UNIVERSIDADE DE SÃO PAULO

Instituto de Ciências Matemáticas e de Computação

Aprendizado de máquina em heurísticas de decomposição para problemas de dimensionamento de lotes.

Fernanda Yuka Ueno

Dissertação de Mestrado do Programa de Pós-Graduação em Ciências
de Computação e Matemática Computacional (PPG-CCMC)

SERVIÇO DE PÓS-GRADUAÇÃO DO ICMC-USP

Data de Depósito:

Assinatura: _____

Fernanda Yuka Ueno

Aprendizado de máquina em heurísticas de decomposição para problemas de dimensionamento de lotes.

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação – ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências – Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maristela Oliveira dos Santos

USP – São Carlos
Dezembro de 2021

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados inseridos pelo(a) autor(a)

U22a Ueno, Fernanda Yuka
 Aprendizado de máquina em heurísticas de
decomposição para problemas de dimensionamento de
lotes. / Fernanda Yuka Ueno; orientadora Maristela
Oliveira dos Santos. -- São Carlos, 2021.
 170 p.

 Dissertação (Mestrado - Programa de Pós-Graduação
em Ciências de Computação e Matemática
Computacional) -- Instituto de Ciências Matemáticas
e de Computação, Universidade de São Paulo, 2021.

 1. Problema de dimensionamento de lotes. 2.
Aprendizado de máquina. 3. Heurística relax-and-fix.
4. Heurística fix-and-optimize. I. Santos, Maristela
Oliveira dos, orient. II. Título.

Fernanda Yuka Ueno

Machine learning on decomposition heuristics for lot sizing problems.

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Maristela Oliveira dos Santos

**USP – São Carlos
December 2021**

AGRADECIMENTOS

Gostaria de agradecer primeiramente aos meus pais por apoiarem, terem me dado a oportunidade de conseguir realizar os meus sonhos e ajudarem-me muito para conseguir realizar os meus estudos.

Agradeço à Universidade de São Paulo por me acolher e possibilitar a realização do mestrado. À Maristela, por ser a minha orientadora e por ter me ajudado muito a realizá-lo, sempre disponível e paciente. Agradeço ao Marcos Mansano Furlan pelas contribuições no trabalho, agradeço ao pessoal do LOt (Laboratório de Otimização), principalmente pelas contribuições do Caio Tomazella, da Lorrany Cristina da Silva e da Oliviana Nascimento.

Agradeço ao Marcel Otoboni por tudo e por estar sempre disponível nas horas difíceis. Aos meus amigos e colegas que conheci ao longo desses anos por terem me ajudado e pela companhia.

Também agradeço à banca composta por Mariá C. V. Nascimento, Alexandre Delbem e Pedro Munari por aceitarem o convite e pelas ótimas sugestões.

O presente trabalho foi realizado com apoio financeiro da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) e a pesquisa desenvolvida com utilização dos recursos computacionais do Centro de Ciências Matemáticas Aplicadas à Indústria (Ce-MEAI), financiados pela FAPESP (proc. 2013/07375-0).

RESUMO

UENO, F. Y. **Aprendizado de máquina em heurísticas de decomposição para problemas de dimensionamento de lotes.** 2021. 134 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

Neste trabalho, são propostas heurísticas baseadas na partição do conjunto de variáveis dos modelos matemáticos, as quais são utilizadas para a resolução de dois problemas de dimensionamento de lotes. As heurísticas desenvolvidas são compostas de duas fases: construção de uma solução inicial e melhoria. As soluções iniciais são obtidas por meio da heurística *relax-and-fix* ou por um resolvedor comercial de otimização (primeira solução factível obtida). As heurísticas de melhoria são do tipo *fix-and-optimize* com partições clássicas e ADN (*Automatically designed neighborhoods*), que constrói uma vizinhança de forma automática, utilizando aprendizado de máquina não supervisionado, ou seja, usamos dois algoritmos de agrupamento: o *k-means* e o *k-medoids*. Nos experimentos computacionais, abordamos o problema de dimensionamento de lotes com múltiplas plantas distintas e o problema de dimensionamento de lotes multiestágio. Para realizar a comparação entre as heurísticas, foram utilizadas instâncias da literatura e as soluções são comparadas com as soluções obtidas por um otimizador comercial.

Palavras-chave: Aprendizado de máquina, Problema de dimensionamento de lotes, Heurística *relax-and-fix*, Heurística *fix-and-optimize*.

ABSTRACT

UENO, F. Y. **Machine learning on decomposition heuristics for lot sizing problems..** 2021. 134 p. Dissertação (Mestrado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, São Carlos – SP, 2021.

In this paper, heuristics based on the partition of the set of variables of the mathematical models are proposed, which are used to solve two lot sizing problems. The developed heuristics are composed of two phases: construction of an initial solution and improvement. The initial solutions are obtained through the relax-and-fix heuristic or by a commercial optimization solver (first feasible solution obtained). The improvement heuristics are of the fix-and-optimize type with classical partitions and ADN (Automatically designed neighborhoods), which builds a neighborhood automatically using unsupervised machine learning, i.e., we use two clustering algorithms, k-means and k-medoids. In the computational experiments we address the lot sizing problem with multiple distinct plants and the multistage lot sizing problem. To perform the comparison between the heuristics, instances from the literature were used and the solutions are compared with the solutions obtained by a commercial optimizer.

Keywords: Machine learning, Lot sizing problem, Fix-and-optimize, Relax-and-fix.

LISTA DE ILUSTRAÇÕES

Figura 1 – Esquema simplificado do aprendizado por reforço.	28
Figura 2 – Estrutura de produto	35
Figura 3 – Fluxograma da abordagem da solução simplificado	38
Figura 4 – Exemplo do RF	40
Figura 5 – Exemplo do FO	42
Figura 6 – Matriz de coeficientes A simplificada de um problema inteiro misto genérico	44
Figura 7 – Matriz de coeficientes A simplificada de um MIP genérico	45
Figura 8 – <i>Box plot</i> do MIPGap em porcentagem das soluções do MPCLSP obtidas pelo <i>Gurobi</i>	52
Figura 9 – <i>Box plot</i> do tempo gasto pelo <i>Gurobi</i> para resolver MPCLSP.	54
Figura 10 – Grafo de adjacência do subconjunto de entidade $E_3 = T$ para MPCLSP.	55
Figura 11 – <i>Box plot</i> do MIPGap do <i>Gurobi</i> e GAPIni das heurísticas para MPCLSP sem <i>outliers</i>	57
Figura 12 – <i>Box plot</i> do número de iterações realizadas pelas heurísticas, exceto as que utilizaram AND1 para MPCLSP.	62
Figura 13 – <i>Box plot</i> do MIPGap em porcentagem das soluções do MLCLSP obtidas pelo <i>Gurobi</i>	64
Figura 14 – <i>Box plot</i> do MIPGap do <i>Gurobi</i> e GAPIni das heurísticas para MLCLSP.	67
Figura 15 – Matriz de coeficientes A dado em A.27 rearranjada	80
Figura 16 – Matriz de coeficientes A dado em A.27 rearranjada	81

LISTA DE ALGORITMOS

Algoritmo 1 – Pseudocódigo da heurística RF considerando particionamento com sobreposição	39
Algoritmo 2 – Pseudocódigo da heurística FO considerando particionamento com sobreposição	41
Algoritmo 3 – Pseudocódigo do <i>k-means</i>	47
Algoritmo 4 – Pseudocódigo do <i>k-medoids</i>	47
Algoritmo 5 – Pseudocódigo do ADN1	49
Algoritmo 6 – Pseudocódigo do ADN2	49
Algoritmo 7 – Pseudocódigo do ADN3	49

LISTA DE TABELAS

Tabela 1 – MIPGap das soluções do MPCLSP obtidas pelo <i>Gurobi</i>	53
Tabela 2 – Tempo gasto pelo <i>Gurobi</i> para resolver MPCLSP.	54
Tabela 3 – GAPIni das heurísticas e o MIPGap do <i>Gurobi</i> para o MPCLSP.	58
Tabela 4 – GAP das heurísticas e pelo <i>Gurobi</i> para MPCLSP.	59
Tabela 5 – Quantidade de instâncias do MPCLSP que tiveram soluções ótimas para cada heurística.	60
Tabela 6 – Tempo gasto pelas heurísticas e pelo <i>Gurobi</i> para MPCLSP.	61
Tabela 7 – Conjunto de instâncias do MLCLSP.	63
Tabela 8 – MIPGap das soluções do MLCLSP obtidas pelo <i>Gurobi</i>	64
Tabela 9 – Número de agrupamentos para itens (N) para MLCLSP.	65
Tabela 10 – Número de agrupamentos para períodos (T) para MLCLSP.	65
Tabela 11 – GAPIni das heurísticas e o MIPGap do <i>Gurobi</i>	67
Tabela 12 – GAP das heurísticas e pelo <i>Gurobi</i> para MPCLSP.	68
Tabela 13 – Tempo gasto pelas heurísticas e pelo <i>Gurobi</i> para MLCLSP.	69
Tabela 14 – Número de iterações realizadas pelas heurísticas.	69
Tabela 15 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 6 itens do MPCLSP.	85
Tabela 16 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 12 itens do MPCLSP.	86
Tabela 17 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 25 itens do MPCLSP.	87
Tabela 18 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 50 itens do MPCLSP.	88
Tabela 19 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 70 itens do MPCLSP.	89
Tabela 20 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 80 itens do MPCLSP.	90
Tabela 21 – GAP das heurísticas e o MIPGap do <i>Gurobi</i> para instâncias com 90 itens do MPCLSP.	91
Tabela 22 – GAPM das heurísticas para instâncias com 6 itens do MPCLSP.	92
Tabela 23 – GAPM das heurísticas para instâncias com 12 itens do MPCLSP.	93
Tabela 24 – GAPM das heurísticas para instâncias com 25 itens do MPCLSP.	94
Tabela 25 – GAPM das heurísticas para instâncias com 50 itens do MPCLSP.	95

Tabela 26 – GAPM das heurísticas para instâncias com 70 itens do MPCLSP.	96
Tabela 27 – GAPM das heurísticas para instâncias com 80 itens do MPCLSP.	97
Tabela 28 – GAPM das heurísticas para instâncias com 90 itens do MPCLSP.	98
Tabela 29 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 6 itens do MPCLSP.	99
Tabela 30 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 12 itens do MPCLSP.	100
Tabela 31 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 25 itens do MPCLSP.	101
Tabela 32 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 50 itens do MPCLSP.	102
Tabela 33 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 70 itens do MPCLSP.	103
Tabela 34 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 80 itens do MPCLSP.	104
Tabela 35 – Tempo gasto pelo <i>Gurobi</i> e pelas heurísticas para instâncias com 90 itens do MPCLSP.	105
Tabela 36 – Número de iterações das heurísticas para instâncias com 6 itens do MPCLSP.	106
Tabela 37 – Número de iterações das heurísticas para instâncias com 12 itens do MPCLSP.	107
Tabela 38 – Número de iterações das heurísticas para instâncias com 25 itens do MPCLSP.	108
Tabela 39 – Número de iterações das heurísticas para instâncias com 50 itens do MPCLSP.	109
Tabela 40 – Número de iterações das heurísticas para instâncias com 70 itens do MPCLSP.	110
Tabela 41 – Número de iterações das heurísticas para instâncias com 80 itens do MPCLSP.	111
Tabela 42 – Número de iterações das heurísticas para instâncias com 90 itens do MPCLSP.	112
Tabela 43 – GAP das heurísticas que utilizaram G e o MIPGap do <i>Gurobi</i> para instâncias do MLCLSP.	114
Tabela 44 – GAP das heurísticas que utilizaram G e o MIPGap do <i>Gurobi</i> para instâncias da classe A+ do MLCLSP.	115
Tabela 45 – GAP das heurísticas que utilizaram G e o MIPGap do <i>Gurobi</i> para instâncias da classe C+ do MLCLSP.	116
Tabela 46 – GAP das heurísticas que utilizaram G e o MIPGap do <i>Gurobi</i> para instâncias da classe E+ do MLCLSP.	117
Tabela 47 – GAP das heurísticas que utilizaram RF10 e o MIPGap do <i>Gurobi</i> para instâncias do MLCLSP.	118
Tabela 48 – GAP das heurísticas que utilizaram RF10 e o MIPGap do <i>Gurobi</i> para instâncias da classe A+ do MLCLSP.	119
Tabela 49 – GAP das heurísticas que utilizaram RF10 e o MIPGap do <i>Gurobi</i> para instâncias da classe C+ do MLCLSP.	120

Tabela 50 – GAP das heurísticas que utilizaram RF10 e o MIPGap do <i>Gurobi</i> para instâncias da classe E+ do MLCLSP.	121
Tabela 51 – GAP das heurísticas que utilizaram RF50 e o MIPGap do <i>Gurobi</i> para instâncias do MLCLSP.	122
Tabela 52 – GAP das heurísticas que utilizaram RF50 e o MIPGap do <i>Gurobi</i> para instâncias da classe A+ do MLCLSP.	123
Tabela 53 – GAP das heurísticas que utilizaram RF50 e o MIPGap do <i>Gurobi</i> para instâncias da classe C+ do MLCLSP.	124
Tabela 54 – GAP das heurísticas que utilizaram RF50 e o MIPGap do <i>Gurobi</i> para instâncias da classe E+ do MLCLSP.	125
Tabela 55 – Número de instâncias do MLCLSP que tiveram GAPini e GAP menor do que o MIPGap do <i>Gurobi</i> para heurísticas que utilizaram o G.	126
Tabela 56 – Número de instâncias do MLCLSP que tiveram GAPini e GAP menor do que o MIPGap do <i>Gurobi</i> para heurísticas que utilizaram o RF10.	127
Tabela 57 – Número de instâncias do MLCLSP que tiveram GAPini e GAP menor do que o MIPGap do <i>Gurobi</i> para heurísticas que utilizaram o RF50.	128
Tabela 58 – Tempo gasto pelas heurísticas que utilizaram o G e pelo <i>Gurobi</i> para MLCLSP.	129
Tabela 59 – Tempo gasto pelas heurísticas que utilizaram o RF10 e pelo <i>Gurobi</i> para MLCLSP.	130
Tabela 60 – Tempo gasto pelas heurísticas que utilizaram o RF50 e pelo <i>Gurobi</i> para MLCLSP.	131
Tabela 61 – Número de iterações realizadas pelas heurísticas que utilizaram o G para MLCLSP.	132
Tabela 62 – Número de iterações realizadas pelas heurísticas que utilizaram o RF10 para MLCLSP.	133
Tabela 63 – Número de iterações realizadas pelas heurísticas que utilizaram o RF50 para MLCLSP.	134

LISTA DE ABREVIATURAS E SIGLAS

ADN *Automatically designed neighborhoods*

FO *Fix-and-optimize*

LSP *Lot Sizing Problem*

ML *Machine Learning*

MLCLSP LSP multietápia

MPCLSP LSP múltiplas plantas

RF *Relax-and-fix*

SUMÁRIO

1	INTRODUÇÃO	23
2	REVISÃO BIBLIOGRÁFICA	25
2.1	Métodos heurísticos clássicos	25
2.2	Introdução ao aprendizado de máquina (ML)	27
2.3	Combinação de técnicas de otimização e aprendizado de máquina (ML)	29
2.4	Problema de dimensionamento de lotes	32
2.4.1	<i>Problema de dimensionamento de lotes com múltiplas plantas (MP-CLSP)</i>	32
2.4.2	<i>Problema de dimensionamento de lotes multiestágio (MLCLSP)</i> . .	34
3	ABORDAGENS DA SOLUÇÃO	37
3.1	Construção de uma solução inicial	37
3.1.1	<i>Relax-and-fix (RF)</i>	37
3.2	Procedimentos de melhoria	40
3.2.1	<i>Fix-and-Optimize (FO)</i>	40
3.2.2	<i>Automatically designed neighborhoods (ADN)</i>	42
3.2.2.1	<i>Matriz de dissimilaridade</i>	43
3.2.2.2	<i>Algoritmos de agrupamento</i>	46
3.2.2.3	<i>Heurística ADN</i>	48
4	TESTES COMPUTACIONAIS	51
4.1	Resultados computacionais para o MPCLSP	51
4.1.1	<i>Análise das soluções: modelo matemático</i>	52
4.1.2	<i>Análise das soluções: heurísticas</i>	53
4.2	Resultados computacionais para o MLCLSP	62
4.2.1	<i>Análise das soluções: modelo matemático</i>	63
4.2.2	<i>Análise das soluções: heurísticas</i>	63
5	CONCLUSÕES E PERSPECTIVAS FUTURAS	71
	REFERÊNCIAS	73

APÊNDICE A	EXEMPLO DE COMO CALCULAR O PESO ENTRE DUAS ENTIDADES	77
APÊNDICE B	ANÁLISES DOS RESULTADOS DAS HEURÍSTICAS PARA O MPCLSP	83
B.1	GAP	84
B.2	GAPM	84
B.3	Tempo gasto	84
B.4	Número de iterações	84
APÊNDICE C	ANÁLISES DOS RESULTADOS DAS HEURÍSTICAS PARA O MLCLSP	113
C.1	GAP	113
C.2	Quantidade de soluções melhores do que o <i>Gurobi</i>	125
C.3	Tempo gasto	128
C.4	Número de iterações	131

INTRODUÇÃO

Na literatura, existem diversas heurísticas baseadas no particionamento do conjunto de variáveis do modelo matemático como, por exemplo, heurísticas *relax-and-fix* (RF) e *fix-and-optimize* (FO) sendo aplicadas principalmente em problema de dimensionamento de lotes (LSP, do inglês *Lot Sizing Problem*). Geralmente, as variáveis binárias, relacionadas com as preparações para a produção, são particionadas considerando períodos, itens ou recursos produtivos.

O LSP consiste em determinar as quantidades a serem produzidas por cada tipo de item, em cada período, ao longo de um horizonte de planejamento finito, de modo que as demandas sejam atendidas e que os recursos sejam utilizados da melhor maneira possível. No LSP, busca-se um plano de produção que minimize os custos de produção, de estoque e de preparação, e que atenda a demanda dos itens em cada período do horizonte finito de planejamento. Dependendo das características do ambiente produtivo, diversos itens podem ser considerado no problema em várias plantas produtivas com a ocorrência de transferência entre plantas ([SAMBASIVAN; SCHMIDT, 2002](#)), ou a produção em vários estágios ([BILLINGTON; MCCLAIN; THOMAS, 1983](#)) ([MAES; MCCLAIN; WASSENHOVE, 1991](#)), dentre outros. Na literatura, revisões gerais sobre o LSP podem ser encontradas em [Karimi, Ghomi e Wilson \(2003\)](#), [Brahimi et al. \(2006\)](#) e [Glock, Grosse e Ries \(2014\)](#).

Exemplos das abordagens de heurísticas baseadas em decomposição para LSP podem ser encontrados nos trabalhos de [Akartunali e Miller \(2009\)](#), [Furlan \(2011\)](#), [Furlan e Santos \(2017\)](#), [Helber e Sahling \(2010\)](#) e [Toledo et al. \(2015\)](#). [Furlan e Santos \(2017\)](#) desenvolveram um método híbrido usando um algoritmo de abelhas para determinar partições não intuitivas das variáveis a serem aplicadas na heurística FO. Recentemente, [Ghiani, Laporte e Manni \(2015\)](#) propuseram uma heurística que utiliza as informações da estrutura do problema e de uma solução inicial em um algoritmo de aprendizado não supervisionado. Isso é feito para auxiliar a definição da estrutura de vizinhança a ser usada no método de busca local que também é utilizada pelos autores.

Atualmente, encontram-se, na literatura, estudos com resultados promissores utilizando o aprendizado de máquina (ML, do inglês *Machine Learning*) na área de otimização, principalmente em problemas com o espaço de busca muito grande ou quando os dados não são fáceis de serem modelados. Então, diante disso, este trabalho de mestrado tem o objetivo de investigar outras formas de determinar partições dos conjuntos de variáveis inteiras e/ou binárias usando técnicas de ML. Dessa forma, visa melhorar a qualidade da solução resultante, bem como determinar partições não intuitivas, principalmente, na heurística de melhoria FO.

A abordagem de ML utilizada nesta pesquisa de mestrado foi o agrupamento, sendo fortemente baseado em modificações do trabalho de Ghiani, Laporte e Manni (2015). Esta técnica foi usada para ajudar a particionar o conjunto de variáveis de dois LSP com objetivo de obter resultados competitivos com os obtidos em relação às partições clássicas utilizadas na literatura. Dois algoritmos de agrupamentos foram investigados: *k-means* (JAIN; DUBES, 1988) e *k-medoids* (PARK; JUN, 2009) para determinar grupos balanceados e não balanceados, resultando, em partições diferentes, o que possibilitou investigar diversas versões das heurísticas.

Para testar as abordagens desenvolvidas, consideramos dois problemas clássicos: o LSP múltiplas plantas (Sambasivan e Schmidt (2002)) (MPCLSP) e o LSP multiestágio (Maes, McClain e Wassenhove (1991)) (MLCLSP). Ghiani, Laporte e Manni (2015) aplicaram a heurística desenvolvida para o problema com múltiplas plantas, porém não no contexto de uma heurística do tipo FO.

Para esses problemas, os testes computacionais, com as instâncias da literatura propostas em Carvalho e Nascimento (2016) e em Stadtler e Sürie (2000), mostraram que as abordagens desenvolvidas são promissoras quando se compara às soluções obtidas com as obtidas pelo *branch-and-cut* de um solver comercial utilizado, tanto em tempo quanto em qualidade das soluções. Para o problema multiestágio, usamos as ideias propostas em Ghiani, Laporte e Manni (2015) para determinar partições por itens e por períodos.

Nesta dissertação será apresentada, no Capítulo 2, uma breve revisão sobre algumas abordagens clássicas de heurísticas de decomposição, também uma breve introdução sobre ML, destacando alguns métodos que combinam essas técnicas de otimização e ML. Também descrevemos brevemente os problemas de dimensionamento de lotes em estudo e os modelos matemáticos utilizados. As abordagens de solução desenvolvidas são apresentadas no Capítulo 3 e os experimentos computacionais no Capítulo 4. As conclusões e perspectivas são descritas no Capítulo 5. No Apêndice A, apresentamos um exemplo de como são obtidas as informações para serem utilizadas nos algoritmos de agrupamento. No Apêndice B e no Apêndice C, apresentamos a análise dos resultados das abordagens para ambos os problemas de dimensionamento de lotes com mais detalhes.

REVISÃO BIBLIOGRÁFICA

Este capítulo apresenta os métodos heurísticos clássicos baseados em decomposição, principalmente os que são aplicados no LSP. Em seguida, há uma introdução ao ML e alguns trabalhos que combinam técnicas de otimização e ML. Depois, apresentaremos dois LSPs clássicos, nos quais aplicamos as abordagens de solução apresentadas no [Capítulo 3](#).

2.1 Métodos heurísticos clássicos

Os problemas de otimização inteira e/ou mista são problemas que podem ser de natureza NP-difícil. Com isso, métodos exatos tendem a ter maior dificuldade para resolvê-los. Dessa maneira, diversos métodos heurísticos foram desenvolvidos para lidar com esses tipos de problemas como, por exemplo, heurísticas construtivas, heurísticas de busca local e meta-heurísticas ([ARENALES et al., 2007](#)).

As heurísticas construtivas têm o objetivo de construir uma solução para um dado problema, adicionando, a cada passo, um elemento ou componentes da solução. Um exemplo dessa heurística é o algoritmo guloso que escolhe, a cada passo da construção da solução, os componentes da solução que leva a melhor vantagem localmente. Heurísticas construtivas podem ser baseadas na formulação matemática dos problemas, como as heurísticas *LP-and-fix* e RF.

A heurística do tipo *LP-and-fix* tenta construir uma solução factível a partir de uma solução da relaxação linear do problema, ou seja, as condições de integralidade são relaxadas e o problema relaxado é resolvido. Da solução obtida, fixam-se as variáveis com valores inteiros, e o MIP resultante é resolvido gerando uma solução heurística. Caso não seja obtida uma solução factível, significa que a heurística falhou ([POCHET; WOLSEY, 2006](#)). Uma variação dessa heurística utiliza um parâmetro para considerar uma faixa de valores para que as variáveis binárias sejam arredondadas e fixadas como feito em [Furlan \(2011\)](#) para o MLCLSP.

A heurística do tipo *relax-and-fix* (RF) particiona o conjunto Q das variáveis binárias em

R conjuntos disjuntos $Q_1, \dots, Q_R$. Para cada r , é definido um subproblema inteiro misto MIP^r associado ao conjunto Q_r com variáveis consideradas inteiras e o restante das variáveis ($Q \setminus Q_r$), que não foram fixadas, são relaxadas, nas quais $r = 1, \dots, R$. Dessa forma, a heurística resolve o problema de forma iterativa, ou seja, logo após a resolução do subproblema MIP^1 . Caso a solução seja factível, são fixadas as variáveis que pertencem a Q_1 considerando a solução encontrada. Na próxima iteração, teremos o subproblema MIP^r com $r = 2, \dots, R$, nas quais as variáveis que pertencem a Q_{r-1} foram fixadas utilizando a solução obtida por MIP^{r-1} , caso a solução seja factível, senão a iteração é interrompida. Quando a iteração chegar ao passo $r = R$, a solução de MIP^R é a solução heurística para o problema original. Caso a solução não seja factível, isso significa que a heurística falhou (POCHET; WOLSEY, 2006).

No particionamento do conjunto de variáveis Q , pode ser considerada a sobreposição (do inglês, *overlapping*) de tal forma que no passo r , o conjunto de variáveis $Q_1 \cup \dots \cup Q_{r-1}$ que foram fixadas e o conjunto Q_r e a sobreposição U_r contida em $Q_{r-1} \cup \dots \cup Q_R$ são integralizados e as variáveis contidas em $Q \setminus U_r$ que não foram fixadas são reais (FURLAN, 2011).

As heurísticas de busca local procuram melhorar uma solução inicial dada (que pode ser obtida por um algoritmo construtivo), fazendo uma busca em sua vizinhança até não encontrar nenhum vizinho melhor que a solução atual ou até satisfazer algum outro critério de parada. Alguns exemplos de heurísticas de busca local são o *local branching*, que combina busca local com estratégias de ramificação (FISCHETTI; LODI, 2003), *RINS* (Relaxation Induced Neighborhood Search), que busca uma solução melhor na vizinhança da solução obtida pelo problema relaxado (DANNA; ROTHBERG; PAPE, 2005) e *feasibility pump*, que, a partir de uma solução $\tilde{x}$ obtida pelo arredondamento da solução da relaxação linear de um problema, é realizado uma busca para encontrar um x^* inteiro que seja o mais próximo possível de $\tilde{x}$ (FISCHETTI; GLOVER; LODI, 2005). Tais heurísticas são também baseadas na formulação matemática do problema.

Outra heurística de melhoria é a *fix-and-optimize* (FO), que é parecida com o RF em relação às estratégias de particionamento das variáveis. Porém, a FO utiliza uma solução factível do problema e tenta melhorá-la, e a RF tenta construir uma solução factível utilizando uma solução da relaxação linear de um problema. Dada uma solução inicial factível, o conjunto de variáveis inteiras Q é particionado em R conjuntos disjuntos $Q_1, \dots, Q_R$ (pode ser considerado a sobreposição também). A cada iteração r com $1 \leq r \leq R$, as variáveis do conjunto Q_r são otimizadas e as demais são fixadas no valor da solução incumbente. Depois, o problema resultante é resolvido sendo verificado se a solução obtida é melhor que a solução incumbente. Caso seja, a solução obtida será considerada a solução incumbente (FURLAN, 2011).

Existem diversos trabalhos, principalmente na área de planejamento da produção, que desenvolveram abordagens utilizando RF para encontrar uma solução inicial e, para melhorar essa solução, aplicaram o FO. Com essas duas etapas e abordando diversas maneiras de particionar as variáveis dos problemas, existem trabalhos como o de Toledo *et al.* (2015) para o MLCLSP,

Stadtler e Sahling (2013) para problemas de dimensionamento e sequenciamento de lotes multiestágio sem *lead time* fixo e de Cunha, Kramer e Melo (2019) para LSP com capacidade para vários itens com remanufatura que, além de apresentar o RF com FO, propuseram uma heurística de arredondamento de geração de colunas com *LP-and-fix*.

As meta-heurísticas têm o objetivo de produzir soluções diferentes das soluções obtidas pelas heurísticas de busca local, ou seja, para sair do mínimo ou máximo local. Existem dois tipos principais de meta-heurísticas: uma baseada em uma solução única e a outra baseada em população. Alguns exemplos de meta-heurísticas baseadas em solução única são o *simulated annealing*, que seria uma extensão da busca local que permite escolher soluções inferiores da solução atual com objetivo de sair do mínimo ou máximo local (KIRKPATRICK, 1984), e a busca tabu que utiliza informações obtidas nas buscas já realizadas para obter alguma vantagem na busca atual (GLOVER, 1986).

Alguns exemplos de meta-heurísticas baseadas em população são algoritmos evolucionários como algoritmo genético, baseado na seleção natural e reprodução genética (HOLLAND *et al.*, 1992); colônia de formigas, baseada no comportamento das formigas que são agentes que buscam alimento, já no caso do algoritmo, há soluções factíveis com objetivo de procurar a melhor solução possível (DORIGO; MANIEZZO; COLORNI, 1996); e algoritmo de abelhas baseado na forma como as abelhas procuram alimento (PHAM *et al.*, 2005). Existem também métodos que combinam meta-heurística com heurísticas de melhoria que utilizaram o algoritmo de abelhas para obter as partições das variáveis utilizadas pelo FO para melhorar a solução, como o proposto em Furlan e Santos (2017) para o MLCLSP com limitação de capacidade.

Além desses tipos de heurísticas, temos as matheurísticas, que combinam o uso de técnicas de métodos exatos e meta-heurísticas. São classificadas em dois tipos: as matheurísticas do tipo (i) são quando as meta-heurísticas controlam as chamadas dos métodos exatos como, por exemplo, nos algoritmos quando a definição da vizinhança é feita pela meta-heurística e a exploração da vizinhança é feita por um método exato. E as matheurísticas do tipo (ii) são quando os métodos exatos controlam as chamadas e controlam o uso de meta-heurística como, por exemplo, os *solvers* de *Branch-and-cut* que utilizam o potencial das meta-heurísticas para tentar buscar soluções de boa qualidade com tempo gasto e menor custo computacional (CASERTA; VOSS, 2009).

Antes de vermos a combinação das técnicas de ML e técnicas de otimização, apresentaremos, na seção a seguir, uma introdução ao ML.

2.2 Introdução ao aprendizado de máquina (ML)

Atualmente, o ML está sendo aplicado em várias áreas com sucesso como classificação de imagens, classificação de sentimentos em textos, desenvolvimento de inteligência artificial em jogos, e tem sido amplamente utilizado em problemas de otimização. Nesta seção serão

apresentados brevemente alguns conceitos sobre o ML.

O ML estuda o processo de “aprender” no abstrato com objetivo de obter regras generalizadas a partir da observação de exemplos vistos e exemplos não vistos para realizar alguma tarefa (LUXBURG; SCHÖLKOPF, 2011). Lodi e Zarpellon (2017) apresentam uma breve revisão sobre os três tipos de ML, os quais são descritos brevemente a seguir.

Os algoritmos de aprendizado supervisionado são aqueles que utilizam dados de entrada com exemplos que possuem rótulos da classe correspondente e o objetivo é realizar tarefas de predição. Há dois tipos de tarefas principais: classificação que utiliza dados de entrada que possuem rótulos qualitativos e a regressão, que usa os dados de entrada que possuem os rótulos quantitativos. Em Mello e Ponti (2018), foi apresentado um algoritmo de classificação da seguinte forma: dado um objeto ou um exemplo $x_i \in X$, em que x_i representa as características do exemplo i e $y_i \in Y$ representa o rótulo da classe correspondente, o algoritmo tem o objetivo de encontrar o melhor classificador possível tal que $f : X \rightarrow Y$ represente o conjunto de exemplos dado.

Algoritmos de aprendizado não supervisionado são aqueles os quais utilizam dados de entrada que não possuem classes ou rótulos e a tarefa é encontrar uma função aproximada que os representa. Há três tipos de tarefas principais: a estimativa de densidade, cuja função é um estimador de distribuição do conjunto de dados de entrada; o agrupamento, cuja proposta é descobrir similaridades entre os elementos dos dados de entrada e assim dividi-lo em um ou mais agrupamentos (do inglês, *clusters*), e a redução de dimensionalidade, cujo objetivo é identificar características importantes nos dados de entrada via seleção ou extração após a redução de dimensionalidade.

Aprendizado por reforço são algoritmos que utilizam agentes os quais conseguem interagir em um ambiente incerto e podem tomar ações de mudança de estado que resultem num sinal de recompensa (ou de punição) através da avaliação do seu comportamento. Assim cada ação que possui alguma probabilidade de ser realizada afeta as entradas posteriores do sistema, e, portanto, as recompensas subsequentes. O objetivo é encontrar uma *policy* (política) que maximize o retorno de recompensas a longo prazo. Na Figura 1, temos um esquema simplificado do aprendizado por reforço.

Figura 1 – Esquema simplificado do aprendizado por reforço.

Fonte: Adaptada de Bengio, Lodi e Prouvost (2020).

2.3 Combinação de técnicas de otimização e aprendizado de máquina (ML)

Bengio, Lodi e Prouvost (2020) fizeram uma revisão sobre os artigos que combinam técnicas de otimização combinatória e ML. O foco foi observar os pontos relevantes dos problemas combinatoriais NP-difícil como sendo um conjunto de dados de entrada para ser utilizado no ML sobre uma determinada tarefa. São citados os três tipos de estruturas de algoritmos: o aprendizado que, atua sozinho para obter uma solução para o problema (não é o ideal); o ML utilizado para obter informações valiosas para um algoritmo de otimização; e o algoritmo de otimização combinatória, que consulta o ML constantemente para tomar decisões.

De uma forma detalhada, Talbi (2020) fez uma pesquisa em diversos artigos relacionados a ML em meta-heurísticas construindo a taxonomia de meta-heurística orientada a dados (do inglês, *Data-driven metaheuristics*). Esses dados gerados pelas meta-heurísticas no processo de busca podem ser estáticos ou dinâmicos. São estáticos se estiverem relacionados ao problema alvo e suas características, e são dinâmicos se os dados são gerados durante o processo de busca iterativo. As aplicações do ML em meta-heurísticas foram separadas em três grupos: nível-problema, nível-baixo e nível-alto.

No nível-problema, a análise é composta pelas seguintes perguntas: quais características? Quais objetivos? Quais métodos de ML? Nesse nível, o ML ajuda na parte da modelagem e decomposição de problemas.

Na parte da modelagem, o ML pode ajudar a modelar a função objetivo para melhorar a convergência e reduzir o custo computacional como, por exemplo, função objetivo aprendível que gera automaticamente funções objetivo a partir da exploração de conhecimento do problema de otimização e características extraídas durante os estados do processo de busca. Existem também as aproximações para as funções objetivo caras e podem ser classificadas em duas formas: aproximação baseada na similaridade (aprendizado preguiçoso ou aprendizado baseado em memória) e aproximação baseada em funções (construção de um novo modelo da função objetivo). E outra parte que o aprendizado pode ajudar na modelagem é manipulando restrições através da aproximação, transformação e manutenção de restrições. Na parte da decomposição de problemas temos a decomposição dos dados de entrada, decomposição de variáveis em subconjuntos independentes e decomposição de problemas multiobjetivos.

No nível-baixo, o ML pode ser utilizado para obter a solução inicial aplicando na incorporação de conhecimento por intensificação, fixação de posições por diversificação e geração de soluções iniciais. Também pode elaborar operadores de pesquisa (do inglês, *Search operators design*): operadores construtivos (por exemplo, busca gulosa), unários (por exemplo, vizinhança de busca local), binários (por exemplo, cruzamento em algoritmos de evolução), indiretos (por exemplo, modelos probabilísticos em algoritmos de distribuição), de intensificação e de diversificação de busca. Além disso, pode realizar ajustes de parâmetros: *offline* ou estático

(antes de executar a meta-heurística) e *online* ou dinâmica (muda os parâmetros durante a busca).

No nível-alto, o ML pode ser utilizado na seleção e geração de meta-heurísticas e na elaboração de meta-heurísticas cooperativas e paralelas.

Outro critério que pode ser considerado no momento do aprendizado: *offline*, quando o processo do ML ocorre antes de resolver o problema e *online*, quando o ML ganha conhecimento durante o processo de busca enquanto resolve o problema. A seguir, serão apresentados vários trabalhos que combinaram técnicas de otimização e ML.

Assim, temos aprendizados de máquinas que visam melhorar a modelagem usando ML em problemas de otimização combinatória. Lombardi e Milano (2018) revisaram esses tipos de abordagens propostos na literatura. Existem casos que incorporam plenamente os modelos de ML em modelo de otimização combinatória e, em casos extremos, podem substituí-los completamente. Foram citados trabalhos que utilizaram informações de um conjunto de exemplos para obter uma restrição ou uma função objetivo. Como exemplo, há o aprendizado via *Native Constraint Language* que substitui um especialista em otimização por um algoritmo para obter restrições. O ML também pode ser utilizado durante a resolução de problema, cujas soluções candidatas promissoras são identificadas e avaliadas para que o modelo interno seja atualizado e o processo se repete; nota-se que o objetivo principal é obter uma solução e não o modelo aprendido.

Outra estratégia é utilizar o ML para imitar e aprender a intuição dos métodos clássicos como em Alvarez, Louveaux e Wehenkel (2017) que utilizaram o *Extremely Randomized Trees*, em que são baseados num conjunto de árvores de regressão para aprender uma função a partir dos dados. Os dados são observações dos problemas obtidos em cada nó escolhido pelo método *strong-branching* e foram baseados no problema de otimização inteira mista binária.

