META-HEURÍSTICAS GRASP E ILS APLICADAS AO PROBLEMA DE LOCALIZAÇÃO DE INSTALAÇÕES INDESEJADAS

WESLEY DE MATOS LANCUNA

META-HEURÍSTICAS GRASP E ILS APLICADAS AO PROBLEMA DE LOCALIZAÇÃO DE INSTALAÇÕES INDESEJADAS

Dissertação apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Modelagem Matemática e Computacional.

Orientador: Elisangela Martins de Sá
Coorientador: Sérgio Ricardo de Souza

Belo Horizonte

Outubro de 2020

Lancuna, Wesley de Matos
L252m Meta-heurísticas GRASP e ILS aplicadas ao problema de localização de instalações indesejadas / Wesley de Matos Lancuna. – 2020.
64 f.

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Modelagem Matemática e Computacional.
Orientadora: Elisangela Martins Sá.
Coorientador: Sérgio Ricardo de Souza
Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais.

1. Programação heurística – Teses. 2. Instalações elétricas – Teses. 3. Sistema de armazenamento e recuperação de informação – Uso do solo – Teses. 4. Algoritmos - Teses. I. Sá, Elisangela Martins. II. Souza, Sérgio Ricardo. III. Centro Federal de Educação Tecnológica de Minas Gerais. IV. Título.

CDD 519.6

Elaboração da ficha catalográfica pela bibliotecária Jane Marangon Duarte, CRB 6º 1592 / Cefet/MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO CURSO DE MESTRADO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"META-HEURÍSTICAS GRASP E ILS APLICADAS AO PROBLEMA DE LOCALIZAÇÃO DE INSTALAÇÕES INDESEJADAS"

Dissertação de Mestrado apresentada por **Wesley de Matos Lancuna**, em 02 de outubro de 2020, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Profª. Drª. Elisângela Martins de Sá (Orientadora)
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. Ricardo Saraiva de Camargo
Universidade Federal de Minas Gerais

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

Prof. Dr. Rodrigo Tomás Nogueira Cardoso
Centro Federal de Educação Tecnológica de Minas Gerais

Visto e permitida a impressão,

Prof. Dr. Thiago de Souza Rodrigues
Coordenador do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

Agradecimentos

Agradeço primeiramente a Deus, pela oportunidade da vida e de poder romper uma fronteira acadêmica com a pesquisa científica. Agradeço aos professores Dornelas e Hanaoka, por fomentar e oportunizar esse sonho por meio da carta de recomendação. Aos meus orientadores, agradeço à Profª. Elisangela, pela paciência, ajuda e todo aprendizado compartilhado. Ao Prof. Sérgio, agradeço pela oportunidade, confiança e compromisso. Agradeço a minha família e minha companheira pelo apoio. Aos amigos e professores do PPGMMC por todo conhecimento compartilhado. Ao CEFET-MG que nos forneceu materiais e condições para desenvolver esta pesquisa. Agradecimento especial a CAPES, pelo apoio financeiro.

"Lá vou eu com minhas pesquisas, minhas brisas,
vontade de dar um grito para as vozes vencidas.
Convencidas do fracasso, se iniciam minhas noites mal dormidas.
Minha liquidez negativada, me ensina a contornar os obstáculos.
Meu oráculo, fugiu e me deixou nessa matrix.
Surge mais uma boa solução aleatória, minha oratória,
e os ratos me mostram o personagem em que eu me fiz."
(Wesley Lancuna, O experimento da pesquisa)

Resumo

Este trabalho tem seu foco no problema de localização de instalações indesejadas. O problema consiste em localizar instalações, de modo que as mesmas estejam o mais afastado possível dos clientes. As instalações serão selecionadas de forma a maximizar a soma das distâncias dos clientes à instalação mais próxima. Possíveis aplicações desse problema são instalações de aterros sanitários, usinas nucleares, barragens de rejeitos de minério e penitenciárias. Como o problema é considerado NP-difícil, para buscar melhores soluções para o problema, propõe-se dois algoritmos meta-heurísticos, um que combina as técnicas do GRASP e do ILS, e o outro que combina as técnicas de inserção mais barata e o ILS. Os resultados mostram que as técnicas propostas apresentam resultados equivalentes aos melhores algoritmos da literatura estatisticamente e são mais simples de serem reproduzidas.

Palavras-chave: Localização de Instalações Indesejadas. GRASP. ILS. Meta-heurísticas.

Abstract

This dissertation addresses the problem of locating obnoxious installations. This problem consists of locating facilities so that they are as far away from customers as possible. The facilities are selected to maximize the sum of customer distances to the nearest facility. Possible applications of this problem are landfill facilities, nuclear power plants, ore tailings dams, and penitentiaries. Since this is an NP-hard problem, two-hybrid metaheuristics are proposed to find better solutions to the problem, one combining Greedy Randomized Adaptive Search Procedure (GRASP) and Iterated Local Search (ILS) metaheuristic and the other combining cheaper insertion techniques and ILS. The results show that the proposed techniques present statistically equivalent results to the best-known algorithms in the literature and are simpler to reproduce.

Keywords: Obnoxious Location Problem. GRASP. ILS. OpMP.

Lista de Figuras

Lista de Tabelas

Sumário

Capítulo 1

Introdução

1.1 Apresentação Inicial

Neste trabalho é estudado o Problema de Localização de p Instalações Indesejadas – p-PLII (*Obnoxious p-Median Problem*). O p-PLII é uma adaptação do problema clássico de p-medianas. O problema de localização de p-medianas consiste em localizar p instalações, com o intuito de minimizar a soma ponderada das distâncias de cada cliente à instalação mais próxima. No p-PLII, por outro lado, deseja-se localizar p instalações o mais afastadas possível dos clientes de modo que elas atendam aos clientes mais próximos de si. Possíveis aplicações para este problema são instalações de aterros sanitários, usinas nucleares, barragens de rejeitos de minério, presídios, dentre outras.

Para Hosseini & Esfahani [2009], em geral, as instalações são divididas em dois grupos: instalações desejadas e indesejadas. As instalações do primeiro grupo são de interesse e desejadas para os habitantes próximos, que gostariam de tê-las próximas, como hospitais, postos de bombeiros, lojas de compras e centros educacionais. O segundo grupo refere-se às que são indesejadas para a população circunvizinha, que as evita, buscando ficar o mais longe possível, como depósitos de lixo, fábricas de produtos químicos, reatores nucleares, instalações militares e presídios. Por questões sanitárias, de segurança ou bem-estar, instalações como estas são indesejadas e tenta-se, do ponto de vista de planejamento de suas localizações, afastá-las dos centros de demanda. Deve-se observar, no entanto, que apesar da maioria destas instalações indesejadas serem necessárias para a comunidade, como o caso de locais de despejo de lixo, sua disposição pode ser desagradável para a população ao seu redor.

Formalmente, o p-PLII consiste em localizar p instalações, dentre um conjunto de instalações candidatas, de forma a maximizar a soma da distância entre os nós de demanda e as instalações mais próximas a cada um deles. Neste caso, assume-se que a

instalação mais próxima do cliente é a instalação que provoca maior efeito negativo.

Conforme citado em Batta & Chiu [1988], o problema p-PLII foi introduzido por Goldman & Dearing [1975]. Desde então, surgiram outras variações de problemas de localização de instalações consideradas indesejadas. Alguns trabalhos abordaram o problema de maneira multiobjetiva [Eiselt & Marianov, 2015; Melachrinoudis et al., 1995; Demesouka et al., 2014]. Enquanto outros focaram em uma abordagem mono-objetiva [Lin & Guan, 2018; Chiang & Lin, 2017; Cheng et al., 2019]. Quanto ao método de resolução, alguns trabalhos usaram métodos matemáticos para resolver o problema [Chiang & Lin, 2017; Drezner et al., 2018a; Belotti et al., 2007]. Outros utilizam algoritmos heurísticos [Colmenar et al., 2016; Herrán et al., 2018; Drezner et al., 2018b].

Como o p-PLII tratado é um problema NP-Difícil [Tamir, 1991], neste trabalho são propostos dois algoritmos meta-heurísticos para buscar boas soluções para instâncias deste problema. O primeiro combina técnicas das meta-heurísticas *Iterated Local Search* (ILS) e do *Greedy Randomized Adaptive Search Procedure* (GRASP) e o outro segundo é uma adaptação da meta-heurística *Iterated Local Search* (ILS). Ambos algoritmos são compostos por duas fases: (i) fase construtiva e (ii) fase de refinamento. A fase construtiva do primeiro foi feita por meio da fase construtiva da meta-heurística GRASP Feo & Resende [1995]. Já a do segundo foi feita de maneira gulosa, usando o método de inserção mais barata. Em seguida, após a construção da solução inicial, ambos utilizam o refinamento da solução com a meta-heurística ILS, proposta por Lourenço et al. [2003]. Os algoritmos foram testados para um conjunto de instâncias e os resultados foram estatisticamente equivalentes aos melhores encontrados na literatura.

1.2 Justificativa

Problemas de localização de instalações indesejadas geralmente esbarram com a pressão social contrária à construção de tais instalações próximas às residências dos moradores. Além de ser um trabalho de planejamento de alto custo, que auxiliaria na viabilização de instalações próximas a habitações e espaços de uso público, o p-PLII é um problema com facetas sociais, haja vista que uma instalação localizada em região próxima a comunidades limítrofes pode gerar um enorme passivo social, caso ocorra algum problema com a instalação e seja necessário realocar as comunidades vizinhas à instalação.

Recentemente, o problema de localização de instalações indesejadas tornou-se ainda mais importante, após acidentes com barragens de rejeito de minério, devastando

bairros por inteiro com inúmeras perdas de vidas humanas e enorme impacto ambiental [Freitas et al., 2019].

Portanto, é fundamental localizar essas instalações de maneira à mitigar os efeitos negativos da pressão social ocasionadas por instalações indesejadas muito próximas aos clientes e reduzir o risco com eventuais acidentes que possam ocorrem com as instalações.

Contudo, não é uma tarefa simples, do ponto de vista técnico, encontrar a melhor localização para uma instalação, que cause o menor efeito negativo possível, num conjunto de possíveis locais disponíveis. Não é incomum que gestores acabem escolhendo um local inadequado para a implantação de instalações altamente indesejadas, ocasionando insegurança psicológica, riscos às comunidades lindeiras e possíveis gastos desnecessários e significativos.

Com isso, surge a necessidade de se desenvolver ferramentas capazes de localizar as instalações com maior assertividade. Para isso, os algoritmos meta-heurísticos são excelentes ferramentas para problemas combinatórios para os quais métodos exatos não conseguem encontrar soluções ótimas dentre de um tempo computacional viável.

1.3 Objetivo Geral e Específicos

O objetivo geral do trabalho é propor algoritmos eficientes para localizar um conjunto de instalações indesejadas, que estejam o mais afastado possível dos clientes, em tempo computacional viável.

Os objetivos específicos são:

- Compreender o problema e os respectivos modelos matemáticos para o mesmo;
- Estudar os principais trabalhos da literatura que abordaram o problema de localizar instalações indesejadas;
- Analisar algoritmos meta-heurísticos que tratam o problema;
- Desenvolver um algoritmo meta-heurístico que gere boas soluções para o problema;
- Comparar os resultados dos algoritmos desenvolvidos com os resultados da literatura.

1.4 Metodologia de Pesquisa

Para se obter os objetivos propostos, esta pesquisa foi desenvolvida com a seguinte metodologia:

1. **Revisão bibliográfica**

   Pesquisar os principais artigos que tratam o problema. Observar nos artigos da literatura qual o tamanho das instâncias que foram resolvidas e quais técnicas foram usadas;

2. **Descrição do problema**

   Realizar uma descrição formal do problema, apresentando uma formulação matemática da literatura que represente as suas características. Examinar problemas que sejam semelhantes e quais as diferenças entre eles;

3. **Definição da abordagem utilizada**

   Definir a abordagem utilizada neste trabalho para buscar boas soluções em tempo viável;

4. **Implementação de algoritmos meta-heurísticos**

   Após analisar as mais variadas técnicas heurísticas para buscar boas soluções para problemas combinatórios, desenvolver algoritmos para o problema.

5. **Análise dos resultados obtidos**

   Para compreender e validar os resultados obtidos, realizar análises estatísticas nos resultados e comparar os mesmos com resultados da literatura.

1.5 Organização do Trabalho

O restante deste trabalho está organizado da seguinte forma: No Capítulo 2 descrevem-se formalmente o problema e os demais relacionados, com suas respectivas diferenças. Além disso, apresenta-se a revisão bibliográfica para o problema, citando os principais trabalhos relacionados ao p-PLII. A fundamentação teórica utilizada nesse trabalho é apresentada no Capítulo 3. Os métodos usados para buscar melhores soluções com as meta-heurísticas propostas são apresentados no Capítulo 4. São apresentados os resultados dos experimentos computacionais realizados e as comparações feitas com algoritmos da literatura no Capítulo 5. O Capítulo 6 apresenta as conclusões advindas do trabalho e discorre a respeito de trabalhos futuros.

Capítulo 2

Descrição do Problema

Neste capítulo é apresentada uma descrição do problema de localização de instalações indesejadas (p-PLII). Inicia-se com o problema clássico das p-Medianas na Seção 2.1, por ser o problema mais conhecido da literatura e que deu origem ao problema tratado nesta dissertação. Parte-se, então, na Seção 2.2, para o problema de p-dispersão, que pode ser confundido com o p-PLII, por apresentar a mesma característica de maximizar distância. Após isso, na Seção 2.3, é apresentado o problema foco desta pesquisa. Sintetizam-se as principais diferenças dos problemas apresentados com o problema p-PLII na Seção 2.4. Ao final, é feita a revisão bibliográfica do problema na Seção 2.5, reportando-se os principais trabalhos relacionados ao tema. Encerra-se com um resumo de tudo que foi tratado no capítulo na Seção 2.6.

2.1 Problema clássico das p-Medianas

2.1.1 Descrição do problema

O problema clássico das p-medianas é considerado um dos principais problemas envolvendo localização de instalações [Jamshidi, 2009]. Esse problema visa encontrar os pontos medianos entre um conjunto de pontos candidatos, de modo que a soma dos custos possa ser minimizada. Já o p-PLII busca encontrar pontos medianos, tendo como objetivo maximizar a soma das menores distâncias entre um conjunto de ponto candidatos e os respectivos clientes.

O problema de localização de p instalações foi apresentado por Hakimi [1964], com o intuito de minimizar o somatório das distâncias mínimas.

Em suma, o problema das p-medianas pode ser descrito da seguinte maneira. Dado um conjunto I de clientes, um conjunto J de pontos de instalação candidatos e

uma matriz D, que armazena a distância dos clientes para as instalações candidatas, deseja-se escolher p instalações de modo que a soma das distâncias das instalações aos clientes seja a menor possível.

2.1.2 Modelo Matemático das p-Medianas

Para apresentar o modelo matemático proposto por Daskin [2011], primeiramente serão descritas as notações utilizadas e as variáveis de decisão:

Parâmetros :

- I : conjunto de nós de demandas;
- J : conjunto de instalações candidatas;
- d_{ij}: distância entre o nó de demanda $i \in I$ e a instalação candidata $j \in J$;
- p: número de instalações a serem localizadas.

Variáveis de Decisão :

- A variável X_j pode assumir os valores 0 ou 1, sendo que 1 significa que a instalação $j \in J$ está ativa e 0 se está inativa;
- A variável Y_{ij} pode assumir os valores 0 ou 1, sendo que 1 significa que a demanda do nó $i \in I$ é satisfeita por uma instalação $j \in J$ e 0, caso contrário.

A partir dessa notação, apresenta-se o seguinte modelo:

$$\min \quad \sum_{i \in I} \sum_{j \in J} d_{ij} Y_{ij} \tag{2.1}$$

$$\text{sujeito a:} \quad \sum_{j \in J} Y_{ij} = 1 \quad \forall\, i \in J \tag{2.2}$$

$$\sum_{j \in J} X_j = p \tag{2.3}$$

$$Y_{ij} \leq X_j \quad \forall\, i \in I, j \in J \tag{2.4}$$

$$X_j \in \{0, 1\} \quad \forall\, j \in J \tag{2.5}$$

$$Y_{ij} \in \{0, 1\} \quad \forall i \in I, j \in J \tag{2.6}$$

A função objetivo (2.1), que deseja-se minimizar, representa o somatório das distâncias ponderadas entre os nós de demanda e as instalações. O conjunto de restrições (2.2) garante que cada nó de demanda seja alocado a somente uma instalação. A restrição (2.3) limita o número de instalações abertas a exatamente p. O conjunto de restrições (2.4) garante que um nó de demanda só pode ser alocado a instalações abertas. Além disso, essa restrição permite que mais de um nó de demanda esteja ligado a uma mesma instalação. Já os conjuntos de restrições (2.5) e (2.6) definem as variáveis de decisão como variáveis binárias.

2.2 Problema de p-Dispersão

2.2.1 Descrição do problema

O problema de p-dispersão é semelhante ao problema de localização de p instalações indesejadas. Porém, no problema de p-dispersão busca-se maximizar a menor distância entre as instalações, enquanto no p-PLII deseja-se maximizar o somatório das distâncias mínimas. O nome p-dispersão surge da ideia de tornar as instalações mais dispersas possíveis entre si [Moon & Chaudhry, 1984]. Em um contexto de logística, o problema é aplicável para instalações cuja proximidade entre si é indesejada [Erkut, 1990]. Nisso, o problema difere-se do p-PLII, no qual a proximidade entre cliente e instalação é indesejada, mas a proximidade entre instalações não é um problema. Por exemplo, no problema de p-dispersão, é indesejável que silos de mísseis estejam dispostos próximos entre si, pois, se vários silos estiverem agrupados na mesma região, em um mesmo ataque todos eles podem ser destruídos. Consequentemente, quanto mais afastados estiverem os silos uns dos outros, menores serão as chances de um possível inimigo eliminar mais de um silo em um único ataque. Em contrapartida, para o p-PLII, não haveria problema em posicionar instalações como barragens de rejeito de minério, ou aterros sanitários, próximos uns dos outros, desde que os mesmos estivessem afastados dos clientes.

O problema de p-dispersão trata também de instalações de interesse da população, pois sua abordagem é aplicável para situações de máxima cobertura. A instalação de um restaurante *fast-food* em uma área urbana é um exemplo disso. Um gestor de negócios pode ter n locais de aluguel disponíveis e ter intenção de abrir p franquias. Nesse caso, portanto, pode não ser interessante ter duas franquias próximas uma das outras compartilhando a mesma base de clientes e, possivelmente, o número total de vendas poderia ser maior se as franquias estiverem espalhadas pela cidade [Erkut, 1990].

2.2.2 Modelo matemático do Problema de p-Dispersão

O modelo matemático apresentado a seguir, relativo ao problema de p-dispersão, foi utilizado em Pisinger [2006]. Dado um conjunto $N = \{1, 2, \ldots, n\}$ de instalações, o problema de p-dispersão consiste em escolher p destas instalações ($1 \leq p \leq n$) de modo que a distância mínima entre estas p instalações escolhidas seja maximizada.

Para descrever o problema, segundo o modelo apresentado em Pisinger [2006] e Sayah & Irnich [2017], considere os parâmetros e as variáveis de decisão a seguir:

Parâmetros :

- d_{ij}: distância entre uma instalação i e uma instalação j;
- p: número de instalações a serem localizadas.

Variáveis de Decisão :

- x_j: variável de decisão binária que pode assumir os valores de 0 ou 1, sendo que 1 significa que a instalação $j \in N$ está ativa e 0, se está inativa;
- t: variável de decisão contínua representa a menor distância entre as instalações j ativas.

O modelo inteiro misto é então dado por:

$$\begin{aligned}
\max \quad & t && (2.7)\\
\text{sujeito a:} \quad & \sum_{j \in N} x_j = p && (2.8)\\
& t x_i x_j \leq d_{ij}, \quad \forall\, i, j \in N && (2.9)\\
& x_j \in \{0, 1\}, \quad \forall\, j \in N && (2.10)\\
& t \in \mathbb{R}_+ && (2.11)
\end{aligned}$$

A função objetivo (2.7), que se deseja maximizar, representa a distância mínima t. A restrição (2.8) garante que apenas p instalações sejam abertas. O conjunto de restrições (2.9) garante que, para todo i e j pertencente a N, a distância d_{ij} seja a menor distância. Já o conjunto de restrições (2.10) exige que as variáveis de decisão sejam binárias e, finalmente, a restrição (2.11) garante que t seja uma variável contínua e não-negativa. Observe, porém, que este modelo é não-linear, dado que o conjunto de restrições (2.9) apresenta, em seu lado esquerdo, um produto de variáveis.

2.3 O problema de localização de p instalações indesejadas

2.3.1 Descrição do problema

O Problema de Localização de p Instalações Indesejadas (p-PLII) consiste em selecionar um subconjunto de p instalações de um determinado conjunto J de possíveis localizações, de forma que a soma da distância de todos os clientes até a instalação mais próxima seja maximizada. Note que o objetivo do problema é encontrar p locais que mantenham os clientes o mais afastados possível das instalações.

