environment: The case of chongqing. *Waste Management*, v. 105, p. 73–83.
- Liu, Kanglin; Liu, Changchun; Xiang, Xi e Tian, Zhili. (2023). Testing facility location and dynamic capacity planning for pandemics with demand uncertainty. *European Journal of Operational Research*, v. 304, n. 1, p. 150–168. doi: <https://doi.org/10.1016/j.ejor.2021.11.028>.
- Liu, Tianqi e Li, Guiyu. (2021). Robust recycling facility location with clustering. *Computers & Operations Research*, v. 136, p. 105466.
- Liu, Y.; Yuan, Y.; Shen, J. e Gao, W. (2021). Emergency response facility location in transportation networks: A literature review. *Journal of Traffic and Transportation Engineering*, v. 8, n. 2, p. 153–169. ISSN 2095-7564.
- López-Ibáñez, Manuel; Dubois-Lacoste, Jérémie; Cáceres, Leslie Pérez; Birattari, Mauro e Stützle, Thomas. (2016). The IRACE package: Iterated racing for automatic algorithm configuration. *Operations Research Perspectives*, v. 3, p. 43–58.

- López-Monzalvo, Francisco e Brizuela, Carlos A. (2014). An experimental analysis of the p -median problem under uncertainty: an evolutionary algorithm approach. *European Journal of Industrial Engineering*, v. 8, p. 554–578.
- Lorena, L. A. N. e Senne, E. L. F. (2004). A column generation approach to capacitated p -median problems. *Computers & Operations Research*, v. 31, n. 6, p. 863–876. ISSN 0305-0548.
- Lorena, L. A. N.; Senne, E. L. F.; Paiva, J. A. C. e Marcones, S. P. B. (2001). Integração de modelos de localização a sistemas de informações geográficas. *Gestão e Produção (UFSCar)*, v. 8, n. 2, p. 180–195.
- Lozano, M. e García-Martínez, C. (2010). Hybrid metaheuristics with evolutionary algorithms specializing in intensification and diversification: Overview and progress report. *Computers & Operations Research*, v. 37, n. 3, p. 481–497. ISSN 0305-0548.
- Ma, Bingshan; Hu, Dawei; Chen, Xiqiong; Wang, Yin e Wu, Xue. (2021). The vehicle routing problem with speed optimization for shared autonomous electric vehicles service. *Computers & Industrial Engineering*, v. 161, p. 107614. ISSN 0360-8352.
- Marla, Lavanya; Vaze, Vikrant e Barnhart, Cynthia. (2018). Robust optimization: Lessons learned from aircraft routing. *Computers & Operations Research*, v. 98, p. 165–184. ISSN 0305-0548.
- Martins, W. A. (2009). Busca em vizinhança variável aplicado na solução do problema de planejamento da expansão do sistema de transmissão de energia elétrica. Dissertação de mestrado, UNESP - Universidade Estadual Paulista.
- Mellouli, A.; Mellouli, R. e Masmoudi, F. (2021). An innovative genetic algorithm for a multi-objective optimization of two-dimensional cutting-stock problem. *Applied Artificial Intelligence*, v. 33, n. 6, p. 531–547. ISSN 0883-9514.
- Melo, M.T.; Nickel, S. e daGama, F. Saldanha. (2009). Facility location and supply chain management – a review. *European Journal of Operational Research*, v. 196, n. 2, p. 401–412. ISSN 0377-2217.
- Menezes, R. C. e Pizzolato, N. D. (2014). Locating Public Schools In Fast Expanding Areas: Application Of The Capacitated P -Median And Maximal Covering Location Models. *Pesquisa Operacional*, v. 34, p. 301–317.
- Menghao, X.; Feng, Y.; Zhong, Y. e Qihong, Z. (2018). A modified p -median model for the emergency facilities location problem and its variable neighbourhood search-based algorithm. *Journal of Applied Mathematics*, v. 2013, p. 10. doi: <https://doi.org/10.1155/2013/375657>.
- Meraklı, Merve e Yaman, Hande. (2016). Robust intermodal hub location under polyhedral demand uncertainty. *Transportation Research Part B: Methodological*, v. 86, p. 66–85.