As informações dos dados são calculadas através de um *score* para cada variável fracionária i baseado no vetor de custo, matriz das restrições, limitantes superiores e inferiores. As características são baseadas no vetor de custo e a matriz das restrições são multiplicadas por um fator de forma que sejam independentes da escala do problema.

Já Afia e Kabbaj (2017) utilizaram *Support Vector Machine (SVM)*, que é normalmente utilizado para operações de classificação e regressão e tem por objetivo encontrar um hiperplano com a melhor margem, ou seja, realizar a separação entre as classes. O *SVM* foi utilizado para aprender as estratégias de ramificação de variáveis e de seleção de nós explorando as informações obtidas pelo *branch-and-cut*. As características baseadas em nós são: ganho da redução da função objetivo, profundidade da árvore de *branch-and-bound*, estimativa do nó e o valor da função objetivo do problema relaxado. E as características baseadas em variáveis são: valor do pseudo-custo e custos reduzidos positivos e negativos.

Em He, III e Eisner (2014), a estratégia do método é aprender imitando o método *branch-and-bound* e, para isso, propuseram uma heurística automática de busca e adaptável para uma

família de problemas, ao invés de ser para cada problema. No método, são usados pequenos conjuntos de problemas resolvidos no treinamento e, no teste, são utilizados problemas que são do mesmo tipo do treinamento. No treinamento, são desenvolvidos dois tipos de *policy*:

- O *node selection policy*, em que se seleciona o nó ativo que possui o maior *score*, ou seja, o que tem maior prioridade. Nas características principais dos nós, são considerados o limite inferior, valor da função objetivo estimado e a profundidade. E, para as características da ramificação são considerados o pseudo-custo, diferença no valor da variável na solução LP atual e na solução LP raiz, diferença entre o seu valor e o seu limite atual.
- O *node pruning policy* decide se descarta ou expande considerando o progresso do *solver* atual. As características principais consideradas na ramificação nas árvore são limitantes inferiores e superiores globais, *integrality gap*, número de soluções encontradas, se o *gap* é infinito.

Podemos utilizar o ML para melhorar os métodos de otimização, como em [Tang, Agrawal e Faenza \(2019\)](#) que utilizaram aprendizado por reforço sendo projetado para ser uma sub-rotina para o método de planos de corte com objetivo de melhorá-lo. Na formulação do plano de corte como aprendizado por reforço, o espaço de estado que consideram é a região factível, os custos, solução ótima do problema relaxado e conjunto de cortes candidatos de *Gomory*. O espaço de ação é dado pelo conjunto de cortes candidatos de *Gomory*. A recompensa é construída de forma que gere cortes agressivos para chegar à solução ótima mais rápido, e a transição de um estado para outro ocorre quando se acrescenta algum corte.

Uma aplicação bem interessante é utilizar o ML para auxiliar na determinação de partições de forma não intuitiva em um procedimento de melhoria baseado em modelagem semelhante ao procedimento do tipo *fix-and-optimize*. Essa técnica foi utilizada em [Ghiani, Laporte e Manni \(2015\)](#) com um algoritmo de agrupamento. Primeiro, são obtidas as soluções iniciais a partir de um resolvedor de otimização, depois são obtidas as informações a partir dessa solução e da estrutura do modelo para serem utilizadas pelo algoritmo de agrupamento, no caso *k-means* balanceado para obter agrupamentos com aproximadamente a mesma quantidade de elementos. Depois, são definidas as vizinhanças a partir das soluções dos subproblemas obtidos por meio das partições.

Mais recentemente, [Adamo et al. \(2020\)](#) desenvolveram um método, a partir de uma solução da relaxação linear de um problema, tem o objetivo de encontrar uma solução factível. [Adamo et al. \(2020\)](#) também utilizam informações da solução do problema relaxado e a estrutura do problema para determinar as partições usando *k-medoids* balanceado.

Observando os trabalhos da literatura, há diversos métodos heurísticos relacionados à decomposição de variáveis, e as escolhas das partições nem sempre é uma tarefa fácil, pois normalmente são necessários vários testes com diversos tipos de decomposições para encontrar

os que têm resultados melhores para cada problema. Como atualmente observamos várias aplicações do ML com técnicas de otimização, a proposta de utilizar o ML é interessante, pois há as informações do problema e/ou da solução para encontrar essas partições de forma automática.

2.4 Problema de dimensionamento de lotes

Agora, apresentaremos dois LSPs clássicos, para os quais aplicamos as abordagens de solução apresentadas no [Capítulo 3](#). Na [Subseção 2.4.1](#), apresentamos o modelo matemático do MPCLSP e, na [Subseção 2.4.2](#), apresentaremos o modelo matemático do MLCLSP.

2.4.1 Problema de dimensionamento de lotes com múltiplas plantas (MPCLSP)

O LSP consiste em determinar as quantidades de itens a serem produzidas em cada período ao longo de um horizonte de planejamento finito, de forma que as demandas de cada período sejam atendidas e que os recursos disponíveis sejam utilizados da melhor maneira possível. No MPCLSP, o objetivo é determinar um plano de produção que minimize os custos de produção, de estoque, de preparação para produção dos itens e de transferência de itens entre plantas, visando o atendimento da demanda em cada planta, em cada período de tempo.

Na literatura, encontramos diversos trabalhos que lidam com esse problema, como, por exemplo, [Melega, Fiorotto e Araujo \(2013\)](#) que investigaram formulações fortes a partir do modelo proposto por [Sambasivan e Schmidt \(2002\)](#); [Nascimento, Resende e Toledo \(2010\)](#) que desenvolveram uma heurística GRASP (do inglês, *Greedy Randomized Adaptive Search*) e GRASP com *path-relinking*; [Carvalho e Nascimento \(2016\)](#) que desenvolveram uma heurística lagrangiana e melhoraram os resultados obtidos em [Nascimento, Resende e Toledo \(2010\)](#); [Carvalho e Nascimento \(2018\)](#); que consideraram o problema com *setup carry-over*; e [Ghiani, Laporte e Manni \(2015\)](#) que propuseram uma heurística baseada em vizinhança.

A seguir, é apresentado o modelo matemático proposto em [Sambasivan e Schmidt \(2002\)](#) para o MPCLSP com múltiplos itens e múltiplos períodos. Nessa modelagem, [Sambasivan e Schmidt \(2002\)](#) consideraram que todos os itens podem ser produzidos em todas as plantas, mas que pode ocorrer transferência de produção entre fábricas para o atendimento da demanda. Os custos de preparação de máquina, produção de itens e estoque variam conforme os itens envolvidos, a planta e o período de planejamento. Já os custos de transferência entre plantas variam conforme as plantas envolvidas e o período.

Parâmetros :

- *NI*: número de itens;
- *NM*: número de plantas;

- NT : número de períodos;
- $N = \{1, \dots, NI\}$: conjunto de índices dos itens;
- $M = \{1, \dots, NM\}$: conjunto de índices das plantas;
- $T = \{1, \dots, NT\}$: conjunto de índices dos períodos;
- cap_{jt} : capacidade produtiva da planta j no período t , com $j \in M$ e $t \in T$;
- b_{ijt} : tempo de produção do item i na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- f_{ijt} : tempo de preparação do item i na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- s_{ijt} : custo de preparação do item i na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- c_{ijt} : custo de produção do item i na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- h_{ijt} : custo de estoque do item i na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- d_{ijt} : demanda do item i na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- $r_{jj't}$: custo de transferência de uma unidade de um item da planta j para a planta j' no período t , com $j, j' \in M$ e $t \in T$;

Variáveis de decisão :

- X_{ijt} : quantidade do item i produzida na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- I_{ijt} : quantidade do item i em estoque na planta j no período t , com $i \in N$, $j \in M$ e $t \in T$;
- $Q_{ijj't}$: quantidade do item i transferido da planta j para a planta j' no período t , com $i \in N$, $j, j' \in M$ e $t \in T$;
- Y_{ijt} : variável binária é igual a 1 se o item i é produzido na planta j e no período t , caso contrário é igual a 0, com $i \in N$, $j \in M$ e $t \in T$.

Função objetivo e restrições :

$$\text{Minimize } \sum_{i \in N} \sum_{j \in M} \sum_{t \in T} (c_{ijt}X_{ijt} + s_{ijt}Y_{ijt} + h_{ijt}I_{ijt} + \sum_{j' \in M, j' \neq j} r_{jj't}Q_{ijj't}) \quad (2.1)$$

$$\text{sujeito a } I_{i,j,t-1} + X_{ijt} - \left(\sum_{j' \in M, j' \neq j} Q_{ijj't} \right) + \left(\sum_{j' \in M, j' \neq j} Q_{ij'jt} \right) - I_{ijt} = d_{ijt}, \quad \forall i, j, t, \quad (2.2)$$

$$X_{ijl} \leq \left(\sum_{l, l \geq t} d_{ijl} \right) Y_{ijt}, \quad \forall i, j, t, \quad (2.3)$$

$$\sum_{i \in N} (b_{ijt}X_{ijt} + f_{ijt}Y_{ijt}) \leq cap_{jt}, \quad \forall j, t, \quad (2.4)$$

$$I_{ij0} = 0, \quad \forall i, j, \quad (2.5)$$

$$X_{ijt}, I_{ijt}, Q_{ijj't} \geq 0, \quad \forall i, j, j', t, \quad (2.6)$$

$$Y_{ijt} \in \{0, 1\}, \quad \forall i, j, t. \quad (2.7)$$

Na formulação (2.1)-(2.7), a função objetivo (2.1) busca minimizar o custo de produção, o custo de preparação, o custo de estoque e o custo de transferência entre plantas. Deve-se produzir todos os itens necessários nas plantas disponíveis para atender a demanda no horizonte de planejamento, obedecendo às restrições (2.2)-(2.7).

As restrições em (2.2) são responsáveis pelo balanceamento do estoque de cada item i em cada período t e planta j para garantir que a demanda (d_{ijt}) seja atendida considerando as quantidades estocadas no período anterior ($I_{i,j,t-1}$), a quantidade do item i produzida (X_{ijt}), a quantidade do item i transferida da planta j para outras plantas ($\sum_{j' \in M, j' \neq j} Q_{ij'jt}$), a quantidade do item i transferida de outras plantas para a planta j ($\sum_{j' \in M, j' \neq j} Q_{ij'jt}$) e a quantidade do item i que será estocada no fim do período atual (I_{ijt}).

As restrições em (2.3) garantem que, se houver produção do item i na planta j no período t , a variável binária (Y_{ijt}) será igual a 1. Assim, a preparação do item i na planta j , no período t é considerada. As restrições em (2.4) garantem que a solução não ultrapasse o limite da capacidade de produção da planta j no período t . As restrições em (2.5) indicam não haver estoques iniciais ($t = 0$). Por fim, as restrições (2.6) e (2.7) definem o domínio das variáveis.

2.4.2 Problema de dimensionamento de lotes multiestágio (MLCLSP)

O MLCLSP ocorre em meios produtivos nos quais, para a produção de determinados itens, há a necessidade de itens componentes que devem ser produzidos ou adquiridos. Ou seja, alguns itens podem ter outros itens como componentes, os quais podem ser componentes de outros itens de acordo com uma estrutura de produto, como a apresentada pela Figura 2, na qual temos os itens representados de 1 a 5. Na figura, temos que o item 3 e 4 são componentes necessários para produzir o produto final 1, e o item 4 e 5 são componentes necessários para produzir o produto final 2.

Existem várias pesquisas sobre técnicas de resolução para o problema multiestágio considerando variações na estrutura do produto e/ou na limitação de capacidade. Pesquisas sobre problemas de MLCLSP e extensões podem ser encontradas em Akartunali e Miller (2009), Furlan (2011), Furlan e Santos (2017), Helber e Sahling (2010), Toledo *et al.* (2015). Furlan e Santos (2017) e Maes, McClain e Wassenhove (1991). Dentre esses trabalhos, destacam-se os de Helber e Sahling (2010), Toledo *et al.* (2015) e Furlan e Santos (2017), pois desenvolveram abordagens semelhantes às estudadas nesta dissertação para lidar com o problema.

A seguir, será apresentado o modelo matemático descrito em (MAES; MCCLAIN; WASSENHOVE, 1991) para o problema clássico de MLCLSP com restrição de capacidade e considerando a possibilidade do uso de horas extras.

Figura 2 – Estrutura de produto

Fonte: Elaborada pela autora.

Parâmetros :

- NK : número de recursos;
- $S(i)$: conjunto de itens que são sucessores ao item i , com $i \in N$;
- K_j : conjunto de produtos que usa o recurso j , com $j \in M$;
- B : valor suficientemente grande;
- cs_i : custo de preparação do item i , com $i \in N$;
- h_i : custo do estoque do item i para mantê-lo por um período, com $i \in N$;
- oc_{jt} : custo de hora extra do recurso j no período t , com $j \in M$ e $t \in T$;
- $r_{ii'}$: número de unidades do item i necessário para produzir uma unidade de i' , com $i, i' \in N$;
- d_{it} : demanda externa para o item i no período t , com $i \in N$ e $t \in T$;
- a_i : tempo de produção de uma unidade do item i , com $i \in N$;
- ts_i : tempo de preparação para a produção do item i , com $i \in N$;
- C_{jt} : capacidade de recurso j disponível no período t , com $j \in M$ e $t \in T$.

Variáveis de decisão :

- O_{jt} : quantidade de hora extra do recurso j no período t , com $j \in M$ e $t \in T$;
- X_{it} : quantidade do item i produzida no período t , com $i \in N$ e $t \in T$;
- I_{it} : estoque do item i no final do período t , com $i \in N$ e $t \in T$;
- Y_{it} : variável binária é igual a 1 se o item i é produzido no período t , caso contrário é igual a 0, com $i \in N$ e $t \in T$.

Função objetivo e restrições :

$$\text{Minimize} \quad \sum_{i \in N} \sum_{t \in T} (cs_i Y_{it} + h_i I_{it}) + \sum_{j \in M} \sum_{t \in T} oc_{jt} O_{jt} \quad (2.8)$$

$$\text{sujeito a } I_{i,t-1} + X_{it} - \sum_{i' \in S(i)} r_{ii'} X_{i't} - I_{it} = d_{it}, \quad \forall i, t, \quad (2.9)$$

$$\sum_{i \in K_j} (a_i X_{it} + t s_i Y_{it}) \leq C_{jt} + O_{jt}, \quad \forall j, t, \quad (2.10)$$

$$X_{it} \leq B Y_{it}, \quad \forall i, t, \quad (2.11)$$

$$X_{it}, I_{it} \geq 0, \quad \forall i, t, \quad (2.12)$$

$$Y_{it} \in \{0, 1\}, \quad \forall i, t, \quad (2.13)$$

Na formulação (2.8)-(2.12), a função objetivo (2.8) busca minimizar a soma dos custos de preparação, de estoque e de horas extras obedecendo às restrições dadas por (2.9) a (2.13). As restrições em (2.9) são responsáveis pelo balanceamento do estoque para cada item i no período t , nas quais são consideradas as quantidades estocadas no período anterior ($I_{i,t-1}$), a produção do período (X_{it}), no final do período atual (I_{it}), demanda interna ($\sum_{i' \in S(i)} r_{ii'} X_{i't}$) e a demanda externa (d_{it}). A demanda interna representa a demanda de produção do item i que deve ser produzida antes da produção de seus sucessores $S(i)$, de forma que atenda suas demandas por componentes. As restrições em (2.10) são responsáveis pela limitação da capacidade de cada recurso j disponível mais a quantidade de horas extras disponíveis em cada período t . As restrições em (2.11) asseguram a necessidade de preparação caso haja produção do item i no período t . O B foi definido de forma que seja igual a 10000. As restrições em (2.13) e (2.12) definem o domínio das variáveis.

ABORDAGENS DA SOLUÇÃO

Neste capítulo, apresentamos as abordagens heurísticas propostas para lidar com o MPCLSP e o MLCLSP, as quais consistem em duas etapas: construção (obtenção) de uma solução inicial e sua melhoria. A solução inicial utilizada é obtida de duas formas: a primeira através de um resolvedor de otimização, o *Gurobi* (G), e a segunda é obtida usando a heurística RF, na sua versão clássica, bastante usada na literatura para diversos problemas de planejamento da produção.

Os procedimentos de melhoria são baseados na decomposição das variáveis binárias e resolução de problemas inteiros menores utilizando um resolvedor. Para esse procedimento, foi utilizada a heurística FO na sua versão clássica, também bastante usada na literatura para diversos problemas de planejamento da produção. Outra heurística usada é a vizinhança construída de forma automática (ADN, do inglês *automatically designed neighborhoods*), que também é uma heurística do tipo FO, porém constrói uma vizinhança de forma automática utilizando ML não supervisionado, a qual é utilizada para particionar as variáveis binárias.

Na [Figura 3](#), temos um fluxograma simplificado da abordagem escolhida e, nas próximas seções, são apresentados os passos e as heurísticas de melhoria detalhadamente.

3.1 Construção de uma solução inicial

3.1.1 Relax-and-fix (RF)

Dado um *MIP* com variáveis binárias, define-se Q como o conjunto de índices relacionado com as variáveis binárias selecionadas. Supomos que seja possível particionar Q em R conjuntos disjuntos $Q_1, \dots, Q_R$.

A heurística RF tenta encontrar uma solução definindo subproblemas a partir desses conjuntos de forma iterativa e sequencial. Nela, todas as variáveis relacionadas com Q são

Figura 3 – Fluxograma da abordagem da solução simplificado

Fonte: Elaborada pela autora.

relaxadas. Na iteração $r = 1$, é definido um subproblema MIP^1 em que todas as variáveis relacionadas com os índices em Q_1 são mantidas no seu domínio original, sendo usado um resolvidor para solucionar esse subproblema. Na próxima iteração r , com $r = 2, \dots, R$, é definido um subproblema MIP^r em que as variáveis relacionadas com Q_r são mantidas no seu domínio original e as relacionadas com Q_{r-1} são fixadas com solução obtida do MIP^{r-1} caso a solução seja factível; senão, a iteração é interrompida. Quando a iteração chegar ao passo $r = R$, a solução de MIP^R é a solução heurística para o problema original. Caso a solução não seja factível, significa que a heurística falhou (POCHET; WOLSEY, 2006).

Na parte do particionamento do conjunto de índices Q , pode ser considerado o *overlapping* $U_r = Q_r \cap Q_{r-1} \neq \emptyset$, com $r = 1, \dots, R-1$. Assim, na iteração r , com $1 < r < R$, as variáveis relacionadas a $Q_{r-1} \setminus U_{r-1}$ são fixadas. As que são relacionadas com conjunto Q_r são mantidas no domínio original e o resto é relaxado (FURLAN, 2011). O Algoritmo 1 apresenta o pseudocódigo do RF considerando a sobreposição.

Algoritmo 1 – Pseudocódigo da heurística RF considerando particionamento com sobreposição

```

1: Entrada:  $Q_r$  e  $U_r$ , com  $r = 1, \dots, R$ ;
2: Relaxe as variáveis binárias relacionadas a  $Q$ ;
3: para  $r \leftarrow 1$  to  $R$  faça
4:   se  $r > 1$  então
5:     Fixe as variáveis relacionadas com  $Q_{r-1} \setminus U_{r-1}$  com a solução obtida do  $MIP^{r-1}$ 
6:   fim se
7:   Variáveis relacionadas a  $Q_r$  são mantidas no seu domínio original;
8:   Resolva o subproblema  $MIP^r$  com o tempo limite igual a  $(TempoMax/R)$ 
9:   O tempo que sobrar é utilizado na próxima iteração.
10:  se  $MIP^r$  é infactível então
11:    retorna solução não encontrada
12:  fim se
13: fim para
14: retorna solução encontrada

```

Definimos que *window* é o tamanho de Q_r , com $r = 1, \dots, R$, ou seja, o *window* indica o número de elementos que tem em Q_r . No qual para todo r , o Q_r tem o mesmo número de elementos exceto o Q_R que pode conter o número de elementos menor ou igual aos outros. Além disso, definimos *overlap* como sendo o tamanho de U_r com $r = 1, \dots, R-1$.

Para exemplificar as partições definidas para a RF, consideramos o MPCLSP. Escolhemos o conjunto Q como sendo os índices do período T , com número de períodos igual a $NT = 12$ para particionar $Q = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$. Se considerarmos *janela* = 4 e *overlap* = 2, temos as seguintes partições: $Q_1 = \{1, 2, 3, 4\}$, $Q_2 = \{3, 4, 5, 6\}$, $Q_3 = \{5, 6, 7, 8\}$, $Q_4 = \{7, 8, 9, 10\}$ e $Q_5 = \{9, 10, 11, 12\}$; e as sobreposições: $U_1 = \{3, 4\}$, $U_2 = \{5, 6\}$, $U_3 = \{7, 8\}$ e $U_4 = \{9, 10\}$.

A Figura 4 indica como seria o domínio das variáveis em todas as iterações feitas pela RF, usando o exemplo apresentado acima e supondo que em todas foram encontradas soluções factíveis. Nessa figura, os índices do período são colocados em uma barra, em que o branco indica que as variáveis relacionadas com esses índices são relaxadas, o cinza-claro indica que as variáveis relacionadas são mantidas no seu domínio original, e o cinza-escuro indica que as variáveis são fixas com a solução encontrada.

Figura 4 – Exemplo do RF

Nota – Esse exemplo mostra como fica o domínio das variáveis relacionadas com o Q em cada iteração do RF. Variáveis relacionadas com índices em branco foram relaxadas. Em cinza claro, as variáveis com domínio original e as variáveis em cinza escuro foram fixadas com as soluções encontradas. Nesse exemplo, foi considerado que todas as iterações obtiveram uma solução factível.

Fonte: Elaborada pela autora.

3.2 Procedimentos de melhoria

3.2.1 Fix-and-Optimize (FO)

A heurística FO foi inicialmente proposta para problemas de planejamento da produção em Pochet e Wolsey (2006), com o nome de *exchange* e, posteriormente, foi renomeada em Helber e Sahling (2010). É similar com o RF, pois a ideia principal por trás dos métodos FO é que, a partir do particionamento das variáveis do problema, podemos resolver subproblemas fixados

mais simples de forma iterativa, gerando melhorias na solução incumbente. Mas, nesse caso, parte de uma solução inicial com objetivo de gerar melhorias é feita através da fixação de parte das variáveis e otimização dos subproblemas resultantes. Para aplicar a heurística, o conjunto de variáveis inteiras e/ou binárias Q é particionado em R subconjuntos $Q_1, \dots, Q_R$.

Na heurística, em cada iteração r , com $1 \leq r \leq R$, é definido um sub-MIP (subproblema inteiro e misto), em que as variáveis do conjunto Q_r são mantidas no domínio original e o restante é fixo com a solução incumbente. O sub-MIP obtido é resolvido e, caso uma solução melhor do que a atual seja encontrada, esta passa a ser a nova solução incumbente. Caso não se encontre uma solução de melhor qualidade após um certo número de iterações ou após um limite de tempo, o método é interrompido.

Neste trabalho, foi implementada a versão do FO baseada em Furlan (2011), em que as variáveis são particionadas por período. Nessa heurística, pode-se particionar as variáveis por períodos, itens, recursos ou fábricas, porém, neste trabalho, consideramos apenas a partição clássica das variáveis por período para o MPCLSP. O particionamento por período foi escolhido para comparar diretamente com os particionamentos usados na heurística ADN, também por período. Dessa forma, pode-se comparar o desempenho das heurísticas do tipo FO na sua versão clássica e com particionamentos obtidos através de ML não supervisionado. Para o MLCLSP, foi considerado particionamento por período e por itens para o FO e o ADN.

O Algoritmo 2 apresenta pseudocódigo do FO baseado em Furlan (2011), e na Figura 5, temos um exemplo indicando o domínio das variáveis em cada iteração feita com mesmo exemplo usado na Figura 4.

Algoritmo 2 – Pseudocódigo da heurística FO considerando particionamento com sobreposição

```

1: Entrada:  $Q_r$  e  $U_r$ , com  $r = 1, \dots, R$ ;
2: Encontre uma solução inicial;
3:  $I = 0$ ;
4: objValNew = objVal;                                     ▷ Valor da função objetivo
5: objVal = objVal + 1000
6: enquanto  $I < IterMax$  e objValNew < objVal e tempo < tempoMax faça
7:   para  $r \leftarrow 1$  to  $R$  faça
8:     Fixe todas as variáveis que não estão relacionadas a  $Q_r$  com a solução incumbente
     objValNew e o restante tem o domínio original;
9:     Resolva o  $MIP^r$ ;
10:    se objVal < objValNew então
11:      objValNew = objVal;
12:    fim se
13:  fim para
14: fim enquanto
15: retorna solução encontrada

```

Figura 5 – Exemplo do FO

Nota – Esse exemplo mostra como fica o domínio das variáveis relacionadas com o Q em cada iteração do FO. Variáveis relacionadas com índices em cinza claro têm o domínio original e o cinza escuro foram fixadas com as soluções encontradas. Nesse exemplo, foi considerado que todas as iterações obtiveram uma solução factível.

Fonte: Elaborada pela autora.

3.2.2 Automatically designed neighborhoods (ADN)

Ghiani, Laporte e Manni (2015) desenvolveram a heurística ADN, em que se extraem informações da estrutura do modelo e da solução corrente (solução factível escolhida para usar construção da vizinhança) para serem utilizadas por um algoritmo de ML não supervisionado a fim de auxiliar na definição da vizinhança. Dado um MIP, a heurística calcula a matriz de dissimilaridade que será utilizada pelo algoritmo de agrupamento para definir a vizinhança. A

seguir, é detalhado o passo a passo dessa heurística.

3.2.2.1 Matriz de dissimilaridade

Para construir a matriz de dissimilaridade, o primeiro passo da heurística consiste em definir um conjunto de índices relacionado com as variáveis binárias e/ou inteiras, chamado conjunto de entidades fundamentais E que pode ser dividido em subconjuntos de entidades disjuntos. Por exemplo, no problema MPCCLSP, temos apenas as variáveis Y_{ijt} com domínio de valores binários. Esse conjunto de variáveis está relacionado a três índices diferentes, sendo que eles estão contidos nos conjuntos N , M e T , respectivamente. Dessa forma, o conjunto de entidades fundamentais é definido como $E = E_1 \cup E_2 \cup E_3$, em que os subconjuntos de entidade são $E_1 = N$, $E_2 = M$ e $E_3 = T$.

Para extrair as informações da estrutura do modelo e da solução corrente $(x^{(h)})$, é construído um grafo de adjacência de entidades separadamente para cada subconjunto de entidades. Os nós são as entidades e existe uma aresta entre os nós se são adjacentes, ou seja, dado duas entidades $e, e' \in E$, elas são adjacentes se possuírem um peso $w_{ee'} > 0$ indicando se possuem alguma similaridade.

O peso $w_{ee'}$ entre a entidade e e e' é definido por duas medidas de similaridade, sendo a primeira baseada na solução corrente ($S_{ee'}(x^{(h)})$) e a segunda baseada na estrutura do modelo ($M_{ee'}$). Assim, o peso é igual a $w_{ee'} = 1$ se $S_{ee'}(x^{(h)}) > 0$; igual a $w_{ee'} = |E|$ se $S_{ee'}(x^{(h)}) = 0$ e $M_{ee'} > 0$; caso contrário, não existe peso entre essas entidades.

Para definir essas duas medidas de similaridade, antes definimos:

- o conjunto de índices associado com conjunto de índices das restrições $I = \{1, \dots, m\}$ e a matriz A como sendo a matriz de coeficientes do modelo matemático de um MIP genérico
- $I(x^{(h)}) \subseteq I$ como sendo o subconjunto de índices associado com as restrições ativas em $x^{(h)}$, ou seja, $m \in I(x^{(h)})$ se $A_m \cdot x^{(h)} = b_m$;
- $I_e \subseteq I$ como sendo o subconjunto de índices associado com as restrições relacionadas com a entidade fundamental $e \in E$, ou seja, $m \in I_e$ se $A_{me} \neq 0$;
- $I_e(x^{(h)}) \subseteq I$ como sendo o subconjunto de índices associado com as restrições relacionadas com a entidade fundamental $e \in E$ e ativas em $x^{(h)}$, ou seja, $m \in I_e(x^{(h)})$ se $A_{me} \neq 0$ e $m \in I(x^{(h)})$;
- J_e : conjunto de variáveis relacionados com a entidade $e \in E$.

A medida de similaridade baseada na solução é dada por $S_{ee'}(x^{(h)}) = P_{S1} + P_{S2} + P_{S3}$, nos quais

- $P_{S1} = \sum_{i \in I(x^{(h)})} \sum_{j \in J_e \cap J_{e'}} H(|a_{ij}|x_j^{(h)}),$
- $P_{S2} = \sum_{i \in I_e(x^{(h)})} \sum_{j \in J_{e'} \setminus J_e} H(|a_{ij}|x_j^{(h)}),$
- $P_{S3} = \sum_{i \in I_{e'}(x^{(h)})} \sum_{j \in J_e \setminus J_{e'}} H(|a_{ij}|x_j^{(h)}).$

E a função H é dada por:

$$H(t) = \begin{cases} 1, & \text{se } t > 0, \\ 0, & \text{caso contrário.} \end{cases}$$

Na [Figura 6](#), temos a matriz de coeficientes A de forma simplificada de um problema inteiro misto genérico, em que as colunas são relacionadas com o conjunto J e as linhas são relacionadas com o conjunto de restrições I . Dados e e $e' \in E$, foram selecionados os subconjuntos utilizados para calcular o $S_{ee'}$. Assim, temos que, para o cálculo de P_{S1} , soma-se o número de parâmetros maiores que zero na região cinza da figura. Para o P_{S2} , é a região cinza-clara e, para o P_{S3} , é a região cinza-escura.

Figura 6 – Matriz de coeficientes A simplificada de um problema inteiro misto genérico

Nota – As colunas são relacionadas com as variáveis do problema e as linhas estão relacionadas com as restrições. Dados e e e' em E , estão em cinza os subconjuntos que são selecionados para calcular $S_{ee'} = P_{S1}$ (cinza) + P_{S2} (cinza-clara) + P_{S3} (cinza-escura).

Fonte: Adaptada de [Ghiani, Laporte e Manni \(2015\)](#).

A medida de similaridade baseada no modelo é dada por $M_{ee'} = P_{M1} + P_{M2} + P_{M3}$, em que:

- $P_{M1} = \sum_{i \in I} \sum_{j \in J_e \cap J_{e'}} H(|a_{ij}|),$
- $P_{M2} = \sum_{i \in I_e} \sum_{j \in J_{e'} \setminus J_e} H(|a_{ij}|),$
- $P_{M3} = \sum_{i \in I_{e'}} \sum_{j \in J_e \setminus J_{e'}} H(|a_{ij}|).$

De forma semelhante a $S_{ee'}(x^{(h)})$, a [Figura 7](#) ilustra, de forma simplificada, a matriz de coeficientes A de um problema inteiro misto genérico. Dados e e $e' \in E$, foram selecionados os subconjuntos utilizados para calcular o $M_{ee'}$. Assim, temos que, para o cálculo de P_{M1} , soma-se o número de parâmetros maiores que zero na região cinza da figura. Para o cálculo de P_{M2} , somam-se os parâmetros maiores que zero na região cinza-clara. Para o cálculo de P_{M3} , soma-se o número de parâmetros maiores que zero da região cinza-escuro.

Figura 7 – Matriz de coeficientes A simplificada de um MIP genérico

Nota – As colunas são relacionadas com as variáveis do problema e as linhas estão relacionadas com as restrições. Dados e e e' em E , estão em cinza os subconjuntos que são selecionados para calcular $M_{ee'} = P_{M1}$ (cinza) + P_{M2} (cinza-clara) + P_{M3} (cinza-escuro).

Fonte: Adaptada de [Ghiani, Laporte e Manni \(2015\)](#).

Para ilustrar, no [Apêndice A](#), temos um exemplo de como é feito o cálculo do peso $w_{1,2}$ de MPCLSP com $N = \{1, 2, 3\}$, $M = \{1\}$ e $T = \{1, 2, 3\}$. A instância usada foi baseada em uma das instâncias geradas por [Carvalho e Nascimento \(2016\)](#).

Após o cálculo das duas medidas de similaridade, considera-se o peso de cada par de entidades em E . Assim, obtemos o grafo de adjacência de entidades. A partir desse grafo, é construída a matriz de dissimilaridade em que, para cada par de entidade, é calculada a menor

distância entre elas sobre o grafo. Observa-se que as entidades são mais dissimilares entre si quanto maior sua distância. Essa matriz é utilizada pelo algoritmo não supervisionado para agrupar as entidades de cada subconjunto de entidades separadamente. No caso, são o algoritmo *k-means* e o *k-medoids* que serão apresentados na subseção seguinte.

3.2.2.2 Algoritmos de agrupamento

Neste trabalho, foram escolhidos dois algoritmos de agrupamento clássico: o *k-means* e o *k-medoids*. Para cada um deles, foi implementada também a versão que obtém os agrupamentos balanceados, ou seja, obtém-se, em todos os agrupamentos, aproximadamente a mesma quantidade de entidades.

O objetivo do *k-means* é obter agrupamentos nos quais cada um tem um centroide que é o "centro" do agrupamento, ou seja, a distância entre o centroide e todos os outros elementos do agrupamento é a mesma. O *k-means* foi implementado baseado em [Jain e Dubes \(1988\)](#) e, no [Algoritmo 3](#), é apresentado o pseudocódigo, em que é definido o conjunto de centroides iniciais para realizar o agrupamento inicial. Para isso, são selecionadas k entidades de forma aleatória de E_r . Cada entidade em E_r que não foi selecionada é designada para o centroide mais próximo; assim obtemos os agrupamentos iniciais.

Depois, calcula-se o *loss* que é a soma de todas as distâncias euclidianas entre as entidades e seus respectivos centroides. Em cada agrupamento, calculam-se novos centroides, as médias da distância entre as entidades e o centroide. É calculado o *lossNew* da mesma forma que o *loss*, mas considerando os novos centroides. O processo se repete até que o número de iterações *iter* atinja o número máximo de iterações *iterMax* definido ou até que o módulo da diferença entre o *loss* e o *lossNew* seja menor que a tolerância máxima *tolerMax*.

Para o *k-means* balanceado, a diferença é apenas no passo 2 do [Algoritmo 3](#). O algoritmo designa os elementos os quais estão mais próximos dos centroides dos agrupamentos que não atingiram o número máximo de elementos que é igual a $|E_r|/k$. Assim, obtemos agrupamentos que possuem aproximadamente a mesma quantidade de elementos, baseados no que foi feito em [Ghiani, Laporte e Manni \(2015\)](#), que utilizaram apenas o *k-means* balanceado.

Para o *k-medoids*, foi escolhida a versão simplificada de [Park e Jun \(2009\)](#), em que autores desenvolveram o algoritmo de forma parecida com o *k-means*, como pode ser observado no pseudocódigo apresentado no [Algoritmo 4](#). Nesse algoritmo, calcula-se, para cada entidade, $j \in E_r$, $v_j = \frac{\sum_{i=1}^{|E_r|} d_{ij}}{\sum_{l=1}^{|E_r|} d_{il}}$, em que d_{ij} é a medida de dissimilaridade entre as entidades i e $j \in E_r$. Depois, são selecionadas k entidades com menores valores de v_j para serem os *medoids* iniciais e o restante das entidades é designado para os *medoids* mais próximos. Calcula-se o *loss*, sendo a soma de todas as distâncias entre as entidades e seus respectivos *medoids*.

São calculados novos *medoids* os quais são as entidades que têm a menor soma de distâncias em relação às outras entidades em cada agrupamento. Obtém-se novos agrupamentos

Algoritmo 3 – Pseudocódigo do *k-means*

```

1: Entrada  $k, E_r, iterMax$  e  $tolerMax$ ;
2: Define um conjunto de centroides iniciais selecionando aleatoriamente  $k$  entidades em  $E_r$ ;
3:  $loss = 0$ ;
4:  $lossNew = 0$ ;
5:  $iter = 0$ ;
6: enquanto  $iter < iterMax$  faça
7:    $loss = lossNew$ 
8:    $iter = iter + 1$ ;
9:   Designe as entidades que estão mais próximos dentre os centroides;
10:  Calcule novos centroides que são as médias das posições das entidades de cada agrupamento;
11:  Calcule  $lossNew$ ;
12:   $toler = |loss - lossNew|$ 
13:  se  $toler < tolerMax$  então
14:    retorna Agrupamentos;
15:  fim se
16: fim enquanto
17: retorna Agrupamentos;

```

designando cada entidade que está mais próxima do *medoid* novo. Calcula-se o $lossNew$ da mesma forma que o $loss$ anterior. O processo se repete até que o número de iterações $iter$ atinja o número máximo de iterações $iterMax$, ou o $lossNew$ seja igual a $loss$. Na implementação do *k-medoids* balanceado, a diferença é que as entidades são designadas para os agrupamentos com o *medoid* mais próximo e que o número de entidades neste agrupamento seja menor que $|E_r|/k$.