2.3.2 Modelo matemático do p-PLII

Considere I como sendo o conjunto de clientes, de modo que $|I| = m$ é a quantidade de clientes. Seja J o conjunto de instalações candidatas, em que $|J| = n$ é a quantidade de instalações candidatas. Uma solução para o problema pode ser representada pelo conjunto de instalações abertas, denotado por S. A distância entre o cliente $i \in I$ e a instalação candidata $j \in J$ é representada por d_{ij}.

Um modelo matemático para representar o problema, apresentado por Colmenar et al. [2016], é dado por:

$$\max \quad \sum_{i \in I} \min\{d_{ij} \quad : \quad j \in S\} \tag{2.12}$$

$$\text{sujeito a:} \qquad S \subseteq J, \quad |S| = p \tag{2.13}$$

A expressão (2.12) representa a função objetivo do problema, que busca maximizar o somatório das distâncias mínimas entre todos os clientes até as instalações pertencentes ao conjunto S das instalações abertas. As restrições (2.13) indicam que o conjunto S das instalações abertas deve estar contido no conjunto J das instalações candidatas e ter cardinalidade igual a p, garantindo que sejam abertas apenas as instalações necessárias.

2.3.3 Exemplo para o p-PLII

Uma maneira de elucidar o problema é apresentar como é feita a avaliação de uma solução segundo suas características. Para isto, será usada a representação visual apresentada por Colmenar et al. [2016].

Considere uma matriz de distância conforme ilustrada na Tabela 2.1. Esta tabela apresenta uma instância com um conjunto com 9 clientes no total e 6 instalações can-

didatas. Perceba que, caso fosse necessário escolher apenas uma instalação, o problema seria facilmente resolvido, com a escolha da instalação que está mais distante de todos os clientes, como é o caso da instalação j_5. Porém, quando a quantidade de instalações aumenta, o problema não pode ser resolvido buscando simplesmente as p instalações mais distantes de todos os clientes. Neste caso busca-se a combinação de instalações candidatas, que gere um conjunto de instalações ativas, que estejam o mais distante possível dos clientes.

Tabela 2.1. Ilustração de uma instância para o p-PLII contendo o somatório das distâncias de cada instalação j para os clientes i

	j_1	j_2	j_3	j_4	j_5	j_6
i_1	3	2	10	13	4	8
i_2	14	14	11	3	15	12
i_3	5	4	6	12	14	6
i_4	3	1	1	3	10	9
i_5	13	9	2	10	9	4
i_6	9	5	11	14	4	11
i_7	4	3	4	13	4	2
i_8	12	2	9	3	15	13
i_9	9	11	1	2	7	4
$\sum$	72	51	55	73	82	69

Para representar como a avaliação de uma solução funciona, pode-se usar essa matriz de distância e considerar duas possíveis soluções com três instalações, conforme apresentado na Tabela 2.2. Perceba que, dada a característica do problema, cada cliente está sujeito a ser alocado à instalação aberta mais próxima de si. Seja $f(S)$ o valor da função objetivo associada à solução com conjunto da instalações ativas S. A maneira de avaliar a função $f(S)$ é somando as distâncias dos clientes à instalação ativa mais próxima dos mesmos. Na Tabela 2.2 isso foi representado pelas distâncias em negrito, na qual a solução S_a, composta por j_2, j_5 e j_6, resulta em $f(S_a) = 35$, que é melhor que a solução S_b, composta por j_2, j_3 e j_4, em que $f(S_b) = 23$.

Tabela 2.2 Ilustração da avaliação de duas soluções para o p-PLII com a soma das menores distâncias em negrito

$S_a = \{j_2, j_5, j_6\}$				$S_b = \{j_2, j_3, j_4\}$		
	j_2	j_5	j_6	j_2	j_3	j_4
i_1	**2**	4	8	**2**	10	13
i_2	14	15	**12**	14	11	**3**
i_3	**4**	14	6	**4**	6	12
i_4	**1**	10	9	**1**	1	3
i_5	9	9	**4**	9	**2**	10
i_6	5	**4**	11	**5**	11	14
i_7	3	4	**2**	**3**	4	13
i_8	**2**	15	13	**2**	9	3
i_9	11	7	**4**	11	**1**	2
$f(S_a) : 35$				$f(S_b) : 23$		

2.4 Principais diferenças entre os problemas das p-medianas, p-dispersão e p-PLII

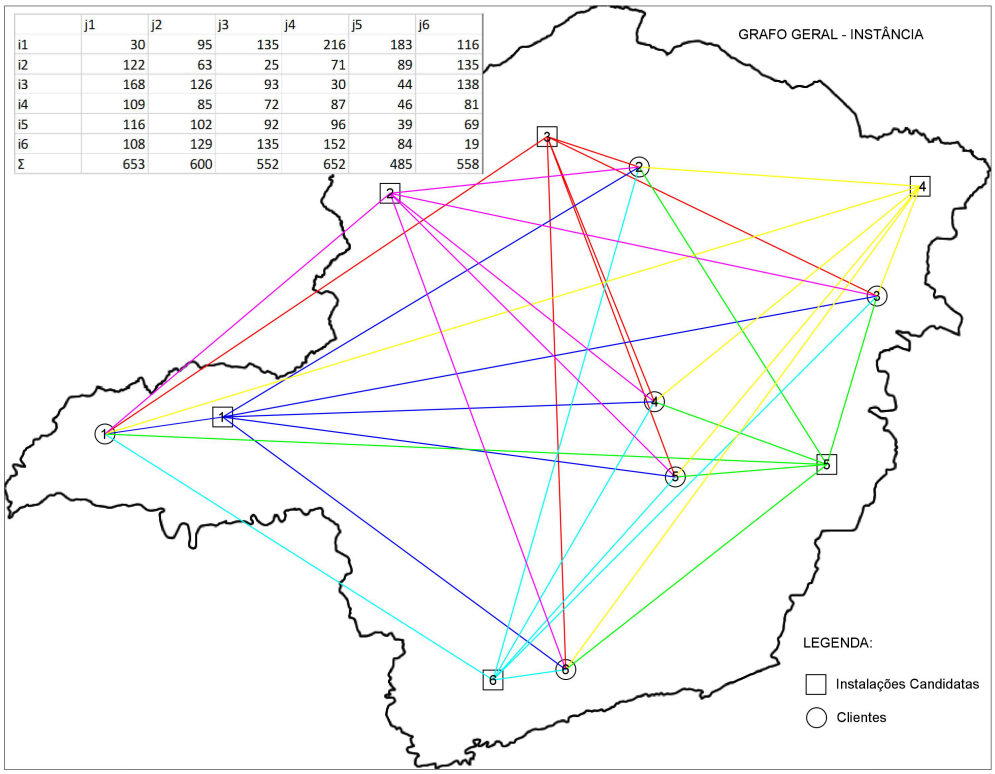
Dentre os problemas de localização de instalações, existem problemas próximos ao problema das p instalações indesejadas. Para esclarecer suas distinções, é interessante fazer uma comparação dos problemas similares ao problema tratado.

2.4.1 O problema das p-medianas e o p-PLII

Conforme foi descrito, o problema de localizar p instalações que são desejadas à população é facilmente representado pelo problema das p-medianas. No caso de instalações desejadas, a logística é facilitada, visto que o custo de transporte também é minimizado automaticamente ao buscar instalações mais próximas. Já no p-PLII deseja-se afastar as instalações dos clientes, mas geralmente esses clientes estão envolvidos logisticamente com as instalações.

Em ambos problemas, a distância a ser considerada é a menor distância entre o cliente e a instalação mais próxima. A diferença é que, para instalações desejadas, em um contexto de minimização, tenta-se localizar instalações mais próximas à população e cada cliente será alocado na instalação mais próxima de si. Ou seja, a condição de que cada cliente esteja alocado à instalação mais próxima beneficia a função objetivo do problema. Para instalações indesejadas, esta condição do problema é prejudicial à função objetivo, visto que deseja-se maximizar o somatório dessas distâncias.

Para ilustrar o problema, pode-se considerar um exemplo, apresentado na Figura 2.1, com seis instalações e seis clientes. Nesta figura, os vértices do grafo são representados pelas circunferências (clientes) e quadrados (instalações). De cada instalação saem seis arestas de uma mesma cor, que representam cada instalação, e de cada cliente saem seis caminhos, cada um com uma cor diferente, que representam o caminho até cada instalação. As instalações candidatas são representadas pelos quadrados e os clientes pelos círculos conforme a Figura 2.1.



	j1	j2	j3	j4	j5	j6
i1	30	95	135	216	183	116
i2	122	63	25	71	89	135
i3	168	126	93	30	44	138
i4	109	85	72	87	46	81
i5	116	102	92	96	39	69
i6	108	129	135	152	84	19
Σ	653	600	552	652	485	558

Figura 2.1. Ilustração de um grafo para o problemas de localização com seis nós clientes e 6 instalações candidatas

Observe que, no grafo formado, todas as instalações têm conexões com todos os clientes. Por exemplo, assumindo que o problema das p-medianas queira minimizar a soma das distâncias, seriam escolhidas as duas instalações cujas somas das distâncias até os clientes é a menor, conforme apresenta a Figura 2.2.

Já no caso de instalações indesejadas, busca-se a melhor combinação de duas instalações para que os clientes fiquem o mais afastado das instalações, como pode ser visto na Figura 2.3.

É interessante observar a diferença no comprimento das arestas e no somatório destas arestas de um problema para o outro. Repare que o problema das p-medianas

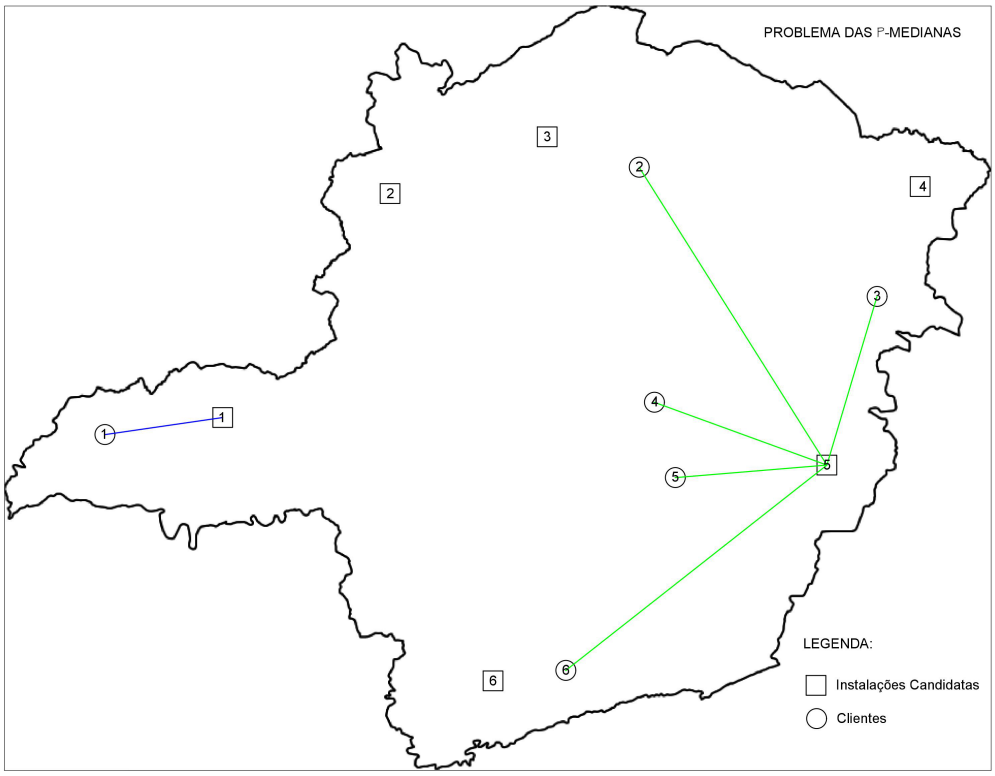


Figura 2.2. Um exemplo de Solução para p-medianas com p=2

fornece uma solução com arestas menores que o p-PLII, por mais que os dois problemas estejam alocando os clientes para as instalações abertas mais próximas.

2.4.2 O problema de p-dispersão e o p-PLII

Apesar do problema de p-dispersão ser um problema de maximizar a distância mínima [Sayah & Irnich, 2017], existem diferenças significativas entre ele e o p-PLII. A principal diferença entre eles é que, para o problema de p-dispersão, a distância que tenta-se maximizar é a distância das instalações entre si e não existem clientes envolvidos como no caso do p-PLII. Por exemplo, considerando a instância apresentada na Seção 2.4.1, retira-se os clientes e mantém-se apenas as instalações. Isso geraria uma nova matriz de distância, pois o foco seria nas distâncias entre instalações e não na distância para os clientes. Neste caso, a solução para o problema seriam as p instalações mais distantes entre si. Em suma, a solução do problema de p-dispersão é uma combinação de p instalações, que faça com que a menor distância entre as instalações seja a maior possível, conforme mostrado no exemplo da Figura 2.4, considerando escolher duas instalações.

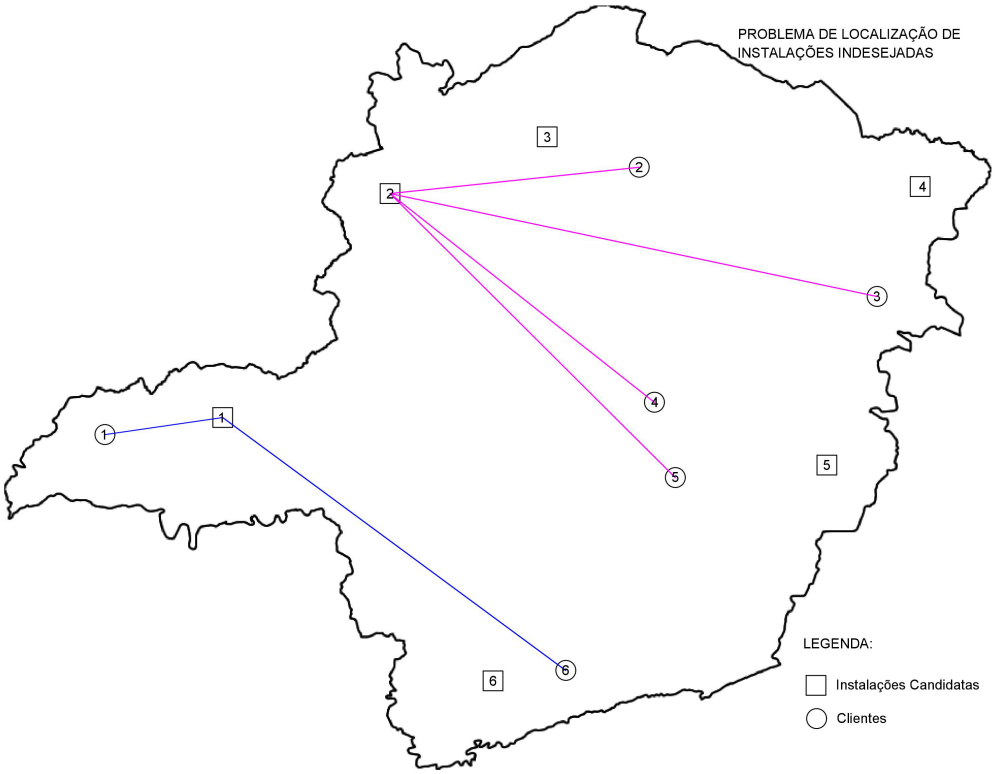
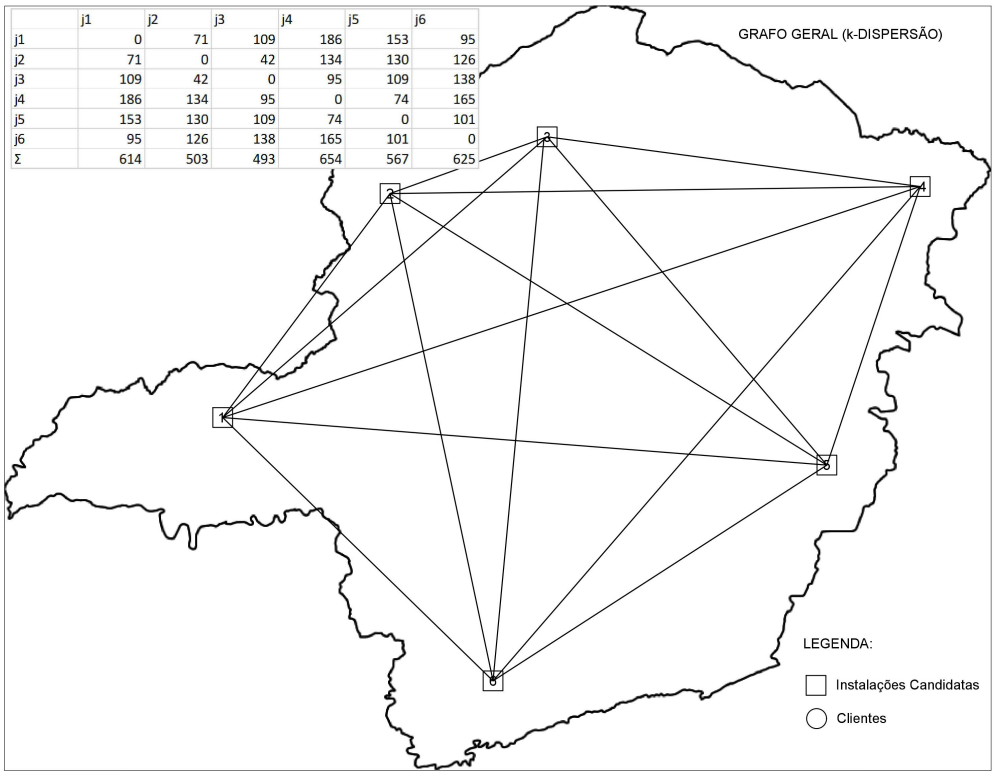


Figura 2.3. Um exemplo de Solução para o p-PLII com p=2

Note que nesse grafo foram apresentadas novas arestas que não são consideradas no p-PLII, que são as arestas entre as instalações. Para o problema p-PLII, não importa se duas instalações estão muito próximas, contando que as mesmas estejam afastadas dos clientes, como é o caso da aplicação de usinas nucleares no Brasil, como mostrado, por exemplo, em Cabral [2012].

Já para o problema de p-dispersão é importante manter determinada distância entre as instalações e não há distância para os clientes a ser respeitada. Um exemplo de solução pode ser dado conforme a Figura 2.5. A solução para o p-dispersão é a identificação das p instalações que proporcionem a maior distância possível para as instalações ativas mais próximas uma das outras. Em outras palavras, busca-se evitar a competição entre as instalações. Os clientes foram apresentados na Figura 2.5 apenas para ilustrar a diferença de uma solução do problema de p-dispersão e uma solução do problema do p-PLII, considerando que os clientes fossem alocados às instalações mais próximas.

Observe que, na solução apresentada na Figura 2.3, as instalações não estão distantes entre si, como é o caso da Figura 2.5, mas o somatório das distâncias dos



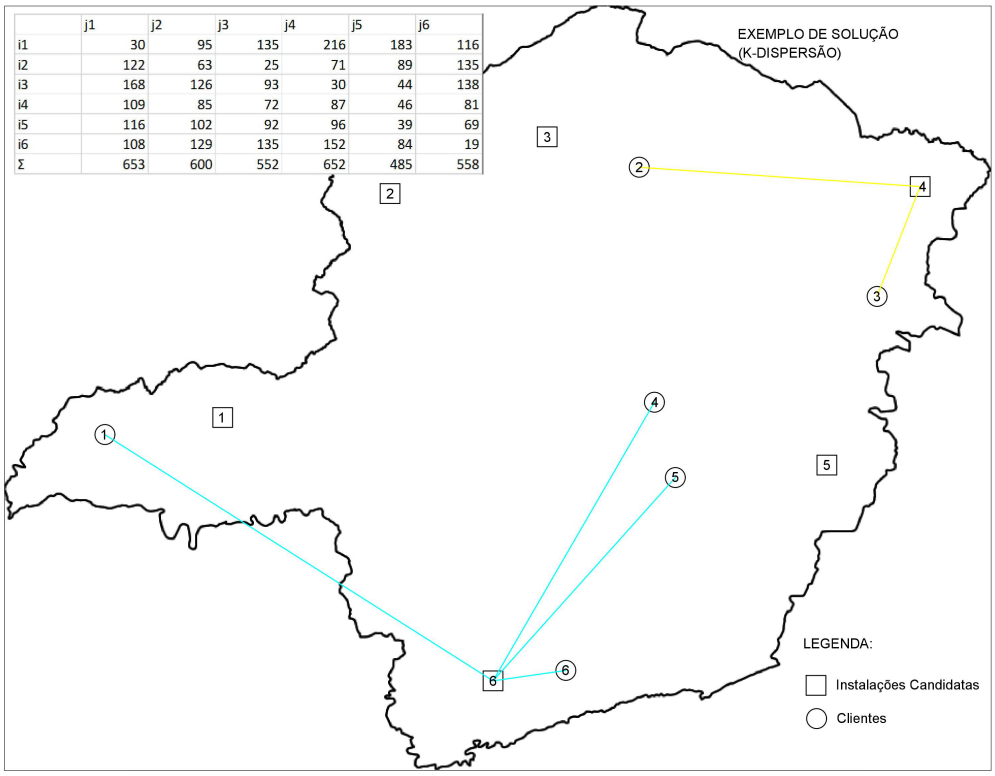
	j1	j2	j3	j4	j5	j6
j1	0	71	109	186	153	95
j2	71	0	42	134	130	126
j3	109	42	0	95	109	138
j4	186	134	95	0	74	165
j5	153	130	109	74	0	101
j6	95	126	138	165	101	0
Σ	614	503	493	654	567	625

Figura 2.4. Ilustração de um grafo para o problemas de p-Dispersão com 6 instalações candidatas.

clientes para as instalações é o maior possível. Outra observação interessante é que, por mais que os problemas sejam significativamente semelhantes, os mesmos também são completamente opostos para determinadas aplicações. Em situações de máxima cobertura, o problema de p-dispersão tenta encontrar as instalações que fiquem mais distantes entre si e, ao mesmo tempo, mais próximas dos clientes os quais elas irão atender. Nesse caso, buscariam-se instalações próximas aos clientes e dispersas entre si, o que é totalmente contrário ao que o problema p-PLII preconiza. Além disso, a imposição de que as instalações sejam dispersas entre si pode fornecer uma solução ruim e temerária para o problema p-PLII. Isso ocorreria em uma situação em que as instalações mais dispersas entre si fossem as mais próximas dos clientes.