- Milano, Michela e Roli, Andrea. (2004). Magma: A multiagent architecture for metaheuristics. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, v. 34, n. 2, p. 925–941.

- Mjirda, Anis; Todosijević, Raca; Hanafi, Saïd; Hansen, Pierre e Mladenović, Nenad. (2017). Sequential variable neighborhood descent variants: an empirical study on the traveling salesman problem. *International Transactions in Operational Research*, v. 24, p. 615–633.
- Mladenović, Nenad; Hansen, Pierre e Moreno-Pérez, José A. (2007). The p -median problem: A survey of metaheuristic approaches. *European Journal of Operational Research*, v. 179, p. 927–939.
- Montgomery, Douglas C. e Runge, George C. (2016). *Estatística aplicada e probabilidade para engenheiros*. LTC, 6 edição.
- Müller, G. I. e Gómez, A. T. (2007). Aplicação de p -medianas ao problema do corte guilhotinado bi-dimensional para peças regulares. *Anais do VI Simpósio de Informática da Região Centro do RS (SIRC/RS 2007)*, (2007).
- Ndiaye, F.; Ndiaye, B. e Idrissa, L. (2012). Application of the p -median problem in school allocation. *American Journal of Operations Research*, v. 02, p. 253–259.
- Nikoofal, Mohammad Ebrahim e Sadjadi, Seyed Jafar. (2010). A robust optimization model for p -median problem with uncertain edge lengths. *The International Journal of Advanced Manufacturing Technology*, v. 50, p. 391–397.
- Pasha, Junayed; Dulebenets, Maxim A.; Kavooosi, Masoud; Abioye, Olumide F.; Wang, Hui e Guo, Weihong. (2020). An optimization model and solution algorithms for the vehicle routing problem with a “factory-in-a-box”. *IEEE Access*, v. 8, p. 134743–134763. doi: 10.1109/ACCESS.2020.3010176.
- Peško, Š. e Majer, T. (2015). Robust p -median problem in changing networks. *Transport Problems*, v. T. 10, z. 3, p. 125–130.
- Pizzolato, N. D.; Barros, A. G.; Barcelos, F. B. e Canen, A. G. (2004). Localização de escolas públicas: síntese de algumas linhas de experiências no brasil. *Pesquisa Operacional*, v. 24, n. 1, p. 111–131.
- Porumbel, D. e Goncalves, G. (2015). Using dual feasible functions to construct fast lower bounds for routing and location problems. *Discrete Applied Mathematics*, v. 196, p. 83–99. ISSN 0166-218X. doi: <https://doi.org/10.1016/j.dam.2014.05.017>. URL <https://www.sciencedirect.com/science/article/pii/S0166218X14002194>. Advances in Combinatorial Optimization.
- Prima, P. e Arymurthy, A. M. (2019). Optimization of school location-allocation using firefly algorithm. *Journal of Physics: Conference Series*, v. 1235, n. 1, p. 012002.
- Qin, Gaoyuan; Tao, Fengming e Li, Lixia. (2019). A vehicle routing optimization problem for cold chain logistics considering customer satisfaction and carbon emissions. *International Journal of Environmental Research and Public Health*, v. 16, n. 4. ISSN 1660-4601. doi: 10.3390/ijerph16040576. URL <https://www.mdpi.com/1660-4601/16/4/576>.

- Quadros, H.; Costa Roboredo, M. e Alves Pessoa, A. (2018). A branch-and-cut algorithm for the multiple allocation r-hub interdiction median problem with fortification. *Expert Systems with Applications*, v. 110, p. 311–322. ISSN 0957-4174. doi: <https://doi.org/10.1016/j.eswa.2018.05.036>.
- Rodriguez, Francisco J.; Garcia-Martinez, Carlos e Lozano, Manuel. (2012). Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: Taxonomy, comparison, and synergy test. *IEEE Transactions on Evolutionary Computation*, v. 16, n. 6, p. 787–800. doi: 10.1109/TEVC.2012.2182773.
- Rostami, Borzou; Buchheim, Christoph; Meier, J. Fabian e Clausen, Uwe. (2016). Lower bounding procedures for the single allocation hub location problem. *Electronic Notes in Discrete Mathematics*, v. 52, p. 69–76. doi: <https://doi.org/10.1016/j.endm.2016.03.010>. INOC 2015 – 7th International Network Optimization Conference.