Algoritmo 4 – Pseudocódigo do *k-medoids*

```

1: Entrada:  $k, E_r, iterMax$  e  $tolerMax$ ;
2: Calcula  $v_j$  para cada entidade  $j \in E_r$ ;
3: Define os medoids iniciais
4: Obtém os agrupamentos iniciais;
5: Calcula  $loss$ ;
6: enquanto  $iter < iterMax$  faça
7:    $iter = iter + 1$ ;
8:   Calcula novos medoids para cada agrupamento;
9:   Obtém agrupamentos novos;
10:  Calcula  $lossNew$ ;
11:   $toler = |loss - lossNew|$ 
12:  se  $toler < tolerMax$  então
13:    retorna Agrupamentos;
14:  fim se
15: fim enquanto
16: retorna Agrupamentos;

```

3.2.2.3 Heurística ADN

Após obter os agrupamentos em cada subconjunto de entidades, é feita a combinação desses agrupamentos. Por exemplo, para o MPCLSP, [Ghiani, Laporte e Manni \(2015\)](#) agruparam apenas o subconjunto E_3 (agrupamento por períodos) em 6 agrupamentos $E_3^{(1)}, E_3^{(2)}, \dots, E_3^{(6)}$, ou seja, os autores utilizaram apenas decomposição por períodos. Assim, temos $NC = 6$ combinações de agrupamentos: $Q^1 = E_1 \cup E_2 \cup E_3^{(1)}, Q^2 = E_1 \cup E_2 \cup E_3^{(2)}, \dots, Q^6 = E_1 \cup E_2 \cup E_3^{(6)}$.

A partir dessas combinações de agrupamentos, em [Ghiani, Laporte e Manni \(2015\)](#), foram definidos dois tipos de vizinhança: uma chamada de vizinhança básica e outra chamada vizinhança de intensificação. A primeira é formada por todas as soluções de *sub-MIPs*, em que cada *sub-MIP^c*, com $c = 1, \dots, NC$, é definido de forma que todas as variáveis que possuem os índices pertencentes à combinação de agrupamentos Q^c são mantidas no domínio original e as restantes são fixas com a solução corrente. A segunda supõe que os subconjuntos de entidades tenham sido agrupados e definimos um conjunto de borda que são conjuntos de entidades pertencentes a agrupamentos diferentes que possuem adjacência. Por exemplo, dados $e, e' \in E$, com $e \in E_3^{(3)}$ e $e' \in E_3^{(4)}$, caso exista aresta entre eles no grafo, então eles são adjacentes e, consequentemente, pertencem ao conjunto de entidades de borda.

A partir deles, é feita a redução do efeito de borda para melhorar a solução localmente. Depois, foi feito o reagrupamento de cada subconjunto considerando o conjunto de entidades de borda para a escolha dos centroides ou *medoids* iniciais no algoritmo de agrupamento; assim, a ideia é diminuir o número de entidades no conjunto de borda, ou seja, reduzir o efeito de borda. Depois, foi construída a vizinhança da mesma forma que foi para a vizinhança básica, mas utilizando o novo agrupamento.

Resumidamente, no algoritmo ADN apresentado em [Ghiani, Laporte e Manni \(2015\)](#) são dados uma solução inicial e um tempo limite. Esse algoritmo é dividido em duas fases. Na primeira fase, utiliza-se a metade do tempo limite para explorar o espaço de busca considerando a vizinhança básica e, na segunda fase, utiliza-se o tempo limite restante para intensificar a busca considerando a vizinhança de intensificação.

Baseado em [Ghiani, Laporte e Manni \(2015\)](#), foram implementadas três versões do ADN e, em todas, foi fornecida uma solução inicial e um tempo limite. O [Algoritmo 5](#) apresenta o pseudocódigo da versão 1 do ADN (ADN1), o qual explora uma vez a vizinhança básica utilizando a metade do tempo limite e uma vez a vizinhança de intensificação com o restante do tempo limite. O [Algoritmo 6](#) apresenta a versão 2 do ADN (ADN2), o qual utiliza a metade do tempo limite para explorar várias vezes a vizinhança básica e, ao término de cada exploração, a vizinhança é atualizada utilizando a solução incumbente. O restante do tempo é utilizado para explorar várias vezes a vizinhança de intensificação e, cada vez que a exploração termina, também atualiza-se a vizinhança utilizando a solução incumbente. O [Algoritmo 7](#) apresenta a versão 3 (ADN3), que explora várias vezes a vizinhança básica atualizando a vizinhança toda

vez que encontra uma solução incumbente e, quando termina de explorar a vizinhança, é definida uma nova vizinhança básica.

Algoritmo 5 – Pseudocódigo do ADN1

FASE 1: define a vizinhança básica com a solução inicial utilizando metade do tempo limite;

FASE 2: define a vizinhança de intensificação com a melhor solução encontrada na fase 1 utilizando o restante do tempo limite.

Algoritmo 6 – Pseudocódigo do ADN2

FASE 1:

Passo 1.1: agrupa os subconjuntos de entidades.

Passo 1.2: utiliza a solução incumbente para definir a vizinhança básica utilizando a combinação de agrupamentos encontrados no passo 1.1.

Passo 1.3: repete o passo 1.2 até que a solução incumbente não melhore ou atinja o restante da metade do tempo limite.

FASE 2:

Passo 2.1: reagrupa os subconjuntos de entidades reduzindo o efeito de borda.

Passo 2.2: utiliza a solução incumbente para definir a vizinhança de intensificação utilizando a combinação de agrupamentos encontrados no passo 2.1.

Passo 2.3: repete os passos 2.2 até que a solução não melhore ou atinja o resto do tempo limite.

Algoritmo 7 – Pseudocódigo do ADN3

FASE 1:

Passo 1.1: agrupa os subconjuntos de entidades considerando a solução incumbente.

Passo 1.2: utiliza a solução incumbente para definir a vizinhança básica utilizando a combinação de agrupamentos encontrados no passo 1.1. Ao resolver um sub-MIP e a solução dele for melhor que a incumbente então a solução incumbente é atualizada.

Passo 1.3: repete o passo 1.1 e 1.2 até que a solução não melhore ou atinja o tempo limite.

TESTES COMPUTACIONAIS

Neste capítulo, vamos apresentar as análises dos resultados dos experimentos computacionais realizados utilizando a heurística FO e três versões da heurística ADN (ADN1, ADN2 e ADN3) que utilizaram algoritmos de agrupamentos (*k-means*, *k-means* balanceado, *k-medoids* e *k-medoids* balanceado) para resolver os LSPs que foram apresentados no [Capítulo 2](#) (MPCLSP e o MLCLSP).

As heurísticas foram implementadas em *Python 3* e o resolvidor comercial utilizado é o *Gurobi Optimizer* versão 9.0.1, na forma padrão. Os experimentos foram realizados no *cluster Euler*, utilizando o processamento sequencial com processadores *Intel Xeon E5 – 2667v4* de 3,2GHz com 8 núcleos, sendo um recurso computacional do Centro de Ciências Matemáticas Aplicadas à Indústria (CeMEAI), financiado pela FAPESP (proc. 2013/07375-0).

Observamos que, para o problema MPCLSP, o *Gurobi* consegue encontrar soluções ótimas para a maioria dos casos. Apesar disso, utilizamos esse problema para realizar os experimentos a fim de estudar as heurísticas e depois escolhemos um problema complexo: o MLCLSP.

4.1 Resultados computacionais para o MPCLSP

As instâncias utilizadas durante os testes computacionais foram geradas de forma randômica por [Carvalho e Nascimento \(2016\)](#) e retiradas do *site* <https://sites.google.com/site/nascimentomcv/downloads/mpclsp>. Para gerar essas instâncias, os valores dos seguintes parâmetros foram considerados a partir de uma distribuição uniforme de um intervalo contínuo:

- custo unitário de produção $c_{ijt} \in [1,5;2,5]$;
- custo unitário de estoque $h_{ijt} \in [0,2;0,4]$;
- custo unitário de transferência $r_{jj't} \in [0,2;0,4]$;

- tempo de produção unitário de produção $b_{ijt} \in [1,0;5,0]$;
- demanda $d_{ijt} \in [0;180]$.

Os parâmetros possuem diferentes características indicando o nível de capacidade disponível (apertada ou normal), custo de preparação (alto ou baixo) e tempo de preparação (alto ou baixo). Todas as instâncias possuem o número de períodos iguais a $NT = 12$. Instâncias com número de itens iguais a $NI \in \{12, 25, 50\}$ possuem o número de plantas iguais a $NM \in \{2, 4, 6\}$ e instâncias com número de itens iguais a $NI \in \{70, 80, 90\}$ possuem o número de plantas iguais a $NM \in \{15, 20\}$. Encontram-se, no total, 18 combinações de cada tipo e cada uma com 40 instâncias, totalizando 720 instâncias. A seguir, serão apresentadas as análises das soluções obtidas pelo *Gurobi* e pelas heurísticas.

4.1.1 Análise das soluções: modelo matemático

Para comparar com as soluções obtidas pelas heurísticas, foram utilizadas as soluções obtidas pelo resolvidor *Gurobi* com o tempo limite de 900 segundos. Para cada instância, foi calculado o MIPGap obtido pelo resolvidor, igual a $(\frac{|z_P - z_D|}{|z_P|} \cdot 100)$, sendo z_P o limitante superior e z_D o limitante inferior. Na Figura 8, é apresentado o gráfico *box plot* (diagrama de caixa) do MIPGap para cada conjunto de instâncias com o mesmo número de itens e, na Tabela 1, apresentamos o MIPGap médio ($MIPGap_{med}$), desvio padrão ($MIPGap_{dp}$), mínimo ($MIPGap_{min}$) e máximo ($MIPGap_{max}$). Para facilitar a visualização, foram destacados, em negrito, os menores valores obtidos em cada coluna.

Figura 8 – *Box plot* do MIPGap em porcentagem das soluções do MPCLSP obtidas pelo *Gurobi*.

Nota – *Box plot* para cada conjunto de instâncias com o mesmo número de itens (NI).

Fonte: Elaborada pela autora.

Tabela 1 – MIPGap das soluções do MPCLSP obtidas pelo *Gurobi*.

<i>NI</i>	MIPGap _{med} (%)	MIPGap _{dp} (%)	MIPGap _{min} (%)	MIPGap _{max} (%)
6	0,00	0,00	0,00	0,01
12	0,00	0,01	0,00	0,01
25	0,01	0,01	0,00	0,01
50	0,01	0,00	0,00	0,01
70	0,01	0,00	0,00	0,02
80	0,01	0,01	0,00	0,07
90	0,01	0,01	0,00	0,13

Fonte: Elaborada pela autora.

Nota – Foram calculados MIPGap médio, desvio padrão, mínimo e máximo para cada conjunto de instâncias com mesmo número de itens (*NI*). Em negrito, estão destacados os menores valores obtidos em cada coluna.

Podemos observar, tanto na [Figura 8](#) quanto na [Tabela 1](#), que os valores de MIPGap obtidos são bem próximos de zero ($\leq 0,13\%$) apesar de alguns atingirem o tempo limite de 900 segundos sem conseguir provar a otimalidade das soluções encontradas. Notamos que, quanto maior o número de itens (*NI*), a tendência é que o MIPGap seja maior. O *Gurobi* obteve o MIPGap igual a 0,00 em 262 instâncias, cerca de 36,4% do total.

Na [Figura 9](#), é apresentado o gráfico *box plot* do tempo gasto pelo *Gurobi* para cada conjunto de instâncias com o mesmo número de itens e, na [Tabela 2](#), é apresentado, em segundos, o tempo médio (Tempo_{med}), desvio padrão (Tempo_{dp}), tempo mínimo (Tempo_{min}) e tempo máximo (Tempo_{max}). Podemos observar que, à medida que o número de itens aumenta, o tempo gasto para resolver também. Além disso, dentre as 720 instâncias, 703 (97,64%) obtiveram soluções ótimas e o restante das 17 instâncias atingiu o tempo limite sendo 5 com 70 itens, 6 com 80 itens e 6 com 90 itens. Vale ressaltar que, mesmo obtendo a solução ótima para uma parte das instâncias, o MIPGap maior ou igual a 0,01% se refere ao arredondamento para duas casas decimais.

4.1.2 Análise das soluções: heurísticas

As heurísticas FO e ADN possuem o tempo limite igual a 900 segundos e, para resolver os sub-MIPs, foi utilizado o *Gurobi*. A fim de encontrar as soluções iniciais, foram escolhidas duas formas: uma utilizando a primeira solução factível encontrada pelo *Gurobi* e a outra utilizando o RF.

Com objetivo de encontrar a solução inicial utilizando o RF, foi considerado $Q = T = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$ para particionar, e consideramos $janela = 4$ e $overlap = 2$ baseados nos estudos feitos por [Mercé e Fontan \(2003\)](#) para LSP monoestágio com restrição de capacidade. Assim, há as seguintes partições: $Q_1 = \{1, 2, 3, 4\}$, $Q_2 = \{3, 4, 5, 6\}$, $Q_3 = \{5, 6, 7, 8\}$, $Q_4 = \{7, 8, 9, 10\}$ e $Q_5 = \{9, 10, 11, 12\}$; e as sobreposições: $U_1 = \{3, 4\}$, $U_2 = \{5, 6\}$, $U_3 =$

Figura 9 – Box plot do tempo gasto pelo *Gurobi* para resolver MPCLSP.

Nota – Box plot para cada conjunto de instâncias com o mesmo número de itens (NI).

Fonte: Elaborada pela autora.

Tabela 2 – Tempo gasto pelo *Gurobi* para resolver MPCLSP.

NI	Tempo _{med} (s)	Tempo _{dp} (s)	Tempo _{min} (s)	Tempo _{max} (s)
6	0,12	0,15	0,01	1,06
12	0,35	0,52	0,02	2,82
25	0,95	1,17	0,03	5,38
50	2,33	3,59	0,10	26,53
70	302,85	269,12	42,84	900,00
80	320,24	280,44	50,91	900,00
90	348,28	284,96	53,54	900,00

Fonte: Elaborada pela autora.

Nota – Foram calculados tempo médio, desvio padrão, mínimo e máximo para cada conjunto de instâncias com mesmo número de itens (NI). Em negrito, estão destacados os menores valores obtidos em cada coluna.

$\{7, 8\}$ e $U_4 = \{9, 10\}$.

Na heurística FO, foi considerado o tamanho de $janela = \{1, 2, 3\}$ e o conjunto de índices dos períodos (T) para particionar. Na heurística ADN definimos o conjunto de entidades $E = E_1 \cup E_2 \cup E_3$, com $E_1 = N$, $E_2 = M$ e $E_3 = T$, que está relacionado com a variável binária Y_{ijt} , com $i \in N$, $j \in M$ e $t \in T$. Foi feito o agrupamento apenas para o subconjunto de entidades $E_3 = T$ em 6 agrupamentos, assim como foi feito em [Ghiani, Laporte e Manni \(2015\)](#) e, para isso, os autores consideraram que dados e e $e' \in E_3$, existe peso $w_{ee'}$ se $|e - e'| = 1$ e:

$$w_{ee'} = \begin{cases} 1, & \text{se existe } i \in N, j \in M, \text{ tal que } I_{ije} > 0 \text{ se } e < e' \text{ ou } I_{ije'} > 0 \text{ se } e' < e, \\ |E|, & \text{caso contrário.} \end{cases}$$

Para todas as instâncias, foi obtido o mesmo grafo de adjacência, pois havia sempre estoque de algum produto em alguma planta. Na Figura 10, encontra-se o grafo obtido e podemos observar que todas as arestas têm o peso igual a 1, ou seja, há uma similaridade forte. Para calcular a matriz de dissimilaridade, foi utilizado o algoritmo de *Floyd-Warshall* implementado na biblioteca *SciPy* do *Python* e a matriz obtida é dada em (4.1).

Figura 10 – Grafo de adjacência do subconjunto de entidade $E_3 = T$ para MPCLSP.

Nota – Os nós do grafo são as entidades pertencentes ao conjunto E_3 e existem arestas entre duas entidades se são adjacentes. Todas as arestas têm o peso igual a 1.

Fonte: Elaborada pela autora.

$$d = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 \\ 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 \\ 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 \\ 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 & 6 \\ 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 & 5 \\ 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 & 4 \\ 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 & 3 \\ 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 & 2 \\ 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 & 1 \\ 11 & 10 & 9 & 8 & 7 & 6 & 5 & 4 & 3 & 2 & 1 & 0 \end{bmatrix} \quad (4.1)$$

Nos cálculos dos centroides a serem utilizados pelo algoritmo *k-means* e pelo *k-means* balanceado, foi considerado o grafo posicionado no plano cartesiano com a entidade $t = 1$ na posição $(0,0)$, $t = 2$ em $(0,1)$, $t = 3$ em $(0,2)$, $t = 4$ em $(0,3)$, $t = 5$ em $(0,4)$, $t = 6$ em $(0,5)$, $t = 7$ em $(0,6)$, $t = 8$ em $(0,7)$, $t = 8$ em $(0,7)$, $t = 9$ em $(0,8)$, $t = 10$ em $(0,9)$, $t = 11$ em $(0,10)$ e $t = 12$ em $(0,11)$. Assim, as distâncias entre eles são de uma unidade de medida, facilitando o cálculo da distância entre os centroides e as outras entidades, pois os centroides são as médias das posições das entidades de cada agrupamento e essas distâncias são euclidianas.

Resumidamente, as heurísticas são compostas por duas partes: encontra-se uma solução inicial (parte 1) e, depois, aplica o procedimento de melhoria (parte 2). A seguir, estão listados, para cada parte, os métodos utilizados e as opções de configurações escolhidas:

- solução inicial:
 - G: primeira solução factível encontrada pelo *Gurobi*;
 - RFp: RF com $janela = 4$ e $overlap = 2$, usando $p\%$ do limite total de 900 segundos, com $p = \{10, 50\}$;

- procedimento de melhoria:
 - FOjanela: $janela = \{1, 2, 3\}$;
 - ADN $_{vk}$: versão v do ADN, com $v = \{1, 2, 3\}$, usando o algoritmo de agrupamento k , com $k = \{K, KB, KM, KMB\}$ (K: *k-means*, KB: *k-means* balanceado, KM: *k-medoids*, KMB: *k-medoids* balanceado);

Assim, temos 45 combinações possíveis de heurísticas. A nomenclatura RF10ADN2KM, por exemplo, é a heurística que encontra uma solução inicial utilizando o RF com tempo limite igual a 10% do tempo limite total, ou seja, 90 segundos. Para melhoria, utilizamos ADN2 (versão 2 do ADN) com o algoritmo de agrupamento *k-medoids*. O tempo reservado para o RF é distribuído de forma igual para resolver os subproblemas MIP^r , com $r = 1, \dots, 5$, ou seja, cada subproblema tem o tempo limite igual a 18 segundos. O tempo que sobrou mais o restante do tempo é utilizado pelas heurísticas de melhoria de ADN2. A seguir, serão apresentados os resultados obtidos por todas as combinações de heurísticas.

A partir dos resultados obtidos pelas heurísticas, foram feitos os seguintes cálculos:

- $GAP_{Ini} = \left(\frac{objVal_{Ini} - z_D}{objVal_{Ini}} \cdot 100 \right) \%$;
- $GAP = \left(\frac{objVal - z_D}{objVal} \cdot 100 \right) \%$;
- $GAPM = \left(\frac{objVal_{Ini} - objVal}{objVal_{Ini}} \cdot 100 \right) \%$.

sendo z_D o limitante superior, conforme definido anteriormente; $objVal$ é o valor da função objetivo obtida pela heurística; $objVal_{Ini}$ é o valor da função objetivo obtida com a solução inicial.

A Figura 11 apresenta o *box plot* do MIPGap do Gurobi e do GAP_{Ini} para cada conjunto de instâncias com o mesmo número de itens para cada método usado para encontrar a solução inicial (G, RF10, RF50). Observamos que as soluções iniciais encontradas pelo G possuem o GAP_{Ini} maior e o restante ficou bem próximo do MIPGap. Notamos também que as instâncias com 70, 80 e 90 itens possuem GAP_{Ini} maiores do que as instâncias com menos itens.

Na Tabela 3, apresentamos o cálculo do GAP_{Ini} médio ($GAP_{Ini_{med}}$), desvio padrão ($GAP_{Ini_{dp}}$), mínimo ($GAP_{Ini_{min}}$) e máximo ($GAP_{Ini_{max}}$) e o $MIPGap_{med}$, $MIPGap_{dp}$, $MIPGap_{min}$ e $MIPGap_{max}$ do Gurobi para cada conjunto de heurísticas, que utilizou o mesmo método para encontrar a solução inicial para cada conjunto de instâncias com o mesmo número de itens. Nessa tabela, estão destacados, em negrito, os menores valores dentre os métodos em cada coluna. Observamos que as soluções iniciais obtidas pelo RF conseguiram GAP_{Ini} bem menores em comparação com as soluções iniciais obtidas pelo G, e algumas heurísticas conseguiram encontrar soluções ótimas já na solução inicial.

Figura 11 – Box plot do MIPGap do Gurobi e GAPIni das heurísticas para MPCLSP sem outliers.

Nota – O GAPIni é em relação ao limitante inferior do *Gurobi* e a solução inicial da heurística. Para o *Gurobi* e cada tipo de método usado para encontrar a solução inicial, encontra-se um conjunto de box plot, que é para cada conjunto de instâncias com o mesmo número de itens (NI).

Fonte: Elaborada pela autora.

Na [Tabela 4](#), apresentamos o cálculo do GAP médio (GAP_{med}), desvio padrão (GAP_{dp}), mínimo (GAP_{min}) e máximo (GAP_{max}) para cada heurística e o $MIPGap_{med}$, $MIPGap_{dp}$, $MIPGap_{min}$ e $MIPGap_{max}$ do *Gurobi*. Nessa tabela, estão destacados, em negrito, os menores valores dentre os métodos em cada coluna. Observamos que a maioria das heurísticas obteve GAP bem próximos ou iguais ao $MIPGap$ do *Gurobi*, exceto as heurísticas que utilizaram o G e AND1.

Na [Tabela 5](#), encontra-se a quantidade de instâncias que obtiveram soluções ótimas para cada heurística já na solução inicial (*solI*) e com a solução após aplicar o procedimento de melhoria (*solM*). Observamos que as heurísticas que utilizaram o RF conseguiram obter soluções ótimas entre 224 e 225 instâncias. Depois do procedimento de melhoria, as heurísticas que utilizaram o G conseguiram melhorar a solução inicial para mais instâncias, exceto as que utilizaram o ADN1, mas, para essas heurísticas, as soluções iniciais tinham o GAPIni bem maior em comparação com as demais. As heurísticas que utilizaram o RF, depois do procedimento de melhoria, em poucas instâncias obtiveram soluções ótimas.

Tabela 3 – GAPIni das heurísticas e o MIPGap do *Gurobi* para o MPCLSP.

Solução inicial	<i>NI</i>	GAPIni _{med} (%)	GAPIni _{dp} (%)	GAPIni _{min} (%)	GAPIni _{max} (%)
G	6	56,85	11,11	22,57	71,22
	12	55,69	12,57	23,03	71,67
	25	52,72	13,81	20,50	71,52
	50	52,49	13,95	21,23	68,70
	70	84,27	2,15	81,38	86,89
	80	84,17	2,17	81,27	86,77
	90	84,02	2,35	74,79	86,49
RF10	6	0,00	0,00	0,00	0,02
	12	0,00	0,00	0,00	0,01
	25	0,01	0,01	0,00	0,07
	50	0,01	0,03	0,00	0,28
	70	1,03	0,92	0,01	4,16
	80	1,24	1,04	0,01	4,07
	90	1,30	1,37	0,01	15,36
RF50	6	0,00	0,00	0,00	0,02
	12	0,00	0,00	0,00	0,01
	25	0,01	0,01	0,00	0,02
	50	0,01	0,00	0,00	0,03
	70	0,06	0,06	0,01	0,51
	80	0,07	0,09	0,01	0,59
	90	0,08	0,10	0,01	0,67
<i>Gurobi</i>	6	0,00	0,00	0,00	0,01
	12	0,00	0,01	0,00	0,01
	25	0,01	0,01	0,00	0,01
	50	0,01	0,00	0,00	0,01
	70	0,01	0,00	0,00	0,02
	80	0,01	0,01	0,00	0,07
	90	0,01	0,01	0,00	0,13

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre os métodos para encontrar a solução inicial para instâncias com mesmo número de itens em cada coluna.

Na [Tabela 6](#), é apresentado, em segundos, o tempo médio (Tempo_{med}), desvio padrão (Tempo_{dp}), mínimo (Tempo_{min}), máximo (Tempo_{max}), o número de instâncias que atingiu o tempo limite ($nInst$) e a porcentagem ($\frac{nInst}{720} \times 100 \%$) que esse número representa do número total de instâncias para cada heurística e pelo *Gurobi*. Nessa tabela, podemos observar que o GADN1KB é o que possui a menor média de tempo gasto. Observa-se também que todas as heurísticas que utilizaram o ADN1 e algumas que utilizaram o AND3 e ADN2 não atingiram o tempo limite para todas as instâncias.

Na [Figura 12](#), temos *box-plot* em relação ao número total de iterações realizadas pelas heurísticas, exceto as que utilizaram o ADN1, pois sempre realizam apenas duas iterações

Tabela 4 – GAP das heurísticas e pelo *Gurobi* para MPCLSP.

Heurísticas	$GAP_{med} (\%)$	$GAP_{dp} (\%)$	$GAP_{min} (\%)$	$GAP_{max} (\%)$	$\frac{solI}{720} \times 100$
GFO1	0,04	0,05	0,00	0,47	100,00
GFO2	0,03	0,04	0,00	0,33	100,00
GFO3	0,02	0,03	0,00	0,28	100,00
GADN1K	7,19	6,11	0,00	30,40	100,00
GADN1KB	10,55	8,11	0,35	34,38	100,00
GADN1KM	10,20	8,49	0,31	32,38	100,00
GADN1KMB	10,55	8,11	0,35	34,38	100,00
GADN2K	0,03	0,05	0,00	0,73	100,00
GADN2KB	0,04	0,07	0,00	0,53	100,00
GADN2KM	0,04	0,10	0,00	0,84	100,00
GADN2KMB	0,04	0,07	0,00	0,65	100,00
GADN3K	0,01	0,02	0,00	0,25	100,00
GADN3KB	0,03	0,04	0,00	0,33	100,00
GADN3KM	0,02	0,03	0,00	0,28	100,00
GADN3KMB	0,03	0,04	0,00	0,33	100,00
RF10FO1	0,16	0,42	0,00	3,49	100,00
RF10FO2	0,14	0,39	0,00	4,00	100,00
RF10FO3	0,18	0,44	0,00	2,45	100,00
RF10ADN1K	0,02	0,03	0,00	0,22	100,00
RF10ADN1KB	0,02	0,04	0,00	0,32	100,00
RF10ADN1KM	0,06	0,15	0,00	0,92	100,00
RF10ADN1KMB	0,02	0,04	0,00	0,40	100,00
RF10ADN2K	0,02	0,03	0,00	0,19	99,58
RF10ADN2KB	0,02	0,04	0,00	0,32	99,72
RF10ADN2KM	0,02	0,04	0,00	0,37	99,72
RF10ADN2KMB	0,02	0,04	0,00	0,53	99,72
RF10ADN3K	0,02	0,03	0,00	0,29	98,89
RF10ADN3KB	0,01	0,02	0,00	0,17	100,00
RF10ADN3KM	0,02	0,04	0,00	0,40	98,75
RF10ADN3KMB	0,02	0,04	0,00	0,44	98,75
RF50FO1	0,01	0,03	0,00	0,20	100,00
RF50FO2	0,01	0,02	0,00	0,20	100,00
RF50FO3	0,02	0,03	0,00	0,20	100,00
RF50ADN1K	0,02	0,03	0,00	0,28	100,00
RF50ADN1KB	0,02	0,04	0,00	0,28	100,00
RF50ADN1KM	0,02	0,04	0,00	0,28	100,00
RF50ADN1KMB	0,02	0,04	0,00	0,28	100,00
RF50ADN2K	0,02	0,03	0,00	0,24	100,00
RF50ADN2KB	0,02	0,03	0,00	0,30	100,00
RF50ADN2KM	0,02	0,03	0,00	0,26	100,00
RF50ADN2KMB	0,02	0,03	0,00	0,30	100,00
RF50ADN3K	0,02	0,03	0,00	0,30	100,00
RF50ADN3KB	0,02	0,04	0,00	0,30	100,00
RF50ADN3KM	0,02	0,04	0,00	0,52	100,00
RF50ADN3KMB	0,02	0,04	0,00	0,30	100,00
Gurobi	0,01	0,01	0,00	0,13	100,00

Fonte: Elaborada pela autora.

Nota – Na coluna ($\frac{solI}{720} \times 100$) indica-se a percentagem do número de instâncias que obtiveram solução factíveis ou ótimas. Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 5 – Quantidade de instâncias do MPCLSP que tiveram soluções ótimas para cada heurística.

Heurísticas	<i>solI</i>	$\frac{solI}{720} \times 100$	<i>solM</i>	$\frac{solM}{720} \times 100$	<i>solI</i> – <i>solM</i>	$\frac{solM - solI}{solM} \times 100$
GFO1	0	0,00	131	18,19	131	100,00
GFO2	0	0,00	132	18,33	132	100,00
GFO3	0	0,00	204	28,33	204	100,00
GADN1K	0	0,00	7	0,97	7	100,00
GADN1KB	0	0,00	0	0,00	0	-
GADN1KM	0	0,00	0	0,00	0	-
GADN1KMB	0	0,00	0	0,00	0	-
GADN2K	0	0,00	211	29,31	211	100,00
GADN2KB	0	0,00	153	21,25	153	100,00
GADN2KM	0	0,00	223	30,97	223	100,00
GADN2KMB	0	0,00	153	21,25	153	100,00
GADN3K	0	0,00	217	30,14	217	100,00
GADN3KB	0	0,00	135	18,75	135	100,00
GADN3KM	0	0,00	197	27,36	197	100,00
GADN3KMB	0	0,00	167	23,19	167	100,00
RF10FO1	225	31,25	240	33,33	15	6,25
RF10FO2	225	31,25	236	32,78	11	4,66
RF10FO3	225	31,25	228	31,67	3	1,32
RF10ADN1K	225	31,25	235	32,64	10	4,26
RF10ADN1KB	225	31,25	236	32,78	11	4,66
RF10ADN1KM	225	31,25	232	32,22	7	3,02
RF10ADN1KMB	225	31,25	237	32,92	12	5,06
RF10ADN2K	225	31,25	237	32,92	12	5,06
RF10ADN2KB	225	31,25	237	32,92	12	5,06
RF10ADN2KM	225	31,25	233	32,36	8	3,43
RF10ADN2KMB	225	31,25	236	32,78	11	4,66
RF10ADN3K	224	31,11	236	32,78	12	5,08
RF10ADN3KB	225	31,25	239	33,19	14	5,86
RF10ADN3KM	225	31,25	231	32,08	6	2,60
RF10ADN3KMB	225	31,25	237	32,92	12	5,06
RF50FO1	225	31,25	238	33,06	13	5,46
RF50FO2	225	31,25	235	32,64	10	4,26
RF50FO3	225	31,25	228	31,67	3	1,32
RF50ADN1K	225	31,25	240	33,33	15	6,25
RF50ADN1KB	225	31,25	236	32,78	11	4,66
RF50ADN1KM	225	31,25	231	32,08	6	2,60
RF50ADN1KMB	225	31,25	236	32,78	11	4,66
RF50ADN2K	225	31,25	235	32,64	10	4,26
RF50ADN2KB	225	31,25	236	32,78	11	4,66
RF50ADN2KM	225	31,25	231	32,08	6	2,60
RF50ADN2KMB	225	31,25	236	32,78	11	4,66
RF50ADN3K	225	31,25	236	32,78	11	4,66
RF50ADN3KB	225	31,25	236	32,78	11	4,66
RF50ADN3KM	225	31,25	230	31,94	5	2,17
RF50ADN3KMB	225	31,25	237	32,92	12	5,06

Fonte: Elaborada pela autora.

Nota – Em negrito, estão os maiores valores obtidos em cada coluna. Encontra-se a quantidade de soluções iniciais encontradas que eram soluções ótimas (*solI*), quantidade de soluções ótimas encontradas pelas heurísticas (*solM*), quantidade de soluções ótimas encontradas depois da melhoria (*solI* – *solM*).

Tabela 6 – Tempo gasto pelas heurísticas e pelo *Gurobi* para MPCLSP.

Heurísticas	Tempo _{med} (s)	Tempo _{dp} (s)	Tempo _{min} (s)	Tempo _{max} (s)	<i>nInst</i>	$\frac{nInst}{720} \times 100$
GFO1	182,11	297,09	0,09	900,00	50	6,94
GFO2	181,15	324,70	0,05	900,00	103	14,31
GFO3	174,43	325,67	0,03	900,00	109	15,14
GADN1K	65,99	103,02	0,05	397,97	0	0,00
GADN1KB	62,56	93,57	0,08	307,33	0	0,00
GADN1KM	83,15	125,27	0,06	418,68	0	0,00
GADN1KMB	82,82	126,10	0,06	450,21	0	0,00
GADN2K	241,66	337,28	0,25	900,00	17	2,36
GADN2KB	258,9	353,29	0,44	900,00	45	6,25
GADN2KM	265,12	358,84	0,23	900,00	44	6,11
GADN2KMB	259,93	354,32	0,22	900,00	43	5,97
GADN3K	141,57	236,76	0,06	872,12	0	0,00
GADN3KB	138,39	218,64	0,08	846,51	0	0,00
GADN3KM	119,51	195,10	0,06	807,32	0	0,00
GADN3KMB	124,01	205,79	0,06	887,86	0	0,00
RF10FO1	161,91	275,21	0,06	900,00	36	5,00
RF10FO2	125,51	225,83	0,05	900,00	22	3,06
RF10FO3	90,14	165,32	0,05	900,00	7	0,97
RF10ADN1K	75,21	113,02	0,07	413,76	0	0,00
RF10ADN1KB	69,39	99,82	0,08	305,46	0	0,00
RF10ADN1KM	113,65	161,25	0,08	522,11	0	0,00
RF10ADN1KMB	71,97	103,01	0,07	316,37	0	0,00
RF10ADN2K	198,87	301,04	0,06	900,00	5	0,69
RF10ADN2KB	203,17	301,93	0,09	900,00	6	0,83
RF10ADN2KM	203,98	304,05	0,08	900,00	5	0,69
RF10ADN2KMB	207,74	310,21	0,07	900,00	10	1,39
RF10ADN3K	181,73	278,79	0,04	900,00	3	0,42
RF10ADN3KB	130,41	214,23	0,06	832,23	0	0,00
RF10ADN3KM	172,67	260,92	0,05	900,00	1	0,14
RF10ADN3KMB	170,42	260,19	0,05	900,00	1	0,14
RF50FO1	93,21	163,25	0,06	900,00	8	1,11
RF50FO2	86,20	172,89	0,04	900,00	20	2,78
RF50FO3	63,45	105,49	0,03	900,00	2	0,28
RF50ADN1K	133,22	202,85	0,06	648,78	0	0,00
RF50ADN1KB	134,55	202,84	0,09	659,19	0	0,00
RF50ADN1KM	134,89	204,14	0,07	645,15	0	0,00
RF50ADN1KMB	136,44	206,25	0,07	663,51	0	0,00
RF50ADN2K	183,61	285,28	0,07	900,00	4	0,56
RF50ADN2KB	171,96	257,69	0,37	894,76	0	0,00
RF50ADN2KM	178,19	269,41	0,07	889,35	0	0,00
RF50ADN2KMB	179,46	272,18	0,07	900,00	1	0,14
RF50ADN3K	177,03	270,80	0,05	900,00	2	0,28
RF50ADN3KB	175,21	263,37	0,06	900,00	3	0,42
RF50ADN3KM	166,75	249,77	0,06	852,23	0	0,00
RF50ADN3KMB	174,64	261,95	0,05	885,50	0	0,00
<i>Gurobi</i>	108,55	220,99	0,01	900,00	17	2,36

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna. Além do tempo gasto, número de instâncias que atingiram o tempo limite (*nInst*) e a porcentagem ($\frac{nInst}{720} \times 100 \%$) do número do total de instâncias.

do total. Observamos que as heurísticas que utilizaram o ADN2 tiveram a média maior em comparação aos demais.

Figura 12 – *Box plot* do número de iterações realizadas pelas heurísticas, exceto as que utilizaram AND1 para MPCLSP.

Fonte: Elaborada pela autora.

Geralmente, observamos que as heurísticas que utilizaram o RF conseguiram soluções melhores já na solução inicial e o GAP médio obtido bem próximo de zero. Por isso, a melhoria não teve muito impacto quanto às heurísticas que utilizaram o G. Em relação ao tempo gasto, a maioria das heurísticas que utiliza o FO teve mais instâncias que atingiram o tempo limite. Em relação ao número de soluções ótimas, o *Gurobi* (97,64%) conseguiu encontrar mais que as heurísticas (31,25%, porcentagem da melhor heurística). No [Apêndice B](#), são apresentados os resultados com mais detalhes.

4.2 Resultados computacionais para o MLCLSP

Os testes computacionais para o MLCLSP foram realizados em instâncias retiradas de [Stadtler e Sürie \(2000\)](#), escolhidas 8 instâncias de cada classe A+, C+ e E+:

- Classe A+: AGC2310, AAC5440, AGC4420, AAC2340, AAC1410, AGC2330, AAC6320, AAC1330;
- Classe C+: CGC4330, CGC1320, CGC4340, CGC1330, CGC1310, CGC1350, CGC4310, CGC4350;
- Classe E+: EGC4340, EGC1310, EGC1340, EGC1330, EGC4320, EGC4330, EGC4350, EGC1350.