2.5 Revisão Bibliográfica do p-PLII

O primeiro trabalho a tratar o problema p-PLII foi Goldman & Dearing [1975], conforme citado em Batta & Chiu [1988]. Após isso, Church & Garfinkel [1978], tam-



	j1	j2	j3	j4	j5	j6
i1	30	95	135	216	183	116
i2	122	63	25	71	89	135
i3	168	126	93	30	44	138
i4	109	85	72	87	46	81
i5	116	102	92	96	39	69
i6	108	129	135	152	84	19
Σ	653	600	552	652	485	558

Figura 2.5. Um exemplo de solução para o problema de p-dispersão, com p=2 envolvendo clientes.

bém segundo Batta & Chiu [1988], abordaram um problema de localizar as instalações o mais afastado possível dos clientes em uma rede, sem considerar a distância de uma instalação para outra, de maneira similar à apresentada nesse trabalho. Porém, o critério que cada cliente deve ser alocado à instalação aberta mais próxima não era tratado.

Partindo desses trabalhos, Batta & Chiu [1988] apresentam um modelo para o problema p-PLII considerando um veículo desagradável, como, por exemplo, um veículo de transporte de material perigoso em uma rede no plano euclidiano, no qual a soma das distâncias de um container até um centro de demanda é minimizada. Gopalan et al. [1990] desenvolveram um modelo matemático e uma heurística baseada em relaxação lagrangeana para gerar um conjunto equitativo de rotas para remessas de materiais perigosos e aplicaram o trabalho na cidade de Albany, no estado de Nova Iorque, EUA. O objetivo era determinar um conjunto de rotas que minimizassem o risco total das viagens e espalhassem o risco de forma equitativa entre as zonas da região geográfica, na qual a rede de transporte está inserida, quando várias viagens são necessárias da origem ao destino.

Melachrinoudis et al. [1995] abordaram um problema específico de localização de aterros sanitários. Os autores desenvolveram um modelo de otimização inteira mista multiobjetivo, cujo objetivo era minimizar os custos de transporte e os riscos ambientais da instalação de um aterro sanitário, buscando uma solução de compromisso entre estes diferentes critérios.

Uma abordagem de localização contínua de instalações indesejáveis dentro de uma determinada região geográfica, considerando os aspectos ambientais, é proposta por Fernández et al. [2000]. Ao mesmo tempo, este artigo inclui um novo modelo matemático, que minimiza a repulsão global dos habitantes da região, tendo em vista as preocupações ambientais que tornam algumas áreas não adequadas para a localização da instalação. Nesse trabalho são abordados instalações que não são prejudiciais à população, porém, ainda assim são consideradas desagradáveis. Cappanera et al. [2003] abordaram o problema de localizar simultaneamente instalações indesejadas e encaminhar materiais desagradáveis entre um conjunto de áreas construídas. Foram propostas duas heurísticas, derivadas de uma relaxação lagrangeana, em que a relaxação permite dividir o problema em dois subproblemas: um problema de roteamento e outro de localização. Rakas et al. [2004] abordam uma modelagem multiobjetivo com critérios técnicos de eliminação, para determinar a localização de instalações indesejadas nos Estados Unidos. Os critérios abordados são: habitação existente, planícies aluviais e zonas úmidas, aeroportos restritos, parques, áreas críticas da Baía de Chesapeake, bacias hidrográficas de reservatórios de água potável, locais históricos, recursos e distritos, áreas de utilização sensível e habitável/ áreas de plantas únicas de estado crítico ou preocupação do município.

De acordo com Chiang & Lin [2017], a maioria das pesquisas sobre a localização de instalações indesejadas concentrou-se na dispersão das instalações, sem considerar as interações entre instalações e clientes (ou outras instalações existentes e centros populacionais). No entanto, os modelos podem fugir da realidade, ao ignorarem os efeitos de instalações indesejadas em centros populacionais ou próximos a clientes.

A variante do p-PLII usada neste trabalho, em que um número fixo de instalações são selecionadas e a soma das distâncias dos clientes à instalação mais próxima é maximizada, foi proposta por Labbé et al. [2001]. Para a solução deste problema, Labbé et al. [2001] propõem um método *branch-and-cut* e um conjunto de desigualdades válidas para o problema. Após isso, compararam com o CPLEX. Porém, os autores só obtiveram resultados satisfatórios para resolver instâncias de pequena dimensão, com quantidade de clientes n até 75. Belotti et al. [2007] também propuseram um método *branch-and-cut* para resolver o p-PLII com um modelo de otimização linear binária, usando uma meta-heurística de busca tabu para os valores de entrada no

solver. Contudo, só conseguiram tempo satisfatório para resolver instâncias de média dimensão, nas quais p corresponde a 1/4 do número de clientes. Chiang & Lin [2017] também propuseram um método *branch-and-cut* para resolver o problema e criaram um novo modelo mais compacto para o problema, mas também só obtiveram resultados satisfatórios para instâncias de média dimensão, com até 200 clientes e 200 locais candidatos para escolher até 20 instalações.

Drezner et al. [2018a] desenvolveram um método exato baseado no problema de Weber, que leva em consideração a distância mínima entre uma instalação e um cliente. O intuito era minimizar o somatório das distâncias. Porém, cada instalação deve estar localizada a uma distância mínima dos clientes, pois as mesmas são indesejadas se muito próximas. Os resultados do método proposto nesse trabalho só foram satisfatórios para instâncias com um número reduzido de instalações a serem abertas.

Colmenar et al. [2016] abordaram o problema usando um método heurístico baseado na metaheurística GRASP (*Greedy Randomized Adaptive Search Procedure*). No método proposto, foi desenvolvido uma técnica de construção de soluções a serem instaladas baseada no benefício de cada instalação. O algoritmo foi executado para um conjunto de instâncias na qual o número de clientes varia de 200 à 450 e as p instalações seguiram a seguinte proporção: $n/2$ (consideradas difíceis), $n/4$ (médias) e $n/8$ (consideradas fáceis). O algoritmo encontrou soluções de boa qualidade em tempo hábil para a maioria das instâncias, sendo, no entanto, superado por métodos exatos em instâncias em que o número de p instalações é reduzido e a quantidade de clientes e locais é elevada. Drezner et al. [2018b] propuseram uma formulação matemática de duas variantes do problema e desenvolveram um algoritmo baseado no diagrama de Voronoy. Além disso, compararam a solução proposta com a encontrada pelo CPLEX para resolver um problema de localização de facilidades no plano. Porém, assim como os outros modelos matemáticos, só conseguiram resultados eficientes para um número significativamente pequeno de instalações.

Em um trabalho mais recente, Colmenar et al. [2018] abordam um problema multiobjetivo, no qual, além de considerar que as instalações fiquem o mais afastado possível dos clientes, levam em conta que as instalações devem se localizar dispersas entre si. O trabalho propõe, para a solução do problema, um Algoritmo Memético Multi-Objetivo (MOMA), que inclui mecanismos de cruzamento e mutação. Eles também estudaram a solução usando os algoritmos NSGA-II e SPEA2.

Lin & Guan [2018] desenvolveram um algoritmo baseado na meta-heurística *Particle Swarm Optimization* para o problema p-PLII. Nesse trabalho, foram criados mecanismos de mutação baseados na metaheurística Busca Tabu de curto prazo. Além disso, foi usada uma implementação de GRASP para intensificar a busca por meio

de um mecanismo de inserção e remoção. Os resultados obtidos foram equivalentes aos anteriores apresentados na literatura, porém, com o valor médio um pouco melhor. Herrán et al. [2018] propuseram um algoritmo baseado em *Variable Neighborhood Search*, usando estratégias de busca local mais rápidas, e utilizaram critérios para o controle do grau de perturbação, ligado à quantidade p de instalações. A construção de soluções foi baseada na técnica proposta por Colmenar et al. [2016]. O algoritmo proposto foi paralelizado e o problema foi resolvido para um conjunto de instâncias distintas e demonstraram, através de análises estatísticas, que a implementação proposta de VNS é superior aos demais trabalhos no estado da arte. Lancuna et al. [2019] apresentaram um algoritmo híbrido meta-heurístico para a solução do p-PLII que combina as técnicas GRASP e ILS. O algoritmo foi testado para um conjunto de instâncias da literatura e demonstrou desempenho equivalente aos demais algoritmos da literatura. Mladenović et al. [2019] apresentaram uma implementação de VNS que explora uma vizinhança de intercâmbio, além da técnica de *less is more*, para tratar o p-PLII.

A Tabela 2.5 apresenta os principais trabalhos relacionados ao problema abordado. Na primeira coluna são apresentados os autores, na segunda coluna em que ano os trabalhos dos autores foram publicados, na terceira coluna qual foi o método utilizado no artigo, na quarta e quinta coluna é sinalizado se o problema é mono ou multi objetivo e na sexta e última coluna é informado se o trabalho está associado a uma aplicação prática.

Tabela 2.3. Principais trabalhos relacionados ao p-PLII até o momento

Autor	**Ano**	**Método**	**Mono-objetivo**	**Multiobjetivo**	**Foi Aplicado**
Goldman & Dearing [1975]	1975	Modelo Matemático	x		
Church & Garfinkel [1978]	1978	Modelo Matemático em uma rede	x		
Batta & Chiu [1988]	1988	Modelo de Transporte de Resíduos Perigosos	x		
Gopalan et al. [1990]	1990	Modelo + Heurística baseados em Relaxação Lagrangeana	x		x
Tamir [1991]	1991	Modelagem Matemática	x		
Melachrinoudis et al. [1995]	1995	Modelo de Otimização Inteira Mista		x	x
Fernández et al. [2000]	2000	*Branch and Bound*	x		
Labbé et al. [2001]	2001	*Branch and Cut*	x		
Cappanera et al. [2003]	2003	*Branch and Bound + Bundle Method*	x		
Rakas et al. [2004]	2004	Modelo Matemático		x	x
Belotti et al. [2007]	2007	Branch and Cut + Busca Tabu (Entrada)	x		
Colmenar et al. [2016]	2016	Heurística GRASP com Busca Local	x		
Chiang & Lin [2017]	2017	Modelo Compacto de Otimização	x		
Drezner et al. [2018a]	2018	Método Exato "*Big Arc Small Arc*"	x		
Lin & Guan [2018]	2018	Heurística PSO (Enxame de Partículas)	x		
Colmenar et al. [2018]	2018	Algoritmo Memético Multi-Objetivo (MOMA)		x	
Drezner et al. [2018b]	2019	Diagrama de Vonoroi	x		
Lancuna et al. [2019]	2019	Heurística Híbrida GRASP/ILS para o p-PLII	x		
Mladenović et al. [2019]	2019	Heurística *Less is more approach*: VNS	x		
Herrán et al. [2018]	2020	Heurística VNS	x		

2.6 Resumo do Capítulo

Neste capítulo foi apresentada a descrição formal do problema de localização de instalações indesejadas (p-PLII). Foram descritos problemas semelhantes e destacadas as principais diferenças entre eles. Apresentou-se o problema clássico das p-Medianas, que é o problema de localização de p instalações mais conhecido. Descreveu-se também o problema de p-dispersão, que é um dos problemas mais abordados para o caso de localização de instalações indesejadas. Percebeu-se que, apesar de terem características semelhantes, os problemas são distintos e poderiam gerar soluções ruins, caso fossem confundidos para determinadas aplicações. Ao final do capítulo, foi apresentada uma revisão bibliográfica para o p-PLII. Notou-se que o trabalho recebeu ampla atenção em pesquisas nos últimos anos e que as principais técnicas para encontrar boas soluções para o problema têm sido as meta-heurísticas.

Capítulo 3

Fundamentação Teórica

Neste capítulo será apresentada a fundamentação teórica usada neste trabalho. Primeiramente, será dada uma introdução sobre o que são métodos meta-heurísticos na Seção 3.1. Após isso, será apresentada a teoria que envolve as meta-heurísticas usadas neste trabalho de uma maneira geral. Inicia-se com a descrição da construção gulosa por meio da técnica de inserção mais barata na Seção 3.2. Parte-se para a descrição da meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) na Seção 3.3. Após isso, faz-se a descrição dos procedimentos da meta-heurística *Iterated Local Search* (ILS) na Seção 3.4.

3.1 Métodos Meta-heurísticos

Uma heurística pode ser entendida como um método de resolução de problemas que encontra boas soluções em tempo computacional viável, porém, sem a garantia de ter encontrado um ótimo global. De acordo com Glover & Kochenberger [2006], meta-heurísticas são métodos de resolução que orquestram uma interação entre procedimentos de melhoria local e estratégias de alto nível para evitar a parada prematura em ótimos locais e realizar uma busca robusta de um espaço de solução.

Em geral, as meta-heurísticas possuem duas fases principais: a fase construtiva e a fase de refinamento. A fase construtiva de uma meta-heurística consiste em construir uma solução inicial elemento por elemento até compor uma solução completa. Já a fase de refinamento é o momento no qual a solução construída é melhorada, por exemplo, por meio de uma busca local.

Para ilustrar como uma meta-heurística funciona, considere a localização de p instalações em um universo de n candidatas. A construção de uma solução pode ser feita ou com uma escolha aleatória de p instalações ou escolhendo uma por uma as

instalações mais distantes dos clientes, até completar uma solução com p instalações. Já um refinamento poderia ser feito alterando o *status* de instalações que estavam inicialmente desativadas para a situação de ativadas, bem como de instalações que estavam ativadas para desativadas, de modo a se determinar uma solução melhor do que a solução inicial. Caso a nova solução seja melhor do que a solução construída, pode-se dizer que ocorreu um refinamento na solução construída.

Em um espaço de busca muito amplo, com muitas instalações candidatas, é extremamente difícil escolher as p melhores instalações existentes. O trabalho de uma meta-heurística é buscar as melhores instalações, seguindo os seus critérios de construção e refinamento.

Nesse trabalho, para a construção da solução inicial, foram exploradas a heurística construtiva por inserção mais barata e a fase construtiva da meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP). Já o refinamento foi feito usando a meta-heurística *Iterated Local Search* (ILS).

3.2 Heurística Construtiva - Inserção Mais Barata

A abordagem de construir uma solução inicial viável para um problema de otimização combinatória usando a inserção mais barata foi proposta por Golden et al. [1985] e posteriormente adaptada por Salhi & Nagy [1999]. Ambos abordaram a técnica para o problema de roteamento de veículos.

Em linhas gerais, o método funciona da seguinte maneira: dado um conjunto de candidatos, seleciona-se um deles para pertencer à solução. Em seguida, é calculado qual é o próximo candidato, que forneça o maior benefício para a função objetivo, para compor a solução. Insere-se esse candidato na solução e repete-se esse procedimento até que a solução esteja completa. Pode-se, ao invés de escolher o primeiro candidato aleatoriamente, optar por escolher de maneira gulosa. O método de construção por inserção mais barata é um método guloso, no qual são selecionados individualmente os candidatos que causem o maior benefício à função objetivo.

Dethloff [2001] utilizou-se das ideias originárias dos artigos acima mencionados e desenvolveu uma construção por inserção mais barata de fácil entendimento, na qual o primeiro cliente escolhido atua como cliente "semente". Primeiramente, uma rota do depósito para o cliente semente e uma rota de volta ao depósito é construída. Em seguida, para todos os clientes não roteados restantes, é calculado o valor de um critério de inserção para todas as posições de inserção possíveis. Então, é realizada a melhor das inserções viáveis (em relação à capacidade do veículo). Essa fase é repetida até que

Algoritmo 1: Fase Construtiva - Inserção mais barata

```
1  S ← ∅
2  r ← SelecionaPrimeiroCandidato(J)
3  J ← J \ {r}
4  S ← S ∪ {r}
5  enquanto Critério de parada não for satisfeito faça
6  |  r ← SelecionaCandidatoQueCausaMaiorBeneficio(J)
7  |  S ← S ∪ {r}
8  |  J ← J \ {r}
9  fim
10 retorna S
```

nenhum outro cliente possa ser inserido na rota.

O pseudocódigo do procedimento de inserção mais barata é apresentado no Algoritmo 1. Na linha 1, representa-se a solução S como um conjunto vazio. Na linha 2, o elemento r recebe o primeiro elemento a fazer parte da solução, derivado da função que seleciona o primeiro candidato. Essa função seleciona um candidato para compor a solução em construção por meio de algum critério definido anteriormente. Nas linhas 3 e 4, este primeiro elemento é retirado do conjunto J de candidatos e inserido ao conjunto S de solução. A partir da linha 5 até a linha 9, é realizada a estrutura de repetição, que insere os elementos que resultam em maior benefício à função objetivo, retirando os mesmos do conjunto dos candidatos. Ao final, é retornada a solução construída (linha 10).

3.3 GRASP

O *Greedy Randomized Adaptive Search Procedure* (GRASP) é uma metaheurística de múltiplos começos, em que cada iteração consiste basicamente de duas fases: (i) fase de construção; e (ii) fase de busca local. A fase de construção fornece uma solução viável. A vizinhança desta solução é investigada até que um mínimo local seja encontrado durante a fase de busca local. Após várias construções, a melhor solução é mantida [Feo & Resende, 1995]. A efetividade de um procedimento de busca local depende de vários aspectos, como a estrutura de vizinhança, a técnica de busca local, a avaliação rápida da função de custo dos vizinhos e a própria solução inicial [Feo & Resende, 1995]. A fase de construção desempenha um papel muito importante em relação a este último aspecto, construindo soluções de partida de alta qualidade para a fase de refinamento.

Algoritmo 2: Algoritmo GRASP

```
1  S* ← ∅
2  enquanto Número de construções não for atingido faça
3  |  S ← ∅
4  |  LRC ← SelecionaCandidatosMaisAptos(J)
5  |  enquanto Critério de parada não for satisfeito faça
6  |  |  r ← EscolheElementoAleatoriamente(LRC)
7  |  |  LRC ← LRC \ {r}
8  |  |  S ← S ∪ {r}
9  |  fim
10 |  S' ← RealizaBuscaLocal(S)
11 |  se f(S') > f(S*) então
12 |  |  S* ← S'
13 |  fim
14 fim
15 retorna S*
```

O método funciona da seguinte forma: a fase construtiva é criada a partir da avaliação de todos os candidatos pertencentes à lista de candidatos (LC), usando uma função gulosa g, normalmente a própria função de avaliação do algoritmo, e determina-se os valores g_{min} e g_{max}, que representam o candidato menos apto a fazer parte da solução e o candidato mais apto a compor a solução respectivamente. Após isso, a lista restrita de candidatos (LRC) é montada, de modo a satisfazer a expressão:

$$LRC = \{t \in \mathcal{C} \mid g(t) \geq g_{\min} + \alpha\,(g_{\max} - g_{\min})\}\,. \tag{3.1}$$

A LCR é formada pelos melhores elementos, ou seja, aqueles cuja incorporação à solução parcial corrente resulta em menores custos incrementais (este é o aspecto guloso do algoritmo). O elemento a ser incorporado na solução parcial é selecionado aleatoriamente na LRC (este é o aspecto probabilístico da heurística). Uma vez que o elemento selecionado é incorporado à solução parcial, a lista de candidatos é atualizada e os custos incrementais são reavaliados (este é o aspecto adaptativo da heurística).

Como o tempo computacional não varia muito de iteração para iteração, ele é previsível e aumenta linearmente com o número de iterações. Consequentemente, quanto maior o número de iterações, maior será o tempo computacional e provavelmente melhor será a solução encontrada [Feo & Resende, 1995].