- Ruslim, Nooradelena Mohd e Ghani, Noraida Abdul. (2006). An application of the p -median problem with uncertainty in demand in emergency medical services. *Proceedings of the 2nd IMT-GT Regional Conference on Mathematics, Statistics and Applications*, (2006). URL <http://math.usm.my/research/OnlineProc/OR06.pdf>.
- Ryu, Jaehyeon e Park, Sungsoo. (2022). A branch-and-price algorithm for the robust single-source capacitated facility location problem under demand uncertainty. *EURO Journal on Transportation and Logistics*, v. 11, p. 100069. doi: <https://doi.org/10.1016/j.ejtl.2021.100069>.
- Sahin, G. e Süral, H. (2007). A review of hierarchical facility location models. *Computers & Operations Research*, v. 34, p. 2310–2331.
- Saif, Ahmed e Delage, Erick. (2021). Data-driven distributionally robust capacitated facility location problem. *European Journal of Operational Research*, v. 291, n. 3, p. 995–1007. ISSN 0377-2217. doi: <https://doi.org/10.1016/j.ejor.2020.09.026>. URL <https://www.sciencedirect.com/science/article/pii/S0377221720308304>.
- Salman, F. Sibel e Yücel, Eda. (2015). Emergency facility location under random network damage: Insights from the istanbul case. *Computers & Operations Research*, v. 62, p. 266–281. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2014.07.015>. URL <https://www.sciencedirect.com/science/article/pii/S0305054814001981>.
- Sanchez, David Torres; Boyacı, Burak e Zografos, Konstantinos G. (2020). An optimisation framework for airline fleet maintenance scheduling with tail assignment considerations. *Transportation research. Part B: methodological*, v. 133, p. 142–164. ISSN 0191-2615.
- Senne, E. L. F. e Lorena, L. A. N. (2003). Abordagens complementares para problemas de p -medianas. *Revista Produção*, v. 13, n. 3, p. 78–87.
- Senne, E. L. F.; Lorena, L. A. N. e Pereira, M. A. (2005)a. A branch-and-price approach to p -median location problems. *Computers & Operations Research*, v. 32, p. 1655–1664.

- Senne, E.L.F.; Lorena, Luiz A. N. e Pereira, M. A. (2005)b. A branch-and-price approach to p -median location problems. *Computers & Operations Research*, v. 32, n. 6, p. 1655–1664. ISSN 0305-0548. doi: <https://doi.org/10.1016/j.cor.2003.11.024>. URL <https://www.sciencedirect.com/science/article/pii/S0305054803003630>.
- Shanjiang, Z; Woon, K e Len, C Gang. (2012). Design and benefit cost analysis of deploying freeway incident response units: Case study for capital beltway in maryland. *Transportation Research Record*, v. 2278, n. 1, p. 104–114.
- Shapiro, Alexander. (2008). Stochastic programming approach to optimization under uncertainty. *Mathematical Programming*, v. 112, n. 1, p. 183–220.
- Shehadeh, K. S. e Sanci, E. (2021). Distributionally robust facility location with bimodal random demand. *Computers & Operations Research*, v. 134, p. 105–257.
- Soltanpour, Akram; Baroughi, Fahimeh e Alizadeh, Behrooz. (2020). A solution algorithm for p -median location problem on uncertain random networks. *Facta Universitatis: Series Mathematics and Informatics*, v. 35, n. 1, p. 55–71. doi: 10.22190/FUMI2001055S.
- Souza, M. J. F; Xavier, A. X. e Araújo, C. R. (2002). Experiência com a utilização de simulated annealing e busca tabu na resolução do problema de alocação de salas. *Simpósio Brasileiro de Pesquisa Operacional*, v. 34.
- Souza, Marccone Jamilson Freitas; Coelho, Igor Machado; Ribas, Sabir; Santos, Haroldo Gambini e Merschmann, Luiz Henrique Campos. (2010). A hybrid heuristic algorithm for the open-pit-mining operational planning problem. *European Journal of Operational Research*, v. 207, n. 2, p. 1041–1051.