Os nomes das instâncias foram adaptados seguindo o padrão das instâncias de [Tempelmeier e Buschkühl \(2009\)](#). A primeira letra indica a classe à qual pertence. A segunda letra indica se a estrutura de produto é geral (G) ou de montagem (A, do inglês *assembly*). A terceira letra indica se o processo é cíclico (C) ou acíclico (N). Na estrutura geral, não há limite de sucessores e predecessores para os itens; já na estrutura de montagem do item, o que muda é que cada item tem, no máximo, um sucessor, mas pode ter predecessores ilimitados. No processo cíclico, alguns itens predecessores são produzidos com o mesmo recurso e, no processo acíclico, os itens são produzidos utilizando diferentes recursos que seus predecessores e seus sucessores ([TEMPELMEIER; BUSCHKÜHL, 2009](#)).

[Ghiani, Laporte e Manni \(2015\)](#) agruparam essas instâncias considerando o número de períodos (T), números de itens (J), número de recursos (M) e se o tempo de preparação é incluído ou não. A [Tabela 7](#) possui algumas informações numéricas a respeito das três classes de instâncias utilizadas.

Tabela 7 – Conjunto de instâncias do MLCLSP.

Classe	NT	NI	NK	Tempo de preparação
A+	24	10	3	Não
C+	48	40	3	Não
E+	48	100	6	Não

Fonte: Adaptada de [Stadtler e Sürie \(2000\)](#).

4.2.1 Análise das soluções: modelo matemático

Para o MLCLSP, também foram utilizadas as soluções obtidas pelo *Gurobi* com o tempo limite de 900 segundos. A [Figura 13](#) apresenta o gráfico *box plot* do MIPGap obtido pelo *Gurobi* para cada conjunto de instâncias da mesma classe e, na [Tabela 8](#), apresentamos o $MIPGap_{med}$, $MIPGap_{dp}$, $MIPGap_{min}$ e $MIPGap_{max}$. Para facilitar a visualização, foram destacados, em negrito, os menores valores obtidos em cada coluna. Podemos observar que a média do MIPGap é menor para a classe A+ com 7,09% e a maior para a classe E+ com 67,68%. Não foi obtida nenhuma solução ótima e, para todas as instâncias, o *Gurobi* atingiu o tempo limite.

4.2.2 Análise das soluções: heurísticas

As heurísticas FO e ADN possuem o tempo limite igual a 900 segundos e, para resolver os sub-MIPs, foi utilizado o *Gurobi*. Para encontrar as soluções, foram escolhidas as mesmas duas formas, uma utilizando a primeira solução factível encontrada pelo *Gurobi* e a outra utilizando a RF com o tamanho de $janela = 4$ e $overlap = 2$, considerando o conjunto de períodos (T) para particionar.

Para heurística FO, foi considerado o tamanho de $janela = \{3, 4\}$ e o conjunto de índices dos períodos ou itens para particionar. Para heurística ADN, foi definido o conjunto de entidades

Figura 13 – *Box plot* do MIPGap em porcentagem das soluções do MLCLSP obtidas pelo *Gurobi*.

Nota – *Box plot* para cada conjunto de instâncias da mesma classe.

Fonte: Elaborada pela autora.

Tabela 8 – MIPGap das soluções do MLCLSP obtidas pelo *Gurobi*.

Classe	MIPGap _{med} (%)	MIPGap _{dp} (%)	MIPGap _{min} (%)	MIPGap _{max} (%)
A+	7,09	2,77	2,93	11,25
C+	35,55	5,51	28,22	45,94
E+	67,68	4,39	61,51	74,22

Fonte: Elaborada pela autora.

Nota – Foram calculados MIPGap médio, desvio padrão, mínimo e máximo para cada conjunto de instâncias da mesma classe. Em negrito, estão destacados os menores valores obtidos em cada coluna.

$E = E_1 \cup E_2$, com $E_1 = N$ e $E_2 = T$, relacionadas com as variáveis Y_{it} , com $i \in N$ e $j \in T$. Os números de agrupamentos escolhidos para cada subconjunto foram baseados em agrupamentos balanceados de forma que tivessem 3 ou 4 entidades em cada agrupamento ($janela = \{3, 4\}$).

Por exemplo, se $|E_1| = 24$ e escolhe-se a opção 3, o número de agrupamentos é 8 ($|E_1|/3 = 24/3 = 8$). Note que, nos agrupamentos não balanceados, há casos em que há cerca de 3 entidades em alguns agrupamentos e há casos em que a divisão não é inteira. Então, o número de agrupamentos é arredondado para o menor número inteiro maior que a divisão. Na [Tabela 9](#) e na [Tabela 10](#), encontramos o número de agrupamentos para $E_1 = N$ e $E_2 = T$, respectivamente.

Para definir o grafo de entidades para cada subconjuntos de entidades, os pesos foram definidos da seguinte forma:

Dados e e $e' \in E_1$ existe peso $w_{ee'}$ se $r_{ee'} > 0$:

Tabela 9 – Número de agrupamentos para itens (N) para MLCLSP.

NI	$janela = 3$	$janela = 4$
10	4	3
40	14	10
100	34	25

Fonte: Elaborada pela autora.

Tabela 10 – Número de agrupamentos para períodos (T) para MLCLSP.

NT	$janela = 3$	$janela = 4$
24	8	6
48	16	12

Fonte: Elaborada pela autora.

$$w_{ee'} = \begin{cases} 1, & \text{se existe } t \in T \text{ tal que } X_{e't} > 0, \\ |E|, & \text{caso contrário.} \end{cases}$$

ou se $r_{e'e} > 0$:

$$w_{ee'} = \begin{cases} 1, & \text{se existe } t \in T \text{ tal que } X_{et} > 0, \\ |E|, & \text{caso contrário.} \end{cases}$$

Dados e e $e' \in E_2$ existe peso $w_{ee'}$ se $|e - e'| = 1$ e:

$$w_{ee'} = \begin{cases} 1, & \text{se existe } i \in N \text{ tal que } I_{ie} > 0 \text{ se } e < e' \text{ ou } I_{ie'} > 0 \text{ se } e' < e, \\ |E|, & \text{caso contrário.} \end{cases}$$

Como os grafos de adjacência obtidos diferem para cada instância e da solução corrente, teria que ser feita uma implementação de forma em que seja possível representar o grafo no plano cartesiano para cada tipo e utilizar em algoritmos de agrupamento *k-means* e *k-means* balanceado. Por isso, para o MLCLSP foram escolhidos apenas os algoritmos de agrupamento *k-medoids* e *k-medoids* balanceado.

Foi definido P como sendo os conjuntos a serem agrupados, com $P = \{N, T, NT\}$, em que NT indica o particionamento de N e T (e não número de períodos). Por exemplo, suponha

que N seja agrupado em $N^{(1)}$, $N^{(2)}$ e $N^{(3)}$ e T seja agrupado em $T^{(1)}$ e $T^{(2)}$. Então, encontra-se $NC = 6$ combinações de agrupamentos: $Q^1 = N^{(1)} \cup T^{(1)}$, $Q^2 = N^{(1)} \cup T^{(2)}$, $Q^3 = N^{(2)} \cup T^{(1)}$, $Q^4 = N^{(2)} \cup T^{(2)}$, $Q^5 = N^{(3)} \cup T^{(1)}$ e $Q^6 = N^{(3)} \cup T^{(2)}$.

A seguir, estão listados, para cada parte das heurísticas, os métodos utilizados e as opções de configurações escolhidas:

- solução inicial:
 - G: primeira solução factível encontrada pelo *Gurobi*;
 - RF p : RF com $janela = 4$ e $overlap = 2$, usando $p\%$ do limite total de 900 segundos, com $p = \{10, 50\}$;
- procedimento de melhoria:
 - FO $janelaP$: $janela = \{3, 4\}$ e $P = \{N, T\}$;
 - ADN $vkjanelaP$: versão v do ADN, com $v = \{1, 2, 3\}$, usando o algoritmo de agrupamento k , com $k = \{KM, KMB\}$ (KM: *k-medoids*, KMB: *k-medoids* balanceado), $janela = \{3, 4\}$ e $P = \{I, T, IT\}$;

No total, há 120 combinações possíveis. O número máximo de iterações para o FO é igual a 1000 e, para os algoritmos de agrupamentos é igual a 300. A seguir, serão apresentados os resultados obtidos pelas heurísticas. Em algumas análises, selecionamos apenas algumas heurísticas para facilitar a visualização. Foram selecionadas as heurísticas que apresentaram os melhores desempenhos nos testes preliminares em relação à solução obtida pelo *Gurobi* e pelas heurísticas.

A [Figura 14](#) apresenta o *box plot* do MIPGap do *Gurobi* e do GAPIni para cada método utilizado para encontrar a solução inicial para cada conjunto de instâncias da mesma classe. Observamos que o método G obteve os maiores GAPIni para todas as classes de instâncias. Os métodos RF10 e RF50 obtiveram o GAPIni próximos do MIPGap do *Gurobi*.

Na [Tabela 11](#) apresentamos o cálculo do $GAPIni_{med}$, $GAPIni_{dp}$, $GAPIni_{min}$ e $GAPIni_{max}$ e o $MIPGap_{med}$, $MIPGap_{dp}$, $MIPGap_{min}$ e $MIPGap_{max}$ do *Gurobi* para cada conjunto de heurísticas que utilizou o mesmo método para encontrar a solução inicial para cada conjunto de instâncias da mesma classe. Observamos na classe A+ que os métodos RF10 e RF50 possuem os menores $GAPIni_{med}$ e, para a classe C+ e E+, o método R50 possui o menor $GAPIni_{med}$. Observamos que para a classe C+, os métodos que utilizaram o RF50 obtiveram o $GAPIni_{med}$ menor do que o $MIPGap_{med}$ e, para a classe E+, foram os métodos que utilizaram o RF10 e RF50.

Figura 14 – Box plot do MIPGap do *Gurobi* e GAPIni das heurísticas para MLCLSP.

Nota – O GAPIni é em relação ao limitante inferior do *Gurobi* e à solução inicial da heurística. Para o *Gurobi* e cada tipo de método usado para encontrar a solução inicial, encontra-se um conjunto de box plot, que é para cada conjunto de instâncias da mesma classe.

Fonte: Elaborada pela autora.

Tabela 11 – GAPIni das heurísticas e o MIPGap do *Gurobi*.

Solução inicial	Classe	$GAPIni_{med} (\%)$	$GAPIni_{dp} (\%)$	$GAPIni_{min} (\%)$	$GAPIni_{max} (\%)$
G	A+	99,81	0,09	99,65	99,90
	C+	100,0	0,00	100,00	100,00
	E+	99,99	0,00	99,99	99,99
RF10	A+	10,72	2,89	4,45	14,55
	C+	36,36	4,06	26,75	48,22
	E+	62,81	3,95	55,77	71,31
RF50	A+	10,72	2,89	4,45	14,55
	C+	32,78	3,57	25,22	38,90
	E+	56,29	3,16	49,79	64,94
<i>Gurobi</i>	A+	7,09	2,77	2,93	11,25
	C+	35,55	5,51	28,22	45,94
	E+	67,68	4,39	61,51	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre os métodos para encontrar a solução inicial para instâncias da mesma classe em cada coluna.

No total, foram 97 heurísticas, dentre 120, que obtiveram GAP_{med} menor que o MIPGap do *Gurobi*. Selecionamos 5 heurísticas que objetivem o menor GAP dentre eles para cada método

Tabela 12 – GAP das heurísticas e pelo *Gurobi* para MPCLSP.

Heurísticas	$GAP_{med} (%)$	$GAP_{dp} (%)$	$GAP_{min} (%)$	$GAP_{max} (%)$	$\frac{solI}{24} \times 100$
GADN3KM3N	30,83	16,68	5,40	54,96	100,00
GADN3KM4N	31,10	16,65	6,71	54,61	100,00
GADN3KM4NT	30,18	16,65	5,53	52,74	100,00
GADN3KMB3N	31,35	17,02	6,08	52,68	100,00
GADN3KMB4N	30,90	17,13	5,32	53,92	100,00
RF10ADN3KM4NT	29,04	16,82	3,85	51,47	100,00
RF10ADN3KM4T	28,66	18,06	3,49	52,36	95,83
RF10ADN3KMB3N	29,39	17,43	3,85	54,22	95,83
RF10ADN3KMB3NT	28,73	17,12	4,22	52,50	91,67
RF10FO4N	29,42	17,64	3,84	55,85	100,00
RF50ADN3KM3N	29,36	16,98	3,69	53,94	100,00
RF50ADN3KM4NT	28,71	16,67	3,85	51,68	100,00
RF50ADN3KMB3N	29,34	17,07	3,85	53,00	100,00
RF50ADN3KMB4N	27,53	16,84	3,49	53,68	91,67
RF50ADN3KMB4NT	29,41	16,95	4,10	51,27	100,00
<i>Gurobi</i>	36,77	25,63	2,93	74,22	100,00

Fonte: Elaborada pela autora.

Nota – A coluna $\frac{solI}{24} \times 100$ indica a porcentagem do número de instâncias que obtiveram solução factíveis ou ótimas. Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

(G, RF10 e RF50) e apresentamos, na [Tabela 12](#), o cálculo do GAP_{med} , GAP_{dp} , GAP_{min} e máximo GAP_{max} e o $MIPGap_{med}$, $MIPGap_{dp}$, $MIPGap_{min}$ e $MIPGap_{max}$ do *Gurobi*. As heurísticas não conseguiram obter solução ótima para nenhuma instância. Nessa tabela, estão destacados, em negrito, os menores valores dentre os métodos em cada coluna. Observamos que a heurística RF50ADN3KMB4N foi a que obteve o menor GAP_{med} .

Calculamos o tempo gasto médio pelas heurísticas selecionadas e os apresentamos na [Tabela 13](#), em segundos, o $Tempo_{med}$, $Tempo_{dp}$, $Tempo_{min}$, $Tempo_{max}$, o número de instâncias que atingiu o tempo limite ($nInst$) e a porcentagem ($\frac{nInst}{24} \times 100 \%$) que esse número representa do número total de instâncias para cada heurística e o *Gurobi*. A heurística que obteve a menor média foi RF10ADN3KMB3NT.

Na [Tabela 14](#), é apresentado o número de iterações médio ($nIter_{med}$), desvio padrão ($nIter_{dp}$), mínimo ($nIter_{min}$), máximo ($nIter_{max}$) e a porcentagem ($\frac{solI}{24} \times 100 \%$) que esse número representa do número total de instâncias para cada heurística selecionada e o *Gurobi*. A heurística RF10ADN3KM4T obteve a menor média dentre todas as heurísticas consideradas.

Geralmente, observamos que a maioria das heurísticas que utilizou o RF50 obteve o GAP_{ini} menor. As heurísticas que utilizaram o ADN3 obtiveram o GAP menor, em destaque, RF50ADN3KMB4N obteve o menor GAP. A heurística que gastou menos tempo em média foi a RF10ADN3KMB3NT. No [Apêndice C](#), temos as análises dos resultados com mais detalhes.

Tabela 13 – Tempo gasto pelas heurísticas e pelo *Gurobi* para MLCLSP.

Heurísticas	Tempo _{med} (s)	Tempo _{dp} (s)	Tempo _{min} (s)	Tempo _{max} (s)	$\frac{nInst}{24} \times 100$
GADN3KM3N	575,33	407,95	3,00	900,00	4,17
GADN3KM4N	615,62	390,02	41,00	900,00	20,83
GADN3KM4NT	496,54	375,50	4,00	900,00	25,00
GADN3KMB3N	425,29	328,32	7,00	900,00	8,33
GADN3KMB4N	552,71	365,14	41,00	900,00	25,00
RF10ADN3KM4NT	408,04	347,30	6,00	900,00	12,50
RF10ADN3KM4T	596,13	425,13	15,00	900,00	50,00
RF10ADN3KMB3N	388,43	360,01	21,00	900,00	20,83
RF10ADN3KMB3NT	330,73	371,83	7,00	900,00	25,00
RF10FO4N	606,25	424,37	12,00	900,00	66,67
RF50ADN3KM3N	583,50	404,92	10,00	900,00	4,17
RF50ADN3KM4NT	522,38	384,21	5,00	900,00	20,83
RF50ADN3KMB3N	466,62	352,31	13,00	900,00	8,33
RF50ADN3KMB4N	527,59	372,19	31,00	900,00	16,67
RF50ADN3KMB4NT	432,92	341,77	4,00	900,00	12,50
<i>Gurobi</i>	900,00	0,00	900,00	900,00	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna. Além do tempo gasto, há o número de instâncias que atingiu o tempo limite ($nInst$) e a porcentagem ($\frac{nInst}{24} \times 100 \%$) do número do total de instâncias.

Tabela 14 – Número de iterações realizadas pelas heurísticas.

Heurísticas	nIter _{med} (s)	nIter _{dp} (s)	nIter _{min} (s)	nIter _{max} (s)	$\frac{solI}{24} \times 100$
GADN3KM3N	6,00	2,34	3,00	12,00	100,00
GADN3KM4N	6,67	2,55	3,00	15,00	100,00
GADN3KM4NT	6,67	2,55	3,00	15,00	100,00
GADN3KMB3N	6,33	2,14	3,00	12,00	100,00
GADN3KMB4N	5,88	1,78	4,00	11,00	100,00
RF10ADN3KM4NT	5,25	2,31	2,00	11,00	100,00
RF10ADN3KM4T	3,22	0,60	2,00	4,00	95,83
RF10ADN3KMB3N	5,17	2,12	3,00	12,00	95,83
RF10ADN3KMB3NT	4,45	1,74	2,00	8,00	91,67
RF10FO4N	3,50	1,29	2,00	5,00	100,00
RF50ADN3KM3N	5,08	1,67	2,00	9,00	100,00
RF50ADN3KM4NT	4,67	1,76	2,00	8,00	100,00
RF50ADN3KMB3N	4,88	1,80	3,00	9,00	100,00
RF50ADN3KMB4N	4,09	1,63	2,00	7,00	91,67
RF50ADN3KMB4NT	4,46	1,96	2,00	8,00	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna. A coluna $\frac{solI}{24} \times 100$ indica a porcentagem de instâncias que obtiveram soluções factíveis.

CONCLUSÕES E PERSPECTIVAS FUTURAS

Neste trabalho, foi abordado o uso de *Machine Learning* (ML), no caso, algoritmos de agrupamento para ajudar a particionar o conjunto de variáveis de dois problemas clássicos: o problema de dimensionamento de lotes com múltiplas plantas (MPCLSP) e o problema de dimensionamento de lotes multiestágio (MLCLSP).

Para isso foram implementadas três versões do *Automatically designed neighborhoods* (ADN) baseados em [Ghiani, Laporte e Manni \(2015\)](#), chamados aqui de ADN1, ADN2 e ADN3. O ADN1 explora a vizinhança básica e a vizinhança de intensificação uma vez, enquanto o ADN2 explora várias vezes e cada vez que termina, atualiza a vizinhança com a melhor solução encontrada. Já o ADN3 explora apenas a vizinhança básica atualizando-a toda vez que encontra uma solução incumbente e, depois que termina de explorar, é definida uma nova vizinhança básica. Os algoritmos de agrupamentos implementados foram o *k-means* e o *k-medoids*; além desses, foi implementada para cada um deles a versão que encontra as partições balanceadas. Para comparar os resultados foi implementado o *Fix-and-Optimize* (FO) baseado em [Furlan \(2011\)](#).

Para o problema MPCLSP, observamos que o *Gurobi* consegue encontrar as soluções ótimas para maioria dos casos e, quando não encontra, o MIPGap é pequeno. A maioria das soluções iniciais encontradas pelo *Relax-and-Fix* (RF) já eram soluções ótimas. E as soluções iniciais encontradas pelo *Gurobi* (G) possuíam GAPIni altas, mas as heurísticas conseguiram melhorá-las significativamente. Observamos que a maioria das heurísticas conseguiu GAP próximos às das heurísticas que utilizaram o FO, exceto as heurísticas que utilizam G e ADN1.

Para o problema MLCLSP, observamos que o *Gurobi* não conseguiu encontrar soluções ótimas e atingiu o tempo limite de 900 segundos em todas as instâncias e nenhuma das heurísticas também conseguiu encontrar soluções ótimas. Para as soluções iniciais encontradas pelo G, as heurísticas conseguiram melhorar bastante; enquanto para soluções encontradas pelo RF o GAPM foi menor. Observamos que a maioria das heurísticas ADN3 consegue melhorar para

mais instâncias em relação às outras heurísticas e consegue GAP menores.

Concluimos que o ADN1 não consegue resultados melhores em comparação com o ADN2 e ADN3, pois realiza menos iterações. A qualidade da solução inicial interfere no procedimento de melhoria. Observando os resultados obtidos para o MPCLSP, não conseguimos concluir qual heurística foi a melhor e para o MLCLSP, a versão ADN3 apresenta resultados promissores, competitivos e próximos aos obtidos com FO.

Trabalhos futuros estarão concentrados na realização de testes computacionais aplicando em outras instâncias. Mesmo que os resultados das heurísticas ADN sejam próximos com o FO, o ganho seria no fato de podermos obter as partições de forma automática. Os agrupamentos não balanceados podem ter diferentes agrupamentos, tornando interessante uma análise do impacto desses agrupamentos nos resultados e a escolha do número de agrupamentos. Recentemente, uma revisão sobre o uso de aprendizado em meta-heurísticas aponta resultados promissores usando aprendizado de máquina por reforço ou transferência de aprendizado (do inglês, *transfer learning*) ([KARIMI-MAMAGHAN et al., 2021](#)). Por isso, usar as heurísticas para a escolha das partições de forma dinâmica parece ser interessante.

REFERÊNCIAS

- ADAMO, T.; GHIANI, G.; GUERRIERO, E.; MANNI, E. A learn-and-construct framework for general mixed-integer programming problems. **International Transactions in Operational Research**, Wiley Online Library, v. 27, n. 1, p. 9–25, 2020. Citado na página 31.
- AFIA, A. E.; KABBAJ, M. M. Supervised learning in branch-and-cut strategies. In: **Proceedings of the 2nd international Conference on Big Data, Cloud and Applications**. [S.l.: s.n.], 2017. p. 1–6. Citado na página 30.
- AKARTUNALI, K.; MILLER, A. J. A heuristic approach for big bucket multi-level production planning problems. **European Journal of Operational Research**, Elsevier, v. 193, n. 2, p. 396–411, 2009. Citado nas páginas 23 e 34.
- ALVAREZ, A. M.; LOUVEAUX, Q.; WEHENKEL, L. A machine learning-based approximation of strong branching. **INFORMS Journal on Computing**, INFORMS, v. 29, n. 1, p. 185–195, 2017. Citado na página 30.
- ARENALES, M.; ARMENTANO, V.; MORABITO, R.; YANASSE, H. **Pesquisa operacional**. Rio de Janeiro: Elsevier: ABEPRO, 2007. Citado na página 25.
- BENGIO, Y.; LODI, A.; PROUVOST, A. Machine learning for combinatorial optimization: a methodological tour d’horizon. **European Journal of Operational Research**, Elsevier, 2020. Citado nas páginas 28 e 29.
- BILLINGTON, P. J.; MCCLAIN, J. O.; THOMAS, L. J. Mathematical programming approaches to capacity-constrained mrp systems: review, formulation and problem reduction. **Management Science**, INFORMS, v. 29, n. 10, p. 1126–1141, 1983. Citado na página 23.
- BRAHIMI, N.; DAUZERE-PERES, S.; NAJID, N. M.; NORDLI, A. Single item lot sizing problems. **European Journal of Operational Research**, Elsevier, v. 168, n. 1, p. 1–16, 2006. Citado na página 23.
- CARVALHO, D. M.; NASCIMENTO, M. C. Lagrangian heuristics for the capacitated multi-plant lot sizing problem with multiple periods and items. **Computers & Operations Research**, Elsevier, v. 71, p. 137–148, 2016. Citado nas páginas 24, 32, 45, 51 e 77.
- _____. A kernel search to the multi-plant capacitated lot sizing problem with setup carry-over. **Computers & Operations Research**, v. 100, p. 43–53, 2018. ISSN 0305-0548. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S030505481830193X>>. Citado na página 32.
- CASERTA, M.; VOSS, S. Metaheuristics: intelligent problem solving. In: **Matheuristics**. [S.l.]: Springer, 2009. p. 1–38. Citado na página 27.
- CUNHA, J. O.; KRAMER, H. H.; MELO, R. A. Effective matheuristics for the multi-item capacitated lot-sizing problem with remanufacturing. **Computers & Operations Research**, Elsevier, v. 104, p. 149–158, 2019. Citado na página 27.

DANNA, E.; ROTHBERG, E.; PAPE, C. L. Exploring relaxation induced neighborhoods to improve mip solutions. **Mathematical Programming**, Springer, v. 102, n. 1, p. 71–90, 2005. Citado na página 26.

DORIGO, M.; MANIEZZO, V.; COLORNI, A. Ant system: optimization by a colony of cooperating agents. **IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)**, IEEE, v. 26, n. 1, p. 29–41, 1996. Citado na página 27.

FISCHETTI, M.; GLOVER, F.; LODI, A. The feasibility pump. **Mathematical Programming**, Springer, v. 104, n. 1, p. 91–104, 2005. Citado na página 26.

FISCHETTI, M.; LODI, A. Local branching. **Mathematical programming**, Springer, v. 98, n. 1-3, p. 23–47, 2003. Citado na página 26.

FURLAN, M. M. **Métodos heurísticos para o problema de dimensionamento de lotes multiestágio com limitação de capacidade**. Tese (Doutorado) — Universidade de São Paulo, 2011. Citado nas páginas 23, 25, 26, 34, 39, 41 e 71.

FURLAN, M. M.; SANTOS, M. O. Bfo: a hybrid bees algorithm for the multi-level capacitated lot-sizing problem. **Journal of Intelligent Manufacturing**, Springer, v. 28, n. 4, p. 929–944, 2017. Citado nas páginas 23, 27 e 34.

GHIANI, G.; LAPORTE, G.; MANNI, E. Model-based automatic neighborhood design by unsupervised learning. **Computers & Operations Research**, Elsevier, v. 54, p. 108–116, 2015. Citado nas páginas 23, 24, 31, 32, 42, 44, 45, 46, 48, 54, 63 e 71.

GLOCK, C. H.; GROSSE, E. H.; RIES, J. M. The lot sizing problem: A tertiary study. **International Journal of Production Economics**, Elsevier, v. 155, p. 39–51, 2014. Citado na página 23.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computers operations research**, v. 13, n. 5, p. 533–549, 1986. Citado na página 27.

HE, H.; III, H. D.; EISNER, J. M. Learning to search in branch and bound algorithms. In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N. D.; WEINBERGER, K. Q. (Ed.). **Advances in Neural Information Processing Systems 27**. Curran Associates, Inc., 2014. p. 3293–3301. Disponível em: <<http://papers.nips.cc/paper/5495-learning-to-search-in-branch-and-bound-algorithms.pdf>>. Citado na página 30.

HELBERT, S.; SAHLING, F. A fix-and-optimize approach for the multi-level capacitated lot sizing problem. **International Journal of Production Economics**, Elsevier, v. 123, n. 2, p. 247–256, 2010. Citado nas páginas 23, 34 e 40.

HOLLAND, J. H. *et al.* **Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence**. [S.l.]: MIT press, 1992. Citado na página 27.

JAIN, A. K.; DUBES, R. C. **Algorithms for clustering data**. [S.l.]: Prentice-Hall, Inc., 1988. Citado nas páginas 24 e 46.

KARIMI, B.; GHOMI, S. M. T. F.; WILSON, J. M. The capacitated lot sizing problem: a review of models and algorithms. **Omega**, Elsevier, v. 31, n. 5, p. 365–378, 2003. Citado na página 23.

- KARIMI-MAMAGHAN, M.; MOHAMMADI, M.; MEYER, P.; KARIMI-MAMAGHAN, A. M.; TALBI, E.-G. Machine learning at the service of meta-heuristics for solving combinatorial optimization problems: A state-of-the-art. **European Journal of Operational Research**, Elsevier, 2021. Citado na página 72.
- KIRKPATRICK, S. Optimization by simulated annealing: Quantitative studies. **Journal of statistical physics**, Springer, v. 34, n. 5-6, p. 975–986, 1984. Citado na página 27.
- LODI, A.; ZARPELLON, G. On learning and branching: a survey. **Top**, Springer, v. 25, n. 2, p. 207–236, 2017. Citado na página 28.
- LOMBARDI, M.; MILANO, M. Boosting combinatorial problem modeling with machine learning. **arXiv preprint arXiv:1807.05517**, 2018. Citado na página 30.
- LUXBURG, U. V.; SCHÖLKOPF, B. Statistical learning theory: Models, concepts, and results. In: **Handbook of the History of Logic**. [S.l.]: Elsevier, 2011. v. 10, p. 651–706. Citado na página 28.
- MAES, J.; MCCLAIN, J. O.; WASSENHOVE, L. N. V. Multilevel capacitated lotsizing complexity and lp-based heuristics. **European Journal of Operational Research**, Elsevier, v. 53, n. 2, p. 131–148, 1991. Citado nas páginas 23, 24 e 34.
- MELEGA, G. M.; FIOROTTO, D. J.; ARAUJO, S. A. de. Formulações fortes para o problema de dimensionamento de lotes com várias plantas. **TEMA (São Carlos)**, SciELO Brasil, v. 14, n. 3, p. 305–318, 2013. Citado na página 32.
- MELLO, R. F.; PONTI, M. A. **Machine Learning: A Practical Approach on the Statistical Learning Theory**. [S.l.]: Springer, 2018. Citado na página 28.
- MERCÉ, C.; FONTAN, G. Mip-based heuristics for capacitated lotsizing problems. **International Journal of Production Economics**, Elsevier, v. 85, n. 1, p. 97–111, 2003. Citado na página 53.
- NASCIMENTO, M. C.; RESENDE, M. G.; TOLEDO, F. M. Grasp heuristic with path-relinking for the multi-plant capacitated lot sizing problem. **European Journal of Operational Research**, Elsevier, v. 200, n. 3, p. 747–754, 2010. Citado na página 32.
- PARK, H.-S.; JUN, C.-H. A simple and fast algorithm for k-medoids clustering. **Expert systems with applications**, Elsevier, v. 36, n. 2, p. 3336–3341, 2009. Citado nas páginas 24 e 46.
- PHAM, D.; GHANBARZADEH, A.; KOC, E.; OTRI, S.; RAHIM, S.; ZAIDI, M. The bees algorithm. **Technical Note, Manufacturing Engineering Centre, Cardiff University, UK**, 2005. Citado na página 27.
- POCHET, Y.; WOLSEY, L. A. **Production planning by mixed integer programming**. [S.l.]: Springer Science & Business Media, 2006. Citado nas páginas 25, 26, 38 e 40.
- SAMBASIVAN, M.; SCHMIDT, C. P. A heuristic procedure for solving multi-plant, multi-item, multi-period capacitated lot-sizing problems. **Asia Pacific Journal of Operational Research**, NATIONAL UNIVERSITY OF SINGAPORE, v. 19, n. 1, p. 87–106, 2002. Citado nas páginas 23, 24 e 32.

- STADTLER, H.; SAHLING, F. A lot-sizing and scheduling model for multi-stage flow lines with zero lead times. **European Journal of Operational Research**, Elsevier, v. 225, n. 3, p. 404–419, 2013. Citado na página [27](#).
- STADTLER, H.; SÜRIE, C. Description of mlclsp test instances. **Technische Universität Darmstadt, Tech. Rep**, 2000. Citado nas páginas [24](#), [62](#) e [63](#).
- TALBI, E.-G. Machine learning into metaheuristics: A survey and taxonomy of data-driven metaheuristics. 2020. Citado na página [29](#).
- TANG, Y.; AGRAWAL, S.; FAENZA, Y. Reinforcement learning for integer programming: Learning to cut. **arXiv preprint arXiv:1906.04859**, 2019. Citado na página [31](#).
- TEMPELMEIER, H.; BUSCHKÜHL, L. A heuristic for the dynamic multi-level capacitated lotsizing problem with linked lotsizes for general product structures. **Or Spectrum**, Springer, v. 31, n. 2, p. 385–404, 2009. Citado na página [63](#).
- TOLEDO, C. F. M.; ARANTES, M. da S.; HOSSOMI, M. Y. B.; FRANÇA, P. M.; AKAR-TUNALI, K. A relax-and-fix with fix-and-optimize heuristic applied to multi-level lot-sizing problems. **Journal of Heuristics**, Springer, v. 21, n. 5, p. 687–717, 2015. Citado nas páginas [23](#), [26](#) e [34](#).

EXEMPLO DE COMO CALCULAR O PESO ENTRE DUAS ENTIDADES

Para exemplificar como são realizados os cálculos dos pesos utilizados para determinar a matriz de dissimilaridade, consideramos um exemplo de MPCLSP. O cálculo do peso $w_{ee'}$ dadas duas entidades $e, e' \in E_r$ é definido na [Subsubseção 3.2.2.1](#). Nesse exemplo, o cálculo foi feito apenas para o $w_{1,2}$, com $1, 2 \in T$ e baseado em uma instância gerada por (CARVALHO; NASCIMENTO, 2016) que foi adaptada de forma que a instância tenha $N = \{1, 2, 3\}$, $M = \{1\}$ e $T = \{1, 2, 3\}$ e a seguinte formulação:

$$\begin{aligned} \text{Min } & 2,4X_{000} + 2,4X_{001} + 2,4X_{002} + 1,5X_{100} + 1,5X_{101} + 1,5X_{102} + 1,5X_{200} + \\ & 1,5X_{201} + 1,5X_{202} + 0,2I_{000} + 0,2I_{001} + 0,2I_{002} + 0,3I_{100} + 0,3I_{101} + \\ & 0,3I_{102} + 0,3I_{200} + 0,3I_{201} + 0,3I_{202} + 78,1Y_{000} + 78,1Y_{001} + 78,1Y_{002} + \\ & 819,8Y_{100} + 819,8Y_{101} + 819,8Y_{102} + 241,2Y_{200} + 241,2Y_{201} + 241,2Y_{202} \end{aligned} \quad (\text{A.1})$$

$$\text{s.a. } X_{000} = 64, \quad (\text{A.2})$$

$$X_{100} = 53, \quad (\text{A.3})$$

$$X_{200} = 127, \quad (\text{A.4})$$

$$X_{001} + I_{000} = 162, \quad (\text{A.5})$$

$$X_{101} + I_{100} = 70, \quad (\text{A.6})$$

$$X_{201} + I_{200} = 169, \quad (\text{A.7})$$

$$X_{002} + I_{001} = 11, \quad (\text{A.8})$$

$$X_{102} + I_{101} = 97, \quad (\text{A.9})$$

$$X_{202} + I_{201} = 160, \quad (\text{A.10})$$

$$1,1X_{000} + 1,7X_{100} + 1,8X_{200} + 52Y_{000} + 54,5Y_{100} + 17,7Y_{200} \leq 1514, \quad (\text{A.11})$$

$$1,1X_{001} + 1,7X_{101} + 1,8X_{201} + 52Y_{001} + 54,5Y_{101} + 17,7Y_{201} \leq 1514, \quad (\text{A.12})$$

$$1, 1X_{002} + 1, 7X_{102} + 1, 8X_{202} + 52Y_{002} + 54, 5Y_{102} + 17, 7Y_{202} \leq 1514, \quad (\text{A.13})$$
$$X_{000} - 192Y_{000} \leq 0, \quad (\text{A.14})$$
$$X_{001} - 324Y_{001} \leq 0, \quad (\text{A.15})$$
$$X_{002} - 11Y_{002} \leq 0, \quad (\text{A.16})$$
$$X_{100} - 159Y_{100} \leq 0, \quad (\text{A.17})$$
$$X_{101} - 140Y_{101} \leq 0, \quad (\text{A.18})$$
$$X_{102} - 97Y_{102} \leq 0, \quad (\text{A.19})$$
$$X_{200} - 381Y_{200} \leq 0, \quad (\text{A.20})$$
$$X_{201} - 338Y_{201} \leq 0, \quad (\text{A.21})$$
$$X_{202} - 160Y_{202} \leq 0, \quad (\text{A.22})$$
$$X_{ijt}, I_{ijt} \geq 0, \quad \forall (i, j, t), \quad (\text{A.23})$$
$$Y_{ijt} \in \{0, 1\}, \quad \forall (i, j, t), \quad (\text{A.24})$$

A solução inicial é dada por:

$$\begin{aligned} x^{(h)} &= (X_{000}, X_{001}, X_{002}, X_{100}, X_{101}, X_{102}, X_{200}, X_{201}, X_{202}, I_{000}, I_{001}, I_{002}, I_{100}, I_{101}, I_{102}, \\ &\quad I_{200}, I_{201}, I_{202}, Y_{000}, Y_{001}, Y_{002}, Y_{100}, Y_{101}, Y_{102}, Y_{200}, Y_{201}, Y_{202}) \\ &= (64, 0, 0, 53, 0, 0, 127, 0, 0, 162, 11, 0, 70, 97, 0, 169, 160, 0, 1, 0, 0, 1, 0, 0, 1, 0, 0) \end{aligned} \quad (\text{A.25})$$
$$b = (64, 53, 127, 162, 70, 169, 11, 97, 160, 1514, 1514, 1514, 0, 0, 0, 0, 0, 0, 0, 0) \quad (\text{A.26})$$
[illegible]

Com a função objetivo (A.1), restrições de (A.2) a (A.24) e solução inicial factível $x^{(h)}$ (A.25), o vetor b (A.26) e a matriz dos coeficientes A (A.27) em que aparecem apenas os valores diferentes de zero para facilitar a visualização, podemos iniciar o processo para construção da matriz que será utilizada nos algoritmos de agrupamento.