O pseudo código do procedimento da metaheurística GRASP é apresentado no Algoritmo 2. Na linha 1, inicia-se o vetor que armazena a melhor solução com um vetor nulo. Nas linhas 2 até a linha 14, ocorre a estrutura de repetição que constrói e refina

as soluções até que determinado número de construções seja atingido. Iniciando-se na linha 3, na qual o vetor que armazena a solução corrente inicia-se com um vetor nulo. Após isso, na linha 4, a lista restrita de candidatos (LRC) recebe um candidato proveniente da função que seleciona candidatos mais aptos. Essa função funciona da seguinte maneira: dado um determinado valor de (α) a função seleciona candidatos de maneira mais ou menos restritiva. Feito isso, da linha 5 até a linha 9 é realizado o procedimento construtivo da solução. Na linha 6 o elemento r recebe um elemento escolhido pela função que escolhe o elemento aleatoriamente. Tal função, seleciona um elemento aleatoriamente da LRC para fazer parte da solução. Na linha 7 este elemento é retirado da LRC e na linha 8 o mesmo é adicionado a solução corrente.

Finalizado esse procedimento de composição de uma solução com a quantidade de elementos desejada, o algoritmo realiza a busca local na linha 10 por meio de uma função que encontra um ótimo local. A verificação se o valor de função objetivo da solução corrente é melhor do que o melhor valor encontrado anteriormente é feito na linha 11. Se o valor for melhor, na linha 12 o vetor que armazena a melhor solução até o momento é atualizado, recebendo a solução corrente. Esse procedimento é feito para cada construção até atingir o critério de parada. Ao final, na linha 15 é retornada a melhor solução encontrada.

O método do GRASP é realizado de modo que haja um equilíbrio entre os fatores de construção aleatório e guloso. Para isso, a lista restrita de candidatos é criada usando um parâmetro α, que varia de 0 a 1. Esse α é um fator multiplicativo na função que cria a LRC. O parâmetro $\alpha = 0$ permite que qualquer um dos elementos pertença a LRC, sendo completamente aleatório. Já o $\alpha = 1$ deve permitir somente a entrada do elemento que traga maior benefício a função objetivo dentre os elementos que ainda não entraram na LRC, sendo totalmente guloso.

Após a construção, o algoritmo parte para a busca local na solução que foi construída, na qual pode ser aplicada uma descida completa na vizinhança, para garantir que o algoritmo encontrou um ótimo local para a vizinhança explorada. Finalmente, o algoritmo compara as soluções de todas as construções após a descida. Ou seja, compara todos os ótimos locais e retorna o melhor dentre eles. A busca local pode ser implementada usando uma estratégia de "melhor vizinho" ou de "primeiro de melhora". No caso da estratégia de melhor vizinho, todos os vizinhos são investigados e a solução atual é substituída pelo melhor vizinho. No caso de uma estratégia de primeiro de melhora, a solução atual é movida para o primeiro vizinho cujo valor de função objetivo é melhor que o da solução atual. De acordo com Feo & Resende [1995], na prática, observa-se, em muitas aplicações, que muitas vezes ambas as estratégias levam à mesma solução final, mas em tempos computacionais menores quando a estratégia

de primeiro de melhora é usada. Observa-se também que a convergência prematura para um mínimo local, não global, é mais provável de ocorrer com uma estratégia de melhor vizinho [Hansen & Mladenović, 2006].

3.4 ILS

A metaheurística *Iterated Local Search* (ILS), proposta por Lourenço et al. [2003], consiste em explorar o espaço de busca por meio de perturbações em ótimos locais. Os desenvolvedores desse método buscavam uma técnica que fosse adaptável a diferentes tipos de problemas, garantindo a sua eficiência [Lourenço et al., 2003].

O método parte do princípio de que uma maneira de tentar melhorar o valor da função objetivo é reiniciar a busca local de outro ponto de partida. Uma possível alternativa seria selecionar aleatoriamente o novo ponto de partida. Contudo, usar essa abordagem pode ser demasiadamente custosa, visto que o tamanho do espaço de busca de soluções tende a aumentar de acordo com o tamanho das instâncias. Isso sugere que não parece ser interessante reiniciar aleatoriamente a busca local para grandes instâncias, devido ao custo computacional. Em contrapartida, apenas a busca local, sem a perturbação, não consegue explorar suficientemente o espaço de busca, ocasionando um método que fica preso em ótimos locais. Baseado nisso, o ILS foi desenvolvido com um procedimento de busca estocástica que combina as duas técnicas, ou seja, o reinício em um novo ponto e a aplicação de busca local para a determinação de uma nova solução.

Para conseguir explorar melhor o espaço de busca e não ser completamente aleatório, o ILS explora as vizinhanças e gera perturbações que permitem ir para vizinhanças intermediárias [Lourenço et al., 2003]. Neste caso, é necessária uma estrutura de vizinhança bem definida, pois vizinhos mais próximos conduzem a reinícios da busca local em regiões mais promissoras. Contudo, buscas locais que usam sempre os vizinhos mais próximos tendem a encontrar os mesmos ótimos locais.

Em linhas gerais, o método funciona da seguinte maneira: a primeira etapa do algoritmo parte de uma solução inicial S_o, na qual será realizada uma busca local. Após isso, é feita uma perturbação na solução corrente encontrada S, resultando em S'. Uma busca local é realizada em s', resultando em S''. A melhor solução corrente S^* pode se atualizada ou não pela solução corrente S'' a partir da avaliação de um dado critério de aceitação. Caso a solução corrente S'' não seja aceita, o nível de perturbação é aumentado e o procedimento é realizado novamente, até que essa perturbação atinja um determinado nível. Este procedimento de busca local conduz a uma boa exploração

do espaço de busca, desde que as perturbações não sejam muito pequenas, nem grandes demais. Se forem muito pequenas, muitas vezes retornará para S e poucas novas soluções S' serão exploradas. Se, ao contrário, as perturbações forem muito grandes, soluções S' serão aleatórias e não haverá direção de caminhada na amostragem e o procedimento será semelhante a um algoritmo do tipo de reinicialização aleatória [Lourenço et al., 2003]. Uma ilustração do comportamento do procedimento ILS global é mostrada na Figura 3.1.

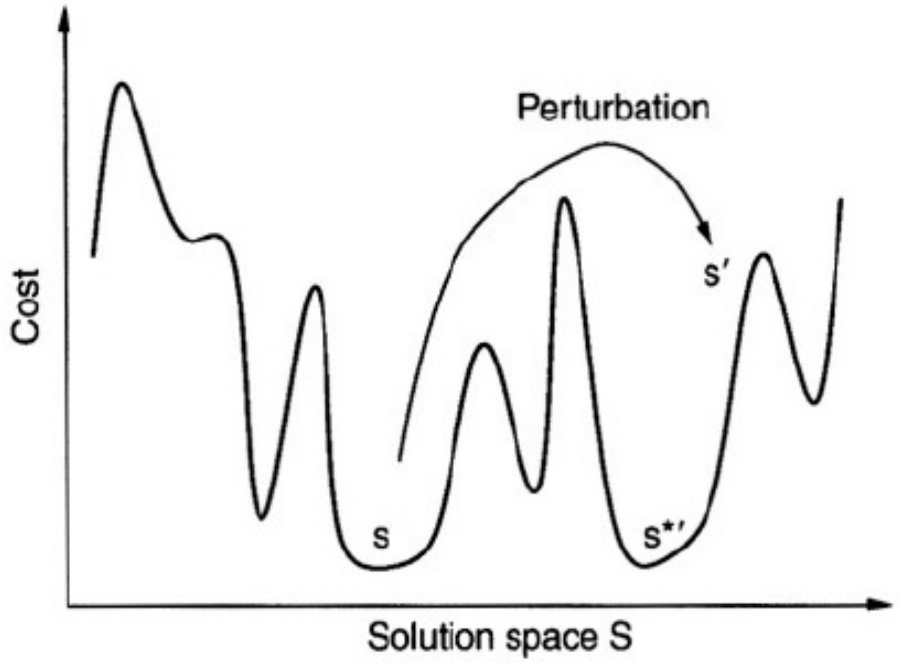


Figura 3.1. Procedimento de exploração do espaço de busca por meio de perturbações [Lourenço et al., 2003]

O procedimento possui duas fases basicamente, intensificação e diversificação. A intensificação ocorre fazendo uma busca local na solução corrente. Já a fase de diversificação consiste em mudar a região de busca, na qual a solução é obtida, aumentando-se gradativamente a quantidade de perturbações na solução ótima local corrente. Por fim, o critério de parada é atingido quando alcança-se o nível máximo de perturbação estabelecido, conforme está representado no Algoritmo 3.

Nas linhas 1 e 2, os vetores que armazenam a solução corrente e a melhor solução iniciam-se com um vetor nulo respectivamente. Já na linha 3, o vetor que armazena a solução corrente recebe uma solução derivada da função que gera uma solução inicial. Esta função tem a finalidade de gerar uma solução inicial viável para o algoritmo. Na linha 4 é realizada uma busca local na solução corrente por meio da função que realiza esse procedimento até encontrar um ótimo local. Na linha 5 o nível de perturbação

Algoritmo 3: Algoritmo ILS

```
1  S ← ∅
2  S* ← ∅
3  S ← GeraSolucaoInicial()
4  S ← BuscaLocal(S)
5  Nivel_Perturba ← 1
6  enquanto Nivel_Perturba < Perturba_Máx faça
7  |  S' ← Perturba(S,Nivel_Perturba)
8  |  S" ← BuscaLocal(S')
9  |  se f(S") > f(S*) então
10 |  |  S* ← S"
11 |  |  S ← S*
12 |  |  Nivel_Perturba ← 1
13 |  fim
14 |  senão
15 |  |  Nivel_Perturba ← Nivel_Perturba + 1
16 |  fim
17 fim
18 retorna S*
```

inicia-se com o valor de 1. A partir da linha 6 até a linha 17 é realizada a estrutura de repetição, que faz o refinamento da solução inicial. Na linha 7 ocorre uma perturbação na solução corrente por meio de uma função que realiza esse procedimento. Esta função realiza a perturbação por meio de movimentos na solução corrente, gerando uma solução perturbada. Na linha 8 é realizada uma busca local na solução perturbada com a função que realiza esse procedimento.

A partir da linha 9 até a linha 16 é realizada a verificação se o valor de função objetivo da solução corrente é melhor que o valor de função objetivo da melhor solução armazenada anteriormente. Se a solução corrente for melhor, nas linhas 10 e 11 o vetor que armazena a melhor solução e o vetor de solução corrente inicial recebem a solução corrente e na linha 12 o nível de perturbação retorna para 1. Caso ocorra o contrário na linha 15 o nível de perturbação é incrementado em uma unidade. Esse procedimento é realizado até que um nível de perturbação máximo seja atingido. Por fim, na linha 18, retorna-se a melhor solução encontrada.

Capítulo 4

Metodologia

Neste capítulo, é apresentada a metodologia específica utilizada para encontrar os resultados dos algoritmos. Para buscar melhores soluções para o problema, foram explorados dois métodos distintos na geração da solução inicial. O primeiro aborda a construção pela fase construtiva do GRASP e o segundo utiliza a construção feita de forma gulosa, por meio da técnica de inserção mais barata. Descreve-se inicialmente como foi feita a representação da solução na Seção 4.1. A Seção 4.2 descreve como foi realizada a geração da solução inicial para os métodos explorados. Após isso, na Seção 4.3 é apresentada a função de avaliação utilizada. A vizinhança e os movimentos de exploração que foram utilizados são apresentados na Seção 4.4. A maneira como foi realizada a busca local é apresentada na Seção 4.5. O procedimento de perturbação característico da técnica ILS foi descrito na Seção 4.6. Ao final nas Seções 4.7 e 4.8 são apresentados os pseudocódigos dos algoritmos implementados.

4.1 Representação da Solução

A solução x é representada através de um vetor binário, em que o valor 1 em uma dada posição representa uma instalação ativa e o valor 0 representa uma instalação inativa. O vetor de solução sempre terá dimensão n, ou seja, a quantidade de instalações candidatas.

Um exemplo para uma solução x com $p = 3$ instalações abertas e $n = 6$ instalações candidatas é dado por:

$$x = [0, 0, 1, 0, 1, 1]$$

em que as posições 3, 5 e 6 representam as instalações abertas e, as demais, as instalações inativas.

4.2 Geração da Solução Inicial

Como mencionado previamente, neste trabalho foram exploradas duas maneiras de gerar a solução inicial. No primeiro método abordou-se a fase construtiva da meta-heurística GRASP adaptada ao problema. Já no segundo método explora-se a geração da solução inicial de maneira gulosa por meio da técnica de inserção mais barata.

4.2.1 Geração via fase construtiva do GRASP

Foram geradas num_Constr construções. A fase construtiva do algoritmo é feita através de uma lista restrita de candidatos, que varia de acordo com o α, da seguinte forma. As instalações foram ordenadas de acordo com a soma da distância de cada cliente para esta instalação. Em seguida, usa-se a função F_grasp, dada por:

$$F\_grasp = menor + \alpha(maior - menor) \tag{4.1}$$

como um ponto de corte entre as instalações candidatas e as instalações que não entram na lista restrita de candidatos de acordo com seus valores de função objetivo. Os candidatos que pertencem à lista restrita de candidatos são sorteados aleatoriamente. O valor de α mais próximo de 1 torna a lista mais restrita e, quanto mais próximo de 0, menos restrita, sendo que α=1 representa uma solução gulosa e α=0 uma solução completamente aleatória.

Note que, caso o valor de α seja 0, todos os elementos da lista de candidatos fazem parte da lista restrita de candidatos. Caso o valor de α seja 1, apenas um elemento fará parte da lista restrita de candidatos. Neste caso, o elemento cuja soma das distâncias para todos os cliente for o maior dentre todos os candidatos.

O algoritmo 4 realiza a fase construtiva do GRASP. O procedimento passo a passo é apresentado a seguir. Na linha 1, inicia-se a variável de melhor solução com valor nulo. A partir da linha 2, começa o *loop* que irá construir várias soluções iniciais. Na linha 3, inicializa-se o vetor que armazena a solução construída, representado como um conjunto inicialmente vazio. Na linha 4, a lista restrita de candidatos recebe os candidatos advindos da função que seleciona os candidatos mais aptos do conjunto J. Da linha 5 até a 9 ocorre o procedimento de escolha das p instalações que pertençam a LRC até completar a solução. Na linha 6, é escolhido um candidato aleatoriamente na LRC. Na linha 7, este candidato é removido da LRC. Na linha 8, a solução recebe esta instalação candidata e o *loop* encerra-se na linha 9. Após isso, na linha 10, ocorre a verificação se a solução construída é melhor do que a melhor solução encontrada anteriormente. Se for verdadeiro, atualiza-se a melhor solução na linha 11. O *loop* que

Algoritmo 4: Fase Construtiva GRASP

```
1  S* ← 0
2  para iter = 1 até numConstr faça
3  |  S ← ∅
4  |  LRC ← SelecionaCandidatosMaisAptos(J, α)
5  |  para iter = 1 até p faça
6  |  |  r ← EscolheElementoAleatoriamente(LRC)
7  |  |  LRC ← LRC \ {r}
8  |  |  S ← S ∪ {r}
9  |  fim
10 |  se f(S) > f(S*) então
11 |  |  S* ← S
12 |  fim
13 fim
14 retorna S*
```

constrói $numConstr$ é encerrado na linha 13 e a melhor construção retornada na linha 14.

4.2.2 Geração via inserção mais barata

A fim de encontrar uma maneira mais rápida e eficiente de gerar uma boa solução inicial, foi explorada a técnica de inserção mais barata para construir a solução inicial do algoritmo. A técnica explorada nesse trabalho foi adaptada ao problema tratado, de maneira que diante das instalações ordenadas de acordo com a soma da distância de cada cliente para as instalações, seleciona-se a instalação mais distante para compor a solução. Em seguida, é verificada qual instalação pertencente ao conjunto das instalações candidatas J traz menos prejuízo à função objetivo da solução parcial em construção. Esse procedimento é feito até que complete-se a solução com as p instalações necessárias. O pseudocódigo do procedimento pode ser visto no algoritmo 5.

Na linha 1, inicializa-se o vetor que armazena a solução S como um conjunto vazio. Na linha 2, o elemento r recebe a primeira instalação a fazer parte da solução, derivado da função que seleciona a instalação mais distante dos clientes. Nas linhas 3 e 4, esta primeira instalação é retirada do conjunto J de candidatos e inserida ao conjunto S de solução. A partir da linha 5 até a linha 9, é realizada a estrutura de repetição que insere as instalações candidatas que causam maior benefício à função objetivo, retirando as mesmas do conjunto das candidatas até completar as p instalações necessárias para o problema. Ao final é retornada a solução construída na linha 10.

Algoritmo 5: Fase Construtiva - Inserção mais barata

```
1  S ← ∅
2  r ← SelecionaInstalacaoMaisDistante(J)
3  J ← J \ {r}
4  S ← S ∪ {r}
5  para j = 1 até p − 1 faça
6  |  r ← SelecionaCandidatoQueCausaMaiorBeneficio(J)
7  |  S ← S ∪ {r}
8  |  J ← J \ {r}
9  fim
10 retorna S
```

4.3 Função de Avaliação

A função de avaliação está apresentada pela própria função objetivo do problema, ou seja:

$$f_{ava} = \sum_{i \in I} \min\{d_{ij} : j \in S\} \tag{4.2}$$

Nesta expressão, S representa o conjunto de instalações abertas, associado a um vetor x binário de solução, conforme descrito na Seção 4.1. Para cada cliente é verificada a distância à instalação aberta mais próxima. Esse valor é somado para todos os clientes para se obter a soma das distâncias.

4.4 Movimento de Exploração

O movimento de exploração do espaço de busca de soluções é dado pelo movimento de troca. A troca é feita dois a dois, fechando uma instalação aberta, substituindo um 1 por um 0, e abrindo uma que esteja fechada, alterando um 0 por um 1.

Por exemplo, considere a solução x^1, em que a instalação 1 está fechada e a instalação 3 aberta:

$$x^1 = [0, 0, 1, 0, 1, 1].$$

Então, após uma troca, a instalação 1 tornou-se aberta e a instalação 3 tornou-se fechada. Assim, a solução resultante após esta troca será a solução:

$$x^2 = [1, 0, 0, 0, 1, 1].$$

Algoritmo 6: Busca Local Primeiro de Melhora

```
1  S_b ← RecebeSolucaoInicial(S)
2  S*_b ← ∅
3  S_b ← Embaralha(S_b)
4  posA ← 0
5  Melhorou ← Falso
6  enquanto Melhorou == Falso e posA < n − 1 faça
7      posB ← posA
8      enquanto Melhorou == Falso e posB < n faça
9          se S_b[posA] ≠ S_b[posB] então
10             S_b ← Troca(S_b[posA] por S_b[posB] e S_b[posB] por S_b[posA])
11             se f(S_b) > f(S*_b) então
12                 S*_b ← S_b
13                 Melhorou ← Verdadeiro
14             fim
15             senão
16                 S_b ← Destroca(S_b)
17             fim
18         fim
19         posB ← posB + 1
20     fim
21     posA ← posA + 1
22 fim
23 retorna S*_b
```

4.5 Busca Local

A vizinhança pode ser explorada com duas estratégias distintas: melhor vizinho (*best improvement*), em que todas as soluções vizinhas são examinadas para identificar a melhor; e o primeiro de melhora (*first improvement*), que tenta evitar a complexidade de tempo de explorar toda a vizinhança, realizando o primeiro movimento de melhoria encontrado durante a exploração da vizinhança correspondente. Portanto, a ordem em que os vizinhos são inspecionados tem uma influência significativa na busca por meio do primeiro de melhora [Colmenar et al., 2016]. Por isso, utilizou-se a estratégia de busca de primeiro de melhora com um procedimento que embaralha a ordem em que os vizinhos são verificados. O procedimento de busca local pode ser representado pelo pseudocódigo do algoritmo 6.

Na linha 1, o vetor S_b, recebe a solução inicial. Já na linha 2, o vetor que armazena a melhor solução da busca local, recebe um valor nulo em todas as posições. O vetor de busca local recebe a solução embaralhada na linha 3. Na linha 5 a variável de controle

$Melhorou$ é inicializada como $Falso$. Na linha 4, a variável que armazena qual posição do vetor será investigada ($posA$) recebe a posição inicial 0. Na linha 6, inicia-se a estrutura de repetição que fará a busca local até que não ocorra mais melhoria e todas as posições da variável $posA$ já tenham sido investigadas. Na linha 7, a variável $posB$ recebe a variável $posA$. Dando sequência na estrutura de repetição, na linha 8, inicia-se o *loop* que fará a busca local até que não ocorra mais melhoria e todas as posições da variável $posB$ já tenham sido investigadas, completando assim, todas as trocas da vizinhança. Na linha 9, é verificado se o valor contido no vetor de busca local na $posA$ é diferente do valor contido na $posB$. Em caso de falso, apenas incrementa-se a variável $posA$, na linha 19. Já em caso verdadeiro, na linha 10, ocorre a troca no vetor. O valor no vetor na $posA$ é substituído pelo valor no vetor na $posB$ e o valor no vetor na $posB$ é substituído pelo valor no vetor na $posA$. Na linha 11, verifica-se se o valor de função objetivo do novo vetor resultante da troca é melhor que o anterior. Se for melhor, atualiza-se o vetor que armazena a melhor solução na linha 12 e a variável $Melhorou$ é confirmada com a resposta $Verdadeiro$ e ao final incrementa-se a variável $posB$ para repetir o processo desde a linha 8. Em caso contrário, é desfeita a troca e o vetor passa a ser como anteriormente, incrementa-se a variável $posB$ e repete-se o processo desde a linha 8 . Após todas as posições $posB$ terem sido investigadas, incrementa-se a $posA$ na linha 21. Esse procedimento é realizado até que todas as posições tenham sido investigadas. Por fim, retorna-se a melhor solução encontrada na linha 23.