- Soyster, A. L. (1973). Technical note: Convex programming with set-inclusive constraints and applications to inexact linear programming. *Operations Research*, v. 21, n. 5, p. 1154–1157. doi: 10.1287/opre.21.5.1154. URL <https://doi.org/10.1287/opre.21.5.1154>.
- Stefanello, Fernando; deAraújo, Olinto César Bassi e Müller, Felipe Martins. (2015). Matheuristics for the capacitated p -median problem. *International Transactions in Operational Research*, v. 22, n. 1, p. 149–167. doi: 10.1111/itor.12103.
- Steglich, Mike. (2019). A hybrid heuristic based on self-organising maps and binary linear programming techniques for the capacitated p -median problem. Iacono, Mauro; Palmieri, Francesco; Gribaudo, Marco e Ficco, Massimo, editors, *Proceedings of the 33rd International ECMS Conference on Modelling and Simulation, ECMS 2019 Caserta, Italy, June 11-14, 2019*, p. 267–276. European Council for Modeling and Simulation, (2019). doi: 10.7148/2019-0267. URL <https://doi.org/10.7148/2019-0267>.
- Subramanian, A.; Drummond, L. M. A.; Bentes, C.; Ochi, L. S. e Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, v. 37, n. 11, p. 1899 – 1911.

- Suckling, J.; Druckman, A.; Small, R.; Cecelja, F. e Bussemaker, M. (2021). Supply chain optimization and analysis of hermetia illucens (black soldier fly) bioconversion of surplus foodstuffs. *Journal of Cleaner Production*, v. 321, p. 128711. ISSN 0959-6526.
- Sylvia, Y. H.; Yong-Hong, K. e Dan, W. (2016). Incorporating institutional and spatial factors in the selection of the optimal locations of public electric vehicle charging facilities: A case study of beijing, china. *Transportation Research Part C: Emerging Technologies*, v. 67, p. 131–148. ISSN 0968-090X. doi: <https://doi.org/10.1016/j.trc.2016.02.003>. URL <http://www.sciencedirect.com/science/article/pii/S0968090X16000486>.
- Talbi, El-Ghazali e Todosijević, Raca. (2017). The robust uncapacitated multiple allocation p-hub median problem. *Computers & Industrial Engineering*, v. 110, p. 322–332. doi: <https://doi.org/10.1016/j.cie.2017.06.017>. URL <https://www.sciencedirect.com/science/article/pii/S0360835217302693>.
- Tayali, Seda Tolun. (2020). A novel backtesting methodology for clustering in mean-variance portfolio optimization. *Knowledge-based systems*, v. 209, p. 106454. ISSN 0950-7051.
- Terouhid, S. A.; Ries, R. e Mirhadi, F. M. (2012). Towards sustainable facility location – a literature review. *Journal of Sustainable Development*, v. 5, n. 7, p. 18–18. ISSN 1913-9063.
- Wietschel, Lars; Messmann, Lukas; Thorenz, Andrea e Tuma, Axel. (2021). Environmental benefits of large-scale second-generation bioethanol production in the eu: An integrated supply chain network optimization and life cycle assessment approach. *Journal of Industrial Ecology*, v. 25, n. 3, p. 677–692.
- Wu, Gang; Li, Ting; Li, Min; Lan, Peng; Ma, Ruiguang; Ma, Tiannan e Jingwei, Deng. (2022). Day-ahead optimal scheduling model of transmission–distribution integrated electricity–gas systems based on convex optimization. *Energy Reports*, v. 8, p. 759–767.
- Xiao, Y. N. e Ning, X. (2022). Optimization model of financial market portfolio using artificial fish swarm model and uniform distribution. *Computational Intelligence and Neuroscience*, v. 2022, p. 7483454–7483454.
- Yaghini, Masoud; Momeni, Mohsen; Sarmadi, Mohammadreza e Ahadi, Hamid Reza. (2013). A hybrid metaheuristic approach for the capacitated p-median problem. *Applied Soft Computing*, v. 13, n. 9, p. 3922–3930. ISSN 1568-4946.
- Ziviane, N. (2004). *Projeto de algoritmos, com implementações em PASCAL e C*. Thomson.