Agora definimos os seguintes subconjuntos:

- $J_1 = \{X_{001}, X_{101}, X_{201}, I_{001}, Y_{101}, I_{201}, Y_{001}, Y_{101}, Y_{201}\}$
- $J_2 = \{X_{002}, X_{102}, I_{202}, I_{002}, Y_{102}, I_{202}, Y_{002}, Y_{102}, Y_{202}\}$
- $J_1 \cap J_2 = \emptyset$
- $I = \{1, 2, \dots, 21\}$
- $I_1 = \{4, 5, 6\}$
- $I_2 = \{7, 8, 9\}$
- $I(x^{(h)}) = \{1, 2, 3, 4, 5, 6, 7, 8, 9\}$
- $I_1(x^{(h)}) = \{1, 2, 3\}$
- $I_2(x^{(h)}) = \{7, 8, 9\}$

Para o cálculo de $w_{1,2}$, precisamos calcular primeiro o $S_{1,2}$. Caso $S_{1,2} > 0$, então temos que $w_{1,2} = 1$; caso $S_{1,2} = 0$, calculamos o $M_{1,2}$. Caso $M_{1,2} > 0$, então temos que $w_{1,2} = |E| = |N| + |M| + |T| = 3 + 1 + 3 = 7$; caso $M_{1,2} = 0$ então $w_{1,2}$ não existe.

Na Figura 15 temos a matriz A em qual foram selecionados os conjuntos que serão usados para o cálculo de $S_{12}(x^{(h)}) = P_{S1} + P_{S2} + P_{S3}$. Note que, nesta figura, o P_{S1} não existe, o P_{S2} é a parte cinza-escuro e o P_{S3} é a parte cinza-claro então temos:

- $P_{S1} = \sum_{i \in I(x^{(h)})} \sum_{j \in J_1 \cap J_2} H(|a_{ij}|x_j^{(h)}) = 0,$
- $P_{S2} = \sum_{i \in I_1(x^{(h)})} \sum_{j \in J_2 \setminus J_1} H(|a_{ij}|x_j^{(h)}) = H(|a_{4,2}|x_2^{(h)}) + H(|a_{5,5}|x_5^{(h)}) + H(|a_{6,8}|x_8^{(h)}) +$
 $H(|a_{7,11}|x_{11}^{(h)}) + H(|a_{8,14}|x_{14}^{(h)}) + H(|a_{9,17}|x_{17}^{(h)}) = H(|1|0) + H(|1|0) +$
 $H(|1|0) + H(|1|11) + H(|1|97) + H(|1|160) = 0 + 0 + 0 + 1 + 1 + 1 = 3,$
- $P_{S3} = \sum_{i \in I_2(x^{(h)})} \sum_{j \in J_1 \setminus J_2} H(|a_{ij}|x_j^{(h)}) = H(|a_{7,3}|x_3^{(h)}) + H(|a_{8,6}|x_6^{(h)}) + H(|a_{9,9}|x_9^{(h)}) =$
 $H(|1|0) + H(|1|0) + H(|1|0) = 0 + 0 + 0 = 0$

Figura 16 – Matriz de coeficientes A dado em A.27 rearranjada

The diagram shows a large square matrix I of size 10x10, partitioned into four 5x5 blocks. The top-left 5x5 block is shaded dark gray and labeled J_1 . The top-right 5x5 block is labeled J_2 . The bottom-left 5x5 block is shaded light gray and labeled J_1 . The bottom-right 5x5 block is labeled J_2 . The matrix I is further partitioned by solid lines into four 2.5x2.5 blocks. The top-left 2.5x2.5 block is labeled I_1 . The top-right 2.5x2.5 block is labeled I_2 . The bottom-left 2.5x2.5 block is labeled I_1 . The bottom-right 2.5x2.5 block is labeled I_2 . The matrix I is further partitioned by solid lines into four 2.5x2.5 blocks. The top-left 2.5x2.5 block is labeled I_1 . The top-right 2.5x2.5 block is labeled I_2 . The bottom-left 2.5x2.5 block is labeled I_1 . The bottom-right 2.5x2.5 block is labeled I_2 .

Nota – Essa matriz foi rearranjada de forma que as colunas e as linhas fossem agrupadas para facilitar a visualização. Seleccionamos os conjuntos que serão usados e, neste caso, temos $M_{1,2} = P_{M2}(\text{cinza-escuro}) + P_{M3}(\text{cinza-claro})$.

Fonte: Elaborada pela autora.

ANÁLISES DOS RESULTADOS DAS HEURÍSTICAS PARA O MPCLSP

Neste apêndice, apresentaremos as análises dos resultados para MPCLSP com mais detalhes. Observamos que algumas heurísticas não conseguiram encontrar solução inicial para algumas instâncias, pois o tempo disponível não era suficiente. Isto aconteceu para as instâncias com 80 e 90 itens e 15 plantas e estão listadas, a seguir, para cada heurística os nomes das instâncias:

- **RF10ADN2K:** ABA0122080_0, ABB0121590_4, NBA0121590_2;
- **RF10ADN2KB:** ABB0121580_1, NBA0121590_2;
- **RF10ADN2KM:** ABB0121580_1, ABB0121590_4;
- **RF10ADN2KMB:** ABB0121580_1, ABB0121590_4;
- **RF10ADN3K:** NBB0122090_1, ABB0121590_4, ABB0122090_2, NBA0121590_2, ABA0122090_1, NBB0122090_3, AAA0122090_1, AAA0122090_0;
- **RF10ADN3KM:** ABB0122080_3, NBA0122080_4, NBB0122080_1, ABB0122090_1, ABB0121590_4, NBA0122090_4, ABA0122090_0, ABB0122090_2, NBA0121590_2;
- **RF10ADN3KMB:** NBB0122080_2, NBB0122080_1, NBA0122090_1, ABB0121590_4, ABA0122090_0, NBA0121590_2, ABA0122090_2, NBB0122090_3, AAA0122090_0.

Nas próximas seções, serão apresentadas as análises dos resultados observando o GAP das instâncias que é em relação ao limitante inferior obtido pelo *Gurobi* e a solução obtida pela heurística; GAPM que é em relação à solução inicial e à solução depois do procedimento de melhoria; tempo gasto pelo *Gurobi* e pelas heurísticas; e o número de iterações no total, realizado pelas heurísticas.

B.1 GAP

Foi realizado o cálculo do GAP médio (GAP_{med}), desvio padrão (GAP_{dp}), GAP mínimo (GAP_{min}) e GAP máximo (GAP_{max}) das heurísticas e do MIPGap médio, desvio padrão, MIPGap mínimo e MIPGap máximo do *Gurobi*. As [Tabela 15](#), [Tabela 16](#), [Tabela 17](#), [Tabela 18](#), [Tabela 19](#), [Tabela 20](#) e [Tabela 21](#) mostram os valores obtidos do GAP para as instâncias com 6 itens, 12 itens, 25 itens, 50 itens, 70 itens, 80 itens e 90 itens, respectivamente.

B.2 GAPM

Foi realizado o cálculo do GAPM médio ($GAPM_{med}$), desvio padrão ($GAPM_{dp}$), GAPM mínimo ($GAPM_{min}$) e máximo GAPM ($GAPM_{max}$) das heurísticas. As [Tabela 22](#), [Tabela 23](#), [Tabela 24](#), [Tabela 25](#), [Tabela 26](#), [Tabela 27](#) e [Tabela 28](#) apresentam os valores obtidos do GAPM para as instâncias com 6 itens, 12 itens, 25 itens, 50 itens, 70 itens, 80 itens e 90 itens, respectivamente.

B.3 Tempo gasto

Foi realizado o cálculo do tempo gasto médio ($Tempo_{med}$), desvio padrão ($Tempo_{dp}$), mínimo ($Tempo_{min}$) e máximo ($Tempo_{max}$) para cada heurística e o *Gurobi*. As [Tabela 29](#), [Tabela 30](#), [Tabela 31](#), [Tabela 32](#), [Tabela 33](#), [Tabela 34](#) e [Tabela 35](#) apresentam o tempo gasto, em segundos, para as instâncias com 6, 12, 25, 50, 70, 80 e 90 itens, respectivamente.

B.4 Número de iterações

Foi realizado o cálculo do número de iterações médio ($nIter_{med}$), desvio padrão ($nIter_{dp}$), mínimo ($nIter_{min}$) e máximo ($nIter_{max}$) para cada heurística. As heurísticas que usam o ADN1 sempre têm duas iterações por isso não foram apresentadas nas tabelas seguintes. Nas [Tabela 36](#), [Tabela 37](#), [Tabela 38](#), [Tabela 39](#), [Tabela 40](#), [Tabela 41](#) e [Tabela 42](#), apresentamos o número de iterações para as instâncias com 6, 12, 25, 50, 70, 80 e 90 itens, respectivamente.

Tabela 15 – GAP das heurísticas e o MIPGap do *Gurobi* para instâncias com 6 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,04	0,08	0,00	0,47
GFO2	0,04	0,07	0,00	0,33
GFO3	0,01	0,05	0,00	0,28
GADN1K	7,35	6,55	0,00	28,84
GADN1KB	10,71	8,55	0,56	34,38
GADN1KM	10,15	8,58	0,49	32,38
GADN1KMB	10,71	8,55	0,56	34,38
GADN2K	0,01	0,03	0,00	0,28
GADN2KB	0,02	0,06	0,00	0,26
GADN2KM	0,00	0,00	0,00	0,01
GADN2KMB	0,02	0,06	0,00	0,26
GADN3K	0,01	0,02	0,00	0,17
GADN3KB	0,03	0,06	0,00	0,33
GADN3KM	0,01	0,05	0,00	0,28
GADN3KMB	0,01	0,04	0,00	0,25
RF10FO1	0,00	0,00	0,00	0,01
RF10FO2	0,00	0,00	0,00	0,02
RF10FO3	0,00	0,00	0,00	0,01
RF10ADN1K	0,00	0,00	0,00	0,01
RF10ADN1KB	0,00	0,00	0,00	0,02
RF10ADN1KM	0,00	0,00	0,00	0,01
RF10ADN1KMB	0,00	0,00	0,00	0,02
RF10ADN2K	0,00	0,00	0,00	0,01
RF10ADN2KB	0,00	0,00	0,00	0,02
RF10ADN2KM	0,00	0,00	0,00	0,01
RF10ADN2KMB	0,00	0,00	0,00	0,02
RF10ADN3K	0,00	0,00	0,00	0,02
RF10ADN3KB	0,00	0,00	0,00	0,02
RF10ADN3KM	0,00	0,00	0,00	0,01
RF10ADN3KMB	0,00	0,00	0,00	0,02
RF50FO1	0,00	0,00	0,00	0,01
RF50FO2	0,00	0,00	0,00	0,02
RF50FO3	0,00	0,00	0,00	0,01
RF50ADN1K	0,00	0,00	0,00	0,02
RF50ADN1KB	0,00	0,00	0,00	0,02
RF50ADN1KM	0,00	0,00	0,00	0,01
RF50ADN1KMB	0,00	0,00	0,00	0,02
RF50ADN2K	0,00	0,00	0,00	0,01
RF50ADN2KB	0,00	0,00	0,00	0,02
RF50ADN2KM	0,00	0,00	0,00	0,01
RF50ADN2KMB	0,00	0,00	0,00	0,02
RF50ADN3K	0,00	0,00	0,00	0,01
RF50ADN3KB	0,00	0,00	0,00	0,02
RF50ADN3KM	0,00	0,00	0,00	0,01
RF50ADN3KMB	0,00	0,00	0,00	0,02
<i>Gurobi</i>	0,00	0,00	0,00	0,01

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 16 – GAP das heurísticas e o MIPGap do Gurobi para instâncias com 12 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,06	0,07	0,00	0,33
GFO2	0,03	0,06	0,00	0,33
GFO3	0,01	0,01	0,00	0,11
GADN1K	8,05	6,40	0,00	27,18
GADN1KB	11,74	8,69	0,35	30,80
GADN1KM	11,24	8,96	0,32	31,40
GADN1KMB	11,74	8,69	0,35	30,80
GADN2K	0,01	0,02	0,00	0,13
GADN2KB	0,04	0,07	0,00	0,33
GADN2KM	0,00	0,01	0,00	0,01
GADN2KMB	0,04	0,07	0,00	0,33
GADN3K	0,01	0,01	0,00	0,06
GADN3KB	0,03	0,06	0,00	0,33
GADN3KM	0,01	0,01	0,00	0,09
GADN3KMB	0,02	0,04	0,00	0,33
RF10FO1	0,00	0,00	0,00	0,01
RF10FO2	0,00	0,01	0,00	0,01
RF10FO3	0,00	0,01	0,00	0,01
RF10ADN1K	0,00	0,01	0,00	0,01
RF10ADN1KB	0,00	0,01	0,00	0,01
RF10ADN1KM	0,00	0,01	0,00	0,01
RF10ADN1KMB	0,00	0,01	0,00	0,01
RF10ADN2K	0,00	0,01	0,00	0,01
RF10ADN2KB	0,00	0,01	0,00	0,01
RF10ADN2KM	0,00	0,01	0,00	0,01
RF10ADN2KMB	0,00	0,01	0,00	0,01
RF10ADN3K	0,00	0,01	0,00	0,01
RF10ADN3KB	0,00	0,01	0,00	0,01
RF10ADN3KM	0,00	0,01	0,00	0,01
RF10ADN3KMB	0,00	0,01	0,00	0,01
RF50FO1	0,00	0,00	0,00	0,01
RF50FO2	0,00	0,01	0,00	0,01
RF50FO3	0,00	0,01	0,00	0,01
RF50ADN1K	0,00	0,01	0,00	0,01
RF50ADN1KB	0,00	0,01	0,00	0,01
RF50ADN1KM	0,00	0,01	0,00	0,01
RF50ADN1KMB	0,00	0,01	0,00	0,01
RF50ADN2K	0,00	0,01	0,00	0,01
RF50ADN2KB	0,00	0,01	0,00	0,01
RF50ADN2KM	0,00	0,01	0,00	0,01
RF50ADN2KMB	0,00	0,01	0,00	0,01
RF50ADN3K	0,00	0,01	0,00	0,01
RF50ADN3KB	0,00	0,01	0,00	0,01
RF50ADN3KM	0,00	0,01	0,00	0,01
RF50ADN3KMB	0,00	0,01	0,00	0,01
<i>Gurobi</i>	0,00	0,00	0,00	0,01

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 17 – GAP das heurísticas e o MIPGap do *Gurobi* para instâncias com 25 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,06	0,07	0,00	0,33
GFO2	0,03	0,06	0,00	0,33
GFO3	0,01	0,01	0,00	0,11
GADN1K	8,05	6,40	0,00	27,18
GADN1KB	11,74	8,69	0,35	30,80
GADN1KM	11,24	8,96	0,32	31,40
GADN1KMB	11,74	8,69	0,35	30,80
GADN2K	0,01	0,02	0,00	0,13
GADN2KB	0,04	0,07	0,00	0,33
GADN2KM	0,00	0,01	0,00	0,01
GADN2KMB	0,04	0,07	0,00	0,33
GADN3K	0,01	0,01	0,00	0,06
GADN3KB	0,03	0,06	0,00	0,33
GADN3KM	0,01	0,01	0,00	0,09
GADN3KMB	0,02	0,04	0,00	0,33
RF10FO1	0,00	0,00	0,00	0,01
RF10FO2	0,00	0,01	0,00	0,01
RF10FO3	0,00	0,01	0,00	0,01
RF10ADN1K	0,00	0,01	0,00	0,01
RF10ADN1KB	0,00	0,01	0,00	0,01
RF10ADN1KM	0,00	0,01	0,00	0,01
RF10ADN1KMB	0,00	0,01	0,00	0,01
RF10ADN2K	0,00	0,01	0,00	0,01
RF10ADN2KB	0,00	0,01	0,00	0,01
RF10ADN2KM	0,00	0,01	0,00	0,01
RF10ADN2KMB	0,00	0,01	0,00	0,01
RF10ADN3K	0,00	0,01	0,00	0,01
RF10ADN3KB	0,00	0,01	0,00	0,01
RF10ADN3KM	0,00	0,01	0,00	0,01
RF10ADN3KMB	0,00	0,01	0,00	0,01
RF50FO1	0,00	0,00	0,00	0,01
RF50FO2	0,00	0,01	0,00	0,01
RF50FO3	0,00	0,01	0,00	0,01
RF50ADN1K	0,00	0,01	0,00	0,01
RF50ADN1KB	0,00	0,01	0,00	0,01
RF50ADN1KM	0,00	0,01	0,00	0,01
RF50ADN1KMB	0,00	0,01	0,00	0,01
RF50ADN2K	0,00	0,01	0,00	0,01
RF50ADN2KB	0,00	0,01	0,00	0,01
RF50ADN2KM	0,00	0,01	0,00	0,01
RF50ADN2KMB	0,00	0,01	0,00	0,01
RF50ADN3K	0,00	0,01	0,00	0,01
RF50ADN3KB	0,00	0,01	0,00	0,01
RF50ADN3KM	0,00	0,01	0,00	0,01
RF50ADN3KMB	0,00	0,01	0,00	0,01
Gurobi	0,00	0,01	0,00	0,01

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 18 – GAP das heurísticas e o MIPGap do Gurobi para instâncias com 50 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,05	0,04	0,00	0,26
GFO2	0,03	0,03	0,00	0,15
GFO3	0,02	0,02	0,00	0,12
GADN1K	7,12	6,98	0,00	30,40
GADN1KB	10,47	9,28	0,51	31,09
GADN1KM	10,23	9,89	0,31	30,99
GADN1KMB	10,47	9,28	0,51	31,09
GADN2K	0,01	0,01	0,00	0,07
GADN2KB	0,03	0,03	0,00	0,15
GADN2KM	0,01	0,01	0,00	0,04
GADN2KMB	0,03	0,03	0,00	0,15
GADN3K	0,01	0,01	0,00	0,05
GADN3KB	0,03	0,03	0,00	0,15
GADN3KM	0,03	0,03	0,00	0,13
GADN3KMB	0,03	0,03	0,00	0,16
RF10FO1	0,01	0,03	0,00	0,25
RF10FO2	0,02	0,04	0,00	0,28
RF10FO3	0,02	0,03	0,00	0,25
RF10ADN1K	0,01	0,00	0,00	0,02
RF10ADN1KB	0,01	0,00	0,00	0,02
RF10ADN1KM	0,01	0,00	0,00	0,02
RF10ADN1KMB	0,01	0,00	0,00	0,02
RF10ADN2K	0,01	0,00	0,00	0,02
RF10ADN2KB	0,01	0,00	0,00	0,02
RF10ADN2KM	0,01	0,00	0,00	0,02
RF10ADN2KMB	0,01	0,00	0,00	0,02
RF10ADN3K	0,01	0,00	0,00	0,02
RF10ADN3KB	0,01	0,00	0,00	0,02
RF10ADN3KM	0,01	0,00	0,00	0,02
RF10ADN3KMB	0,01	0,00	0,00	0,02
RF50FO1	0,01	0,00	0,00	0,02
RF50FO2	0,01	0,00	0,00	0,02
RF50FO3	0,01	0,00	0,00	0,02
RF50ADN1K	0,01	0,00	0,00	0,02
RF50ADN1KB	0,01	0,00	0,00	0,02
RF50ADN1KM	0,01	0,00	0,00	0,02
RF50ADN1KMB	0,01	0,00	0,00	0,02
RF50ADN2K	0,01	0,00	0,00	0,02
RF50ADN2KB	0,01	0,00	0,00	0,02
RF50ADN2KM	0,01	0,00	0,00	0,02
RF50ADN2KMB	0,01	0,00	0,00	0,02
RF50ADN3K	0,01	0,00	0,00	0,02
RF50ADN3KB	0,01	0,00	0,00	0,02
RF50ADN3KM	0,01	0,00	0,00	0,02
RF50ADN3KMB	0,01	0,00	0,00	0,02
Gurobi	0,01	0,00	0,00	0,01

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 19 – GAP das heurísticas e o MIPGap do *Gurobi* para instâncias com 70 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,03	0,01	0,01	0,08
GFO2	0,02	0,01	0,01	0,06
GFO3	0,01	0,01	0,00	0,04
GADN1K	6,52	4,61	0,01	15,37
GADN1KB	10,07	6,42	3,24	17,98
GADN1KM	9,94	7,06	2,36	18,75
GADN1KMB	10,07	6,42	3,24	17,98
GADN2K	0,05	0,05	0,01	0,26
GADN2KB	0,07	0,07	0,01	0,32
GADN2KM	0,09	0,12	0,01	0,61
GADN2KMB	0,08	0,11	0,01	0,65
GADN3K	0,02	0,02	0,00	0,08
GADN3KB	0,03	0,03	0,01	0,12
GADN3KM	0,02	0,02	0,01	0,12
GADN3KMB	0,03	0,02	0,01	0,11
RF10FO1	0,54	0,67	0,00	2,44
RF10FO2	0,48	0,60	0,00	2,44
RF10FO3	0,54	0,62	0,01	2,44
RF10ADN1K	0,04	0,04	0,00	0,17
RF10ADN1KB	0,04	0,05	0,01	0,25
RF10ADN1KM	0,13	0,17	0,00	0,80
RF10ADN1KMB	0,05	0,04	0,00	0,19
RF10ADN2K	0,04	0,04	0,00	0,16
RF10ADN2KB	0,05	0,05	0,01	0,18
RF10ADN2KM	0,06	0,06	0,00	0,37
RF10ADN2KMB	0,05	0,05	0,01	0,18
RF10ADN3K	0,03	0,03	0,00	0,13
RF10ADN3KB	0,03	0,03	0,01	0,16
RF10ADN3KM	0,04	0,04	0,00	0,22
RF10ADN3KMB	0,04	0,04	0,01	0,29
RF50FO1	0,03	0,03	0,01	0,14
RF50FO2	0,03	0,03	0,01	0,14
RF50FO3	0,03	0,03	0,01	0,14
RF50ADN1K	0,04	0,04	0,01	0,18
RF50ADN1KB	0,04	0,04	0,01	0,16
RF50ADN1KM	0,04	0,04	0,01	0,18
RF50ADN1KMB	0,04	0,04	0,01	0,18
RF50ADN2K	0,03	0,03	0,01	0,12
RF50ADN2KB	0,04	0,04	0,01	0,16
RF50ADN2KM	0,05	0,04	0,01	0,16
RF50ADN2KMB	0,04	0,04	0,01	0,16
RF50ADN3K	0,02	0,02	0,01	0,11
RF50ADN3KB	0,02	0,02	0,01	0,11
RF50ADN3KM	0,02	0,02	0,01	0,09
RF50ADN3KMB	0,02	0,02	0,01	0,10
Gurobi	0,01	0,00	0,00	0,02

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 20 – GAP das heurísticas e o MIPGap do Gurobi para instâncias com 80 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,03	0,02	0,01	0,11
GFO2	0,02	0,01	0,01	0,07
GFO3	0,02	0,01	0,01	0,08
GADN1K	6,79	4,71	1,51	15,48
GADN1KB	10,22	6,48	3,39	18,93
GADN1KM	10,00	7,07	2,31	19,80
GADN1KMB	10,22	6,48	3,39	18,93
GADN2K	0,07	0,10	0,01	0,73
GADN2KB	0,09	0,11	0,01	0,53
GADN2KM	0,13	0,19	0,00	0,84
GADN2KMB	0,09	0,11	0,01	0,50
GADN3K	0,03	0,02	0,01	0,10
GADN3KB	0,04	0,03	0,01	0,11
GADN3KM	0,04	0,05	0,01	0,26
GADN3KMB	0,03	0,02	0,01	0,12
RF10FO1	0,43	0,55	0,01	2,31
RF10FO2	0,41	0,66	0,01	4,00
RF10FO3	0,51	0,61	0,01	2,27
RF10ADN1K	0,04	0,04	0,01	0,21
RF10ADN1KB	0,05	0,06	0,01	0,32
RF10ADN1KM	0,18	0,22	0,01	0,84
RF10ADN1KMB	0,06	0,06	0,01	0,24
RF10ADN2K	0,05	0,04	0,01	0,15
RF10ADN2KB	0,06	0,07	0,01	0,32
RF10ADN2KM	0,06	0,07	0,01	0,32
RF10ADN2KMB	0,06	0,08	0,01	0,53
RF10ADN3K	0,03	0,03	0,01	0,11
RF10ADN3KB	0,04	0,04	0,01	0,24
RF10ADN3KM	0,04	0,04	0,01	0,25
RF10ADN3KMB	0,04	0,04	0,01	0,24
RF50FO1	0,04	0,03	0,01	0,17
RF50FO2	0,03	0,04	0,01	0,18
RF50FO3	0,04	0,03	0,01	0,17
RF50ADN1K	0,05	0,06	0,01	0,28
RF50ADN1KB	0,05	0,06	0,01	0,28
RF50ADN1KM	0,05	0,06	0,00	0,28
RF50ADN1KMB	0,05	0,06	0,01	0,28
RF50ADN2K	0,04	0,05	0,01	0,24
RF50ADN2KB	0,05	0,05	0,01	0,30
RF50ADN2KM	0,05	0,05	0,00	0,22
RF50ADN2KMB	0,05	0,05	0,01	0,30
RF50ADN3K	0,03	0,03	0,01	0,10
RF50ADN3KB	0,03	0,03	0,01	0,24
RF50ADN3KM	0,03	0,04	0,01	0,24
RF50ADN3KMB	0,03	0,03	0,01	0,24
Gurobi	0,01	0,01	0,00	0,07

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 21 – GAP das heurísticas e o MIPGap do *Gurobi* para instâncias com 90 itens do MPCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO1	0,04	0,03	0,01	0,15
GFO2	0,03	0,02	0,01	0,12
GFO3	0,02	0,02	0,01	0,12
GADN1K	7,08	5,04	0,01	17,11
GADN1KB	10,26	6,67	0,92	18,65
GADN1KM	10,14	7,28	0,56	19,52
GADN1KMB	10,26	6,67	0,92	18,65
GADN2K	0,05	0,05	0,01	0,24
GADN2KB	0,07	0,07	0,01	0,43
GADN2KM	0,08	0,12	0,01	0,65
GADN2KMB	0,07	0,06	0,01	0,27
GADN3K	0,03	0,03	0,01	0,15
GADN3KB	0,03	0,02	0,01	0,12
GADN3KM	0,03	0,04	0,01	0,24
GADN3KMB	0,04	0,03	0,01	0,14
RF10FO1	0,43	0,67	0,01	3,49
RF10FO2	0,34	0,48	0,01	2,20
RF10FO3	0,55	0,68	0,01	2,45
RF10ADN1K	0,04	0,04	0,01	0,22
RF10ADN1KB	0,05	0,05	0,01	0,22
RF10ADN1KM	0,22	0,25	0,01	0,92
RF10ADN1KMB	0,05	0,06	0,01	0,40
RF10ADN2K	0,04	0,04	0,00	0,19
RF10ADN2KB	0,06	0,06	0,00	0,29
RF10ADN2KM	0,05	0,06	0,00	0,24
RF10ADN2KMB	0,05	0,06	0,01	0,40
RF10ADN3K	0,03	0,02	0,00	0,14
RF10ADN3KB	0,04	0,04	0,00	0,21
RF10ADN3KM	0,04	0,03	0,00	0,15
RF10ADN3KMB	0,04	0,03	0,00	0,17
RF50FO1	0,04	0,04	0,01	0,20
RF50FO2	0,04	0,04	0,01	0,20
RF50FO3	0,04	0,04	0,01	0,20
RF50ADN1K	0,05	0,05	0,01	0,21
RF50ADN1KB	0,05	0,05	0,01	0,22
RF50ADN1KM	0,05	0,05	0,01	0,22
RF50ADN1KMB	0,05	0,05	0,01	0,22
RF50ADN2K	0,04	0,04	0,01	0,21
RF50ADN2KB	0,05	0,05	0,01	0,21
RF50ADN2KM	0,05	0,05	0,01	0,26
RF50ADN2KMB	0,05	0,05	0,01	0,21
RF50ADN3K	0,03	0,03	0,01	0,12
RF50ADN3KB	0,03	0,03	0,01	0,15
RF50ADN3KM	0,03	0,03	0,01	0,19
RF50ADN3KMB	0,03	0,03	0,01	0,13
Gurobi	0,01	0,01	0,00	0,13

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 22 – GAPM das heurísticas para instâncias com 6 itens do MPCLSP.

Heurísticas	$GAPM_{med} (%)$	$GAPM_{dp} (%)$	$GAPM_{min} (%)$	$GAPM_{max} (%)$
GFO1	56,83	11,16	22,57	71,22
GFO2	56,83	11,17	22,57	71,22
GFO3	56,85	11,16	22,57	71,22
GADN1K	53,26	11,98	22,28	71,00
GADN1KB	51,27	12,88	21,63	69,78
GADN1KM	51,57	12,80	21,63	69,97
GADN1KMB	51,27	12,88	21,63	69,78
GADN2K	56,85	11,15	22,57	71,22
GADN2KB	56,84	11,16	22,57	71,22
GADN2KM	56,85	11,15	22,57	71,22
GADN2KMB	56,84	11,16	22,57	71,22
GADN3K	56,85	11,15	22,57	71,22
GADN3KB	56,84	11,16	22,57	71,22
GADN3KM	56,85	11,16	22,57	71,22
GADN3KMB	56,85	11,15	22,57	71,22
RF10FO1	0,00	0,00	0,00	0,00
RF10FO2	0,00	0,00	0,00	0,00
RF10FO3	0,00	0,00	0,00	0,01
RF10ADN1K	0,00	0,00	0,00	0,01
RF10ADN1KB	0,00	0,00	0,00	0,00
RF10ADN1KM	0,00	0,00	0,00	0,01
RF10ADN1KMB	0,00	0,00	0,00	0,00
RF10ADN2K	0,00	0,00	0,00	0,01
RF10ADN2KB	0,00	0,00	0,00	0,00
RF10ADN2KM	0,00	0,00	0,00	0,01
RF10ADN2KMB	0,00	0,00	0,00	0,00
RF10ADN3K	0,00	0,00	0,00	0,00
RF10ADN3KB	0,00	0,00	0,00	0,00
RF10ADN3KM	0,00	0,00	0,00	0,01
RF10ADN3KMB	0,00	0,00	0,00	0,00
RF50FO1	0,00	0,00	0,00	0,00
RF50FO2	0,00	0,00	0,00	0,00
RF50FO3	0,00	0,00	0,00	0,01
RF50ADN1K	0,00	0,00	0,00	0,00
RF50ADN1KB	0,00	0,00	0,00	0,00
RF50ADN1KM	0,00	0,00	0,00	0,01
RF50ADN1KMB	0,00	0,00	0,00	0,00
RF50ADN2K	0,00	0,00	0,00	0,01
RF50ADN2KB	0,00	0,00	0,00	0,00
RF50ADN2KM	0,00	0,00	0,00	0,01
RF50ADN2KMB	0,00	0,00	0,00	0,00
RF50ADN3K	0,00	0,00	0,00	0,01
RF50ADN3KB	0,00	0,00	0,00	0,00
RF50ADN3KM	0,00	0,00	0,00	0,01
RF50ADN3KMB	0,00	0,00	0,00	0,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o $GAPM_{dp}$.

Tabela 23 – GAPM das heurísticas para instâncias com 12 itens do MPCLSP.

Heurísticas	$GAPM_{med}$ (%)	$GAPM_{dp}$ (%)	$GAPM_{min}$ (%)	$GAPM_{max}$ (%)
GFO1	55,66	12,64	23,01	71,67
GFO2	55,67	12,63	23,01	71,67
GFO3	55,68	12,62	23,01	71,67
GADN1K	51,91	12,51	22,35	69,68
GADN1KB	49,71	13,16	21,99	67,39
GADN1KM	49,99	13,07	22,06	67,90
GADN1KMB	49,71	13,16	21,99	67,39
GADN2K	55,68	12,62	23,02	71,67
GADN2KB	55,67	12,63	23,00	71,67
GADN2KM	55,68	12,62	23,02	71,67
GADN2KMB	55,67	12,63	23,00	71,67
GADN3K	55,68	12,62	23,02	71,67
GADN3KB	55,67	12,63	23,01	71,67
GADN3KM	55,68	12,62	23,01	71,67
GADN3KMB	55,67	12,63	23,00	71,67
RF10FO1	0,00	0,00	0,00	0,01
RF10FO2	0,00	0,00	0,00	0,00
RF10FO3	0,00	0,00	0,00	0,00
RF10ADN1K	0,00	0,00	0,00	0,01
RF10ADN1KB	0,00	0,00	0,00	0,00
RF10ADN1KM	0,00	0,00	0,00	0,00
RF10ADN1KMB	0,00	0,00	0,00	0,00
RF10ADN2K	0,00	0,00	0,00	0,01
RF10ADN2KB	0,00	0,00	0,00	0,00
RF10ADN2KM	0,00	0,00	0,00	0,00
RF10ADN2KMB	0,00	0,00	0,00	0,00
RF10ADN3K	0,00	0,00	0,00	0,01
RF10ADN3KB	0,00	0,00	0,00	0,00
RF10ADN3KM	0,00	0,00	0,00	0,00
RF10ADN3KMB	0,00	0,00	0,00	0,00
RF50FO1	0,00	0,00	0,00	0,01
RF50FO2	0,00	0,00	0,00	0,00
RF50FO3	0,00	0,00	0,00	0,00
RF50ADN1K	0,00	0,00	0,00	0,01
RF50ADN1KB	0,00	0,00	0,00	0,00
RF50ADN1KM	0,00	0,00	0,00	0,00
RF50ADN1KMB	0,00	0,00	0,00	0,00
RF50ADN2K	0,00	0,00	0,00	0,01
RF50ADN2KB	0,00	0,00	0,00	0,00
RF50ADN2KM	0,00	0,00	0,00	0,00
RF50ADN2KMB	0,00	0,00	0,00	0,00
RF50ADN3K	0,00	0,00	0,00	0,01
RF50ADN3KB	0,00	0,00	0,00	0,00
RF50ADN3KM	0,00	0,00	0,00	0,00
RF50ADN3KMB	0,00	0,00	0,00	0,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o $GAPM_{dp}$.

Tabela 24 – GAPM das heurísticas para instâncias com 25 itens do MPCLSP.

Heurísticas	$GAPM_{med} (%)$	$GAPM_{dp} (%)$	$GAPM_{min} (%)$	$GAPM_{max} (%)$
GFO1	52,70	13,88	20,49	71,52
GFO2	52,71	13,87	20,49	71,52
GFO3	52,72	13,87	20,49	71,52
GADN1K	49,43	13,10	19,80	70,21
GADN1KB	47,77	12,92	19,67	68,18
GADN1KM	48,03	12,83	19,79	68,92
GADN1KMB	47,77	12,92	19,67	68,18
GADN2K	52,72	13,87	20,49	71,52
GADN2KB	52,71	13,87	20,47	71,52
GADN2KM	52,72	13,87	20,49	71,52
GADN2KMB	52,71	13,87	20,47	71,52
GADN3K	52,72	13,87	20,49	71,52
GADN3KB	52,71	13,87	20,49	71,52
GADN3KM	52,72	13,87	20,49	71,52
GADN3KMB	52,71	13,87	20,45	71,52
RF10FO1	0,00	0,00	0,00	0,01
RF10FO2	0,00	0,00	0,00	0,01
RF10FO3	0,00	0,00	0,00	0,01
RF10ADN1K	0,00	0,00	0,00	0,01
RF10ADN1KB	0,00	0,00	0,00	0,01
RF10ADN1KM	0,00	0,00	0,00	0,01
RF10ADN1KMB	0,00	0,00	0,00	0,01
RF10ADN2K	0,00	0,00	0,00	0,01
RF10ADN2KB	0,00	0,00	0,00	0,01
RF10ADN2KM	0,00	0,00	0,00	0,01
RF10ADN2KMB	0,00	0,00	0,00	0,01
RF10ADN3K	0,00	0,00	0,00	0,01
RF10ADN3KB	0,00	0,00	0,00	0,01
RF10ADN3KM	0,00	0,00	0,00	0,05
RF10ADN3KMB	0,00	0,00	0,00	0,01
RF50FO1	0,00	0,00	0,00	0,01
RF50FO2	0,00	0,00	0,00	0,01
RF50FO3	0,00	0,00	0,00	0,01
RF50ADN1K	0,00	0,00	0,00	0,01
RF50ADN1KB	0,00	0,00	0,00	0,01
RF50ADN1KM	0,00	0,00	0,00	0,01
RF50ADN1KMB	0,00	0,00	0,00	0,01
RF50ADN2K	0,00	0,00	0,00	0,01
RF50ADN2KB	0,00	0,00	0,00	0,01
RF50ADN2KM	0,00	0,00	0,00	0,01
RF50ADN2KMB	0,00	0,00	0,00	0,01
RF50ADN3K	0,00	0,00	0,00	0,01
RF50ADN3KB	0,00	0,00	0,00	0,01
RF50ADN3KM	0,00	0,00	0,00	0,01
RF50ADN3KMB	0,00	0,00	0,00	0,01

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o $GAPM_{dp}$.

Tabela 25 – GAPM das heurísticas para instâncias com 50 itens do MPCLSP.