É importante notar que quando ocorre uma melhoria na solução o vetor que armazena a solução é atualizado e o algoritmo retorna ao inicio da estrutura de repetição. Ou seja, cada melhoria na solução pode ser entendida como um passo na caminhada até o ótimo local. Por isso, pode-se dizer que ao final de todo o procedimento tem-se uma descida completa.

4.6 Procedimento de Perturbação

Para Blum et al. [2005] o procedimento da perturbação é muito importante. Uma perturbação muito pequena pode não permitir que o sistema escape do mínimo local recém-encontrado. Por outro lado, uma perturbação muito grande tornaria o algoritmo semelhante a uma busca local de reinício aleatório. A perturbação geralmente não é determinística para evitar ciclos, sua característica mais importante é a força, definida como a quantidade de mudanças infligidas na solução atual. A força pode ser fixa ou variável. No primeiro caso, a distância entre a solução inicial e a perturbada é mantida constante, independentemente do tamanho do problema. No entanto, uma

Algoritmo 7: Perturbação ILS

```
1  S_p ← RecebeSolucaoInicial(S_b)
2  Nivel_Perturba ← RecebeNivelPerturbacao(ILS)
3  i ← 0
4  para i < 2 × Nivel_Perturba faça
5      posA ← SorteiaPosicao(S_p)
6      posB ← posA
7      enquanto S_p[posA] ≠ 1 faça
8          posA ← SorteiaPosicaoAtiva(S_p)
9      fim
10     enquanto S_p[posB] ≠ 0 faça
11         posB ← SorteiaPosicaoInativa(S_p)
12     fim
13     S_p[posA] ← 0
14     S_p[posB] ← 1
15     i ← i + 1
16 fim
17 retorna S_p
```

perturbação de força variável é em geral mais eficaz, uma vez que foi verificado experimentalmente que, na maioria dos problemas, quanto maior o tamanho da instância do problema, maior deve ser a força da perturbação. Um mecanismo mais sofisticado consiste em alterar a força da perturbação de forma adaptativa. Por exemplo, a força pode ser aumentada quando a diversificação for mais necessária ou diminuída quando a intensificação parecer ser mais preferível [Blum et al., 2005].

Diante disso, optou-se por utilizar uma perturbação de força variável. Essa força foi representada como níveis de perturbação. Para cada nível de perturbação foram realizadas o dobro de quantidade de trocas na solução para vizinhança explorada. O procedimento de perturbação pode ser representado pelo pseudocódigo do algoritmo 7.

Na linha 1, o vetor de perturbação recebe a solução derivada da busca local. Na linha 2, a variável (Nível_Perturba) recebe o nível de perturbação a ser utilizado derivado do ILS. Na linha 3, a variável i que controla a estrutura de repetição inicia-se com o valor nulo. A estrutura de repetição que é delimitada pelo nível de perturbação multiplicado por 2, inicia-se na linha 4. Primeiramente, na linha 5, a variável que armazena a posição a ser investigada ($posA$) recebe uma posição sorteada aleatoriamente. Apenas para não iniciar-se vazia, a variável $posB$ recebe a $posA$, na linha 6. Da linha 7 até a linha 9 ocorre a estrutura de repetição que garante que será sorteada uma posição que contenha o valor 0. Já da linha 10 até a linha 12 ocorre a estrutura de repetição que garante que será sorteada uma posição que contenha o valor 1. Após isso,

Algoritmo 8: Algoritmo Híbrido GRASP-ILS

```
1  S* ← 0
2  S ← RecebeSolucaoInicial(GRASP)
3  S ← BuscaLocal(S)
4  Nivel_Perturba ← 1
5  enquanto Nivel_Perturba < Perturba_Max faça
6  |  S' ← Perturba(S)
7  |  S" ← BuscaLocal(S')
8  |  se f(S") > f(S*) então
9  |  |  S* ← S"
10 |  |  S ← S*
11 |  |  Nivel_Perturba ← 1
12 |  fim
13 |  senão
14 |  |  Nivel_Perturba ← Nivel_Perturba + 1
15 |  fim
16 fim
17 retorna S*
```

ocorrem as trocas das posições selecionadas nas linhas 13 e 14 e a variável de controle i é incrementada na linha 15. Por fim, na linha 17 retorna-se a solução perturbada.

4.7 Algoritmo Híbrido GRASP-ILS

Para encontrar melhores soluções para o p-PLII, inicialmente foi proposto um Algoritmo Híbrido GRASP/ILS. Na primeira fase o algoritmo utiliza a fase construtiva do GRASP, na qual cria-se uma lista restrita de candidatos, em que a cada iteração uma instalação é escolhida para compor a solução. Ao final foram feitas um número fixo de gerações, dado por num_Constr, e manteve-se apenas a melhor solução construída encontrada. Já na segunda fase do algoritmo é realizada a busca local por meio do algoritmo ILS, usando a primeira solução de melhora. Ou seja, o algoritmo realiza os movimentos de troca na vizinhança da solução corrente até que a primeira melhoria na função objetivo ocorra. Quando o algoritmo fica preso em um ótimo local é realizada uma perturbação que vai sendo incrementada em diferentes níveis, conforme a dificuldade de encontrar outro ótimo local, melhor que o anterior. Para cada nível de perturbação é realizado o dobro de trocas na solução pelo algoritmo.

A heurística híbrida GRASP-ILS é apresentada no Algoritmo 8. Na linha 1, inicia-se a variável de melhor solução com valor nulo. Na linha 2, o vetor de soluções, recebe a solução oriunda da fase construtiva do GRASP. Na linha 3, aplica-se uma

Algoritmo 9: Algoritmo Híbrido - Inserção mais Barata-ILS

```
1  S ← ∅
2  S ← RecebeSolucaoInicial(InsercaoMaisBarata)
3  S ← BuscaLocal(S)
4  Nivel_Perturba ← 1
5  enquanto Nivel_Perturba < Perturba_Max faça
6  |  S' ← Perturba(S)
7  |  S" ← BuscaLocal(S')
8  |  se f(S") > f(S*) então
9  |  |  S* ← S"
10 |  |  S ← S*
11 |  |  Nivel_Perturba ← 1
12 |  fim
13 |  senão
14 |  |  Nivel_Perturba ← Nivel_Perturba + 1
15 |  fim
16 fim
17 retorna S*
```

busca local nessa solução que foi recebida. Na linha 4, inicia-se o nível de perturbação com o valor de um. Na linha 5, inicia-se a estrutura de repetição delimitada pelo nível de perturbação máximo. Na linha 6, aplica-se uma perturbação na solução corrente. Na linha 7, é aplicada uma busca local na solução perturbada afim de chegar a um ótimo local. Da linha 8 até a linha 12, é realizada a verificação se a solução perturbada é melhor do que a melhor solução encontrada anteriormente. Se a solução perturbada for melhor, então, o vetor de soluções recebe esta solução e o nível de perturbação retorna para o valor de um (linha 11). Caso o contrário aconteça, incrementa-se o nível de perturbação (linha 14). Esse procedimento é realizado até que o nível de perturbação máximo seja atingido. Após isso, a melhor solução é retornada (linha 17).

Em linhas gerais, o algoritmo híbrido GRASP-ILS constrói uma solução através da fase construtiva do GRASP e volta procurando melhores soluções causando perturbações para escapar de ótimos locais. O algoritmo se torna mais aleatório a cada nível de perturbação que não encontra uma solução melhor. Quando o algoritmo encontra uma solução melhor, ele retorna ao primeiro nível de perturbação.

4.8 Algoritmo Híbrido Inserção Mais Barata-ILS

Apesar da construção da solução inicial pelo método proposto por Feo & Resende [1995] ser relativamente barato computacionalmente, ainda assim, são necessárias várias

construções iniciais para conseguir gerar uma boa solução para o refinamento. Com isso, foi proposta uma construção gulosa pelo método de inserção mais barata. Pois, além de ser um método mais rápido de realizar a construção, visto que só uma construção é suficiente, a técnica analisa a vizinhança das instalações para inserir uma instalação na solução. O algoritmo 9 representa a união da fase construtiva de inserção mais barata com o refinamento usando ILS.

Capítulo 5

Resultados Computacionais

Neste capítulo serão descritos os resultados alcançados pelos algoritmos desenvolvidos nesta pesquisa. Na Seção 5.1 serão apresentadas as características das instâncias que foram executadas. A metodologia de análise e quais foram os parâmetros utilizados na execução de cada algoritmo para encontrar os resultados serão apresentados na Seção 5.2. Nas Seções 5.3 e 5.4, serão demonstrados os testes estatísticos dos resultados para as execuções de cada algoritmo. Na Seção 5.5, serão apresentados os resultados dos algoritmos e comparou-se os mesmos entre si. Já na Seção 5.6, foi realizada a comparação dos resultados com os principais algoritmos da literatura. Na Seção 5.7, serão feitas considerações finais sobre os resultados.

5.1 Descrição das Instâncias

Para a realização dos experimentos foram utilizadas 144 instâncias. Conforme citado em Mladenović et al. [2019], essas instâncias foram propostas por Belotti et al. [2007], transformando instâncias de *benchmark*, do conjunto de instâncias para o problema clássico de p-medianas existentes na *OR Library* (`http://people.brunel.uk/~mastjjb/jeb/orlib/pmedinfo.html`).

Em particular, foram escolhidas 24 instâncias (de pmed17 a pmed40) do conjunto de instâncias para o problema de p-medianas não-capacitado e, a partir destas, foram criadas 72 instâncias para o p-PLII. A transformação de cada instância foi realizada conforme o seguinte procedimento, descrito em Belotti et al. [2007].

Dada a instância original com n nós, o procedimento gera três instâncias para o p-PLII considerando três valores de p: $n/2$ (consideradas difíceis), $n/4$ (médias) e $n/8$ (consideradas fáceis), conforme pode ser visto na Tabela 5.1. Após essa etapa, seleciona-se $n/2$ nós aleatoriamente para constituir o conjunto de clientes e declarando

os $n/2$ nós de instalação. O procedimento original apresentado em Belotti et al. [2007] gerou o conjunto de instâncias A. Este mesmo procedimento foi, recentemente, reproduzido por Herrán et al. [2018], o que possibilitou a geração do conjunto de instâncias B e, portanto, estender os conjuntos de *benchmark* para o p-PLII, introduzindo 72 novas instâncias de testes geradas da mesma maneira [Mladenović et al., 2019].

A Tabela 5.1 apresenta as características destes dois conjuntos de instâncias usados para os testes computacionais realizados para a presente dissertação. A coluna "Instância" mostra a identificação da instância. A coluna "n" mostra o número de nós da instância. A coluna "$|I|/|J|$ mostra a razão da quantidade de clientes por instalações candidatas. E a coluna "p" mostra a quantidade de instalações necessárias para a instância.

5.2 Metodologia de Análise

A análise estatística dos resultados é uma das etapas mais importantes de um trabalho científico. Segundo Montgomery [2017] a abordagem estatística do projeto experimental é necessária se desejarmos tirar conclusões significativas dos dados.

Para comparar os resultados encontrados pelos algoritmos desenvolvidos e os resultados da literatura, optou-se por comparar o melhor resultado das execuções. Como tratam-se de algoritmos estocásticos, para validar tal metodologia de análise, será verificada a normalidade das execuções para cada instância. Após isso, será comparado cada melhor resultado com o conjunto de resultados da instância, para verificar se existe variação significativa entre o conjunto de resultados. Caso essa resposta seja negativa, ou seja, não exista variação significativa entre o melhor resultado de cada instância e os demais resultados para a mesma instância, pode-se usar o melhor resultado para representar o conjunto de execuções e compará-lo com os demais resultados da literatura.

Para executar o algoritmo serão utilizadas 30 sementes diferentes, uma em cada execução, totalizando 30 execuções. Em seguida, será armazenado o melhor resultado entre as execuções e guardado o tempo computacional para essa execução. Além disso, será calculada a média das soluções para cada instância e qual foi o *gap* da melhor solução encontrada em relação a melhor solução da literatura.

Primeiramente, para analisar o conjunto das soluções que os algoritmos alcançaram, optou-se por realizar a análise da distribuição da amostra dos resultados encontrados. Sendo assim, será realizado o teste estatístico desenvolvido por Shapiro & Wilk [1965]. O teste de Shapiro & Wilk [1965] é um teste de hipótese, no qual existem

Tabela 5.1. Caraterísticas das Instâncias A e B usadas nos experimentos

Instância	n	|I|/|J|	p	Instância	n	|I|/|J|	p
pmed17-p100	400	200	100	pmed29-p150	600	300	150
pmed17-p25	400	200	25	pmed29-p37	600	300	37
pmed17-p50	400	200	50	pmed29-p75	600	300	75
pmed18-p100	400	200	100	pmed30-p150	600	300	150
pmed18-p25	400	200	25	pmed30-p37	600	300	37
pmed18-p50	400	200	50	pmed30-p75	600	300	75
pmed19-p100	400	200	100	pmed31-p175	700	350	175
pmed19-p25	400	200	25	pmed31-p43	700	350	43
pmed19-p50	400	200	50	pmed31-p87	700	350	87
pmed20-p100	400	200	100	pmed32-p175	700	350	175
pmed20-p25	400	200	25	pmed32-p43	700	350	43
pmed20-p50	400	200	50	pmed32-p87	700	350	87
pmed21-p125	500	250	125	pmed33-p175	700	350	175
pmed21-p31	500	250	31	pmed33-p43	700	350	43
pmed21-p62	500	250	62	pmed33-p87	700	350	87
pmed22-p125	500	250	125	pmed34-p175	700	350	175
pmed22-p31	500	250	31	pmed34-p43	700	350	43
pmed22-p62	500	250	62	pmed34-p87	700	350	87
pmed23-p125	500	250	125	pmed35-p100	800	400	100
pmed23-p31	500	250	31	pmed35-p200	800	400	200
pmed23-p62	500	250	62	pmed35-p50	800	400	50
pmed24-p125	500	250	125	pmed36-p100	800	400	100
pmed24-p31	500	250	31	pmed36-p200	800	400	200
pmed24-p62	500	250	62	pmed36-p50	800	400	50
pmed25-p125	500	250	125	pmed37-p100	800	400	100
pmed25-p31	500	250	31	pmed37-p200	800	400	200
pmed25-p62	500	250	62	pmed37-p50	800	400	50
pmed26-p150	600	300	150	pmed38-p112	900	450	112
pmed26-p37	600	300	37	pmed38-p225	900	450	225
pmed26-p75	600	300	75	pmed38-p56	900	450	56
pmed27-p150	600	300	150	pmed39-p112	900	450	112
pmed27-p37	600	300	37	pmed39-p225	900	450	225
pmed27-p75	600	300	75	pmed39-p56	900	450	56
pmed28-p150	600	300	150	pmed40-p112	900	450	112
pmed28-p37	600	300	37	pmed40-p225	900	450	225
pmed28-p75	600	300	75	pmed40-p56	900	450	56

sempre duas hipóteses: a hipótese nula (Ho) e a hipótese alternativa (Ha). Caso o resultado do p-valor seja maior que o nível de significância, a hipótese nula é aceita, caso contrário, rejeita-se a hipótese nula. O nível de significância adotado foi de 5%.

Para a análise de distribuição amostral, a hipótese nula é que a distribuição dos dados é normal. Se a amostra possuir normalidade, será proposto o teste *t-Student* Gosset [1908], que é feito dois a dois e verificando se existe variação significativa entre os conjuntos de dados. Caso a amostra não possua normalidade, será proposto o

teste desenvolvido por Wilcoxon [1992] que é feito da mesma maneira que o *t-Student*, porém, para amostras sem garantia de normalidade. A análise dos resultados dos testes *t-Student* e Wilcoxon [1992] também seguem a mesma metodologia do Shapiro & Wilk [1965] quanto à verificação do p-valor e o nível de significância adotado.

Em suma, a metodologia de análise será feita da seguinte maneira. Primeiramente será verificada a normalidade das 30 execuções de cada instância. Após isso, será realizado um teste estatístico para verificar se o melhor resultado de cada instância diverge da média do conjunto de resultados. Finalmente, os melhores resultados das 30 execuções da cada instância serão comparados com os melhores resultados da literatura.

Os algoritmos propostos foram desenvolvidos na linguagem C++. Todos os experimentos foram executados em um computador com processador Intel Xeon E5506 @ 2.13 GHz com 32 GB de memória RAM e sistema operacional CentOS 6.6 64 bits.

Os resultados obtidos pelos algoritmos foram separados em duas subseções para apresentar os resultados alcançados pelo algoritmo híbrido G/ILS e também do algoritmo ILS que utiliza a construção pela técnica de inserção mais barata.

5.3 Resultados do Algoritmo Híbrido GRASP-ILS

Os parâmetros utilizados pelo algoritmo foram α= 0,9 e nível máximo de perturbação $Perturba\_MAX$ =10 e foram geradas num_Constr= 10000 soluções iniciais na fase construtiva do GRASP. Esses parâmetros foram escolhidos de maneira empírica. Executou-se o algoritmo várias vezes para um conjunto de instâncias variado, buscando encontrar os melhores parâmetros para superar ou igualar os algoritmos da literatura em relação a qualidade das soluções e eficiência de tempo computacional. Nesse desenvolvimento não foram feitas análises estatísticas para descobrir quais parâmetros seriam mais adequados ao algoritmo.

Conforme descrito na Seção 5.2 foram comparados os melhores resultados das 30 execuções do algoritmo com os melhores apresentados na literatura. Para validar essa proposta, foi verificada a normalidade do conjunto das execuções e analisado se o melhor resultado das execuções diverge da média dos 30 resultados para aquela instância. Esse procedimento foi feito para todas as instância executadas e o resultado foi apresentado nas Tabela 5.2 e Tabela 5.3, para as instâncias A e B, respectivamente.

As colunas "Instâncias" informam o nome das instâncias que foram executadas. As colunas "Melhor" apresentam o melhor resultado encontrado pelo algoritmo para as 30 execuções. As colunas "Média" apresentam a média dos resultados das execuções. As colunas "Shapiro" apresentam o resultado do p-Valor para o teste estatístico de Shapiro,

no qual valores menores que $0,05$ rejeitam a hipótese nula. Os resultados desta coluna que aparecem com hífen (-) representam instâncias cujas execuções apresentaram os mesmos resultados para todas as 30 execuções. Neste caso, não foi possível usar o teste, porém tais amostras são claramente não normais.

Como todas as instâncias não apresentaram normalidade, foi feito o teste de Wilcoxon [1992] para verificar se o melhor resultado é representativo para as execuções. Nas colunas "Wilcox" é apresentado o p-Valor do teste e, para simplificar a visualização do resultado nas colunas "H0", é sinalizado um "Ac" para as instâncias nas quais o melhor resultado não diverge das execuções, ou seja, aceita-se a hipótese nula, garantindo que o melhor resultado é um bom representante para ser usado como resultado daquela instância. Já os resultados desta coluna que são representados com um "Re", rejeitam a hipótese nula, indicando que o melhor resultado não é um bom representante para os resultados daquela instância. A aplicação do teste de Wilcoxon demonstra que, para a maioria das instâncias, foi aceita a hipótese nula. Com essa validade estatística, os melhores resultados das execuções serão comparados aos melhores resultados da literatura.

5.4 Resultados do Algoritmo Híbrido Inserção Mais Barata-ILS

Os parâmetros utilizados pelo algoritmo foram nível máximo de perturbação $Perturba\_MAX = 10$ e foi gerada somente uma solução inicial gulosa por meio da técnica de inserção mais barata. Por mais que este algoritmo construa apenas uma solução inicial, seu modo de construção demanda de significativo tempo computacional.