Heurísticas	$GAPM_{med}$ (%)	$GAPM_{dp}$ (%)	$GAPM_{min}$ (%)	$GAPM_{max}$ (%)
GFO1	52,46	14,02	21,19	68,69
GFO2	52,47	14,02	21,19	68,69
GFO3	52,48	14,01	21,21	68,70
GADN1K	49,09	13,22	21,06	68,61
GADN1KB	47,14	13,17	20,66	68,51
GADN1KM	47,27	13,02	20,74	68,61
GADN1KMB	47,14	13,17	20,66	68,51
GADN2K	52,48	14,01	21,22	68,70
GADN2KB	52,47	14,01	21,17	68,69
GADN2KM	52,48	14,01	21,22	68,70
GADN2KMB	52,47	14,01	21,17	68,69
GADN3K	52,48	14,01	21,22	68,70
GADN3KB	52,47	14,02	21,20	68,69
GADN3KM	52,48	14,02	21,20	68,70
GADN3KMB	52,48	14,01	21,19	68,69
RF10FO1	0,00	0,03	0,00	0,27
RF10FO2	0,00	0,00	0,00	0,01
RF10FO3	0,00	0,00	0,00	0,00
RF10ADN1K	0,00	0,02	0,00	0,23
RF10ADN1KB	0,00	0,00	0,00	0,03
RF10ADN1KM	0,01	0,03	0,00	0,23
RF10ADN1KMB	0,00	0,01	0,00	0,12
RF10ADN2K	0,01	0,03	0,00	0,23
RF10ADN2KB	0,01	0,03	0,00	0,27
RF10ADN2KM	0,01	0,04	0,00	0,27
RF10ADN2KMB	0,01	0,03	0,00	0,23
RF10ADN3K	0,01	0,04	0,00	0,27
RF10ADN3KB	0,00	0,01	0,00	0,08
RF10ADN3KM	0,01	0,04	0,00	0,27
RF10ADN3KMB	0,01	0,04	0,00	0,27
RF50FO1	0,00	0,00	0,00	0,00
RF50FO2	0,00	0,00	0,00	0,01
RF50FO3	0,00	0,00	0,00	0,00
RF50ADN1K	0,00	0,00	0,00	0,02
RF50ADN1KB	0,00	0,00	0,00	0,01
RF50ADN1KM	0,00	0,00	0,00	0,01
RF50ADN1KMB	0,00	0,00	0,00	0,01
RF50ADN2K	0,00	0,00	0,00	0,02
RF50ADN2KB	0,00	0,00	0,00	0,01
RF50ADN2KM	0,00	0,00	0,00	0,01
RF50ADN2KMB	0,00	0,00	0,00	0,01
RF50ADN3K	0,00	0,00	0,00	0,01
RF50ADN3KB	0,00	0,00	0,00	0,01
RF50ADN3KM	0,00	0,00	0,00	0,02
RF50ADN3KMB	0,00	0,00	0,00	0,02

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o $GAPM_{dp}$.

Tabela 26 – GAPM das heurísticas para instâncias com 70 itens do MPCLSP.

Heurísticas	GAPM _{med} (%)	GAPM _{dp} (%)	GAPM _{min} (%)	GAPM _{max} (%)
GFO1	84,27	2,17	81,38	86,89
GFO2	84,27	2,17	81,38	86,89
GFO3	84,27	2,16	81,38	86,89
GADN1K	83,13	2,48	78,85	86,66
GADN1KB	82,41	2,79	77,99	86,45
GADN1KM	82,42	2,85	77,76	86,53
GADN1KMB	82,41	2,79	77,99	86,45
GADN2K	84,26	2,16	81,38	86,89
GADN2KB	84,26	2,16	81,38	86,89
GADN2KM	84,26	2,16	81,38	86,84
GADN2KMB	84,26	2,16	81,38	86,89
GADN3K	84,27	2,16	81,38	86,89
GADN3KB	84,27	2,16	81,38	86,89
GADN3KM	84,27	2,16	81,38	86,89
GADN3KMB	84,27	2,16	81,38	86,89
RF10FO1	0,98	1,26	0,00	3,75
RF10FO2	0,90	1,16	0,00	3,83
RF10FO3	0,81	1,19	0,00	4,15
RF10ADN1K	0,35	0,41	0,00	2,68
RF10ADN1KB	0,35	0,48	0,00	2,32
RF10ADN1KM	1,28	1,00	0,02	3,19
RF10ADN1KMB	0,43	0,58	0,00	2,47
RF10ADN2K	0,99	0,87	0,05	2,81
RF10ADN2KB	1,01	0,90	0,03	2,69
RF10ADN2KM	0,97	0,93	0,00	3,74
RF10ADN2KMB	0,99	0,93	0,00	3,76
RF10ADN3K	1,20	0,94	0,11	3,80
RF10ADN3KB	0,59	0,73	0,00	2,68
RF10ADN3KM	1,16	0,95	0,09	3,80
RF10ADN3KMB	1,30	0,93	0,05	3,31
RF50FO1	0,00	0,02	0,00	0,12
RF50FO2	0,00	0,01	0,00	0,05
RF50FO3	0,00	0,01	0,00	0,05
RF50ADN1K	0,01	0,02	0,00	0,11
RF50ADN1KB	0,01	0,01	0,00	0,05
RF50ADN1KM	0,01	0,01	0,00	0,06
RF50ADN1KMB	0,01	0,02	0,00	0,06
RF50ADN2K	0,03	0,04	0,00	0,20
RF50ADN2KB	0,03	0,03	0,00	0,14
RF50ADN2KM	0,02	0,03	0,00	0,13
RF50ADN2KMB	0,02	0,03	0,00	0,12
RF50ADN3K	0,04	0,05	0,00	0,24
RF50ADN3KB	0,03	0,05	0,00	0,23
RF50ADN3KM	0,04	0,08	0,00	0,49
RF50ADN3KMB	0,03	0,05	0,00	0,29

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o GAPM_{dp}.

Tabela 27 – GAPM das heurísticas para instâncias com 80 itens do MPCLSP.

Heurísticas	$GAPM_{med}$ (%)	$GAPM_{dp}$ (%)	$GAPM_{min}$ (%)	$GAPM_{max}$ (%)
GFO1	84,16	2,18	81,27	86,77
GFO2	84,16	2,18	81,27	86,77
GFO3	84,16	2,18	81,27	86,77
GADN1K	82,97	2,48	78,44	86,49
GADN1KB	82,27	2,78	77,98	86,31
GADN1KM	82,30	2,84	77,58	86,46
GADN1KMB	82,27	2,78	77,98	86,31
GADN2K	84,16	2,18	81,27	86,77
GADN2KB	84,15	2,18	81,27	86,77
GADN2KM	84,15	2,17	81,27	86,77
GADN2KMB	84,15	2,18	81,27	86,77
GADN3K	84,16	2,18	81,27	86,77
GADN3KB	84,16	2,18	81,27	86,77
GADN3KM	84,16	2,18	81,27	86,77
GADN3KMB	84,16	2,18	81,27	86,77
RF10FO1	1,17	1,40	0,00	4,06
RF10FO2	1,05	1,32	0,00	3,98
RF10FO3	0,95	1,30	0,00	3,93
RF10ADN1K	0,63	0,78	0,00	2,65
RF10ADN1KB	0,74	0,89	0,01	2,68
RF10ADN1KM	1,27	1,07	0,00	3,65
RF10ADN1KMB	0,67	0,82	0,00	2,68
RF10ADN2K	1,25	0,92	0,05	3,02
RF10ADN2KB	1,10	0,98	0,01	3,79
RF10ADN2KM	1,24	0,97	0,07	3,82
RF10ADN2KMB	1,28	1,06	0,08	3,79
RF10ADN3K	1,49	1,11	0,13	3,93
RF10ADN3KB	0,89	0,92	0,02	2,68
RF10ADN3KM	1,43	1,13	0,06	3,98
RF10ADN3KMB	1,38	1,07	0,13	3,86
RF50FO1	0,00	0,01	0,00	0,04
RF50FO2	0,00	0,01	0,00	0,06
RF50FO3	0,00	0,00	0,00	0,01
RF50ADN1K	0,01	0,02	0,00	0,07
RF50ADN1KB	0,01	0,02	0,00	0,13
RF50ADN1KM	0,01	0,02	0,00	0,12
RF50ADN1KMB	0,01	0,03	0,00	0,16
RF50ADN2K	0,04	0,07	0,00	0,36
RF50ADN2KB	0,03	0,05	0,00	0,20
RF50ADN2KM	0,04	0,06	0,00	0,30
RF50ADN2KMB	0,04	0,06	0,00	0,23
RF50ADN3K	0,06	0,09	0,00	0,46
RF50ADN3KB	0,04	0,06	0,00	0,23
RF50ADN3KM	0,04	0,06	0,00	0,22
RF50ADN3KMB	0,05	0,08	0,00	0,45

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o $GAPM_{dp}$.

Tabela 28 – GAPM das heurísticas para instâncias com 90 itens do MPCLSP.

Heurísticas	$GAPM_{med} (%)$	$GAPM_{dp} (%)$	$GAPM_{min} (%)$	$GAPM_{max} (%)$
GFO1	84,01	2,36	74,78	86,48
GFO2	84,01	2,36	74,78	86,48
GFO3	84,01	2,36	74,79	86,48
GADN1K	82,76	2,65	74,66	86,03
GADN1KB	82,10	2,91	74,56	85,93
GADN1KM	82,10	2,96	74,65	86,00
GADN1KMB	82,10	2,91	74,56	85,93
GADN2K	84,01	2,36	74,79	86,48
GADN2KB	84,01	2,36	74,79	86,47
GADN2KM	84,01	2,36	74,79	86,48
GADN2KMB	84,01	2,36	74,79	86,47
GADN3K	84,01	2,36	74,79	86,47
GADN3KB	84,01	2,36	74,78	86,47
GADN3KM	84,01	2,36	74,79	86,48
GADN3KMB	84,01	2,36	74,79	86,48
RF10FO1	1,02	1,31	0,00	3,99
RF10FO2	1,24	1,38	0,00	3,90
RF10FO3	1,03	1,65	0,00	9,25
RF10ADN1K	0,62	0,83	0,00	2,55
RF10ADN1KB	0,59	0,79	0,02	2,55
RF10ADN1KM	1,35	1,07	0,00	3,16
RF10ADN1KMB	0,78	0,92	0,02	2,55
RF10ADN2K	1,17	1,03	0,10	3,85
RF10ADN2KB	1,11	0,98	0,08	2,94
RF10ADN2KM	1,09	0,99	0,05	3,11
RF10ADN2KMB	1,05	0,99	0,03	3,68
RF10ADN3K	2,41	3,26	0,12	15,25
RF10ADN3KB	0,94	0,92	0,02	2,55
RF10ADN3KM	1,66	1,42	0,14	8,56
RF10ADN3KMB	1,54	1,10	0,11	3,99
RF50FO1	0,00	0,01	0,00	0,05
RF50FO2	0,01	0,02	0,00	0,10
RF50FO3	0,00	0,00	0,00	0,00
RF50ADN1K	0,01	0,02	0,00	0,09
RF50ADN1KB	0,01	0,02	0,00	0,07
RF50ADN1KM	0,01	0,01	0,00	0,05
RF50ADN1KMB	0,01	0,01	0,00	0,07
RF50ADN2K	0,04	0,05	0,00	0,19
RF50ADN2KB	0,03	0,04	0,00	0,15
RF50ADN2KM	0,03	0,06	0,00	0,46
RF50ADN2KMB	0,03	0,04	0,00	0,25
RF50ADN3K	0,09	0,13	0,00	0,56
RF50ADN3KB	0,07	0,11	0,00	0,50
RF50ADN3KM	0,06	0,10	0,00	0,46
RF50ADN3KMB	0,08	0,13	0,00	0,55

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna, exceto para o $GAPM_{dp}$.

Tabela 29 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 6 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	0,58	0,38	0,09	1,53
GFO2	0,39	0,28	0,05	1,27
GFO3	0,32	0,24	0,03	1,36
GADN1K	0,30	0,20	0,05	0,85
GADN1KB	0,35	0,23	0,08	0,98
GADN1KM	0,32	0,23	0,06	1,00
GADN1KMB	0,33	0,23	0,06	0,97
GADN2K	2,77	2,26	0,25	10,42
GADN2KB	3,50	2,40	0,44	11,06
GADN2KM	3,19	2,35	0,23	11,30
GADN2KMB	3,47	2,69	0,22	11,75
GADN3K	0,42	0,30	0,06	1,33
GADN3KB	0,46	0,31	0,08	1,47
GADN3KM	0,42	0,29	0,06	1,38
GADN3KMB	0,44	0,32	0,06	1,30
RF10FO1	0,93	0,64	0,06	3,36
RF10FO2	0,72	0,51	0,05	2,39
RF10FO3	0,67	0,48	0,05	2,08
RF10ADN1K	0,44	0,34	0,07	1,55
RF10ADN1KB	0,50	0,33	0,08	1,42
RF10ADN1KM	0,45	0,31	0,08	1,32
RF10ADN1KMB	0,47	0,35	0,07	1,50
RF10ADN2K	0,65	0,50	0,06	2,53
RF10ADN2KB	0,69	0,55	0,09	3,23
RF10ADN2KM	0,47	0,34	0,08	1,67
RF10ADN2KMB	0,47	0,34	0,07	1,52
RF10ADN3K	0,32	0,24	0,04	1,18
RF10ADN3KB	0,37	0,28	0,06	1,55
RF10ADN3KM	0,34	0,25	0,05	1,16
RF10ADN3KMB	0,33	0,25	0,05	1,18
RF50FO1	0,34	0,22	0,06	0,89
RF50FO2	0,26	0,16	0,04	0,59
RF50FO3	0,24	0,15	0,03	0,53
RF50ADN1K	0,42	0,29	0,06	1,21
RF50ADN1KB	0,46	0,31	0,09	1,34
RF50ADN1KM	0,45	0,32	0,07	1,42
RF50ADN1KMB	0,44	0,31	0,07	1,23
RF50ADN2K	0,45	0,34	0,07	1,72
RF50ADN2KB	2,74	1,81	0,45	9,57
RF50ADN2KM	0,48	0,36	0,07	1,54
RF50ADN2KMB	0,48	0,36	0,07	1,59
RF50ADN3K	0,33	0,24	0,05	1,18
RF50ADN3KB	0,34	0,25	0,06	1,19
RF50ADN3KM	0,36	0,27	0,06	1,30
RF50ADN3KMB	0,36	0,29	0,05	1,71
Gurobi	0,12	0,15	0,01	1,06

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 30 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 12 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	1,41	0,94	0,18	3,19
GFO2	0,92	0,61	0,10	2,41
GFO3	0,74	0,54	0,08	2,49
GADN1K	0,71	0,59	0,11	3,16
GADN1KB	0,78	0,59	0,12	3,17
GADN1KM	0,77	0,61	0,12	3,32
GADN1KMB	0,76	0,57	0,11	2,65
GADN2K	5,46	4,07	0,48	16,60
GADN2KB	7,13	5,32	0,72	23,65
GADN2KM	6,55	4,89	0,71	24,09
GADN2KMB	7,16	5,33	0,43	22,16
GADN3K	0,97	0,70	0,13	3,41
GADN3KB	1,05	0,72	0,13	3,03
GADN3KM	1,01	0,74	0,13	3,04
GADN3KMB	1,03	0,72	0,12	3,55
RF10FO1	2,13	1,60	0,16	6,65
RF10FO2	1,49	1,05	0,11	4,72
RF10FO3	1,35	0,91	0,15	4,84
RF10ADN1K	0,99	0,76	0,15	3,67
RF10ADN1KB	1,10	0,80	0,17	3,74
RF10ADN1KM	1,07	0,79	0,15	4,03
RF10ADN1KMB	1,07	0,80	0,15	3,71
RF10ADN2K	1,59	1,27	0,21	6,03
RF10ADN2KB	1,67	1,42	0,18	8,91
RF10ADN2KM	1,12	0,85	0,15	3,95
RF10ADN2KMB	1,12	0,88	0,16	5,07
RF10ADN3K	0,79	0,58	0,10	2,61
RF10ADN3KB	0,83	0,59	0,13	2,44
RF10ADN3KM	0,86	0,63	0,10	3,06
RF10ADN3KMB	0,87	0,65	0,11	2,91
RF50FO1	0,81	0,57	0,14	2,13
RF50FO2	0,64	0,43	0,10	1,95
RF50FO3	0,57	0,38	0,09	1,45
RF50ADN1K	0,97	0,75	0,15	3,72
RF50ADN1KB	1,01	0,74	0,16	3,19
RF50ADN1KM	1,00	0,76	0,15	3,54
RF50ADN1KMB	1,01	0,75	0,16	3,55
RF50ADN2K	1,04	0,83	0,13	4,42
RF50ADN2KB	4,43	2,95	0,73	14,85
RF50ADN2KM	1,14	0,88	0,16	4,07
RF50ADN2KMB	1,14	0,85	0,16	4,19
RF50ADN3K	0,86	0,65	0,10	3,52
RF50ADN3KB	0,94	0,70	0,13	3,67
RF50ADN3KM	0,95	0,78	0,12	4,09
RF50ADN3KMB	0,95	0,74	0,10	3,72
Gurobi	0,35	0,52	0,02	2,82

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 31 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 25 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	3,50	2,73	0,52	15,93
GFO2	2,34	2,05	0,29	13,60
GFO3	1,91	1,83	0,18	13,93
GADN1K	2,00	2,48	0,19	19,80
GADN1KB	2,13	2,28	0,25	17,47
GADN1KM	2,18	2,46	0,23	18,63
GADN1KMB	2,12	2,28	0,22	16,84
GADN2K	14,71	15,52	1,13	80,97
GADN2KB	17,58	19,56	1,03	146,37
GADN2KM	17,16	20,84	1,12	163,93
GADN2KMB	17,74	19,94	0,80	152,99
GADN3K	2,90	3,01	0,22	20,93
GADN3KB	3,06	2,73	0,23	18,01
GADN3KM	2,69	2,46	0,25	17,11
GADN3KMB	2,98	2,99	0,21	22,37
RF10FO1	6,06	4,67	0,92	24,32
RF10FO2	4,51	3,82	0,70	22,31
RF10FO3	4,34	3,67	0,21	21,29
RF10ADN1K	2,67	2,67	0,24	19,86
RF10ADN1KB	2,63	2,51	0,29	18,15
RF10ADN1KM	3,95	4,21	0,42	32,05
RF10ADN1KMB	2,74	2,76	0,28	20,78
RF10ADN2K	5,29	5,60	0,29	31,90
RF10ADN2KB	4,57	4,48	0,32	28,53
RF10ADN2KM	3,87	3,83	0,30	23,60
RF10ADN2KMB	3,71	4,27	0,33	29,86
RF10ADN3K	5,25	4,35	0,29	23,05
RF10ADN3KB	2,20	2,04	0,21	14,45
RF10ADN3KM	5,79	6,02	0,53	50,44
RF10ADN3KMB	5,50	5,18	0,33	33,72
RF50FO1	1,94	1,41	0,26	7,47
RF50FO2	1,48	1,17	0,19	6,85
RF50FO3	1,38	1,09	0,16	6,67
RF50ADN1K	2,54	2,33	0,25	13,64
RF50ADN1KB	2,62	2,35	0,30	13,99
RF50ADN1KM	2,65	2,60	0,29	19,34
RF50ADN1KMB	2,63	2,56	0,29	18,90
RF50ADN2K	3,69	4,07	0,23	27,56
RF50ADN2KB	7,10	6,45	0,37	43,31
RF50ADN2KM	3,81	3,67	0,28	25,90
RF50ADN2KMB	3,79	3,97	0,31	24,85
RF50ADN3K	5,63	4,79	0,43	31,69
RF50ADN3KB	5,76	5,01	0,63	31,53
RF50ADN3KM	5,90	4,93	0,65	33,36
RF50ADN3KMB	5,82	4,90	0,70	31,50
<i>Gurobi</i>	0,95	1,17	0,03	5,38

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 32 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 50 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	8,99	8,93	0,82	58,65
GFO2	6,49	7,93	0,51	51,15
GFO3	6,53	10,46	0,40	85,04
GADN1K	5,23	7,92	0,47	55,43
GADN1KB	5,64	7,62	0,51	50,14
GADN1KM	5,27	6,55	0,52	43,82
GADN1KMB	5,23	6,50	0,48	39,32
GADN2K	44,82	66,19	2,11	409,14
GADN2KB	49,87	68,47	2,94	496,84
GADN2KM	52,14	74,13	2,23	512,90
GADN2KMB	50,00	68,14	4,04	492,06
GADN3K	10,28	16,49	0,60	133,62
GADN3KB	10,60	13,78	0,50	85,85
GADN3KM	9,24	11,50	0,76	64,11
GADN3KMB	9,99	13,38	0,93	75,39
RF10FO1	15,46	18,12	1,11	135,34
RF10FO2	10,93	7,97	0,75	28,66
RF10FO3	10,17	7,44	1,09	25,83
RF10ADN1K	8,08	10,89	0,59	64,51
RF10ADN1KB	7,30	9,76	0,72	55,89
RF10ADN1KM	17,02	22,59	0,83	117,08
RF10ADN1KMB	8,43	11,51	0,82	72,61
RF10ADN2K	18,98	25,22	1,50	169,70
RF10ADN2KB	19,19	27,79	1,18	228,61
RF10ADN2KM	18,18	28,22	0,84	237,48
RF10ADN2KMB	17,71	29,94	0,73	224,43
RF10ADN3K	20,21	27,37	1,43	150,31
RF10ADN3KB	6,73	9,61	0,50	59,68
RF10ADN3KM	21,86	40,64	1,09	370,22
RF10ADN3KMB	20,52	31,42	1,55	226,13
RF50FO1	4,82	6,10	0,59	51,61
RF50FO2	3,72	3,86	0,44	29,21
RF50FO3	3,45	3,59	0,41	26,62
RF50ADN1K	6,87	9,84	0,64	78,61
RF50ADN1KB	7,15	9,04	0,67	47,93
RF50ADN1KM	7,50	10,50	0,70	72,65
RF50ADN1KMB	7,44	10,79	0,73	65,87
RF50ADN2K	16,50	21,75	0,97	130,97
RF50ADN2KB	15,28	19,21	0,98	108,62
RF50ADN2KM	17,86	24,21	0,81	186,49
RF50ADN2KMB	16,80	23,49	0,75	147,86
RF50ADN3K	19,46	27,06	1,50	199,45
RF50ADN3KB	19,81	31,95	1,74	270,31
RF50ADN3KM	19,67	29,81	1,59	270,65
RF50ADN3KMB	19,49	30,44	1,36	253,82
<i>Gurobi</i>	2,33	3,59	0,10	26,53

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 33 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 70 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	484,71	268,82	101,07	900,00
GFO2	514,30	369,39	60,25	900,00
GFO3	497,29	388,06	43,06	900,00
GADN1K	180,89	93,15	47,18	345,74
GADN1KB	164,12	72,76	52,12	307,33
GADN1KM	164,70	69,19	51,03	294,30
GADN1KMB	161,82	70,17	48,11	295,16
GADN2K	677,87	203,11	245,71	900,00
GADN2KB	720,79	166,91	313,48	900,00
GADN2KM	736,89	149,67	342,48	900,00
GADN2KMB	724,22	169,55	314,29	900,00
GADN3K	411,94	237,73	51,93	872,12
GADN3KB	381,55	200,53	114,82	846,51
GADN3KM	343,70	200,26	75,45	753,28
GADN3KMB	352,61	196,74	67,49	694,95
RF10FO1	451,53	290,95	136,35	900,00
RF10FO2	318,45	240,87	77,70	900,00
RF10FO3	238,16	187,71	80,65	900,00
RF10ADN1K	209,63	90,03	71,88	403,88
RF10ADN1KB	189,30	58,01	77,69	267,12
RF10ADN1KM	304,38	79,45	137,65	460,05
RF10ADN1KMB	195,47	60,59	81,39	275,25
RF10ADN2K	569,77	222,80	207,05	892,47
RF10ADN2KB	576,50	208,03	223,47	900,00
RF10ADN2KM	569,18	215,14	153,83	900,00
RF10ADN2KMB	582,47	214,78	158,21	900,00
RF10ADN3K	508,77	220,68	118,51	885,68
RF10ADN3KB	365,47	209,23	75,84	832,23
RF10ADN3KM	487,86	194,43	138,12	869,89
RF10ADN3KMB	481,20	197,79	157,19	833,31
RF50FO1	245,89	198,38	59,32	900,00
RF50FO2	204,91	189,30	46,34	900,00
RF50FO3	167,40	121,71	42,59	900,00
RF50ADN1K	297,98	154,73	74,86	608,28
RF50ADN1KB	279,72	122,16	78,94	480,72
RF50ADN1KM	294,51	146,83	72,55	628,46
RF50ADN1KMB	297,95	149,21	77,15	663,51
RF50ADN2K	530,26	253,49	118,47	900,00
RF50ADN2KB	478,95	197,65	130,29	894,76
RF50ADN2KM	480,07	204,15	118,86	854,63
RF50ADN2KMB	495,99	213,69	134,05	897,97
RF50ADN3K	480,46	218,39	111,04	869,33
RF50ADN3KB	461,13	196,11	102,69	900,00
RF50ADN3KM	455,61	199,21	117,12	846,71
RF50ADN3KMB	453,48	191,23	117,17	839,38
Gurobi	302,85	269,12	42,84	900,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 34 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 80 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	512,38	270,49	141,58	900,00
GFO2	528,96	357,71	90,70	900,00
GFO3	511,16	382,17	56,05	900,00
GADN1K	199,31	81,99	54,67	322,12
GADN1KB	188,79	60,45	74,95	283,83
GADN1KM	272,28	60,19	140,13	384,16
GADN1KMB	273,21	68,35	136,69	389,67
GADN2K	703,62	167,22	266,95	900,00
GADN2KB	743,93	162,26	366,12	900,00
GADN2KM	780,72	134,76	321,45	900,00
GADN2KMB	745,43	159,85	381,16	900,00
GADN3K	410,99	230,62	71,12	847,83
GADN3KB	403,88	181,42	123,56	753,81
GADN3KM	336,32	168,72	93,29	692,63
GADN3KMB	347,52	188,25	85,72	701,61
RF10FO1	476,66	282,24	92,83	900,00
RF10FO2	370,77	262,10	91,17	900,00
RF10FO3	269,33	186,79	60,39	900,00
RF10ADN1K	218,04	76,62	97,42	371,57
RF10ADN1KB	208,59	55,80	95,21	305,46
RF10ADN1KM	339,55	88,76	104,22	522,11
RF10ADN1KMB	210,84	54,92	101,99	298,19
RF10ADN2K	601,29	218,04	146,52	900,00
RF10ADN2KB	611,24	186,29	237,14	897,62
RF10ADN2KM	628,19	184,49	251,95	900,00
RF10ADN2KMB	645,10	197,92	201,61	900,00
RF10ADN3K	533,40	204,60	163,81	900,00
RF10ADN3KB	377,30	184,59	110,59	715,79
RF10ADN3KM	493,93	173,62	211,59	900,00
RF10ADN3KMB	491,64	188,24	195,85	897,19
RF50FO1	241,44	115,18	80,82	709,76
RF50FO2	243,31	213,59	55,68	900,00
RF50FO3	180,80	103,53	52,68	900,00
RF50ADN1K	426,14	120,11	178,77	605,53
RF50ADN1KB	441,50	108,17	206,93	625,36
RF50ADN1KM	434,48	114,80	170,11	605,55
RF50ADN1KMB	437,69	110,56	205,73	610,93
RF50ADN2K	532,75	216,81	139,75	898,92
RF50ADN2KB	495,04	193,15	201,54	893,46
RF50ADN2KM	534,59	192,51	239,54	844,07
RF50ADN2KMB	546,89	200,99	189,56	900,00
RF50ADN3K	500,92	208,20	145,46	864,30
RF50ADN3KB	497,66	193,77	150,83	883,48
RF50ADN3KM	470,26	182,24	165,80	848,97
RF50ADN3KMB	499,69	188,28	163,20	864,71
<i>Gurobi</i>	320,24	280,44	50,91	900,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 35 – Tempo gasto pelo *Gurobi* e pelas heurísticas para instâncias com 90 itens do MPCLSP.

Heurísticas	Tempo _{med} (%)	Tempo _{dp} (%)	Tempo _{min} (%)	Tempo _{max} (%)
GFO1	620,15	258,09	164,37	900,00
GFO2	571,93	334,11	118,12	900,00
GFO3	547,15	357,11	94,31	900,00
GADN1K	201,32	79,46	82,97	397,97
GADN1KB	196,79	60,56	80,52	298,49
GADN1KM	298,54	67,14	182,28	418,68
GADN1KMB	297,72	75,21	160,34	450,21
GADN2K	691,77	191,50	210,72	900,00
GADN2KB	748,28	164,66	402,31	900,00
GADN2KM	749,92	147,50	375,58	900,00
GADN2KMB	752,15	159,00	394,15	900,00
GADN3K	429,38	230,15	92,70	861,46
GADN3KB	437,35	172,35	139,32	814,66
GADN3KM	375,53	175,26	124,06	807,32
GADN3KMB	394,27	212,96	99,49	887,86
RF10FO1	492,10	284,95	97,23	900,00
RF10FO2	413,91	255,49	96,55	900,00
RF10FO3	278,99	206,54	71,31	900,00
RF10ADN1K	230,92	82,99	85,35	413,76
RF10ADN1KB	209,33	55,91	101,16	303,45
RF10ADN1KM	345,15	87,78	139,18	499,76
RF10ADN1KMB	222,37	54,55	114,40	316,37
RF10ADN2K	593,92	211,46	215,40	900,00
RF10ADN2KB	611,83	197,97	230,74	900,00
RF10ADN2KM	613,40	192,54	193,88	900,00
RF10ADN2KMB	618,19	193,77	231,09	900,00
RF10ADN3K	594,85	188,82	215,34	900,00
RF10ADN3KB	415,72	200,52	137,93	825,53
RF10ADN3KM	570,89	167,62	192,38	899,89
RF10ADN3KMB	562,40	176,94	215,24	900,00
RF50FO1	339,64	178,65	117,20	900,00
RF50FO2	318,43	230,45	100,67	900,00
RF50FO3	214,41	66,08	92,97	505,53
RF50ADN1K	458,66	103,95	216,08	648,78
RF50ADN1KB	472,91	97,64	257,11	659,19
RF50ADN1KM	467,66	98,08	260,76	645,15
RF50ADN1KMB	475,02	98,21	263,47	643,23
RF50ADN2K	557,00	221,80	160,06	900,00
RF50ADN2KB	529,37	179,34	203,72	886,08
RF50ADN2KM	554,14	189,95	210,48	889,35
RF50ADN2KMB	539,00	184,92	210,76	891,10
RF50ADN3K	572,48	201,18	181,36	900,00
RF50ADN3KB	577,83	169,10	197,19	900,00
RF50ADN3KM	534,58	150,83	204,88	852,23
RF50ADN3KMB	578,69	167,92	222,93	885,50
<i>Gurobi</i>	348,28	284,96	53,54	900,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 36 – Número de iterações das heurísticas para instâncias com 6 itens do MPCLSP.

Heurísticas	$nIter_{med}$ (%)	$nIter_{dp}$ (%)	$nIter_{min}$ (%)	$nIter_{max}$ (%)
GFO1	2,79	0,72	2,00	5,00
GFO2	2,66	0,69	2,00	5,00
GFO3	2,54	0,56	2,00	4,00
GADN2K	7,47	1,52	4,00	12,00
GADN2KB	8,55	1,26	5,00	13,00
GADN2KM	8,28	1,22	5,00	11,00
GADN2KMB	8,55	1,26	5,00	13,00
GADN3K	2,58	0,71	2,00	5,00
GADN3KB	2,73	0,82	2,00	6,00
GADN3KM	2,62	0,55	2,00	4,00
GADN3KMB	2,57	0,59	2,00	4,00
RF10FO1	1,09	0,29	1,00	2,00
RF10FO2	1,08	0,26	1,00	2,00
RF10FO3	1,09	0,29	1,00	2,00
RF10ADN2K	2,17	0,46	2,00	4,00
RF10ADN2KB	2,10	0,30	2,00	3,00
RF10ADN2KM	2,14	0,42	2,00	4,00
RF10ADN2KMB	2,10	0,30	2,00	3,00
RF10ADN3K	1,12	0,32	1,00	2,00
RF10ADN3KB	1,10	0,30	1,00	2,00
RF10ADN3KM	1,12	0,32	1,00	2,00
RF10ADN3KMB	1,10	0,30	1,00	2,00
RF50FO1	1,09	0,29	1,00	2,00
RF50FO2	1,08	0,26	1,00	2,00
RF50FO3	1,09	0,29	1,00	2,00
RF50ADN2K	2,14	0,40	2,00	4,00
RF50ADN2KB	2,10	0,30	2,00	3,00
RF50ADN2KM	2,14	0,42	2,00	4,00
RF50ADN2KMB	2,10	0,30	2,00	3,00
RF50ADN3K	1,11	0,31	1,00	2,00
RF50ADN3KB	1,10	0,30	1,00	2,00
RF50ADN3KM	1,12	0,32	1,00	2,00
RF50ADN3KMB	1,10	0,30	1,00	2,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 37 – Número de iterações das heurísticas para instâncias com 12 itens do MPCLSP.

Heurísticas	$nIter_{med} (\%)$	$nIter_{dp} (\%)$	$nIter_{min} (\%)$	$nIter_{max} (\%)$
GFO1	3,11	0,68	2,00	5,00
GFO2	2,96	0,54	2,00	4,00
GFO3	2,74	0,56	2,00	5,00
GADN2K	7,98	1,71	4,00	12,00
GADN2KB	9,03	1,25	5,00	12,00
GADN2KM	8,62	1,26	5,00	12,00
GADN2KMB	9,03	1,25	5,00	12,00
GADN3K	2,78	0,69	2,00	4,00
GADN3KB	2,98	0,65	2,00	5,00
GADN3KM	2,75	0,58	2,00	5,00
GADN3KMB	2,71	0,57	2,00	5,00
RF10FO1	1,27	0,45	1,00	2,00
RF10FO2	1,28	0,47	1,00	3,00
RF10FO3	1,28	0,47	1,00	3,00
RF10ADN2K	2,32	0,55	2,00	5,00
RF10ADN2KB	2,30	0,50	2,00	4,00
RF10ADN2KM	2,32	0,53	2,00	5,00
RF10ADN2KMB	2,30	0,50	2,00	4,00
RF10ADN3K	1,32	0,52	1,00	3,00
RF10ADN3KB	1,28	0,45	1,00	2,00
RF10ADN3KM	1,27	0,45	1,00	2,00
RF10ADN3KMB	1,29	0,47	1,00	3,00
RF50FO1	1,27	0,45	1,00	2,00
RF50FO2	1,28	0,47	1,00	3,00
RF50FO3	1,28	0,47	1,00	3,00
RF50ADN2K	2,33	0,55	2,00	5,00
RF50ADN2KB	2,30	0,50	2,00	4,00
RF50ADN2KM	2,32	0,53	2,00	5,00
RF50ADN2KMB	2,30	0,50	2,00	4,00
RF50ADN3K	1,31	0,52	1,00	3,00
RF50ADN3KB	1,29	0,47	1,00	3,00
RF50ADN3KM	1,27	0,45	1,00	2,00
RF50ADN3KMB	1,29	0,47	1,00	3,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 38 – Número de iterações das heurísticas para instâncias com 25 itens do MPCLSP.

Heurísticas	$nIter_{med}$ (%)	$nIter_{dp}$ (%)	$nIter_{min}$ (%)	$nIter_{max}$ (%)
GFO1	3,38	0,85	2,00	5,00
GFO2	3,08	0,60	2,00	5,00
GFO3	2,96	0,56	2,00	5,00
GADN2K	8,12	2,26	4,00	16,00
GADN2KB	9,10	1,45	6,00	12,00
GADN2KM	8,81	1,81	5,00	13,00
GADN2KMB	9,11	1,46	6,00	12,00
GADN3K	3,18	0,81	2,00	6,00
GADN3KB	3,28	0,73	2,00	5,00
GADN3KM	2,98	0,63	2,00	5,00
GADN3KMB	3,06	0,63	2,00	5,00
RF10FO1	1,46	0,62	1,00	3,00
RF10FO2	1,37	0,55	1,00	3,00
RF10FO3	1,39	0,49	1,00	2,00
RF10ADN2K	2,58	0,71	2,00	5,00
RF10ADN2KB	2,41	0,56	2,00	4,00
RF10ADN2KM	2,50	0,61	2,00	4,00
RF10ADN2KMB	2,41	0,56	2,00	4,00
RF10ADN3K	1,53	0,65	1,00	4,00
RF10ADN3KB	1,40	0,54	1,00	3,00
RF10ADN3KM	1,43	0,50	1,00	2,00
RF10ADN3KMB	1,38	0,49	1,00	2,00
RF50FO1	1,46	0,62	1,00	3,00
RF50FO2	1,37	0,55	1,00	3,00
RF50FO3	1,39	0,49	1,00	2,00
RF50ADN2K	2,68	0,72	2,00	4,00
RF50ADN2KB	2,41	0,56	2,00	4,00
RF50ADN2KM	2,50	0,61	2,00	4,00
RF50ADN2KMB	2,41	0,56	2,00	4,00
RF50ADN3K	1,42	0,57	1,00	4,00
RF50ADN3KB	1,38	0,49	1,00	2,00
RF50ADN3KM	1,42	0,50	1,00	2,00
RF50ADN3KMB	1,38	0,49	1,00	2,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 39 – Número de iterações das heurísticas para instâncias com 50 itens do MPCLSP.

Heurísticas	$nIter_{med} (%)$	$nIter_{dp} (%)$	$nIter_{min} (%)$	$nIter_{max} (%)$
GFO1	3,65	0,82	2,00	8,00
GFO2	3,42	0,72	2,00	5,00
GFO3	3,63	0,84	2,00	6,00
GADN2K	9,02	2,81	4,00	20,00
GADN2KB	9,93	1,44	7,00	15,00
GADN2KM	10,26	1,88	6,00	15,00
GADN2KMB	9,92	1,42	7,00	15,00
GADN3K	3,59	1,08	2,00	7,00
GADN3KB	3,60	0,81	2,00	6,00
GADN3KM	3,43	0,72	2,00	6,00
GADN3KMB	3,28	0,52	2,00	5,00
RF10FO1	1,40	0,60	1,00	4,00
RF10FO2	1,39	0,58	1,00	4,00
RF10FO3	1,43	0,69	1,00	4,00
RF10ADN2K	2,91	1,09	2,00	8,00
RF10ADN2KB	2,59	0,72	2,00	5,00
RF10ADN2KM	2,91	0,99	2,00	6,00
RF10ADN2KMB	2,60	0,71	2,00	5,00
RF10ADN3K	1,61	0,69	1,00	4,00
RF10ADN3KB	1,48	0,55	1,00	3,00
RF10ADN3KM	1,68	0,74	1,00	4,00
RF10ADN3KMB	1,54	0,56	1,00	3,00
RF50FO1	1,37	0,52	1,00	3,00
RF50FO2	1,39	0,58	1,00	4,00
RF50FO3	1,43	0,69	1,00	4,00
RF50ADN2K	2,79	0,92	2,00	6,00
RF50ADN2KB	2,53	0,69	2,00	5,00
RF50ADN2KM	2,86	1,00	2,00	6,00
RF50ADN2KMB	2,53	0,69	2,00	5,00
RF50ADN3K	1,66	0,88	1,00	5,00
RF50ADN3KB	1,48	0,59	1,00	4,00
RF50ADN3KM	1,56	0,71	1,00	4,00
RF50ADN3KMB	1,46	0,53	1,00	3,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 40 – Número de iterações das heurísticas para instâncias com 70 itens do MPCLSP.

Heurísticas	$nIter_{med}$ (%)	$nIter_{dp}$ (%)	$nIter_{min}$ (%)	$nIter_{max}$ (%)
GFO1	4,10	1,11	2,00	7,00
GFO2	3,58	0,76	2,00	6,00
GFO3	3,18	0,63	2,00	6,00
GADN2K	9,60	2,03	6,00	16,00
GADN2KB	9,84	1,69	7,00	14,00
GADN2KM	10,29	1,58	8,00	15,00
GADN2KMB	9,90	1,58	8,00	14,00
GADN3K	3,84	1,25	2,00	8,00
GADN3KB	3,68	1,29	2,00	9,00
GADN3KM	3,51	0,95	2,00	7,00
GADN3KMB	3,29	1,10	2,00	6,00
RF10FO1	2,12	1,29	1,00	6,00
RF10FO2	2,15	1,39	1,00	6,00
RF10FO3	1,94	1,27	1,00	6,00
RF10ADN2K	6,01	1,77	3,00	10,00
RF10ADN2KB	5,85	1,92	3,00	11,00
RF10ADN2KM	5,92	1,86	2,00	10,00
RF10ADN2KMB	5,76	1,84	3,00	11,00
RF10ADN3K	3,39	1,16	2,00	7,00
RF10ADN3KB	3,00	1,31	1,00	9,00
RF10ADN3KM	3,12	1,05	2,00	6,00
RF10ADN3KMB	3,02	1,09	2,00	6,00
RF50FO1	1,72	1,16	1,00	6,00
RF50FO2	1,60	0,98	1,00	5,00
RF50FO3	1,41	0,71	1,00	4,00
RF50ADN2K	4,99	2,11	2,00	10,00
RF50ADN2KB	4,29	1,86	2,00	12,00
RF50ADN2KM	4,32	1,82	2,00	9,00
RF50ADN2KMB	4,49	2,06	2,00	13,00
RF50ADN3K	2,68	1,31	1,00	7,00
RF50ADN3KB	2,49	1,18	1,00	7,00
RF50ADN3KM	2,54	1,30	1,00	7,00
RF50ADN3KMB	2,42	1,06	1,00	5,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 41 – Número de iterações das heurísticas para instâncias com 80 itens do MPCLSP.

Heurísticas	$nIter_{med}$ (%)	$nIter_{dp}$ (%)	$nIter_{min}$ (%)	$nIter_{max}$ (%)
GFO1	3,66	0,91	3,00	6,00
GFO2	3,41	0,59	2,00	5,00
GFO3	2,92	0,76	2,00	5,00
GADN2K	9,48	1,77	6,00	14,00
GADN2KB	9,69	1,70	7,00	14,00
GADN2KM	10,54	1,66	8,00	16,00
GADN2KMB	9,69	1,71	7,00	14,00
GADN3K	3,68	1,28	2,00	8,00
GADN3KB	3,61	0,89	2,00	6,00
GADN3KM	3,19	0,80	2,00	6,00
GADN3KMB	2,99	0,92	2,00	6,00
RF10FO1	2,16	1,31	1,00	6,00
RF10FO2	2,24	1,30	1,00	5,00
RF10FO3	2,02	1,38	1,00	6,00
RF10ADN2K	6,06	1,88	3,00	11,00
RF10ADN2KB	5,95	1,87	3,00	11,00
RF10ADN2KM	6,29	1,67	3,00	10,00
RF10ADN2KMB	6,18	1,91	3,00	11,00
RF10ADN3K	3,35	1,09	2,00	7,00
RF10ADN3KB	2,90	0,98	2,00	5,00
RF10ADN3KM	3,01	0,98	2,00	6,00
RF10ADN3KMB	2,88	1,07	2,00	7,00
RF50FO1	1,48	0,78	1,00	4,00
RF50FO2	1,58	0,85	1,00	4,00
RF50FO3	1,40	0,67	1,00	3,00
RF50ADN2K	4,42	1,94	2,00	13,00
RF50ADN2KB	4,03	1,81	2,00	9,00
RF50ADN2KM	4,29	1,77	2,00	9,00
RF50ADN2KMB	4,46	2,17	2,00	11,00
RF50ADN3K	2,50	1,09	1,00	6,00
RF50ADN3KB	2,48	1,11	1,00	7,00
RF50ADN3KM	2,31	1,06	1,00	6,00
RF50ADN3KMB	2,50	1,08	1,00	6,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

Tabela 42 – Número de iterações das heurísticas para instâncias com 90 itens do MPCLSP.

Heurísticas	$nIter_{med}$ (%)	$nIter_{dp}$ (%)	$nIter_{min}$ (%)	$nIter_{max}$ (%)
GFO1	3,75	0,82	2,00	6,00
GFO2	3,40	0,63	2,00	6,00
GFO3	2,98	0,86	2,00	7,00
GADN2K	9,41	2,04	5,00	14,00
GADN2KB	9,84	1,83	7,00	15,00
GADN2KM	10,21	1,79	6,00	14,00
GADN2KMB	9,86	1,84	8,00	15,00
GADN3K	3,56	1,22	2,00	7,00
GADN3KB	3,59	0,90	2,00	6,00
GADN3KM	3,44	0,76	2,00	6,00
GADN3KMB	3,34	1,41	2,00	8,00
RF10FO1	2,24	1,26	1,00	5,00
RF10FO2	2,34	1,20	1,00	5,00
RF10FO3	1,94	1,16	1,00	5,00
RF10ADN2K	6,10	1,99	3,00	11,00
RF10ADN2KB	6,16	1,89	3,00	12,00
RF10ADN2KM	6,29	1,90	3,00	11,00
RF10ADN2KMB	6,22	1,82	3,00	11,00
RF10ADN3K	3,25	1,10	2,00	7,00
RF10ADN3KB	3,05	1,30	2,00	8,00
RF10ADN3KM	3,16	1,03	2,00	6,00
RF10ADN3KMB	3,01	1,02	2,00	6,00
RF50FO1	1,70	1,02	1,00	5,00
RF50FO2	1,82	1,03	1,00	5,00
RF50FO3	1,40	0,72	1,00	4,00
RF50ADN2K	4,58	2,10	2,00	11,00
RF50ADN2KB	4,22	1,73	2,00	9,00
RF50ADN2KM	4,39	1,84	2,00	9,00
RF50ADN2KMB	4,40	1,93	2,00	11,00
RF50ADN3K	2,65	1,16	1,00	6,00
RF50ADN3KB	2,59	1,08	1,00	5,00
RF50ADN3KM	2,38	0,99	1,00	5,00
RF50ADN3KMB	2,55	0,90	1,00	5,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os maiores valores dentre as heurísticas em cada coluna.

ANÁLISES DOS RESULTADOS DAS HEURÍSTICAS PARA O MLCLSP

Para o problema MLCLSP, algumas heurísticas também não conseguiram encontrar uma solução inicial para todas as instâncias, pois o tempo disponível não era suficiente para geração de uma solução. A seguir, são listados os nomes das instâncias e os nomes das heurísticas que não conseguiram encontrar a solução:

- **EGC1310:** RF10ADN2KMB4NT, RF50ADN1KM4N, RF10ADN1KM4T, RF10ADN2KMB4T, RF10ADN3KM4T, RF10ADN1KMB3N, RF10ADN3KMBN3T
- **EGC1350:** RF10ADN3KM3T
- **EGC4340:** RF10ADN3KMB3N, RF10ADN3KMBN3T

Nas seções seguintes, são apresentadas as análises dos resultados para MLCLSP. Uma seção para cada heurística, a quantidade de instâncias que obtiveram o GAP menor do que o MIPGap do *Gurobi* e outra seção com o tempo gasto para cada instâncias e a quantidade de instâncias em que cada uma atingiu o tempo limite.

C.1 GAP

Foi realizado o cálculo do GAP médio (GAP_{med}), desvio padrão (GAP_{dp}), GAP mínimo (GAP_{min}) e GAP máximo (GAP_{max}) das heurísticas e do MIPGap médio, desvio padrão, MIPGap mínimo e MIPGap máximo do *Gurobi*. Na [Tabela 43](#), [Tabela 44](#), [Tabela 45](#), [Tabela 46](#), [Tabela 47](#), [Tabela 48](#), [Tabela 49](#), [Tabela 50](#), [Tabela 51](#), [Tabela 52](#), [Tabela 53](#), [Tabela 54](#), apresentamos os valores obtidos do GAP para heurísticas que utilizaram o G para MLCLSP, para classe A+ do MLCLSP, para classe B+ do MLCLSP, para classe C+ do MLCLSP; para heurísticas que

utilizaram o RF10 para MLCLSP, para classe A+ do MLCLSP, para classe B+ do MLCLSP, para classe C+ do MLCLSP; para heurísticas que utilizaram o R50 para MLCLSP, para classe A+ do MLCLSP, para classe B+ do MLCLSP e para classe C+ do MLCLSP, respectivamente. Em todas as tabelas, temos o *Gurobi*.

Tabela 43 – GAP das heurísticas que utilizaram G e o MIPGap do *Gurobi* para instâncias do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO3N	32,86	19,79	5,08	61,25
GFO4N	32,96	20,05	4,87	61,39
GFO3T	33,06	20,00	5,08	61,39
GFO4T	32,44	19,92	4,87	59,74
GADN1KM3N	68,65	22,09	15,98	99,50
GADN1KM4N	59,63	26,24	13,03	98,86
GADN1KM3T	67,57	20,67	31,08	99,61
GADN1KM4T	67,11	21,25	29,36	99,53
GADN1KM3NT	73,25	18,32	36,68	99,64
GADN1KM4NT	72,57	18,94	35,47	99,64
GADN1KMB3N	67,43	21,17	20,03	99,24
GADN1KMB4N	68,84	22,39	15,52	99,26
GADN1KMB3T	68,83	20,03	30,86	99,59
GADN1KMB4T	67,95	20,59	29,74	99,53
GADN1KMB3NT	73,70	18,08	38,13	99,71
GADN1KMB4NT	73,35	18,41	36,25	99,62
GADN2KM3N	36,43	23,05	5,69	68,02
GADN2KM4N	36,37	23,26	6,44	69,24
GADN2KM3T	38,50	23,92	4,43	68,51
GADN2KM4T	41,27	25,13	4,80	70,90
GADN2KM3NT	45,26	26,72	5,92	78,08
GADN2KM4NT	44,45	26,61	6,06	76,57
GADN2KMB3N	43,38	26,35	5,48	76,29
GADN2KMB4N	42,30	26,25	4,88	74,69
GADN2KMB3T	45,65	26,94	4,42	74,76
GADN2KMB4T	45,05	26,92	4,31	75,24
GADN2KMB3NT	48,15	26,95	5,49	78,24
GADN2KMB4NT	47,79	27,39	7,50	77,88
GADN3KM3N	30,83	16,68	5,40	54,96
GADN3KM4N	31,10	16,65	6,71	54,61
GADN3KM3T	31,45	18,43	5,38	59,32
GADN3KM4T	36,37	21,77	6,96	80,20
GADN3KM3NT	31,45	16,80	5,85	54,81
GADN3KM4NT	30,18	16,65	5,53	52,74
GADN3KMB3N	31,35	17,02	6,08	52,68
GADN3KMB4N	30,90	17,13	5,32	53,92
GADN3KMB3T	32,95	18,91	7,17	72,49
GADN3KMB4T	32,07	18,72	4,65	58,56
GADN3KMB3NT	32,44	16,99	6,23	54,04
GADN3KMB4NT	31,99	16,72	5,23	53,51
<i>Gurobi</i>	36,77	25,63	2,93	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 44 – GAP das heurísticas que utilizaram G e o MIPGap do *Gurobi* para instâncias da classe A+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO3N	9,59	2,64	5,08	13,96
GFO4N	8,88	2,23	4,87	11,42
GFO3T	9,59	2,64	5,08	13,96
GFO4T	8,81	2,18	4,87	11,42
GADN1KM3N	80,12	34,20	15,98	99,50
GADN1KM4N	56,81	44,45	13,03	98,86
GADN1KM3T	74,78	33,88	31,08	99,61
GADN1KM4T	73,77	35,13	29,36	99,53
GADN1KM3NT	85,27	26,48	36,68	99,64
GADN1KM4NT	84,46	27,77	35,47	99,64
GADN1KMB3N	72,58	35,27	20,03	99,24
GADN1KMB4N	78,07	36,05	15,52	99,26
GADN1KMB3T	75,04	33,27	30,86	99,59
GADN1KMB4T	74,17	34,34	29,74	99,53
GADN1KMB3NT	85,35	26,41	38,13	99,71
GADN1KMB4NT	84,77	27,21	36,25	99,62
GADN2KM3N	10,15	2,86	5,69	14,09
GADN2KM4N	9,68	2,55	6,44	13,15
GADN2KM3T	8,83	2,52	4,43	11,53
GADN2KM4T	9,08	2,76	4,80	12,34
GADN2KM3NT	11,42	3,18	5,92	16,39
GADN2KM4NT	10,64	2,80	6,06	15,01
GADN2KMB3N	10,27	3,22	5,48	14,83
GADN2KMB4N	9,22	2,94	4,88	14,51
GADN2KMB3T	10,00	3,09	4,42	13,92
GADN2KMB4T	9,62	3,14	4,31	13,72
GADN2KMB3NT	12,65	3,80	5,49	18,07
GADN2KMB4NT	11,50	2,75	7,50	15,14
GADN3KM3N	10,89	3,61	5,40	17,77
GADN3KM4N	10,82	3,38	6,71	17,00
GADN3KM3T	9,92	2,89	5,38	14,25
GADN3KM4T	18,19	25,14	6,96	80,20
GADN3KM3NT	11,70	3,61	5,85	16,92
GADN3KM4NT	10,53	2,76	5,53	13,63
GADN3KMB3N	11,08	3,62	6,08	16,81
GADN3KMB4N	10,37	3,42	5,32	17,00
GADN3KMB3T	17,75	22,21	7,17	72,49
GADN3KMB4T	9,91	3,11	4,65	13,33
GADN3KMB3NT	12,30	3,49	6,23	16,82
GADN3KMB4NT	12,08	3,34	5,23	15,25
<i>Gurobi</i>	7,09	2,77	2,93	11,25

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 45 – GAP das heurísticas que utilizaram G e o MIPGap do *Gurobi* para instâncias da classe C+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO3N	32,38	2,92	27,16	37,04
GFO4N	33,42	2,83	29,43	38,24
GFO3T	32,53	3,29	26,75	37,47
GFO4T	32,27	2,55	28,65	36,47
GADN1KM3N	54,19	5,04	45,15	60,18
GADN1KM4N	50,83	4,95	41,67	56,24
GADN1KM3T	56,33	5,11	47,61	62,39
GADN1KM4T	55,98	5,27	47,35	62,30
GADN1KM3NT	60,01	5,03	51,05	65,85
GADN1KM4NT	59,10	5,12	49,98	65,02
GADN1KMB3N	56,85	5,26	47,49	62,96
GADN1KMB4N	55,71	5,16	46,51	61,77
GADN1KMB3T	58,72	4,99	49,64	64,20
GADN1KMB4T	57,71	4,95	48,77	63,16
GADN1KMB3NT	60,97	5,10	51,86	66,83
GADN1KMB4NT	60,59	5,10	51,45	66,44
GADN2KM3N	34,21	2,73	31,16	39,26
GADN2KM4N	34,51	3,21	30,43	39,71
GADN2KM3T	41,18	4,50	35,41	46,86
GADN2KM4T	46,60	4,54	38,26	53,35
GADN2KM3NT	49,88	4,05	43,75	55,72
GADN2KM4NT	49,50	5,36	41,21	57,21
GADN2KMB3N	47,41	4,93	39,40	53,14
GADN2KMB4N	46,45	4,36	38,99	51,67
GADN2KMB3T	54,92	4,67	46,94	58,57
GADN2KMB4T	53,81	5,02	45,87	57,78
GADN2KMB3NT	57,15	5,19	47,97	62,98
GADN2KMB4NT	57,51	5,42	47,84	63,45
GADN3KM3N	31,38	3,15	26,04	36,70
GADN3KM4N	32,28	2,78	28,41	36,77
GADN3KM3T	30,87	3,17	25,67	34,66
GADN3KM4T	33,16	2,90	29,31	37,45
GADN3KM3NT	31,25	3,30	26,36	36,02
GADN3KM4NT	30,00	3,37	24,97	34,59
GADN3KMB3N	31,78	3,99	25,44	37,51
GADN3KMB4N	31,49	3,65	26,22	36,39
GADN3KMB3T	29,74	3,63	25,25	35,20
GADN3KMB4T	31,97	3,37	28,27	37,64
GADN3KMB3NT	32,60	3,78	25,89	36,47
GADN3KMB4NT	32,24	3,42	26,56	36,03
<i>Gurobi</i>	35,55	5,51	28,22	45,94

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 46 – GAP das heurísticas que utilizaram G e o MIPGap do *Gurobi* para instâncias da classe E+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
GFO3N	56,62	2,54	53,32	61,25
GFO4N	56,57	2,91	51,75	61,39
GFO3T	57,07	2,76	52,69	61,39
GFO4T	56,25	2,57	53,06	59,74
GADN1KM3N	71,65	2,97	67,74	75,41
GADN1KM4N	71,24	3,11	67,10	74,88
GADN1KM3T	71,60	2,67	67,16	74,65
GADN1KM4T	71,59	2,64	67,39	74,63
GADN1KM3NT	74,47	2,98	70,49	78,00
GADN1KM4NT	74,14	2,96	70,08	77,67
GADN1KMB3N	72,88	3,04	68,73	76,58
GADN1KMB4N	72,75	3,03	68,64	76,47
GADN1KMB3T	72,73	2,85	68,69	76,12
GADN1KMB4T	71,95	2,76	67,98	75,24
GADN1KMB3NT	74,77	2,99	70,68	78,36
GADN1KMB4NT	74,70	3,00	70,60	78,29
GADN2KM3N	64,93	2,51	60,91	68,02
GADN2KM4N	64,90	3,49	58,23	69,24
GADN2KM3T	65,50	2,52	61,61	68,51
GADN2KM4T	68,14	2,13	64,40	70,90
GADN2KM3NT	74,50	2,97	70,47	78,08
GADN2KM4NT	73,22	2,87	69,15	76,57
GADN2KMB3N	72,46	3,08	68,23	76,29
GADN2KMB4N	71,25	2,95	66,86	74,69
GADN2KMB3T	72,02	2,71	67,43	74,76
GADN2KMB4T	71,73	3,15	66,15	75,24
GADN2KMB3NT	74,65	3,00	70,61	78,24
GADN2KMB4NT	74,35	2,92	70,36	77,88
GADN3KM3N	50,22	2,62	46,82	54,96
GADN3KM4N	50,18	2,08	48,36	54,61
GADN3KM3T	53,57	2,79	50,72	59,32
GADN3KM4T	57,75	2,32	53,85	61,52
GADN3KM3NT	51,39	1,72	49,15	54,81
GADN3KM4NT	49,99	1,49	47,54	52,74
GADN3KMB3N	51,21	1,66	47,96	52,68
GADN3KMB4N	50,84	1,81	47,63	53,92
GADN3KMB3T	51,36	2,26	47,19	55,31
GADN3KMB4T	54,32	1,86	52,24	58,56
GADN3KMB3NT	52,42	1,41	49,29	54,04
GADN3KMB4NT	51,64	1,29	49,58	53,51
<i>Gurobi</i>	67,68	4,39	61,51	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 47 – GAP das heurísticas que utilizaram RF10 e o MIPGap do *Gurobi* para instâncias do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF10FO3N	30,47	18,65	3,49	60,17
RF10FO4N	29,42	17,64	3,84	55,85
RF10FO3T	29,84	17,96	3,49	57,22
RF10FO4T	29,62	17,92	3,84	57,00
RF10ADN1KM3N	34,06	21,28	4,01	64,50
RF10ADN1KM4N	34,14	21,31	4,01	65,44
RF10ADN1KM3T	34,65	21,39	3,72	65,40
RF10ADN1KM4T	36,15	23,64	3,49	68,67
RF10ADN1KM3NT	35,31	21,87	4,18	66,15
RF10ADN1KM4NT	34,09	21,07	3,85	65,13
RF10ADN1KMB3N	32,92	20,61	4,10	62,80
RF10ADN1KMB4N	34,45	21,65	3,65	66,89
RF10ADN1KMB3T	34,73	21,30	3,72	66,06
RF10ADN1KMB4T	37,17	23,52	4,10	67,36
RF10ADN1KMB3NT	35,31	21,48	4,22	67,34
RF10ADN1KMB4NT	35,73	21,75	4,10	67,35
RF10ADN2KM3N	34,92	22,80	4,01	66,99
RF10ADN2KM4N	33,93	21,81	4,01	66,34
RF10ADN2KM3T	32,31	20,55	3,84	63,44
RF10ADN2KM4T	33,11	21,46	3,49	64,70
RF10ADN2KM3NT	32,55	20,80	3,85	64,73
RF10ADN2KM4NT	34,13	21,83	3,85	65,50
RF10ADN2KMB3N	34,48	22,83	3,85	68,09
RF10ADN2KMB4N	32,29	20,73	3,49	65,89
RF10ADN2KMB3T	33,68	21,41	3,65	64,63
RF10ADN2KMB4T	32,97	21,21	4,10	63,97
RF10ADN2KMB3NT	36,07	23,71	4,22	69,48
RF10ADN2KMB4NT	31,83	20,56	4,10	61,48
RF10ADN3KM3N	30,48	17,72	3,69	55,62
RF10ADN3KM4N	30,45	17,96	3,85	57,23
RF10ADN3KM3T	29,47	18,76	3,84	60,14
RF10ADN3KM4T	28,66	18,06	3,49	52,36
RF10ADN3KM3NT	29,85	17,23	4,22	54,30
RF10ADN3KM4NT	29,04	16,82	3,85	51,47
RF10ADN3KMB3N	29,39	17,43	3,85	54,22
RF10ADN3KMB4N	30,34	18,17	3,49	54,66
RF10ADN3KMB3T	29,68	18,16	3,65	56,69
RF10ADN3KMB4T	30,99	18,90	4,10	60,51
RF10ADN3KMB3NT	28,73	17,12	4,22	52,50
RF10ADN3KMB4NT	30,05	17,29	4,10	52,37
<i>Gurobi</i>	36,77	25,63	2,93	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 48 – GAP das heurísticas que utilizaram RF10 e o MIPGap do *Gurobi* para instâncias da classe A+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF10FO3N	8,75	2,92	3,49	12,81
RF10FO4N	8,46	2,63	3,84	12,06
RF10FO3T	8,75	2,92	3,49	12,81
RF10FO4T	8,46	2,63	3,84	12,06
RF10ADN1KM3N	9,22	2,88	4,01	12,95
RF10ADN1KM4N	9,10	2,88	4,01	12,95
RF10ADN1KM3T	9,42	2,80	3,72	12,89
RF10ADN1KM4T	8,99	2,72	3,49	12,12
RF10ADN1KM3NT	9,93	2,87	4,18	12,90
RF10ADN1KM4NT	9,64	3,06	3,85	12,89
RF10ADN1KMB3N	9,43	2,66	4,10	12,90
RF10ADN1KMB4N	9,17	3,02	3,65	12,78
RF10ADN1KMB3T	9,70	3,03	3,72	13,29
RF10ADN1KMB4T	9,65	2,85	4,10	12,91
RF10ADN1KMB3NT	10,03	2,95	4,22	13,30
RF10ADN1KMB4NT	9,92	2,91	4,10	13,01
RF10ADN2KM3N	8,75	2,72	4,01	12,95
RF10ADN2KM4N	8,43	2,74	4,01	12,95
RF10ADN2KM3T	8,07	2,48	3,84	11,16
RF10ADN2KM4T	7,72	2,46	3,49	10,94
RF10ADN2KM3NT	9,24	2,65	3,85	12,87
RF10ADN2KM4NT	8,93	2,76	3,85	12,61
RF10ADN2KMB3N	8,87	2,59	3,85	12,78
RF10ADN2KMB4N	8,56	2,84	3,49	12,78
RF10ADN2KMB3T	8,41	2,85	3,65	12,81
RF10ADN2KMB4T	8,56	2,56	4,10	12,35
RF10ADN2KMB3NT	9,45	2,61	4,22	13,06
RF10ADN2KMB4NT	9,41	2,67	4,10	13,01
RF10ADN3KM3N	9,03	2,73	3,69	12,82
RF10ADN3KM4N	8,56	2,79	3,85	12,95
RF10ADN3KM3T	8,25	2,64	3,84	11,97
RF10ADN3KM4T	7,84	2,49	3,49	10,86
RF10ADN3KM3NT	9,45	2,57	4,22	12,90
RF10ADN3KM4NT	9,04	2,77	3,85	12,61
RF10ADN3KMB3N	9,01	2,62	3,85	12,78
RF10ADN3KMB4N	8,72	2,83	3,49	12,78
RF10ADN3KMB3T	8,38	2,94	3,65	12,81
RF10ADN3KMB4T	8,88	2,64	4,10	13,01
RF10ADN3KMB3NT	9,64	2,69	4,22	13,06
RF10ADN3KMB4NT	9,51	2,67	4,10	13,01
<i>Gurobi</i>	7,09	2,77	2,93	11,25

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 49 – GAP das heurísticas que utilizaram RF10 e o MIPGap do *Gurobi* para instâncias da classe C+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF10FO3N	29,90	3,78	24,13	34,88
RF10FO4N	29,54	3,35	25,27	34,61
RF10FO3T	29,52	3,53	24,30	34,21
RF10FO4T	29,53	3,54	24,24	34,61
RF10ADN1KM3N	33,24	3,57	28,72	38,21
RF10ADN1KM4N	33,61	3,05	29,69	37,27
RF10ADN1KM3T	34,38	3,87	29,08	40,62
RF10ADN1KM4T	37,24	3,51	32,71	42,03
RF10ADN1KM3NT	34,09	2,94	30,26	37,51
RF10ADN1KM4NT	33,05	3,96	26,25	37,16
RF10ADN1KMB3N	33,56	3,09	29,76	37,26
RF10ADN1KMB4N	33,67	3,50	29,21	38,29
RF10ADN1KMB3T	34,35	4,15	29,47	41,19
RF10ADN1KMB4T	36,25	3,84	31,30	42,78
RF10ADN1KMB3NT	34,90	3,78	30,65	40,37
RF10ADN1KMB4NT	35,75	3,31	31,73	41,73
RF10ADN2KM3N	33,10	3,10	29,02	38,48
RF10ADN2KM4N	33,37	3,77	28,02	39,55
RF10ADN2KM3T	31,98	3,39	27,10	36,01
RF10ADN2KM4T	32,94	3,52	28,88	38,16
RF10ADN2KM3NT	29,99	3,73	24,48	34,85
RF10ADN2KM4NT	32,80	4,21	27,24	39,03
RF10ADN2KMB3N	31,82	4,06	27,35	39,63
RF10ADN2KMB4N	30,83	3,82	26,20	37,21
RF10ADN2KMB3T	33,33	3,28	28,80	37,20
RF10ADN2KMB4T	34,21	3,96	29,15	40,88
RF10ADN2KMB3NT	33,20	4,23	28,50	41,85
RF10ADN2KMB4NT	30,56	3,01	26,75	34,08
RF10ADN3KM3N	31,41	3,33	26,84	37,46
RF10ADN3KM4N	31,79	3,22	27,89	37,04
RF10ADN3KM3T	30,01	3,47	24,78	34,34
RF10ADN3KM4T	29,76	3,55	24,56	33,94
RF10ADN3KM3NT	29,88	3,75	23,78	34,22
RF10ADN3KM4NT	29,22	3,57	24,07	34,23
RF10ADN3KMB3N	31,13	3,58	25,77	35,68
RF10ADN3KMB4N	31,63	3,26	27,02	36,22
RF10ADN3KMB3T	29,29	3,46	24,20	33,04
RF10ADN3KMB4T	30,44	3,17	25,48	34,68
RF10ADN3KMB3NT	30,91	3,43	25,93	35,13
RF10ADN3KMB4NT	30,16	3,69	24,08	34,05
<i>Gurobi</i>	35,55	5,51	28,22	45,94

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 50 – GAP das heurísticas que utilizaram RF10 e o MIPGap do *Gurobi* para instâncias da classe E+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF10FO3N	52,76	3,57	48,13	60,17
RF10FO4N	50,26	2,47	47,14	55,85
RF10FO3T	51,23	2,66	48,23	57,22
RF10FO4T	50,87	2,73	47,40	57,00
RF10ADN1KM3N	59,73	2,78	56,78	64,50
RF10ADN1KM4N	59,71	3,15	56,04	65,44
RF10ADN1KM3T	60,16	3,08	56,92	65,40
RF10ADN1KM4T	65,94	2,30	63,43	68,67
RF10ADN1KM3NT	61,91	3,01	58,37	66,15
RF10ADN1KM4NT	59,58	2,70	56,04	65,13
RF10ADN1KMB3N	59,05	2,21	56,07	62,80
RF10ADN1KMB4N	60,51	3,36	56,50	66,89
RF10ADN1KMB3T	60,14	3,21	56,72	66,06
RF10ADN1KMB4T	65,61	1,78	62,10	67,36
RF10ADN1KMB3NT	60,98	3,18	57,35	67,34
RF10ADN1KMB4NT	61,51	3,75	55,87	67,35
RF10ADN2KM3N	62,92	2,85	59,21	66,99
RF10ADN2KM4N	60,01	4,46	54,24	66,34
RF10ADN2KM3T	56,88	2,95	53,21	63,44
RF10ADN2KM4T	58,67	3,27	54,10	64,70
RF10ADN2KM3NT	58,41	2,88	55,12	64,73
RF10ADN2KM4NT	60,67	2,81	57,68	65,50
RF10ADN2KMB3N	62,74	4,24	57,64	68,09
RF10ADN2KMB4N	57,49	4,19	53,36	65,89
RF10ADN2KMB3T	59,30	2,69	56,46	64,63
RF10ADN2KMB4T	59,46	2,96	55,54	63,97
RF10ADN2KMB3NT	65,57	3,04	60,08	69,48
RF10ADN2KMB4NT	58,90	2,05	55,43	61,48
RF10ADN3KM3N	51,01	2,30	47,63	55,62
RF10ADN3KM4N	51,00	3,01	46,90	57,23
RF10ADN3KM3T	53,11	3,93	49,35	60,14
RF10ADN3KM4T	51,20	1,22	48,63	52,36
RF10ADN3KM3NT	50,23	1,94	47,21	54,30
RF10ADN3KM4NT	48,86	1,72	45,44	51,47
RF10ADN3KMB3N	50,69	1,98	47,95	54,22
RF10ADN3KMB4N	50,83	2,15	46,88	54,66
RF10ADN3KMB3T	51,36	2,66	46,95	56,69
RF10ADN3KMB4T	53,66	3,29	48,77	60,51
RF10ADN3KMB3NT	51,28	1,63	48,06	52,50
RF10ADN3KMB4NT	50,48	1,57	47,15	52,37
<i>Gurobi</i>	67,68	4,39	61,51	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 51 – GAP das heurísticas que utilizaram RF50 e o MIPGap do *Gurobi* para instâncias do MLCLSP.

Heurísticas	GAP _{med} (%)	GAP _{dp} (%)	GAP _{min} (%)	GAP _{max} (%)
RF50FO3N	29,72	17,87	3,49	55,88
RF50FO4N	29,63	17,94	3,84	56,84
RF50FO3T	30,66	18,82	3,49	58,84
RF50FO4T	29,90	18,02	3,84	58,01
RF50ADN1KM3N	32,47	19,68	4,01	62,16
RF50ADN1KM4N	31,54	19,46	4,01	57,87
RF50ADN1KM3T	31,32	18,67	3,72	58,40
RF50ADN1KM4T	32,65	19,94	3,49	62,73
RF50ADN1KM3NT	31,87	18,75	4,18	56,90
RF50ADN1KM4NT	31,46	18,77	3,85	60,46
RF50ADN1KMB3N	33,03	20,09	4,10	63,16
RF50ADN1KMB4N	32,96	20,25	3,65	63,28
RF50ADN1KMB3T	31,89	18,81	3,72	60,26
RF50ADN1KMB4T	32,87	19,66	4,10	62,67
RF50ADN1KMB3NT	33,31	20,00	4,22	64,40
RF50ADN1KMB4NT	31,34	18,42	4,10	56,51
RF50ADN2KM3N	30,78	18,55	4,01	59,06
RF50ADN2KM4N	31,81	19,68	4,01	61,99
RF50ADN2KM3T	32,53	20,77	3,84	63,20
RF50ADN2KM4T	32,21	20,40	3,49	62,71
RF50ADN2KM3NT	32,01	20,08	3,85	62,56
RF50ADN2KM4NT	30,69	18,58	3,85	56,45
RF50ADN2KMB3N	31,09	19,29	3,85	61,54
RF50ADN2KMB4N	31,88	20,11	3,49	63,80
RF50ADN2KMB3T	33,07	20,91	3,65	64,20
RF50ADN2KMB4T	31,86	19,52	4,10	60,36
RF50ADN2KMB3NT	33,17	20,68	4,22	63,75
RF50ADN2KMB4NT	30,87	18,71	4,10	56,91
RF50ADN3KM3N	29,36	16,98	3,69	53,94
RF50ADN3KM4N	29,94	17,71	3,85	55,90
RF50ADN3KM3T	29,59	18,01	3,84	55,74
RF50ADN3KM4T	30,78	19,39	3,49	59,72
RF50ADN3KM3NT	31,67	19,74	4,22	62,55
RF50ADN3KM4NT	28,71	16,67	3,85	51,68
RF50ADN3KMB3N	29,34	17,07	3,85	53,00
RF50ADN3KMB4N	27,53	16,84	3,49	53,68
RF50ADN3KMB3T	30,63	19,03	3,65	58,40
RF50ADN3KMB4T	30,66	18,47	4,10	60,31
RF50ADN3KMB3NT	31,62	19,28	4,22	63,25
RF50ADN3KMB4NT	29,41	16,95	4,10	51,27
<i>Gurobi</i>	36,77	25,63	2,93	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 52 – GAP das heurísticas que utilizaram RF50 e o MIPGap do *Gurobi* para instâncias da classe A+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF50FO3N	8,75	2,92	3,49	12,81
RF50FO4N	8,46	2,63	3,84	12,06
RF50FO3T	8,75	2,92	3,49	12,81
RF50FO4T	8,46	2,63	3,84	12,06
RF50ADN1KM3N	9,22	2,88	4,01	12,95
RF50ADN1KM4N	9,10	2,88	4,01	12,95
RF50ADN1KM3T	9,42	2,80	3,72	12,89
RF50ADN1KM4T	8,99	2,72	3,49	12,12
RF50ADN1KM3NT	9,93	2,87	4,18	12,90
RF50ADN1KM4NT	9,64	3,06	3,85	12,89
RF50ADN1KMB3N	9,43	2,66	4,10	12,90
RF50ADN1KMB4N	9,17	3,02	3,65	12,78
RF50ADN1KMB3T	9,70	3,03	3,72	13,29
RF50ADN1KMB4T	9,65	2,85	4,10	12,91
RF50ADN1KMB3NT	10,03	2,95	4,22	13,30
RF50ADN1KMB4NT	9,92	2,91	4,10	13,01
RF50ADN2KM3N	8,75	2,72	4,01	12,95
RF50ADN2KM4N	8,43	2,74	4,01	12,95
RF50ADN2KM3T	8,07	2,48	3,84	11,16
RF50ADN2KM4T	7,72	2,46	3,49	10,94
RF50ADN2KM3NT	9,24	2,65	3,85	12,87
RF50ADN2KM4NT	8,93	2,76	3,85	12,61
RF50ADN2KMB3N	8,87	2,59	3,85	12,78
RF50ADN2KMB4N	8,56	2,84	3,49	12,78
RF50ADN2KMB3T	8,41	2,85	3,65	12,81
RF50ADN2KMB4T	8,56	2,56	4,10	12,35
RF50ADN2KMB3NT	9,45	2,61	4,22	13,06
RF50ADN2KMB4NT	9,41	2,67	4,10	13,01
RF50ADN3KM3N	9,03	2,73	3,69	12,82
RF50ADN3KM4N	8,56	2,79	3,85	12,95
RF50ADN3KM3T	8,25	2,64	3,84	11,97
RF50ADN3KM4T	7,84	2,49	3,49	10,86
RF50ADN3KM3NT	9,45	2,57	4,22	12,90
RF50ADN3KM4NT	9,04	2,77	3,85	12,61
RF50ADN3KMB3N	9,01	2,62	3,85	12,78
RF50ADN3KMB4N	8,72	2,83	3,49	12,78
RF50ADN3KMB3T	8,38	2,94	3,65	12,81
RF50ADN3KMB4T	8,88	2,64	4,10	13,01
RF50ADN3KMB3NT	9,64	2,69	4,22	13,06
RF50ADN3KMB4NT	9,51	2,67	4,10	13,01
<i>Gurobi</i>	7,09	2,77	2,93	11,25

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 53 – GAP das heurísticas que utilizaram RF50 e o MIPGap do *Gurobi* para instâncias da classe C+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF50FO3N	29,37	3,57	24,25	34,20
RF50FO4N	29,52	3,72	24,83	34,96
RF50FO3T	29,96	3,45	24,93	33,92
RF50FO4T	30,13	3,39	25,54	35,78
RF50ADN1KM3N	32,38	3,43	27,53	36,60
RF50ADN1KM4N	32,74	3,48	26,91	37,10
RF50ADN1KM3T	30,88	3,75	25,80	35,42
RF50ADN1KM4T	32,83	3,90	26,71	37,47
RF50ADN1KM3NT	31,24	3,56	25,94	35,50
RF50ADN1KM4NT	30,70	3,37	26,63	34,97
RF50ADN1KMB3N	32,59	3,78	26,12	36,82
RF50ADN1KMB4N	32,60	3,78	26,23	36,83
RF50ADN1KMB3T	31,78	3,89	26,10	36,21
RF50ADN1KMB4T	32,80	3,79	26,76	36,84
RF50ADN1KMB3NT	32,53	3,64	27,44	37,10
RF50ADN1KMB4NT	30,57	3,80	24,80	35,13
RF50ADN2KM3N	30,90	3,34	26,17	34,72
RF50ADN2KM4N	31,95	3,24	27,32	35,71
RF50ADN2KM3T	32,11	3,50	26,82	36,32
RF50ADN2KM4T	32,91	3,75	27,00	37,02
RF50ADN2KM3NT	30,04	3,67	24,40	34,21
RF50ADN2KM4NT	30,17	3,73	25,43	35,15
RF50ADN2KMB3N	29,89	3,59	25,47	35,31
RF50ADN2KMB4N	31,03	3,58	26,10	36,31
RF50ADN2KMB3T	32,74	3,46	27,77	36,65
RF50ADN2KMB4T	32,20	3,69	26,90	36,60
RF50ADN2KMB3NT	31,54	3,23	27,15	35,71
RF50ADN2KMB4NT	29,46	3,59	24,38	33,70
RF50ADN3KM3N	29,82	3,37	24,72	34,27
RF50ADN3KM4N	30,87	3,50	25,90	35,83
RF50ADN3KM3T	29,59	3,54	23,98	34,14
RF50ADN3KM4T	30,69	3,66	25,17	34,96
RF50ADN3KM3NT	29,50	3,52	23,82	33,70
RF50ADN3KM4NT	28,63	3,61	23,33	32,82
RF50ADN3KMB3N	29,61	3,48	24,32	33,94
RF50ADN3KMB4N	29,85	3,68	24,02	33,76
RF50ADN3KMB3T	30,01	3,19	25,40	34,47
RF50ADN3KMB4T	30,58	3,57	25,36	35,59
RF50ADN3KMB3NT	30,08	3,39	24,24	34,12
RF50ADN3KMB4NT	29,08	3,70	23,25	33,44
<i>Gurobi</i>	35,55	5,51	28,22	45,94

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 54 – GAP das heurísticas que utilizaram RF50 e o MIPGap do *Gurobi* para instâncias da classe E+ do MLCLSP.