Conforme descrito na Seção 5.2 foram comparados os melhores resultados das 30 execuções do algoritmo com os melhores apresentados na literatura. Para validar essa proposta, foi verificada a normalidade do conjunto das execuções e analisado se o melhor resultado das execuções diverge da média dos 30 resultados para aquela instância. Esse procedimento foi feito para todas as instância executadas e o resultado foi apresentado nas Tabela 5.4 e Tabela 5.5, para as instâncias A e B respectivamente. As colunas "Instâncias" informam o nome das instâncias que foram executadas. As colunas "Melhor" apresentam o melhor resultado encontrado pelo algoritmo para as 30 execuções. As colunas "Média" apresentam a média dos resultados das execuções. As colunas "Shapiro" apresentam o resultado do p-Valor para o teste estatístico de Shapiro, no qual valores menores que 0,05 rejeitam a hipótese nula. Ou seja, não apresentam normalidade. Os resultados desta coluna que aparecem com hífen (-) representam ins-

Tabela 5.2. Análise dos resultados: GRASP-ILS (Instâncias Grupo A)

			p-Valor						p-Valor		
Instâncias	Melhor	Média	Shapiro	Wilcox	H0	Instâncias	Melhor	Média	Shapiro	Wilcox	H0
pmed17-p100	4054	4054,0	-	1,00	Ac	pmed29-p150	4101	4100,2	<0,05	0,77	Ac
pmed17-p25	7317	7317,0	-	1,00	Ac	pmed29-p37	7404	7404,0	-	1,00	Ac
pmed17-p50	5411	5410,9	<0,05	0,77	Ac	pmed29-p75	5880	5880,0	-	1,00	Ac
pmed18-p100	4220	4212,1	<0,05	0,77	Ac	pmed30-p150	4366	4343,5	<0,05	0,05	Ac
pmed18-p25	7432	7432,0	-	1,00	Ac	pmed30-p37	7704	7685,7	<0,05	0,05	Ac
pmed18-p50	5736	5698,7	<0,05	0,03	Re	pmed30-p75	6165	6165,0	-	1,00	Ac
pmed19-p100	3985	3984,5	<0,05	0,77	Ac	pmed31-p175	4108	4091,3	<0,05	0,05	Ac
pmed19-p25	7020	7020,0	-	1,00	Ac	pmed31-p43	7424	7424,0	-	1,00	Ac
pmed19-p50	5340	5339,1	<0,05	0,03	Re	pmed31-p87	5905	5885,0	<0,05	0,05	Ac
pmed20-p100	4062	4062,0	-	1,00	Ac	pmed32-p175	4239	4230,3	<0,05	0,73	Ac
pmed20-p25	7648	7648,0	-	1,00	Ac	pmed32-p43	7752	7752,0	-	1,00	Ac
pmed20-p50	5863	5861,4	<0,05	0,77	Ac	pmed32-p87	5889	5883,7	<0,05	0,73	Ac
pmed21-p125	4145	4145,0	-	1,00	Ac	pmed33-p175	4099	4092,5	<0,05	0,73	Ac
pmed21-p31	7304	7304,0	-	1,00	Ac	pmed33-p43	7598	7597,7	<0,05	0,73	Ac
pmed21-p62	5743	5742,7	<0,05	0,77	Ac	pmed33-p87	5769	5767,2	<0,05	0,73	Ac
pmed22-p125	4341	4340,6	<0,05	0,77	Ac	pmed34-p175	4285	4272,5	<0,05	0,05	Ac
pmed22-p31	7900	7897,5	<0,05	0,77	Ac	pmed34-p43	7725	7722,7	<0,05	0,73	Ac
pmed22-p62	5974	5970,5	<0,05	0,03	Re	pmed34-p87	5835	5832,8	<0,05	0,73	Ac
pmed23-p125	4106	4103,4	<0,05	0,03	Re	pmed35-p100	5844	5844,0	-	1,00	Ac
pmed23-p31	7839	7839,0	-	1,00	Ac	pmed35-p200	3970	3964,0	<0,05	0,73	Ac
pmed23-p62	5776	5775,5	<0,05	0,77	Ac	pmed35-p50	7155	7155,0	-	1,00	Ac
pmed24-p125	4091	4082,3	<0,05	0,03	Re	pmed36-p100	6451	6450,7	<0,05	0,73	Ac
pmed24-p31	7425	7425,0	-	1,00	Ac	pmed36-p200	4302	4300,2	<0,05	0,73	Ac
pmed24-p62	5527	5527,0	-	1,00	Ac	pmed36-p50	8173	8173,0	-	1,00	Ac
pmed25-p125	4137	4130,9	<0,05	0,03	Re	pmed37-p100	6168	6163,0	<0,05	0,05	Ac
pmed25-p31	7552	7552,0	-	1,00	Ac	pmed37-p200	4568	4553,8	<0,05	0,05	Ac
pmed25-p62	5767	5752,3	<0,05	0,03	Re	pmed37-p50	7829	7758,2	<0,05	0,05	Ac
pmed26-p150	4326	4325,3	<0,05	0,77	Ac	pmed38-p112	5903	5888,8	<0,05	0,05	Ac
pmed26-p37	8112	8112,0	-	1,00	Ac	pmed38-p225	4402	4400,2	<0,05	0,73	Ac
pmed26-p75	5780	5776,5	<0,05	0,03	Re	pmed38-p56	7410	7405,0	<0,05	0,05	Ac
pmed27-p150	4020	4018,9	<0,05	0,77	Ac	pmed39-p112	5935	5933,3	<0,05	0,05	Ac
pmed27-p37	7556	7556,0	-	1,00	Ac	pmed39-p225	4343	4337,5	<0,05	0,73	Ac
pmed27-p75	5643	5642,7	<0,05	0,77	Ac	pmed39-p56	7711	7711,0	-	1,00	Ac
pmed28-p150	4074	4072,5	<0,05	0,77	Ac	pmed40-p112	6264	6263,0	<0,05	0,73	Ac
pmed28-p37	7366	7366,0	-	1,00	Ac	pmed40-p225	4550	4549,3	<0,05	0,73	Ac
pmed28-p75	5662	5643,8	<0,05	0,03	Re	pmed40-p56	8190	8190,0	-	1,00	Ac

tâncias cujas execuções apresentaram os mesmos resultados para todas as 30 execuções. Neste caso, não foi possível usar o teste, porém tais amostras são claramente não normais. Como todas as instâncias não apresentaram normalidade, foi feito o teste de Wilcoxon para verificar se o melhor resultado diverge da média das execuções. Nas colunas "Wilcox" é apresentado o p-Valor do teste e para simplificar a visualização do resultado nas colunas "H0" é sinalizado um "Ac" para as instâncias nas quais o melhor resultado não diverge das execuções, ou seja, aceita a hipótese nula, garantindo que o melhor resultado é um bom representante para ser usado como resultado daquela instância. Já os resultados desta coluna que são representados com um "Re", rejeitam a hipótese nula, indicando que o melhor resultado não é um bom representante para os resultados daquela instância. A aplicação do teste de Wilcoxon, demonstra que para

Tabela 5.3. Análise dos resultados: GRASP-ILS (Instâncias Grupo B)

			p-Valor						p-Valor		
Instâncias	Melhor	Média	Shapiro	Wilcox	H0	Instâncias	Melhor	Média	Shapiro	Wilcox	H0
pmed17-p100.B	3992	3988,5	<0,05	0,03	Re	pmed29-p150.B	4139	4138,2	<0,05	0,77	Ac
pmed17-p25.B	6905	6903,4	<0,05	0,77	Ac	pmed29-p37.B	7497	7497,0	-	1,00	Ac
pmed17-p50.B	5563	5556,6	<0,05	0,77	Ac	pmed29-p75.B	5700	5699,1	<0,05	0,77	Ac
pmed18-p100.B	4113	4102,6	<0,05	0,03	Re	pmed30-p150.B	4305	4304,5	<0,05	0,77	Ac
pmed18-p25.B	7662	7662,0	-	1,00	Ac	pmed30-p37.B	8048	8048,0	-	1,00	Ac
pmed18-p50.B	5852	5852,0	-	1,00	Ac	pmed30-p75.B	6037	6034,6	<0,05	0,77	Ac
pmed19-p100.B	4015	3973,4	<0,05	0,03	Re	pmed31-p175.B	4121	4116,1	<0,05	0,77	Ac
pmed19-p25.B	6816	6816,0	-	1,00	Ac	pmed31-p43.B	7320	7313,9	<0,05	0,03	Re
pmed19-p50.B	5413	5413,0	-	1,00	Ac	pmed31-p87.B	5585	5570,3	<0,05	0,03	Re
pmed20-p100.B	4031	4021,5	<0,05	0,03	Re	pmed32-p175.B	4210	4185,7	<0,05	0,03	Re
pmed20-p25.B	7349	7349,0	-	1,00	Ac	pmed32-p43.B	7899	7899,0	-	1,00	Ac
pmed20-p50.B	5665	5655,0	<0,05	0,77	Ac	pmed32-p87.B	5844	5844,0	-	1,00	Ac
pmed21-p125.B	4004	4003,3	<0,05	0,77	Ac	pmed33-p175.B	4128	4118,5	<0,05	0,03	Re
pmed21-p31.B	7331	7331,0	-	1,00	Ac	pmed33-p43.B	7554	7552,3	<0,05	0,77	Ac
pmed21-p62.B	5870	5870,0	-	1,00	Ac	pmed33-p87.B	5802	5796,7	<0,05	0,77	Ac
pmed22-p125.B	4326	4322,1	<0,05	0,77	Ac	pmed34-p175.B	4270	4270,0	-	1,00	Ac
pmed22-p31.B	7695	7695,0	-	1,00	Ac	pmed34-p43.B	7514	7514,0	-	1,00	Ac
pmed22-p62.B	6259	6259,0	-	1,00	Ac	pmed34-p87.B	5855	5848,1	<0,05	0,03	Re
pmed23-p125.B	4095	4095,0	-	1,00	Ac	pmed35-p100.B	5621	5615,0	<0,05	0,77	Ac
pmed23-p31.B	7122	7121,5	<0,05	0,77	Ac	pmed35-p200.B	4088	4088,0	-	1,00	Ac
pmed23-p62.B	5724	5706,7	<0,05	0,03	Re	pmed35-p50.B	7570	7570,0	-	1,00	Ac
pmed24-p125.B	4024	4023,7	<0,05	0,77	Ac	pmed36-p100.B	6205	6161,7	<0,05	0,03	Re
pmed24-p31.B	7190	7190,0	-	1,00	Ac	pmed36-p200.B	4307	4235,9	<0,05	0,03	Re
pmed24-p62.B	5752	5709,5	<0,05	0,03	Re	pmed36-p50.B	8139	8134,9	<0,05	0,77	Ac
pmed25-p125.B	4211	4207,8	<0,05	0,77	Ac	pmed37-p100.B	6195	6191,1	<0,05	0,77	Ac
pmed25-p31.B	7552	7547,5	<0,05	0,77	Ac	pmed37-p200.B	4606	4604,4	<0,05	0,77	Ac
pmed25-p62.B	5692	5691,1	<0,05	0,77	Ac	pmed37-p50.B	8375	8363,7	<0,05	0,03	Re
pmed26-p150.B	4154	4146,2	<0,05	0,03	Re	pmed38-p112.B	5930	5929,6	<0,05	0,77	Ac
pmed26-p37.B	7643	7643,0	-	1,00	Ac	pmed38-p225.B	4414	4399,3	<0,05	0,03	Re
pmed26-p75.B	5863	5859,8	<0,05	0,77	Ac	pmed38-p56.B	7531	7531,0	-	1,00	Ac
pmed27-p150.B	4140	4136,1	<0,05	0,77	Ac	pmed39-p112.B	6174	6173,9	<0,05	0,77	Ac
pmed27-p37.B	7448	7448,0	-	1,00	Ac	pmed39-p225.B	4247	4246,6	<0,05	0,77	Ac
pmed27-p75.B	5824	5820,5	<0,05	0,03	Re	pmed39-p56.B	7625	7625,0	-	1,00	Ac
pmed28-p150.B	4053	4053,0	-	1,00	Ac	pmed40-p112.B	6183	6141,4	<0,05	0,03	Re
pmed28-p37.B	7388	7388,0	-	1,00	Ac	pmed40-p225.B	4507	4506,1	<0,05	0,03	Re
pmed28-p75.B	5627	5626,3	<0,05	0,77	Ac	pmed40-p56.B	8021	8020,6	<0,05	0,77	Ac

todas as instâncias foi aceita a hipótese nula. Com essa validade estatística, os melhores resultados das execuções serão comparados aos melhores resultados da literatura.

5.5 Comparação entre os Algoritmos Desenvolvidos

Nesta pesquisa foram desenvolvidos dois algoritmos com técnicas de construção de soluções distintas. Para fins de comparação de qual técnica apresentou melhor desempenho foi feita uma análise dos resultados.

A Tabelas 5.6 e 5.7 apresentam a comparação entre os resultados obtidos entre os algoritmos propostos. Foi calculado o desvio percentual (*gap*) entre os resultados encontrados para este conjunto de instâncias. O algoritmo híbrido GRASP-ILS será referenciado como G-ILS. Já o algoritmo híbrido Inserção Mais Barata-ILS será apre-

Tabela 5.4. Análise dos resultados: ILS (Instâncias do Grupo A)

			p-Valor						p-Valor		
Instâncias	Melhor	Média	Shapiro	Wilcox	H0	Instâncias	Melhor	Média	Shapiro	Wilcox	H0
pmed17-p100	4054	4050,0	<0,05	0,61084	Ac	pmed29-p150	4131	4112,9	<0,05	0,16094	Ac
pmed17-p25	7317	7315,4	<0,05	0,33292	Ac	pmed29-p37	7404	7377,1	<0,05	0,13629	Ac
pmed17-p50	5411	5400,2	<0,05	0,10288	Ac	pmed29-p75	5880	5858,3	<0,05	0,15218	Ac
pmed18-p100	4220	4217,6	<0,05	0,51265	Ac	pmed30-p150	4385	4370,2	<0,05	0,12692	Ac
pmed18-p25	7432	7432,0	-	1	Ac	pmed30-p37	7704	7704,0	-	1	Ac
pmed18-p50	5746	5738,7	<0,05	0,39645	Ac	pmed30-p75	6189	6176,2	<0,05	0,10541	Ac
pmed19-p100	4033	4015,0	<0,05	0,15559	Ac	pmed31-p175	4136	4112,3	<0,05	0,11715	Ac
pmed19-p25	7020	7020,0	-	1	Ac	pmed31-p43	7424	7421,0	<0,05	0,56967	Ac
pmed19-p50	5386	5350,3	<0,05	0,13059	Ac	pmed31-p87	5905	5898,6	<0,05	0,29645	Ac
pmed20-p100	4062	4059,0	<0,05	0,77322	Ac	pmed32-p175	4240	4190,3	<0,05	0,12848	Ac
pmed20-p25	7648	7648,0	-	1	Ac	pmed32-p43	7794	7764,7	<0,05	0,16036	Ac
pmed20-p50	5872	5860,8	<0,05	0,09094	Ac	pmed32-p87	5905	5901,7	<0,05	0,21618	Ac
pmed21-p125	4155	4144,7	<0,05	0,10375	Ac	pmed33-p175	4096	4067,8	<0,05	0,11534	Ac
pmed21-p31	7304	7304,0	-	1	Ac	pmed33-p43	7598	7596,6	<0,05	0,50995	Ac
pmed21-p62	5784	5778,4	<0,05	0,1379	Ac	pmed33-p87	5790	5749,0	<0,05	0,11248	Ac
pmed22-p125	4358	4339,5	<0,05	0,14508	Ac	pmed34-p175	4286	4275,0	<0,05	0,12887	Ac
pmed22-p31	7900	7894,7	<0,05	0,34284	Ac	pmed34-p43	7725	7702,0	<0,05	0,45422	Ac
pmed22-p62	5995	5993,2	<0,05	0,72572	Ac	pmed34-p87	5842	5831,2	<0,05	0,11188	Ac
pmed23-p125	4108	4095,5	<0,05	0,11512	Ac	pmed35-p100	5845	5837,2	<0,05	0,18754	Ac
pmed23-p31	7841	7840,7	<0,05	0,77283	Ac	pmed35-p200	3992	3965,9	<0,05	0,13051	Ac
pmed23-p62	5785	5782,0	<0,05	0,45246	Ac	pmed35-p50	7155	7141,8	<0,05	0,20075	Ac
pmed24-p125	4091	4078,6	<0,05	0,2517	Ac	pmed36-p100	6461	6454,4	<0,05	0,34723	Ac
pmed24-p31	7425	7422,4	<0,05	0,82745	Ac	pmed36-p200	4318	4308,0	<0,05	0,11564	Ac
pmed24-p62	5528	5515,4	<0,05	0,21332	Ac	pmed36-p50	8173	8173,0	-	1	Ac
pmed25-p125	4155	4123,8	<0,05	0,13074	Ac	pmed37-p100	6203	6154,2	<0,05	0,10998	Ac
pmed25-p31	7552	7549,0	<0,05	1	Ac	pmed37-p200	4587	4574,1	<0,05	0,14312	Ac
pmed25-p62	5767	5765,6	<0,05	0,64343	Ac	pmed37-p50	7830	7814,7	<0,05	0,09589	Ac
pmed26-p150	4334	4322,8	<0,05	0,11497	Ac	pmed38-p112	5909	5907,1	<0,05	0,34437	Ac
pmed26-p37	8112	8110,5	<0,05	0,77342	Ac	pmed38-p225	4425	4412,1	<0,05	0,11649	Ac
pmed26-p75	5789	5777,9	<0,05	0,10307	Ac	pmed38-p56	7432	7420,6	<0,05	0,42456	Ac
pmed27-p150	4062	4047,2	<0,05	0,11682	Ac	pmed39-p112	5935	5930,8	<0,05	0,21981	Ac
pmed27-p37	7556	7548,9	<0,05	0,39165	Ac	pmed39-p225	4363	4345,8	<0,05	0,11645	Ac
pmed27-p75	5668	5650,3	<0,05	0,11641	Ac	pmed39-p56	7712	7711,4	<0,05	0,72676	Ac
pmed28-p150	4099	4074,3	<0,05	0,11653	Ac	pmed40-p112	6272	6262,4	<0,05	0,1156	Ac
pmed28-p37	7366	7359,1	<0,05	0,57364	Ac	pmed40-p225	4555	4536,5	<0,05	0,11641	Ac
pmed28-p75	5667	5653,6	<0,05	0,1279	Ac	pmed40-p56	8211	8197,0	<0,05	0,28722	Ac

sentado como ILS. O cálculo dos valores percentuais de *gap* é dado por:

$$gap\% = \left(\frac{f(ILS) - f(\text{G-ILS})}{f(ILS)} \right) \times 100 \tag{5.1}$$

Nas Tabelas 5.6 e 5.7, as colunas "Instâncias" informam o nome das instâncias que foram executadas. As colunas "ILS" apresentam os resultados do algoritmo híbrido ILS para as instâncias. As colunas "G-ILS" apresentam os resultados do algoritmo híbrido GRASP-ILS. As colunas "t(ILS)" e "t(G-ILS)" apresentam o tempo computacional dos algoritmos. E por fim as colunas *gap* representam a diferença percentual entre o algoritmo ILS e o algoritmo G-ILS. É importante ressaltar que valor de $gap\% = 0$ representa que os algoritmos alcançaram os mesmos resultados. Valores de *gap* negativos indicam que o algoritmo G-ILS superou o resultado do ILS. Já os valores de *gap* positivos

Tabela 5.5. Análise dos resultados: ILS (Instâncias do Grupo B)

			p-Valor						p-Valor		
Instâncias	Melhor	Média	Shapiro	Wilcox	H0	Instâncias	Melhor	Média	Shapiro	Wilcox	H0
pmed17-p100.B	3992	3988,0	<0,05	0,2117	Ac	pmed29-p150.B	4155	4132,2	<0,05	0,1153	Ac
pmed17-p25.B	6905	6897,8	<0,05	0,48032	Ac	pmed29-p37.B	7529	7523,8	<0,05	0,54388	Ac
pmed17-p50.B	5563	5549,7	<0,05	0,29445	Ac	pmed29-p75.B	5709	5703,6	<0,05	0,48304	Ac
pmed18-p100.B	4122	4107,6	<0,05	0,12969	Ac	pmed30-p150.B	4312	4298,0	<0,05	0,12864	Ac
pmed18-p25.B	7662	7662,0	-	1	Ac	pmed30-p37.B	8048	8046,5	<0,05	0,77283	Ac
pmed18-p50.B	5852	5852,0	-	1	Ac	pmed30-p75.B	6041	6023,6	<0,05	0,14382	Ac
pmed19-p100.B	4016	3987,0	<0,05	0,12833	Ac	pmed31-p175.B	4130	4118,9	<0,05	0,11508	Ac
pmed19-p25.B	6816	6816,0	-	1	Ac	pmed31-p43.B	7320	7312,7	<0,05	0,18402	Ac
pmed19-p50.B	5423	5421,6	<0,05	0,77322	Ac	pmed31-p87.B	5617	5588,2	<0,05	0,11645	Ac
pmed20-p100.B	4067	4045,3	<0,05	0,11382	Ac	pmed32-p175.B	4241	4227,4	<0,05	0,1166	Ac
pmed20-p25.B	7349	7349,0	-	1	Ac	pmed32-p43.B	7899	7896,5	<0,05	0,54222	Ac
pmed20-p50.B	5665	5633,1	<0,05	0,32263	Ac	pmed32-p87.B	5843	5809,1	<0,05	0,1004	Ac
pmed21-p125.B	4033	4010,8	<0,05	0,11701	Ac	pmed33-p175.B	4145	4121,6	<0,05	0,11697	Ac
pmed21-p31.B	7331	7331,0	-	1	Ac	pmed33-p43.B	7611	7609,3	<0,05	1	Ac
pmed21-p62.B	5870	5858,9	<0,05	0,16989	Ac	pmed33-p87.B	5839	5818,8	<0,05	0,11625	Ac
pmed22-p125.B	4336	4329,9	<0,05	0,1761	Ac	pmed34-p175.B	4270	4266,2	<0,05	0,2513	Ac
pmed22-p31.B	7695	7695,0	-	1	Ac	pmed34-p43.B	7514	7496,7	<0,05	0,57673	Ac
pmed22-p62.B	6259	6259,0	-	1	Ac	pmed34-p87.B	5857	5846,6	<0,05	0,15538	Ac
pmed23-p125.B	4095	4075,6	<0,05	0,1166	Ac	pmed35-p100.B	5625	5617,4	<0,05	0,12911	Ac
pmed23-p31.B	7137	7134,2	<0,05	0,09094	Ac	pmed35-p200.B	4105	4087,1	<0,05	0,11693	Ac
pmed23-p62.B	5724	5709,7	<0,05	0,2765	Ac	pmed35-p50.B	7570	7567,4	<0,05	0,43277	Ac
pmed24-p125.B	4072	4049,7	<0,05	0,1286	Ac	pmed36-p100.B	6219	6189,0	<0,05	0,14549	Ac
pmed24-p31.B	7190	7183,0	<0,05	0,48206	Ac	pmed36-p200.B	4319	4306,2	<0,05	0,11693	Ac
pmed24-p62.B	5752	5752,0	-	1	Ac	pmed36-p50.B	8144	8126,0	<0,05	0,34589	Ac
pmed25-p125.B	4233	4227,7	<0,05	0,34659	Ac	pmed37-p100.B	6208	6187,3	<0,05	0,13047	Ac
pmed25-p31.B	7552	7546,2	<0,05	0,57667	Ac	pmed37-p200.B	4605	4586,1	<0,05	0,13024	Ac
pmed25-p62.B	5689	5677,9	<0,05	0,07882	Ac	pmed37-p50.B	8379	8379,0	-	1	Ac
pmed26-p150.B	4162	4154,1	<0,05	0,11575	Ac	pmed38-p112.B	5949	5939,2	<0,05	0,11415	Ac
pmed26-p37.B	7643	7632,3	<0,05	0,41703	Ac	pmed38-p225.B	4437	4415,9	<0,05	0,11597	Ac
pmed26-p75.B	5923	5915,6	<0,05	0,19277	Ac	pmed38-p56.B	7532	7531,6	<0,05	0,72676	Ac
pmed27-p150.B	4144	4130,7	<0,05	0,12997	Ac	pmed39-p112.B	6198	6183,7	<0,05	0,11411	Ac
pmed27-p37.B	7448	7445,3	<0,05	0,72572	Ac	pmed39-p225.B	4262	4250,7	<0,05	0,11686	Ac
pmed27-p75.B	5844	5835,4	<0,05	0,17778	Ac	pmed39-p56.B	7611	7585,3	<0,05	0,20601	Ac
pmed28-p150.B	4069	4065,4	<0,05	0,25145	Ac	pmed40-p112.B	6200	6172,4	<0,05	0,11701	Ac
pmed28-p37.B	7388	7387,9	<0,05	1	Ac	pmed40-p225.B	4522	4487,8	<0,05	0,11667	Ac
pmed28-p75.B	5642	5634,5	<0,05	0,20095	Ac	pmed40-p56.B	8022	8018,5	<0,05	0,29651	Ac

indicam que o algoritmo ILS foi superior.