Heurísticas	GAP_{med} (%)	GAP_{dp} (%)	GAP_{min} (%)	GAP_{max} (%)
RF50FO3N	51,05	2,30	47,73	55,88
RF50FO4N	50,90	2,70	47,71	56,84
RF50FO3T	53,27	3,14	48,74	58,84
RF50FO4T	51,09	3,12	47,27	58,01
RF50ADN1KM3N	55,80	3,46	51,21	62,16
RF50ADN1KM4N	55,83	1,90	52,77	57,87
RF50ADN1KM3T	53,65	2,35	49,74	58,40
RF50ADN1KM4T	56,12	3,74	51,71	62,73
RF50ADN1KM3NT	54,44	1,45	52,47	56,90
RF50ADN1KM4NT	54,03	3,09	49,36	60,46
RF50ADN1KMB3N	57,07	2,79	54,19	63,16
RF50ADN1KMB4N	57,12	3,18	52,25	63,28
RF50ADN1KMB3T	54,18	2,78	50,26	60,26
RF50ADN1KMB4T	56,15	3,40	52,29	62,67
RF50ADN1KMB3NT	57,38	3,23	54,28	64,40
RF50ADN1KMB4NT	53,55	1,82	49,82	56,51
RF50ADN2KM3N	52,69	3,02	48,80	59,06
RF50ADN2KM4N	55,06	3,47	50,67	61,99
RF50ADN2KM3T	57,41	2,84	54,06	63,20
RF50ADN2KM4T	56,01	3,75	50,35	62,71
RF50ADN2KM3NT	56,74	2,81	53,73	62,56
RF50ADN2KM4NT	52,96	1,99	49,41	56,45
RF50ADN2KMB3N	54,50	3,20	51,40	61,54
RF50ADN2KMB4N	56,06	4,11	52,96	63,80
RF50ADN2KMB3T	58,06	2,87	54,86	64,20
RF50ADN2KMB4T	54,83	2,91	50,46	60,36
RF50ADN2KMB3NT	58,53	2,67	54,86	63,75
RF50ADN2KMB4NT	53,75	1,77	50,63	56,91
RF50ADN3KM3N	49,22	2,28	45,72	53,94
RF50ADN3KM4N	50,40	2,84	45,79	55,90
RF50ADN3KM3T	50,91	2,55	46,45	55,74
RF50ADN3KM4T	53,80	2,96	50,42	59,72
RF50ADN3KM3NT	56,07	3,39	52,64	62,55
RF50ADN3KM4NT	48,46	2,03	44,55	51,68
RF50ADN3KMB3N	49,41	2,38	44,41	53,00
RF50ADN3KMB4N	49,51	2,66	45,35	53,68
RF50ADN3KMB3T	53,49	2,71	50,03	58,40
RF50ADN3KMB4T	52,51	3,67	47,70	60,31
RF50ADN3KMB3NT	55,13	4,04	51,63	63,25
RF50ADN3KMB4NT	49,62	1,87	45,35	51,27
<i>Gurobi</i>	67,68	4,39	61,51	74,22

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

C.2 Quantidade de soluções melhores do que o *Gurobi*

Foi calculado, para cada heurística, o número de instâncias que obteve o GAP_{ini} menor do que o MIPGap (gI), o número de instâncias que o GAP menor do que o MIPGap (gM), a diferença entre $gM - gI$ e a porcentagem que cada um representa no número total de instâncias que é 24. Nas [Tabela 55](#), [Tabela 56](#), [Tabela 57](#), apresentamos os resultados para as heurísticas

que utilizaram o G, para as heurísticas que utilizaram o RF10 e para as heurísticas que utilizaram o RF50, respectivamente.

Tabela 55 – Número de instâncias do MLCLSP que tiveram GAP_{Ini} e GAP menor do que o MIPGap do *Gurobi* para heurísticas que utilizaram o G.

Heurísticas	gI	$\frac{gI}{24} \times 100$	gM	$\frac{gM}{24} \times 100$	$gI - gM$	$\frac{gM - gI}{gM} \times 100$
GFO3N	0	0,00	15	62,50	15	62,50
GFO4N	0	0,00	14	58,33	14	58,33
GFO3T	0	0,00	15	62,50	15	62,50
GFO4T	0	0,00	15	62,50	15	62,50
GADN1KM3N	0	0,00	2	8,33	2	8,33
GADN1KM4N	0	0,00	2	8,33	2	8,33
GADN1KM3T	0	0,00	2	8,33	2	8,33
GADN1KM4T	0	0,00	2	8,33	2	8,33
GADN1KM3NT	0	0,00	0	0,00	0	0,00
GADN1KM4NT	0	0,00	0	0,00	0	0,00
GADN1KMB3N	0	0,00	2	8,33	2	8,33
GADN1KMB4N	0	0,00	2	8,33	2	8,33
GADN1KMB3T	0	0,00	2	8,33	2	8,33
GADN1KMB4T	0	0,00	2	8,33	2	8,33
GADN1KMB3NT	0	0,00	0	0,00	0	0,00
GADN1KMB4NT	0	0,00	0	0,00	0	0,00
GADN2KM3N	0	0,00	10	41,67	10	41,67
GADN2KM4N	0	0,00	10	41,67	10	41,67
GADN2KM3T	0	0,00	6	25,00	6	25,00
GADN2KM4T	0	0,00	6	25,00	6	25,00
GADN2KM3NT	0	0,00	0	0,00	0	0,00
GADN2KM4NT	0	0,00	1	4,17	1	4,17
GADN2KMB3N	0	0,00	3	12,50	3	12,50
GADN2KMB4N	0	0,00	2	8,33	2	8,33
GADN2KMB3T	0	0,00	2	8,33	2	8,33
GADN2KMB4T	0	0,00	3	12,50	3	12,50
GADN2KMB3NT	0	0,00	0	0,00	0	0,00
GADN2KMB4NT	0	0,00	0	0,00	0	0,00
GADN3KM3N	0	0,00	17	70,83	17	70,83
GADN3KM4N	0	0,00	15	62,50	15	62,50
GADN3KM3T	0	0,00	16	66,67	16	66,67
GADN3KM4T	0	0,00	13	54,17	13	54,17
GADN3KM3NT	0	0,00	16	66,67	16	66,67
GADN3KM4NT	0	0,00	16	66,67	16	66,67
GADN3KMB3N	0	0,00	16	66,67	16	66,67
GADN3KMB4N	0	0,00	17	70,83	17	70,83
GADN3KMB3T	0	0,00	16	66,67	16	66,67
GADN3KMB4T	0	0,00	15	62,50	15	62,50
GADN3KMB3NT	0	0,00	16	66,67	16	66,67
GADN3KMB4NT	0	0,00	16	66,67	16	66,67

Fonte: Elaborada pela autora.

Nota – Em negrito, estão os maiores valores obtidos em cada coluna, exceto quando é igual a zero. Encontra-se, para cada heurística, a quantidade de instâncias (gI) que obteve GAP_{Ini} menor do que MIPGap, a quantidade de instâncias (gM) que obteve GAP menor do que MIPGap, a quantidade de instâncias que obteve GAP menor do que o MIPGap depois da melhoria ($gI - gM$).

Tabela 56 – Número de instâncias do MLCLSP que tiveram GAPIni e GAP menor do que o MIPGap do Gurobi para heurísticas que utilizaram o RF10.

Heurísticas	gI	$\frac{gI}{24} \times 100$	gM	$\frac{gM}{24} \times 100$	$gI - gM$	$\frac{gM - gI}{gM} \times 100$
RF10FO3N	6	25,00	16	66,67	10	41,67
RF10FO4N	13	54,17	17	70,83	4	16,67
RF10FO3T	10	41,67	16	66,67	6	25,00
RF10FO4T	15	62,50	17	70,83	2	8,33
RF10ADN1KM3N	14	58,33	15	62,50	1	4,17
RF10ADN1KM4N	12	50,00	15	62,50	3	12,50
RF10ADN1KM3T	12	50,00	13	54,17	1	4,17
RF10ADN1KM4T	4	16,67	6	25,00	2	8,33
RF10ADN1KM3NT	12	50,00	13	54,17	1	4,17
RF10ADN1KM4NT	15	62,50	15	62,50	0	0,00
RF10ADN1KMB3N	11	45,83	14	58,33	3	12,50
RF10ADN1KMB4N	11	45,83	15	62,50	4	16,67
RF10ADN1KMB3T	13	54,17	13	54,17	0	0,00
RF10ADN1KMB4T	6	25,00	8	33,33	2	8,33
RF10ADN1KMB3NT	12	50,00	12	50,00	0	0,00
RF10ADN1KMB4NT	10	41,67	12	50,00	2	8,33
RF10ADN2KM3N	5	20,83	14	58,33	9	37,50
RF10ADN2KM4N	5	20,83	16	66,67	11	45,83
RF10ADN2KM3T	13	54,17	17	70,83	4	16,67
RF10ADN2KM4T	13	54,17	17	70,83	4	16,67
RF10ADN2KM3NT	15	62,50	17	70,83	2	8,33
RF10ADN2KM4NT	9	37,50	16	66,67	7	29,17
RF10ADN2KMB3N	6	25,00	15	62,50	9	37,50
RF10ADN2KMB4N	12	50,00	17	70,83	5	20,83
RF10ADN2KMB3T	12	50,00	16	66,67	4	16,67
RF10ADN2KMB4T	8	33,33	14	58,33	6	25,00
RF10ADN2KMB3NT	10	41,67	13	54,17	3	12,50
RF10ADN2KMB4NT	12	50,00	15	62,50	3	12,50
RF10ADN3KM3N	6	25,00	17	70,83	11	45,83
RF10ADN3KM4N	5	20,83	16	66,67	11	45,83
RF10ADN3KM3T	8	33,33	16	66,67	8	33,33
RF10ADN3KM4T	11	45,83	16	66,67	5	20,83
RF10ADN3KM3NT	9	37,50	16	66,67	7	29,17
RF10ADN3KM4NT	13	54,17	16	66,67	3	12,50
RF10ADN3KMB3N	4	16,67	16	66,67	12	50,00
RF10ADN3KMB4N	5	20,83	16	66,67	11	45,83
RF10ADN3KMB3T	13	54,17	17	70,83	4	16,67
RF10ADN3KMB4T	5	20,83	17	70,83	12	50,00
RF10ADN3KMB3NT	8	33,33	14	58,33	6	25,00
RF10ADN3KMB4NT	15	62,50	16	66,67	1	4,17

Fonte: Elaborada pela autora.

Nota – Em negrito, estão os maiores valores obtidos em cada coluna. Encontra-se, para cada heurística, a quantidade de instâncias (gI) que obteve GAPIni menor do que MIPGap, a quantidade de instâncias (gM) que obteve GAP menor do que MIPGap, a quantidade de instâncias que obteve GAP menor do que o MIPGap depois da melhoria ($gI - gM$).

Tabela 57 – Número de instâncias do MLCLSP que tiveram GAP_{Ini} e GAP menor do que o MIPGap do *Gurobi* para heurísticas que utilizaram o RF50.

Heurísticas	gI	$\frac{gI}{24} \times 100$	gM	$\frac{gM}{24} \times 100$	$gI - gM$	$\frac{gM - gI}{gM} \times 100$
RF50FO3N	16	66,67	16	66,67	0	0,00
RF50FO4N	16	66,67	17	70,83	1	4,17
RF50FO3T	16	66,67	16	66,67	0	0,00
RF50FO4T	16	66,67	17	70,83	1	4,17
RF50ADN1KM3N	16	66,67	16	66,67	0	0,00
RF50ADN1KM4N	14	58,33	15	62,50	1	4,17
RF50ADN1KM3T	16	66,67	16	66,67	0	0,00
RF50ADN1KM4T	15	62,50	15	62,50	0	0,00
RF50ADN1KM3NT	16	66,67	16	66,67	0	0,00
RF50ADN1KM4NT	16	66,67	16	66,67	0	0,00
RF50ADN1KMB3N	15	62,50	16	66,67	1	4,17
RF50ADN1KMB4N	14	58,33	16	66,67	2	8,33
RF50ADN1KMB3T	16	66,67	16	66,67	0	0,00
RF50ADN1KMB4T	15	62,50	16	66,67	1	4,17
RF50ADN1KMB3NT	15	62,50	16	66,67	1	4,17
RF50ADN1KMB4NT	16	66,67	16	66,67	0	0,00
RF50ADN2KM3N	16	66,67	17	70,83	1	4,17
RF50ADN2KM4N	13	54,17	17	70,83	4	16,67
RF50ADN2KM3T	15	62,50	17	70,83	2	8,33
RF50ADN2KM4T	15	62,50	18	75,00	3	12,50
RF50ADN2KM3NT	16	66,67	17	70,83	1	4,17
RF50ADN2KM4NT	16	66,67	17	70,83	1	4,17
RF50ADN2KMB3N	16	66,67	17	70,83	1	4,17
RF50ADN2KMB4N	14	58,33	17	70,83	3	12,50
RF50ADN2KMB3T	14	58,33	17	70,83	3	12,50
RF50ADN2KMB4T	16	66,67	17	70,83	1	4,17
RF50ADN2KMB3NT	16	66,67	17	70,83	1	4,17
RF50ADN2KMB4NT	16	66,67	16	66,67	0	0,00
RF50ADN3KM3N	16	66,67	17	70,83	1	4,17
RF50ADN3KM4N	15	62,50	17	70,83	2	8,33
RF50ADN3KM3T	16	66,67	17	70,83	1	4,17
RF50ADN3KM4T	14	58,33	17	70,83	3	12,50
RF50ADN3KM3NT	15	62,50	16	66,67	1	4,17
RF50ADN3KM4NT	16	66,67	16	66,67	0	0,00
RF50ADN3KMB3N	16	66,67	17	70,83	1	4,17
RF50ADN3KMB4N	12	50,00	15	62,50	3	12,50
RF50ADN3KMB3T	15	62,50	17	70,83	2	8,33
RF50ADN3KMB4T	15	62,50	17	70,83	2	8,33
RF50ADN3KMB3NT	15	62,50	16	66,67	1	4,17
RF50ADN3KMB4NT	16	66,67	16	66,67	0	0,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão os maiores valores obtidos em cada coluna. Encontra-se, para cada heurística, a quantidade de instâncias (gI) que obteve GAP_{Ini} menor do que MIPGap, a quantidade de instâncias (gM) que obteve GAP menor do que MIPGap, a quantidade de instâncias que obteve GAP menor do que o MIPGap depois da melhoria ($gI - gM$).

C.3 Tempo gasto

Foi realizado o cálculo do tempo gasto médio (Tempo_{med}), desvio padrão (Tempo_{dp}), mínimo (Tempo_{min}), máximo (Tempo_{max}) para cada heurística e o *Gurobi*. As [Tabela 58](#), [Tabela 59](#),

Tabela 60 mostram o tempo gasto, em segundos, para as heurísticas que utilizaram o G, para as heurísticas que utilizaram o RF10 e para as heurísticas que utilizaram o RF50, respectivamente.

Tabela 58 – Tempo gasto pelas heurísticas que utilizaram o G e pelo *Gurobi* para MLCLSP.

Heurísticas	Tempo _{med} (s)	Tempo _{dp} (s)	Tempo _{min} (s)	Tempo _{max} (s)	nInst	$\frac{nInst}{24} \times 100$
GFO3N	613,08	414,80	11,00	900,00	16	66,67
GFO4N	611,42	413,13	17,00	900,00	15	62,50
GFO3T	601,67	414,79	9,00	900,00	15	62,50
GFO4T	603,17	420,41	9,00	900,00	15	62,50
GADN1KM3N	348,67	248,05	3,00	612,00	0	0,00
GADN1KM4N	376,33	240,95	2,00	640,00	0	0,00
GADN1KM3T	386,54	279,76	2,00	690,00	0	0,00
GADN1KM4T	510,71	363,41	5,00	840,00	0	0,00
GADN1KM3NT	292,17	256,27	2,00	701,00	0	0,00
GADN1KM4NT	232,71	170,10	1,00	436,00	0	0,00
GADN1KMB3N	515,92	372,22	5,00	861,00	0	0,00
GADN1KMB4N	556,42	377,60	9,00	856,00	0	0,00
GADN1KMB3T	562,38	406,08	2,00	900,00	4	16,67
GADN1KMB4T	566,75	408,65	4,00	900,00	4	16,67
GADN1KMB3NT	366,25	346,91	3,00	891,00	0	0,00
GADN1KMB4NT	253,25	196,34	1,00	590,00	0	0,00
GADN2KM3N	626,50	395,28	14,00	900,00	15	62,50
GADN2KM4N	724,62	317,07	74,00	900,00	16	66,67
GADN2KM3T	616,88	409,30	16,00	900,00	16	66,67
GADN2KM4T	640,08	377,60	42,00	900,00	16	66,67
GADN2KM3NT	627,08	399,66	46,00	900,00	16	66,67
GADN2KM4NT	628,92	394,16	52,00	900,00	16	66,67
GADN2KMB3N	623,67	400,86	32,00	900,00	16	66,67
GADN2KMB4N	648,33	365,23	87,00	900,00	16	66,67
GADN2KMB3T	627,88	394,34	38,00	900,00	16	66,67
GADN2KMB4T	633,33	387,20	56,00	900,00	16	66,67
GADN2KMB3NT	629,54	402,47	49,00	900,00	16	66,67
GADN2KMB4NT	622,96	402,67	39,00	900,00	16	66,67
GADN3KM3N	575,33	407,95	3,00	900,00	1	4,17
GADN3KM4N	615,62	390,02	41,00	900,00	5	20,83
GADN3KM3T	604,83	424,75	5,00	900,00	11	45,83
GADN3KM4T	614,29	412,98	15,00	900,00	16	66,67
GADN3KM3NT	476,12	375,88	5,00	900,00	8	33,33
GADN3KM4NT	496,54	375,50	4,00	900,00	6	25,00
GADN3KMB3N	425,29	328,32	7,00	900,00	2	8,33
GADN3KMB4N	552,71	365,14	41,00	900,00	6	25,00
GADN3KMB3T	604,50	427,39	4,00	900,00	16	66,67
GADN3KMB4T	607,88	422,77	10,00	900,00	16	66,67
GADN3KMB3NT	415,88	380,76	9,00	900,00	8	33,33
GADN3KMB4NT	313,04	298,50	7,00	900,00	1	4,17
<i>Gurobi</i>	900,00	0,00	900,00	900,00	24	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 59 – Tempo gasto pelas heurísticas que utilizaram o RF10 e pelo *Gurobi* para MLCLSP.

Heurísticas	Tempo _{med} (s)	Tempo _{dp} (s)	Tempo _{min} (s)	Tempo _{max} (s)	nInst	$\frac{nInst}{24} \times 100$
RF10FO3N	607,67	422,33	15,00	900,00	16	66,67
RF10FO4N	606,25	424,37	12,00	900,00	16	66,67
RF10FO3T	600,62	421,73	12,00	900,00	15	62,50
RF10FO4T	603,00	429,06	6,00	900,00	16	66,67
RF10ADN1KM3N	270,58	193,78	9,00	597,00	0	0,00
RF10ADN1KM4N	362,38	212,44	9,00	683,00	0	0,00
RF10ADN1KM3T	457,62	327,02	9,00	867,00	0	0,00
RF10ADN1KM4T	556,78	397,60	14,00	900,00	3	12,50
RF10ADN1KM3NT	283,17	271,70	7,00	752,00	0	0,00
RF10ADN1KM4NT	195,96	154,36	4,00	493,00	0	0,00
RF10ADN1KMB3N	217,74	229,64	6,00	821,00	0	0,00
RF10ADN1KMB4N	301,83	286,65	7,00	876,00	0	0,00
RF10ADN1KMB3T	562,38	400,67	9,00	900,00	6	25,00
RF10ADN1KMB4T	578,92	406,21	14,00	900,00	6	25,00
RF10ADN1KMB3NT	267,33	286,60	7,00	882,00	0	0,00
RF10ADN1KMB4NT	204,08	210,04	9,00	788,00	0	0,00
RF10ADN2KM3N	622,88	401,47	19,00	900,00	16	66,67
RF10ADN2KM4N	689,29	330,84	22,00	900,00	15	62,50
RF10ADN2KM3T	611,92	416,30	16,00	900,00	16	66,67
RF10ADN2KM4T	617,79	408,64	14,00	900,00	16	66,67
RF10ADN2KM3NT	603,12	430,14	5,00	900,00	16	66,67
RF10ADN2KM4NT	610,00	420,41	19,00	900,00	16	66,67
RF10ADN2KMB3N	605,21	408,40	25,00	900,00	14	58,33
RF10ADN2KMB4N	577,42	380,69	20,00	900,00	12	50,00
RF10ADN2KMB3T	612,21	416,40	20,00	900,00	16	66,67
RF10ADN2KMB4T	591,61	432,06	5,00	900,00	15	62,50
RF10ADN2KMB3NT	608,58	427,70	10,00	900,00	16	66,67
RF10ADN2KMB4NT	560,22	417,51	7,00	900,00	12	50,00
RF10ADN3KM3N	583,75	399,63	15,00	900,00	6	25,00
RF10ADN3KM4N	611,75	392,85	23,00	900,00	6	25,00
RF10ADN3KM3T	593,43	426,43	17,00	900,00	9	37,50
RF10ADN3KM4T	596,13	425,13	15,00	900,00	12	50,00
RF10ADN3KM3NT	451,92	372,15	10,00	900,00	7	29,17
RF10ADN3KM4NT	408,04	347,30	6,00	900,00	3	12,50
RF10ADN3KMB3N	388,43	360,01	21,00	900,00	5	20,83
RF10ADN3KMB4N	495,26	368,21	37,00	900,00	6	25,00
RF10ADN3KMB3T	606,12	425,01	9,00	900,00	15	62,50
RF10ADN3KMB4T	609,25	420,78	18,00	900,00	16	66,67
RF10ADN3KMB3NT	330,73	371,83	7,00	900,00	6	25,00
RF10ADN3KMB4NT	285,38	328,80	3,00	900,00	3	12,50
<i>Gurobi</i>	900,00	0,00	900,00	900,00	24	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

Tabela 60 – Tempo gasto pelas heurísticas que utilizaram o RF50 e pelo *Gurobi* para MLCLSP.

Heurísticas	Tempo _{med} (s)	Tempo _{dp} (s)	Tempo _{min} (s)	Tempo _{max} (s)	nInst	$\frac{nInst}{24} \times 100$
RF50FO3N	605,83	424,97	12,00	900,00	16	66,67
RF50FO4N	604,04	427,56	8,00	900,00	16	66,67
RF50FO3T	607,08	423,17	15,00	900,00	16	66,67
RF50FO4T	606,79	423,59	16,00	900,00	16	66,67
RF50ADN1KM3N	428,33	294,19	13,00	788,00	0	0,00
RF50ADN1KM4N	485,91	282,86	21,00	753,00	0	0,00
RF50ADN1KM3T	513,38	365,58	6,00	853,00	0	0,00
RF50ADN1KM4T	587,62	401,46	18,00	900,00	2	8,33
RF50ADN1KM3NT	436,58	336,96	4,00	900,00	1	4,17
RF50ADN1KM4NT	394,04	288,47	4,00	781,00	0	0,00
RF50ADN1KMB3N	435,21	316,59	14,00	900,00	1	4,17
RF50ADN1KMB4N	485,58	321,64	25,00	900,00	1	4,17
RF50ADN1KMB3T	586,92	414,70	10,00	900,00	6	25,00
RF50ADN1KMB4T	595,42	414,19	16,00	900,00	7	29,17
RF50ADN1KMB3NT	474,88	378,80	5,00	900,00	6	25,00
RF50ADN1KMB4NT	349,75	257,59	3,00	710,00	0	0,00
RF50ADN2KM3N	619,62	404,24	17,00	900,00	14	58,33
RF50ADN2KM4N	688,92	332,16	22,00	900,00	16	66,67
RF50ADN2KM3T	620,67	403,87	31,00	900,00	16	66,67
RF50ADN2KM4T	636,71	383,01	43,00	900,00	16	66,67
RF50ADN2KM3NT	608,79	425,41	13,00	900,00	16	66,67
RF50ADN2KM4NT	603,33	429,57	6,00	900,00	16	66,67
RF50ADN2KMB3N	592,33	412,01	16,00	900,00	13	54,17
RF50ADN2KMB4N	653,04	361,88	38,00	900,00	15	62,50
RF50ADN2KMB3T	613,00	415,28	19,00	900,00	16	66,67
RF50ADN2KMB4T	615,17	411,81	18,00	900,00	16	66,67
RF50ADN2KMB3NT	610,71	425,27	13,00	900,00	16	66,67
RF50ADN2KMB4NT	589,88	420,66	6,00	900,00	13	54,17
RF50ADN3KM3N	583,50	404,92	10,00	900,00	1	4,17
RF50ADN3KM4N	611,21	400,93	19,00	900,00	5	20,83
RF50ADN3KM3T	605,29	422,52	16,00	900,00	10	41,67
RF50ADN3KM4T	614,08	413,37	23,00	900,00	16	66,67
RF50ADN3KM3NT	570,62	406,94	11,00	900,00	8	33,33
RF50ADN3KM4NT	522,38	384,21	5,00	900,00	5	20,83
RF50ADN3KMB3N	466,62	352,31	13,00	900,00	2	8,33
RF50ADN3KMB4N	527,59	372,19	31,00	900,00	4	16,67
RF50ADN3KMB3T	608,83	421,37	18,00	900,00	16	66,67
RF50ADN3KMB4T	610,00	419,34	20,00	900,00	16	66,67
RF50ADN3KMB3NT	513,00	384,16	10,00	900,00	8	33,33
RF50ADN3KMB4NT	432,92	341,77	4,00	900,00	3	12,50
<i>Gurobi</i>	900,00	0,00	900,00	900,00	24	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna.

C.4 Número de iterações

Foi realizado o cálculo do número de iterações médio ($nIter_{med}$), desvio padrão ($nIter_{dp}$), mínimo ($nIter_{min}$), máximo ($nIter_{max}$) e a porcentagem ($\frac{solI}{24} \times 100\%$). As [Tabela 61](#), [Tabela 62](#), [Tabela 63](#) apresentam o número de iterações para as heurísticas que utilizaram o G, para as heurísticas que utilizaram o RF10 e para as heurísticas que utilizaram o RF50, respectivamente. Desconsideramos as heurísticas que utilizaram o ADN1, pois este sempre realiza duas iterações.

Tabela 61 – Número de iterações realizadas pelas heurísticas que utilizaram o G para MLCLSP.

Heurísticas	nIter _{med} (s)	nIter _{dp} (s)	nIter _{min} (s)	nIter _{max} (s)	$\frac{solI}{24} \times 100$
GFO3N	5,08	3,71	2,00	18,00	100,00
GFO4N	4,71	3,34	2,00	16,00	100,00
GFO3T	5,08	3,66	2,00	18,00	100,00
GFO4T	4,79	3,36	2,00	16,00	100,00
GADN2KM3N	16,21	5,54	9,00	27,00	100,00
GADN2KM4N	12,67	3,80	8,00	22,00	100,00
GADN2KM3T	14,83	10,09	6,00	49,00	100,00
GADN2KM4T	12,08	10,12	4,00	52,00	100,00
GADN2KM3NT	20,58	16,95	4,00	59,00	100,00
GADN2KM4NT	19,83	16,12	6,00	63,00	100,00
GADN2KMB3N	8,67	6,00	4,00	30,00	100,00
GADN2KMB4N	8,58	4,37	5,00	20,00	100,00
GADN2KMB3T	11,17	13,79	2,00	63,00	100,00
GADN2KMB4T	10,04	11,65	2,00	50,00	100,00
GADN2KMB3NT	19,71	17,35	4,00	67,00	100,00
GADN2KMB4NT	17,12	15,58	4,00	59,00	100,00
GADN3KM3N	6,67	2,04	3,00	10,00	100,00
GADN3KM4N	6,25	1,82	4,00	12,00	100,00
GADN3KM3T	5,33	3,21	3,00	18,00	100,00
GADN3KM4T	4,00	2,80	2,00	14,00	100,00
GADN3KM3NT	6,00	2,34	3,00	12,00	100,00
GADN3KM4NT	6,67	2,55	3,00	15,00	100,00
GADN3KMB3N	6,33	2,14	3,00	12,00	100,00
GADN3KMB4N	5,88	1,78	4,00	11,00	100,00
GADN3KMB3T	3,62	3,65	1,00	17,00	100,00
GADN3KMB4T	3,42	3,35	1,00	17,00	100,00
GADN3KMB3NT	6,00	2,41	3,00	14,00	100,00
GADN3KMB4NT	8,46	5,74	4,00	34,00	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna. A coluna $\frac{solI}{24} \times 100$ indica a percentagem de instâncias que obteve soluções factíveis.

Tabela 62 – Número de iterações realizadas pelas heurísticas que utilizaram o RF10 para MLCLSP.

Heurísticas	nIter _{med} (s)	nIter _{dp} (s)	nIter _{min} (s)	nIter _{max} (s)	$\frac{solI}{24} \times 100$
RF10FO3N	3,38	1,35	2,00	5,00	100,00
RF10FO4N	3,50	1,29	2,00	5,00	100,00
RF10FO3T	3,38	1,38	2,00	6,00	100,00
RF10FO4T	3,50	1,29	2,00	5,00	100,00
RF10ADN2KM3N	10,00	4,57	4,00	18,00	100,00
RF10ADN2KM4N	9,46	3,61	3,00	15,00	100,00
RF10ADN2KM3T	9,29	3,94	4,00	20,00	100,00
RF10ADN2KM4T	6,58	3,26	3,00	15,00	100,00
RF10ADN2KM3NT	18,00	13,73	3,00	40,00	100,00
RF10ADN2KM4NT	10,62	4,47	4,00	19,00	100,00
RF10ADN2KMB3N	11,21	8,30	4,00	31,00	100,00
RF10ADN2KMB4N	13,29	9,04	4,00	32,00	100,00
RF10ADN2KMB3T	5,67	3,70	2,00	17,00	100,00
RF10ADN2KMB4T	5,70	3,66	2,00	14,00	95,83
RF10ADN2KMB3NT	12,96	13,08	2,00	43,00	100,00
RF10ADN2KMB4NT	25,09	24,72	3,00	81,00	95,83
RF10ADN3KM3N	5,17	1,76	2,00	8,00	100,00
RF10ADN3KM4N	4,67	1,66	2,00	8,00	100,00
RF10ADN3KM3T	4,09	1,04	3,00	6,00	95,83
RF10ADN3KM4T	3,22	0,60	2,00	4,00	95,83
RF10ADN3KM3NT	4,42	1,69	2,00	8,00	100,00
RF10ADN3KM4NT	5,25	2,31	2,00	11,00	100,00
RF10ADN3KMB3N	5,17	2,12	3,00	12,00	95,83
RF10ADN3KMB4N	4,74	2,16	2,00	8,00	95,83
RF10ADN3KMB3T	2,79	1,25	1,00	6,00	100,00
RF10ADN3KMB4T	2,38	1,06	1,00	5,00	100,00
RF10ADN3KMB3NT	4,45	1,74	2,00	8,00	91,67
RF10ADN3KMB4NT	5,21	2,72	2,00	11,00	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna. A coluna $\frac{solI}{24} \times 100$ indica a percentagem de instâncias que obteve soluções factíveis.

Tabela 63 – Número de iterações realizadas pelas heurísticas que utilizaram o RF50 para MLCLSP.

Heurísticas	nIter _{med} (s)	nIter _{dp} (s)	nIter _{min} (s)	nIter _{max} (s)	$\frac{solI}{24} \times 100$
RF50FO3N	2,67	0,76	1,00	4,00	100,00
RF50FO4N	2,83	0,70	2,00	4,00	100,00
RF50FO3T	2,62	0,82	1,00	4,00	100,00
RF50FO4T	2,83	0,70	2,00	4,00	100,00
RF50ADN2KM3N	7,96	2,37	4,00	13,00	100,00
RF50ADN2KM4N	6,29	1,97	3,00	12,00	100,00
RF50ADN2KM3T	6,17	4,98	2,00	20,00	100,00
RF50ADN2KM4T	4,62	4,22	1,00	15,00	100,00
RF50ADN2KM3NT	8,46	5,52	2,00	16,00	100,00
RF50ADN2KM4NT	9,12	3,57	4,00	17,00	100,00
RF50ADN2KMB3N	11,21	8,28	2,00	34,00	100,00
RF50ADN2KMB4N	7,50	5,00	2,00	22,00	100,00
RF50ADN2KMB3T	4,33	4,27	1,00	17,00	100,00
RF50ADN2KMB4T	4,42	4,13	1,00	14,00	100,00
RF50ADN2KMB3NT	7,21	5,49	1,00	19,00	100,00
RF50ADN2KMB4NT	18,17	15,85	3,00	48,00	100,00
RF50ADN3KM3N	5,08	1,67	2,00	9,00	100,00
RF50ADN3KM4N	4,25	1,26	2,00	7,00	100,00
RF50ADN3KM3T	3,88	0,90	2,00	6,00	100,00
RF50ADN3KM4T	2,79	0,72	1,00	4,00	100,00
RF50ADN3KM3NT	3,33	1,63	2,00	7,00	100,00
RF50ADN3KM4NT	4,67	1,76	2,00	8,00	100,00
RF50ADN3KMB3N	4,88	1,80	3,00	9,00	100,00
RF50ADN3KMB4N	4,09	1,63	2,00	7,00	91,67
RF50ADN3KMB3T	2,46	1,32	1,00	6,00	100,00
RF50ADN3KMB4T	2,29	1,04	1,00	5,00	100,00
RF50ADN3KMB3NT	3,08	1,53	1,00	7,00	100,00
RF50ADN3KMB4NT	4,46	1,96	2,00	8,00	100,00

Fonte: Elaborada pela autora.

Nota – Em negrito, estão destacados os menores valores dentre as heurísticas em cada coluna. A coluna $\frac{solI}{24} \times 100$ indica a percentagem de instâncias que obteve soluções factíveis.