Percebe-se que as diferenças percentuais entre os resultados encontrados pelos algoritmos não é significativa, uma evidência disso é que a maior diferença apresentada foi de 1,2% tanto para o conjunto de instância do tipo A, quanto do tipo B. Os algoritmos desenvolvidos encontraram resultados com pequena diferença percentual, porém, como o algoritmo ILS apresentou um desempenho em relação ao *gap* um pouco melhor, optou-se por compará-lo com os principais algoritmos da literatura.

Tabela 5.6. Comparação dos Resultados - ILS e G-ILS (Instâncias do Grupo A)

Instâncias	ILS	G-ILS	t (ILS)	t (G-ILS)	GAP %	Instâncias	ILS	G-ILS	t (ILS)	t (G-ILS)	GAP %
pmed17-p100.A	4054	4054	388	212	0,0	pmed29-p150.A	4131	4101	1825	1724	0,7
pmed17-p25.A	7317	7317	84	163	0,0	pmed29-p37.A	7404	7404	1767	1388	0,0
pmed17-p50.A	5411	5411	262	389	0,0	pmed29-p75.A	5880	5880	2116	1418	0,0
pmed18-p100.A	4220	4220	218	158	0,0	pmed30-p150.A	4385	4366	1965	2773	0,4
pmed18-p25.A	7432	7432	108	221	0,0	pmed30-p37.A	7704	7704	449	873	0,0
pmed18-p50.A	5746	5736	287	266	0,2	pmed30-p75.A	6189	6165	2128	927	0,4
pmed19-p100.A	4033	3985	274	955	1,2	pmed31-p175.A	4136	4108	2859	4388	0,7
pmed19-p25.A	7020	7020	116	106	0,0	pmed31-p43.A	7424	7424	1087	1029	0,0
pmed19-p50.A	5386	5340	326	244	0,9	pmed31-p87.A	5905	5905	3488	9183	0,0
pmed20-p100.A	4062	4062	254	253	0,0	pmed32-p175.A	4240	4239	11632	7727	0,0
pmed20-p25.A	7648	7648	171	149	0,0	pmed32-p43.A	7794	7752	2836	691	0,5
pmed20-p50.A	5872	5863	333	375	0,2	pmed32-p87.A	5905	5889	1470	1822	0,3
pmed21-p125.A	4155	4145	1279	1089	0,2	pmed33-p175.A	4096	4099	4602	4736	-0,1
pmed21-p31.A	7304	7304	233	241	0,0	pmed33-p43.A	7598	7598	1714	1969	0,0
pmed21-p62.A	5784	5743	711	497	0,7	pmed33-p87.A	5790	5769	3265	3253	0,4
pmed22-p125.A	4358	4341	710	868	0,4	pmed34-p175.A	4286	4285	4636	6418	0,0
pmed22-p31.A	7900	7900	404	336	0,0	pmed34-p43.A	7725	7725	2412	1048	0,0
pmed22-p62.A	5995	5974	744	661	0,4	pmed34-p87.A	5842	5835	3129	1579	0,1
pmed23-p125.A	4108	4106	1180	905	0,0	pmed35-p100.A	5845	5844	5499	4703	0,0
pmed23-p31.A	7841	7839	598	276	0,0	pmed35-p200.A	3992	3970	13498	8385	0,6
pmed23-p62.A	5785	5776	831	819	0,2	pmed35-p50.A	7155	7155	2863	3661	0,0
pmed24-p125.A	4091	4091	1433	2386	0,0	pmed36-p100.A	6461	6451	2671	4287	0,2
pmed24-p31.A	7425	7425	330	206	0,0	pmed36-p200.A	4318	4302	4900	7122	0,4
pmed24-p62.A	5528	5527	1667	719	0,0	pmed36-p50.A	8173	8173	1871	1961	0,0
pmed25-p125.A	4155	4137	1580	720	0,4	pmed37-p100.A	6203	6168	7055	4093	0,6
pmed25-p31.A	7552	7552	413	643	0,0	pmed37-p200.A	4587	4568	7507	5203	0,4
pmed25-p62.A	5767	5767	476	718	0,0	pmed37-p50.A	7830	7829	2575	1815	0,0
pmed26-p150.A	4334	4326	2613	2133	0,2	pmed38-p112.A	5909	5903	7763	3642	0,1
pmed26-p37.A	8112	8112	577	445	0,0	pmed38-p225.A	4425	4402	8786	13336	0,5
pmed26-p75.A	5789	5780	2012	1937	0,2	pmed38-p56.A	7432	7410	7295	2502	0,3
pmed27-p150.A	4062	4020	5056	1757	1,0	pmed39-p112.A	5935	5935	11682	7427	0,0
pmed27-p37.A	7556	7556	1141	982	0,0	pmed39-p225.A	4363	4343	6628	8479	0,5
pmed27-p75.A	5668	5643	3236	1004	0,4	pmed39-p56.A	7712	7711	1297	2360	0,0
pmed28-p150.A	4099	4074	3040	1384	0,6	pmed40-p112.A	6272	6264	11085	3486	0,1
pmed28-p37.A	7366	7366	718	601	0,0	pmed40-p225.A	4555	4550	19907	10898	0,1
pmed28-p75.A	5667	5662	1958	1498	0,1	pmed40-p56.A	8211	8190	1148	1113	0,3

5.6 Comparação entre o Algoritmo ILS e os Resultados da Literatura

O Algoritmo que será utilizado para comparação com a literatura é o Algoritmo Híbrido Inserção Mais Barata-ILS, pois o mesmo apresentou resultados ligeiramente superiores aos do Algoritmo Híbrido GRASP-ILS. Conforme descrito na Seção 5.4, os parâmetros utilizados pelo algoritmo foram nível máximo de perturbação $Perturba\_MAX = 10$ e foi gerada somente uma solução inicial gulosa por meio da técnica de inserção mais barata.

A Tabelas 5.8 e 5.9 apresentam a comparação entre os resultados obtidos pela aplicação do algoritmo proposto com as melhores soluções encontradas pelos algoritmos heurísticos GRASP (1000), B&C + XTS, XTS e VNS, conforme apresentado e disponibilizado por Herrán et al. [2018] e Mladenović et al. [2019]. Foi calculado o desvio percentual (*gap*) entre os resultados encontrados e os melhores resultados da literatura

Tabela 5.7. Comparação dos Resultados - ILS e G-ILS (Instâncias do Grupo B)

Instâncias	ILS	G-ILS	t (ILS)	t (G-ILS)	GAP %	Instâncias	ILS	G-ILS	t (ILS)	t (G-ILS)	GAP %
pmed17-p100.B	3992	3992	232	350	0,0	pmed29-p150.B	4155	4139	5667	3189	0,4
pmed17-p25.B	6905	6905	210	244	0,0	pmed29-p37.B	7529	7497	392	378	0,4
pmed17-p50.B	5563	5563	267	336	0,0	pmed29-p75.B	5709	5700	995	1959	0,2
pmed18-p100.B	4122	4113	985	538	0,2	pmed30-p150.B	4312	4305	2273	1145	0,2
pmed18-p25.B	7662	7662	120	78	0,0	pmed30-p37.B	8048	8048	921	539	0,0
pmed18-p50.B	5852	5852	146	134	0,0	pmed30-p75.B	6041	6037	2028	2214	0,1
pmed19-p100.B	4016	4015	484	452	0,0	pmed31-p175.B	4130	4121	3800	4789	0,2
pmed19-p25.B	6816	6816	245	132	0,0	pmed31-p43.B	7320	7320	1482	1232	0,0
pmed19-p50.B	5423	5413	246	237	0,2	pmed31-p87.B	5617	5585	7254	2331	0,6
pmed20-p100.B	4067	4031	658	324	0,9	pmed32-p175.B	4241	4210	4358	2924	0,7
pmed20-p25.B	7349	7349	139	99	0,0	pmed32-p43.B	7899	7899	1524	860	0,0
pmed20-p50.B	5665	5665	197	403	0,0	pmed32-p87.B	5843	5844	2404	2339	0,0
pmed21-p125.B	4033	4004	959	733	0,7	pmed33-p175.B	4145	4128	6044	2948	0,4
pmed21-p31.B	7331	7331	362	552	0,0	pmed33-p43.B	7611	7554	893	1689	0,7
pmed21-p62.B	5870	5870	1246	1260	0,0	pmed33-p87.B	5839	5802	1987	2085	0,6
pmed22-p125.B	4336	4326	761	903	0,2	pmed34-p175.B	4270	4270	2362	1070	0,0
pmed22-p31.B	7695	7695	304	340	0,0	pmed34-p43.B	7514	7514	1707	1545	0,0
pmed22-p62.B	6259	6259	412	758	0,0	pmed34-p87.B	5857	5855	2823	2662	0,0
pmed23-p125.B	4095	4095	1946	1564	0,0	pmed35-p100.B	5625	5621	3580	9840	0,1
pmed23-p31.B	7137	7122	339	383	0,2	pmed35-p200.B	4105	4088	9864	5859	0,4
pmed23-p62.B	5724	5724	632	1379	0,0	pmed35-p50.B	7570	7570	2744	2566	0,0
pmed24-p125.B	4072	4024	569	579	1,2	pmed36-p100.B	6219	6205	5764	5646	0,2
pmed24-p31.B	7190	7190	318	345	0,0	pmed36-p200.B	4319	4307	7105	8476	0,3
pmed24-p62.B	5752	5752	405	887	0,0	pmed36-p50.B	8144	8139	1865	2349	0,1
pmed25-p125.B	4233	4211	949	1596	0,5	pmed37-p100.B	6208	6195	14318	2873	0,2
pmed25-p31.B	7552	7552	387	805	0,0	pmed37-p200.B	4605	4606	8270	7029	0,0
pmed25-p62.B	5689	5692	793	649	-0,1	pmed37-p50.B	8379	8375	2210	1775	0,0
pmed26-p150.B	4162	4154	2709	4843	0,2	pmed38-p112.B	5949	5930	7238	4588	0,3
pmed26-p37.B	7643	7643	489	611	0,0	pmed38-p225.B	4437	4414	20731	8344	0,5
pmed26-p75.B	5923	5863	1661	1373	1,0	pmed38-p56.B	7532	7531	2706	1973	0,0
pmed27-p150.B	4144	4140	3276	2767	0,1	pmed39-p112.B	6198	6174	5280	4027	0,4
pmed27-p37.B	7448	7448	474	510	0,0	pmed39-p225.B	4262	4247	15332	6335	0,4
pmed27-p75.B	5844	5824	2312	1786	0,3	pmed39-p56.B	7611	7625	6771	2671	-0,2
pmed28-p150.B	4069	4053	1493	1197	0,4	pmed40-p112.B	6200	6183	6935	7572	0,3
pmed28-p37.B	7388	7388	726	795	0,0	pmed40-p225.B	4522	4507	14581	12205	0,3
pmed28-p75.B	5642	5627	1515	1044	0,3	pmed40-p56.B	8022	8021	2284	2603	0,0

para esse conjunto de instâncias. Considera-se Mladenović et al. [2019] como o trabalho mais recente para o conjunto de instâncias. Os melhores resultados da literatura serão referenciados como BKS. O cálculo dos valores percentuais de *gap* é dado por:

$$gap\% = \left(\frac{F(BKS) - F(ILS)}{F(BKS)} \right) \times 100 \tag{5.2}$$

em que $F(BKS)$ representa o valor da melhor solução da literatura e $F(ILS)$ representa o valor da solução obtida pela heurística proposta.

Nas Tabelas 5.8 e 5.9, as colunas "Instâncias" informam o nome das instâncias que foram executadas. As colunas "BKS" apresentam os melhores resultados da literatura para as instâncias. Já nas demais colunas são exibidos os resultados do algoritmo proposto na seguinte ordem: melhor resultado encontrado, média entre as 30 execuções, tempo computacional do melhor resultado, *gap* percentual entre o melhor resultado do algoritmo proposto e o melhor da literatura. É importante ressaltar que valor de $gap\% = 0$ representa que o algoritmo proposto encontrou o mesmo resultado da litera-

tura. Valores de *gap* negativos indicam que o algoritmo proposto superou o resultado da literatura. Já os valores de *gap* positivos indicam que o algoritmo proposto foi superado pelos resultados da literatura.

Tabela 5.8. Comparação dos Resultados - ILS (Instâncias do Grupo A)

		ILS						ILS			
Instâncias	BKS	Melhor	Média	Tempo	GAP%	Instâncias	BKS	Melhor	Média	Tempo	GAP%
pmed17-p100.A	4054	4054	4050,0	388	0,000	pmed29-p150.A	4141	4131	4112,9	1825	0,241
pmed17-p25.A	7317	7317	7315,4	84	0,000	pmed29-p37.A	7404	7404	7377,1	1767	0,000
pmed17-p50.A	5411	5411	5400,2	262	0,000	pmed29-p75.A	5880	5880	5858,3	2116	0,000
pmed18-p100.A	4220	4220	4217,6	218	0,000	pmed30-p150.A	4385	4385	4370,2	1965	0,000
pmed18-p25.A	7432	7432	7432,0	108	0,000	pmed30-p37.A	7704	7704	7704,0	449	0,000
pmed18-p50.A	5746	5746	5738,7	287	0,000	pmed30-p75.A	6189	6189	6176,2	2128	0,000
pmed19-p100.A	4033	4033	4015,0	274	0,000	**pmed31-p175.A**	**4135**	**4136**	**4112,3**	**2859**	**-0,024**
pmed19-p25.A	7020	7020	7020,0	116	0,000	pmed31-p43.A	7424	7424	7421,0	1087	0,000
pmed19-p50.A	5387	5386	5350,3	326	0,019	pmed31-p87.A	5905	5905	5898,6	3488	0,000
pmed20-p100.A	4063	4062	4059,0	254	0,025	pmed32-p175.A	4242	4240	4190,3	11632	0,047
pmed20-p25.A	7648	7648	7648,0	171	0,000	pmed32-p43.A	7794	7794	7764,7	2836	0,000
pmed20-p50.A	5872	5872	5860,8	333	0,000	pmed32-p87.A	5925	5905	5901,7	1470	0,338
pmed21-p125.A	4155	4155	4144,7	1279	0,000	pmed33-p175.A	4105	4096	4067,8	4602	0,219
pmed21-p31.A	7304	7304	7304,0	233	0,000	pmed33-p43.A	7598	7598	7596,6	1714	0,000
pmed21-p62.A	5784	5784	5778,4	711	0,000	pmed33-p87.A	5793	5790	5749,0	3265	0,052
pmed22-p125.A	4358	4358	4339,5	710	0,000	pmed34-p175.A	4287	4286	4275,0	4636	0,023
pmed22-p31.A	7900	7900	7894,7	404	0,000	pmed34-p43.A	7725	7725	7702,0	2412	0,000
pmed22-p62.A	5995	5995	5993,2	744	0,000	pmed34-p87.A	5849	5842	5831,2	3129	0,120
pmed23-p125.A	4114	4108	4095,5	1180	0,146	pmed35-p100.A	5845	5845	5837,2	5499	0,000
pmed23-p31.A	7841	7841	7840,7	598	0,000	pmed35-p200.A	4007	3992	3965,9	13498	0,374
pmed23-p62.A	5785	5785	5782,0	831	0,000	pmed35-p50.A	7155	7155	7141,8	2863	0,000
pmed24-p125.A	4091	4091	4078,6	1433	0,000	pmed36-p100.A	6461	6461	6454,4	2671	0,000
pmed24-p31.A	7425	7425	7422,4	330	0,000	pmed36-p200.A	4319	4318	4308,0	4900	0,023
pmed24-p62.A	5528	5528	5515,4	1667	0,000	pmed36-p50.A	8179	8173	8173,0	1871	0,073
pmed25-p125.A	4155	4155	4123,8	1580	0,000	pmed37-p100.A	6203	6203	6154,2	7055	0,000
pmed25-p31.A	7552	7552	7549,0	413	0,000	pmed37-p200.A	4593	4587	4574,1	7507	0,131
pmed25-p62.A	5767	5767	5765,6	476	0,000	pmed37-p50.A	7830	7830	7814,7	2575	0,000
pmed26-p150.A	4341	4334	4322,8	2613	0,161	pmed38-p112.A	5915	5909	5907,1	7763	0,101
pmed26-p37.A	8112	8112	8110,5	577	0,000	pmed38-p225.A	4428	4425	4412,1	8786	0,068
pmed26-p75.A	5789	5789	5777,9	2012	0,000	pmed38-p56.A	7432	7432	7420,6	7295	0,000
pmed27-p150.A	4062	4062	4047,2	5056	0,000	pmed39-p112.A	5935	5935	5930,8	11682	0,000
pmed27-p37.A	7556	7556	7548,9	1141	0,000	pmed39-p225.A	4369	4363	4345,8	6628	0,137
pmed27-p75.A	5668	5668	5650,3	3236	0,000	pmed39-p56.A	7712	7712	7711,4	1297	0,000
pmed28-p150.A	4099	4099	4074,3	3040	0,000	pmed40-p112.A	6272	6272	6262,4	11085	0,000
pmed28-p37.A	7366	7366	7359,1	718	0,000	pmed40-p225.A	4571	4555	4536,5	19907	0,350
pmed28-p75.A	5681	5667	5653,6	1958	0,246	pmed40-p56.A	8211	8211	8197,0	1148	0,000

Tabela 5.9. Comparação dos Resultados - ILS (Instâncias do Grupo B)

		ILS						ILS			
Instâncias	BKS	Melhor	Média	Tempo	GAP	Instâncias	BKS	Melhor	Média	Tempo	GAP
pmed17-p100.B	3992	3992	3988,0	232.03	0,000	pmed29-p150.B	4157	4155	4132,2	5666.81	0,048
pmed17-p25.B	6905	6905	6897,8	209.64	0,000	pmed29-p37.B	7529	7529	7523,8	392.22	0,000
pmed17-p50.B	5563	5563	5549,7	266.6	0,000	pmed29-p75.B	5709	5709	5703,6	994.6	0,000
pmed18-p100.B	4122	4122	4107,6	984.97	0,000	pmed30-p150.B	4313	4312	4298,0	2272.56	0,023
pmed18-p25.B	7662	7662	7662,0	119.95	0,000	pmed30-p37.B	8048	8048	8046,5	921.38	0,000
pmed18-p50.B	5852	5852	5852,0	145.89	0,000	pmed30-p75.B	6041	6041	6023,6	2028.31	0,000
pmed19-p100.B	4016	4016	3987,0	484.13	0,000	pmed31-p175.B	4138	4130	4118,9	3800.31	0,193
pmed19-p25.B	6816	6816	6816,0	244.79	0,000	pmed31-p43.B	7320	7320	7312,7	1481.6	0,000
pmed19-p50.B	5423	5423	5421,6	245.96	0,000	pmed31-p87.B	5621	5617	5588,2	7253.85	0,071
pmed20-p100.B	4067	4067	4045,3	657.8	0,000	pmed32-p175.B	4244	4241	4227,4	4357.93	0,071
pmed20-p25.B	7349	7349	7349,0	138.9	0,000	pmed32-p43.B	7899	7899	7896,5	1523.56	0,000
pmed20-p50.B	5665	5665	5633,1	196.81	0,000	pmed32-p87.B	5852	5843	5809,1	2403.69	0,154
pmed21-p125.B	4033	4033	4010,8	959.15	0,000	pmed33-p175.B	4156	4145	4121,6	6044.31	0,265
pmed21-p31.B	7331	7331	7331,0	361.99	0,000	pmed33-p43.B	7611	7611	7609,3	892.76	0,000
pmed21-p62.B	5870	5870	5858,9	1245.67	0,000	pmed33-p87.B	5840	5839	5818,8	1986.57	0,017
pmed22-p125.B	4338	4336	4329,9	761.03	0,046	pmed34-p175.B	4270	4270	4266,2	2362.19	0,000
pmed22-p31.B	7695	7695	7695,0	303.96	0,000	pmed34-p43.B	7514	7514	7496,7	1706.64	0,000
pmed22-p62.B	6259	6259	6259,0	412.4	0,000	pmed34-p87.B	5857	5857	5846,6	2823.14	0,000
pmed23-p125.B	4095	4095	4075,6	1946.18	0,000	pmed35-p100.B	5639	5625	5617,4	3580.29	0,248
pmed23-p31.B	7137	7137	7134,2	339.21	0,000	pmed35-p200.B	4109	4105	4087,1	9864.45	0,097
pmed23-p62.B	5724	5724	5709,7	631.68	0,000	pmed35-p50.B	7570	7570	7567,4	2744.2	0,000
pmed24-p125.B	4072	4072	4049,7	568.85	0,000	pmed36-p100.B	6219	6219	6189,0	5764.39	0,000
pmed24-p31.B	7190	7190	7183,0	318.06	0,000	pmed36-p200.B	4319	4319	4306,2	7105.09	0,000
pmed24-p62.B	5752	5752	5752,0	405.23	0,000	pmed36-p50.B	8144	8144	8126,0	1864.74	0,000
pmed25-p125.B	4233	4233	4227,7	948.94	0,000	pmed37-p100.B	6211	6208	6187,3	14317.5	0,048
pmed25-p31.B	7552	7552	7546,2	386.86	0,000	pmed37-p200.B	4609	4605	4586,1	8269.99	0,087
pmed25-p62.B	5692	5689	5677,9	792.66	0,053	pmed37-p50.B	8379	8379	8379,0	2210.02	0,000
pmed26-p150.B	4173	4162	4154,1	2709.24	0,264	pmed38-p112.B	5949	5949	5939,2	7237.52	0,000
pmed26-p37.B	7643	7643	7632,3	488.51	0,000	pmed38-p225.B	4446	4437	4415,9	20730.9	0,202
pmed26-p75.B	5923	5923	5915,6	1661.43	0,000	pmed38-p56.B	7535	7532	7531,6	2706.08	0,040
pmed27-p150.B	4144	4144	4130,7	3276.43	0,000	pmed39-p112.B	6198	6198	6183,7	5279.62	0,000
pmed27-p37.B	7448	7448	7445,3	473.56	0,000	pmed39-p225.B	4266	4262	4250,7	15331.6	0,094
pmed27-p75.B	5844	5844	5835,4	2311.85	0,000	pmed39-p56.B	7625	7611	7585,3	6770.79	0,184
pmed28-p150.B	4069	4069	4065,4	1492.96	0,000	pmed40-p112.B	6200	6200	6172,4	6934.79	0,000
pmed28-p37.B	7388	7388	7387,9	725.51	0,000	pmed40-p225.B	4525	4522	4487,8	14581.2	0,066
pmed28-p75.B	5642	5642	5634,5	1515.22	0,000	pmed40-p56.B	8022	8022	8018,5	2284.42	0,000

Os resultados mostram que a heurística apresentada encontrou boas soluções comparada aos principais algoritmos da literatura. Uma evidência disso, é que das 144 instâncias que foram testadas o algoritmo encontrou os mesmos resultados da literatura em 104 delas e teve um (*gap*) baixo nas que não foi possível igualar a literatura. O *gap* percentual foi inferior a 0,4% em relação às melhores soluções encontradas pela literatura. Além disso, o algoritmo conseguiu explorar uma nova solução para a instância pmed31-p175.A, na qual superou os melhores resultados da literatura. Porém, o fator desfavorável a este algoritmo é o tempo computacional que foi superior aos demais da literatura. Mas, por se tratar de um problema de planejamento, o fator tempo não é o principal aspecto a ser mensurado. Não faz muita diferença para um gestor se o algoritmo demora um minuto ou uma hora para ser executado, desde que o mesmo entregue uma solução de boa qualidade.

5.6.1 Comparação do Estado da Arte dos Métodos

A fim de analisar melhor o desempenho dos principais algoritmos da literatura em relação ao algoritmo desenvolvido nesta pesquisa, foi feita uma comparação do resultado geral médio do algoritmo e do número de vezes que os algoritmos encontraram o BKS para cada instância, separados por instâncias fáceis, médias e difíceis.

Os resultados obtidos foram comparados com o TS e o BC de Belotti et al. [2007], o GRASP de Colmenar et al. [2016], o P-VNS de Herrán et al. [2018], o VNS de Mladenović et al. [2019] e o algoritmo desenvolvido nesta pesquisa foram apresentados como ILS. A seguir são apresentadas as Tabelas 5.10 e 5.11 de comparação dos métodos.

Tabela 5.10. Comparação do Estado da Arte

							ILS		
Instância	BKS	B&C	TS	GRASP	P-VNS	VNS	BEST	MÉDIA	*gap*%
Pequenas	7582	7452	7463	7582	7582	7582,4	7581,9	7575,2	0,01
Médias	5857	5784,8	5814,5	5856	5857	5856,9	5855,1	5835,9	0,03
Grandes	4213	4090,3	4169,8	4205	4213	4212,8	4210,2	4188,6	0,07
Total	5884	5775,7	5815,8	5881	5884	5884	5882,4	5866,6	0,03

Tabela 5.11. Comparação do Estado da Arte (Números de BKS)

Tipo de Instâncias	B&C	TS	GRASP	P-VNS	VNS	ILS	"Novos BKS"
Pequenas	9	16	48	48	48	45	0
Médias	6	6	35	43	45	36	0
Grandes	1	1	9	46	40	23	1
Total	16	23	92	137	133	104	1

Analisando os resultados comparativos dos principais algoritmos da literatura, percebe-se que o algoritmo ILS desenvolvido apresenta excelentes resultados. No total, os resultados foram melhores do que os algoritmos BC, TS e o GRASP, tanto em relação ao número de BKS encontrado, quanto aos valores médios dos resultados. A diferença percentual de *gap* em relação ao BKS também foi muito pequena. Em relação ao estado da arte, os algoritmos VNS mostraram-se um pouco melhores que o ILS desenvolvido, porém, essa melhoria não representa uma diferença significativa estatisticamente.

5.7 Discussão dos Resultados

Os dois algoritmos desenvolvidos foram baseados em técnicas de simples implementações, isso favorece a reprodução do método para aplicações em cenários reais. Além disso, os algoritmos alcançaram a literatura na maioria das instâncias e não

houve diferença estatística entre os resultados encontrados pelos algoritmos desenvolvidos e a literatura.

Ademais, nas instâncias consideradas médias e pequenas os algoritmos desenvolvidos foram eficientes tanto em tempo computacional quanto as soluções encontradas. Isso é um outro indicador que os algoritmos desenvolvidos são eficientes caso sejam implantados em situações reais, visto que para instalações indesejadas o conjunto de instalações abertas é bem discreto em relação a quantidade de clientes.

Por exemplo, no cenário atual brasileiro existem aproximadamente 1700 aterros sanitários e 1478 presídios, sendo que 5570 são os municípios existentes no Brasil [IBGE, 2008]. Isso indica que a relação mais próxima à realidade, são instâncias nas quais o número de p instalações segue uma razão de aproximadamente 25% em relação ao tamanho da instância. Pensando nesse sentido os resultados mais relevantes dessa heurística são os que tratam ($p = n/4$).

Em relação ao desempenho dos algoritmos, percebeu-se que o algoritmo ILS, foi mais eficiente que o G-ILS devido a fase construtiva por meio da técnica de inserção mais barata, na qual foi analisada a interação de cada instalação que foi inserida com a solução parcial em construção. Já a fase construtiva do algoritmo G-ILS, dada pela meta-heurística GRASP, não faz uma análise das instalações já constantes na solução para inserir a próxima instalação, analisando apenas características individuais de cada instalação.

Já na comparação do Estado da Arte, percebeu-se claramente que o algoritmo ILS é tão eficiente quando os principais algoritmos da literatura, uma prova disso foi que o maior *gap* percentual em relação ao BKS foi de menos de 0,1%.

Capítulo 6

Conclusão

Neste capítulo serão descritas as conclusões do trabalho, na Seção 6.1. Na Seção 6.2, são discriminados os artigos derivados da pesquisa realizada até o momento para a dissertação. Por fim, na Seção 6.3 foram apresentados os trabalhos futuros que darão continuidade a esta pesquisa.

6.1 Considerações Finais

Este trabalho teve seu foco no problema de localização de instalações indesejadas p-PLII, que consiste em localizar p instalações de modo a maximizar o somatório das distâncias mínimas de cada cliente as instalações abertas.

Foram propostos dois algoritmos meta-heurísticos, que combinam técnicas heurísticas do GRASP e do ILS. Na primeira etapa em um dos algoritmos ocorre a fase construtiva do GRASP, já no outro algoritmo é feita uma construção gulosa com a técnica de inserção mais barata. Na segunda fase ocorre a busca local usando a técnica de pertubações do algoritmo ILS. Os algoritmos foram testados em um conjunto de instâncias da literatura e os resultados foram comparados primeiramente entre si, e o melhor deles foi comparado com os melhores encontrados na literatura para o problema até o momento.

Os resultados obtidos indicam que o algoritmo ILS mostra-se tão eficiente quanto os melhores algoritmos da literatura. Além disso, os algoritmos propostos são de fáceis reprodução e adaptação para aplicações reais. A comparação percentual dos algoritmo ILS com os resultados da literatura também indica equivalência, pois a maior diferença de GAP percentual para uma instância foi de 0,4% . Além disso, comparando-se os resultados gerais do algoritmo no estado da arte, percebeu-se que ele é equivalente aos melhores algoritmos da literatura para o problema em relação aos resultados obtidos.

Uma evidência disso foi que a maior diferença percentual para determinado grupo de instâncias foi menor que 0,1%.

Apesar do tempo computacional ser um pouco maior do que os demais algoritmos da literatura, isso não inviabiliza sua utilização, visto que para um problema de planejamento é mais interessante uma solução de boa qualidade do que uma solução rápida. A facilidade de implementação e reprodução das técnicas utilizadas são pontos significativamente positivos, pois num caso de aplicação real, a adaptação do código não será um empecilho para a utilização do método.

Além disso, neste trabalho discutiu-se a relevância de testar instâncias com o número de p instalações muito alto (valores superiores a 25%) da quantidade de instalações candidatas, visto que na maioria dos casos de localização de instalações indesejadas a relação é de até 25%. Para validar essa observação, comparou-se os dados referentes a quantidades de presídios e quantidades de aterros sanitários no Brasil em relação a quantidades de municípios existentes no país.

6.2 Publicações

A realização deste trabalho resultou em 2 publicações:

(i) um artigo completo publicado no LI Simpósio Brasileiro de Pesquisa Operacional (SBPO), em Setembro de 2019;

(ii) um artigo completo aceito para o LII Simpósio Brasileiro de Pesquisa Operacional (SBPO), em Novembro de 2020;

Para a primeira publicação foi apresentado o algoritmo híbrido entre GRASP e ILS, tratando um conjunto limitado das instâncias exploradas nesta dissertação. Já na segunda publicação foi apresentada a meta-heurística ILS com solução inicial por meio da técnica de inserção mais barata e tratou-se o conjunto completo das instâncias anteriormente tratadas.

6.3 Trabalhos Futuros

Como trabalhos futuros, apontam-se os seguintes: (i) Implementar um método exato que consiga partir da solução inicial oriunda desta heurística; (ii) Executar a calibração dos parâmetros; (iii) Desenvolver um modelo específico para alguma instalação com suas respectivas restrições e particularidades.

Além disso, percebe-se que o problema é tipicamente multicritério. A consideração de que os clientes vão ser alocados nas instalações mais próximas de si, que estiverem abertas, é o que contorna a entrave logística característica do problema. Pois, almeja-se localizar as instalações o mais afastado possível, mas não deseja-se aumentar os custos de transporte. Pensando nisso, como trabalhos futuros podem ser investigados problemas multicritérios que também modelem o problema tratado.

Referências Bibliográficas

Batta, R. & Chiu, S. S. (1988). Optimal obnoxious paths on a network: transportation of hazardous materials. *Operations Research*, 36(1):84--92.

Belotti, P.; Labbé, M.; Maffioli, F. & Ndiaye, M. M. (2007). A branch-and-cut method for the obnoxious p-median problem. *4OR*, 5(4):299--314.

Blum, C.; Roli, A. & Alba, E. (2005). An introduction to metaheuristic techniques. *Parallel Metaheuristics: A New Class of Algorithms*, 47:1.

Cabral, A. (2012). História das usinas nucleoelétricas no brasil. *Revista Eletrônica de Energia*, 1(1).

Cappanera, P.; Gallo, G. & Maffioli, F. (2003). Discrete facility location and routing of obnoxious activities. *Discrete Applied Mathematics*, 133(1-3):3--28.

Cheng, Y.; Han, Q.; Yu, W. & Zhang, G. (2019). Strategy-proof mechanisms for obnoxious facility game with bounded service range. *Journal of Combinatorial Optimization*, 37(2):737--755.

Chiang, Y.-I. & Lin, C.-C. (2017). Compact model for the obnoxious p-median problem. *American Journal of Operations Research*, 7(06):348.

Church, R. L. & Garfinkel, R. S. (1978). Locating an obnoxious facility on a network. *Transportation science*, 12(2):107--118.

Colmenar, J. M.; Greistorfer, P.; Martí, R. & Duarte, A. (2016). Advanced greedy randomized adaptive search procedure for the obnoxious p-median problem. *European Journal of Operational Research*, 252(2):432--442.

Colmenar, J. M.; Martí, R. & Duarte, A. (2018). Multi-objective memetic optimization for the bi-objective obnoxious p-median problem. *Knowledge-Based Systems*, 144:88--101.

Daskin, M. S. (2011). *Network and discrete location: models, algorithms, and applications*. John Wiley & Sons.

Demesouka, O.; Vavatsikos, A. & Anagnostopoulos, K. (2014). Gis-based multicriteria municipal solid waste landfill suitability analysis: A review of the methodologies performed and criteria implemented. *Waste Management & Research*, 32(4):270--296.

Dethloff, J. (2001). Vehicle routing and reverse logistics: the vehicle routing problem with simultaneous delivery and pick-up. *OR-Spektrum*, 23(1):79--96.

Drezner, T.; Drezner, Z. & Schöbel, A. (2018a). The weber obnoxious facility location model: A big arc small arc approach. *Computers & Operations Research*, 98:240--250.

Drezner, Z.; Kalczynski, P. & Salhi, S. (2018b). The planar multiple obnoxious facilities location problem: A voronoi based heuristic. *Omega*.

Eiselt, H. A. & Marianov, V. (2015). Location modeling for municipal solid waste facilities. *Computers & Operations Research*, 62:305--315.

Erkut, E. (1990). The discrete p-dispersion problem. *European Journal of Operational Research*, 46(1):48--60.

Feo, T. A. & Resende, M. G. (1995). Greedy randomized adaptive search procedures. *Journal of global optimization*, 6(2):109--133.

Fernández, J.; Fernández, P. & Pelegrın, B. (2000). A continuous location model for siting a non-noxious undesirable facility within a geographical region. *European Journal of Operational Research*, 121(2):259--274.

Freitas, C. M. d.; Barcellos, C.; Asmus, C. I. R. F.; Silva, M. A. d. & Xavier, D. R. (2019). Da samarco em mariana à vale em brumadinho: desastres em barragens de mineração e saúde coletiva. *Cadernos de Saúde Pública*, 35:e00052519.

Glover, F. W. & Kochenberger, G. A. (2006). *Handbook of metaheuristics*, volume 57. Springer Science & Business Media.

Golden, B.; Baker, E.; Alfaro, J. & Schaffer, J. (1985). The vehicle routing problem with backhauling: two approaches. Em *Proceedings of the twenty-first annual meeting of the SE TIMS, Myrtle Beach, SC, USA*.

Goldman, A. & Dearing, P. M. (1975). Concepts of optimal location for partially noxious facilities. *ORSA Bulletin*, 23(1).

Gopalan, R.; Kolluri, K. S.; Batta, R. & Karwan, M. H. (1990). Modeling equity of risk in the transportation of hazardous materials. *Operations Research*, 38(6):961--973.

Gosset, W. S. (1908). Student. *The Application of the'Law of Error'to the Work of the Brewery*, pp. 3--6.

Hakimi, S. L. (1964). Optimum locations of switching centers and the absolute centers and medians of a graph. *Oper. Res.*, 12(3):450--459. ISSN 0030-364X.

Hansen, P. & Mladenović, N. (2006). First vs. best improvement: An empirical study. *Discrete Applied Mathematics*, 154(5):802 – 817. ISSN 0166-218X. IV ALIO/EURO Workshop on Applied Combinatorial Optimization.

Herrán, A.; Colmenar, J. M.; Martí, R. & Duarte, A. (2018). A parallel variable neighborhood search approach for the obnoxious p-median problem. *International Transactions in Operational Research*.

Hosseini, S. & Esfahani, A. M. (2009). Obnoxious facility location. Em *Facility Location*, pp. 315--345. Springer.

IBGE, I. (2008). Pesquisa nacional de saneamento básico 2008. *Instituto Brasileiro de Geografia e Estatistica–IBGE*.

Jamshidi, M. (2009). Median location problem. Em *Facility Location*, pp. 177--191. Springer.

Labbé, M.; Maffioli, F.; Ndiaye, M. & Belotti, P. (2001). Obnoxious p-median problems: valid inequalities and a branch-and-cut approach. *The OR Peripatetic Post-Graduate Programme*, pp. 26--29.

Lancuna, W.; Martins de Sá, E. & Souza, S. (2019). Heurística híbrida grasp/ils para o problema localização de p instalações indesejadas. *LI Simpósio Brasileiro De Pesquisa Operacional*.

Lin, G. & Guan, J. (2018). A hybrid binary particle swarm optimization for the obnoxious p-median problem. *Information Sciences*, 425:1--17.

Lourenço, H. R.; Martin, O. C. & Stützle, T. (2003). Iterated local search. Em *Handbook of metaheuristics*, pp. 320--353. Springer.

Melachrinoudis, E.; Min, H. & Wu, X. (1995). A multiobjective model for the dynamic location of landfills. *Location Science*, 3(3):143--166.

Mladenović, N.; Alkandari, A.; Pei, J.; Todosijević, R. & Pardalos, P. M. (2019). Less is more approach: basic variable neighborhood search for the obnoxious p-median problem. *International Transactions in Operational Research*.

Montgomery, D. C. (2017). *Design and analysis of experiments*. John wiley & sons.

Moon, I. D. & Chaudhry, S. S. (1984). An analysis of network location problems with distance constraints. *Management Science*, 30(3):290--307.

Pisinger, D. (2006). Upper bounds and exact algorithms for -dispersion problems. *Computers & Operations Research*, 33(5):1380--1398.

Rakas, J.; Teodorović, D. & Kim, T. (2004). Multi-objective modeling for determining location of undesirable facilities. *Transportation Research Part D: Transport and Environment*, 9(2):125--138.

Salhi, S. & Nagy, G. (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling. *Journal of the Operational Research Society*, 50(10):1034--1042. ISSN 1476-9360.

Sayah, D. & Irnich, S. (2017). A new compact formulation for the discrete p-dispersion problem. *European Journal of Operational Research*, 256(1):62--67.

Shapiro, S. S. & Wilk, M. B. (1965). An analysis of variance test for normality (complete samples). *Biometrika*, 52(3/4):591--611.

Tamir, A. (1991). Obnoxious facility location on graphs. *SIAM Journal on Discrete Mathematics*, 4(4):550--567.

Wilcoxon, F. (1992). Individual comparisons by ranking methods. Em *Breakthroughs in statistics*, pp. 196--202. Springer.