



Programa de Pós-Graduação em

Computação Aplicada

Mestrado/Doutorado Acadêmico

Luan Carlos Nesi

Modelo hiper-heurístico baseado em
Redes de Autômatos Estocásticos com Aprendizado

São Leopoldo, 2020

UNIVERSIDADE DO VALE DO RIO DOS SINOS — UNISINOS
UNIDADE ACADÊMICA DE PESQUISA E PÓS-GRADUAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO APLICADA
NÍVEL DOUTORADO

LUAN CARLOS NESI

**MODELO HIPER-HEURÍSTICO BASEADO EM
REDES DE AUTÔMATOS ESTOCÁSTICOS COM APRENDIZADO**

São Leopoldo
2020

Luan Carlos Nesi

**MODELO HIPER-HEURÍSTICO BASEADO EM
REDES DE AUTÔMATOS ESTOCÁSTICOS COM APRENDIZADO**

Tese de doutorado apresentada como requisito
parcial para a obtenção do título de Doutor
pelo Programa de Pós-Graduação em
Computação Aplicada da Universidade do Vale
do Rio dos Sinos — UNISINOS

Orientador:
Prof. Dr. Rodrigo da Rosa Righi

São Leopoldo
2020

N459m Nesi, Luan Carlos.
Modelo hiper-heurístico baseado em redes de autômatos estocásticos com aprendizado / Luan Carlos Nesi. – 2020.
217 f. : il. ; 30 cm.

Tese (doutorado) – Universidade do Vale do Rio dos Sinos, Programa de Pós-Graduação em Computação Aplicada, 2020.
“Orientador: Prof. Dr. Rodrigo da Rosa Righi.”

1. Hiper-heurística. 2. Meta-heurística. 3. Metaespaço.
4. Redes de autômatos estocásticos de aprendizagem. I.
Título.

CDU 004.732

Dados Internacionais de Catalogação na Publicação (CIP)
(Bibliotecária: Amanda Schuster – CRB 10/2517)

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001.

Dedico esta tese aos meus grandes heróis, meus pais, Rejane e Ademar.

Com carinho, Luan.

AGRADECIMENTOS

Este trabalho só foi possível graças a diversos apoiadores. Gostaria de transmitir meus sinceros agradecimentos...

... aos meus Pais Rejane e Ademir,
meus alicerces, pelo apoio incondicional e
por acreditarem neste trabalho muitas vezes mais do que eu mesmo;

... a minha amada companheira Daniela,
pela paciência, por segurar a barra e não soltar a minha mão;

... a minha pequenina Vênus, fiel companheira,
por aguentar meus piores momentos e me aquecer nas noites frias de escrita;

... aos amigos Cláudio e Ângelo,
pela parceria envolvida e por aguardarem pacientemente o fim deste ciclo;

... ao amigo Márcio,
por dividirmos essa etapa de vida juntos;

... a minha grande amiga Lorena, por tudo o que aprendemos e fomos juntos,
pelo respeito e admiração que levamos conosco;

... aos meus orientadores nesta jornada,
Prof. Dr. Arthur Tórgo Gómez, Prof. Dra. Marta Becker Villamil e
Prof. Dr. Rodrigo da Rosa Righi pelos conselhos e envolvimento;

... à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (Capes) e
ao Programa de Pós-Graduação em Computação Aplicada,
pelo apoio aos meus interesses de pesquisa e
pela bolsa de estudos que tornou possível a realização do Curso de Doutorado;

... a Unisinos, minha *alma mater*,
pelas oportunidades e aprendizados ao longo destes 11 anos de casa;

... a todos os demais colegas acadêmicos, amigos, professores que
enriqueceram meu aprendizado e de uma forma ou outra contribuíram
com opiniões, críticas, questionamentos sobre esta pesquisa;

... a todos citados e não citados (desculpem-me!) que dividiram
um café e um instante de vida comigo, só tenho a lhes agradecer;

... por fim, agradeço ao Luan que iniciou esta jornada,
aos muitos que me tornei nestas andanças e ao Luan que finda esta trajetória!

*“Os grandes pensadores são um tanto sísmicos;
não evoluem, mas avançam por crises, por abalos.”
(Gilles Deleuze, 1990)*

RESUMO

Vivencia-se, na última década, um aumento substancial de trabalhos sobre meta-heurísticas fundamentadas em metáforas. Ainda que estes trabalhos tenham produzido soluções eficazes dentro de suas propostas, a comunidade científica critica a validade dos embasamentos metafóricos, como simples processos de camuflagem. Neste sentido, a pesquisa investigou o processo de movimentação pelo espaço de busca realizado pelos mecanismos heurísticos, livre de abstrações empregadas em técnicas. Ou, nos termos conceituais propostos pela tese, desenvolvemos um modelo hiper-heurístico baseado em redes de autômatos estocásticos de aprendizagem, para seleção e parametrização de heurísticas de baixo nível. Para tanto, a pesquisa partiu de uma revisão epistêmica identificando perspectivas modelares, com um apanhado sobre conceitos e características. Após, avaliou-se oito meta-heurísticas em prol da identificação dos mecanismos heurísticos, ou seja, os processos conducentes da busca. Essa identificação aliada à teoria de Redes de Autômatos Estocásticos com se consolidam na figura do metaespaço, uma parte da arquitetura da hiper-heurística proposta no trabalho, nomeada modelo H2-SLAN. Conseqüente, apresentou-se o processo de experimentação e validação do modelo proposto. A técnica de validação tomou por base a resolução de instâncias do problema do caixeiro viajante, por meio da aplicação dos modelos canônicos das meta-heurísticas. Este processo serviu para a construção de uma base comparativa de resultados, considerando o Grão Computacional como unidade de medida de desempenho quantitativo e qualitativo. Tendo estabelecido os padrões-alvo de execução, a hiper-heurística foi aplicada sobre o mesmo contexto, visando obter melhores desempenhos. Os resultados obtidos pelo modelo H2-SLAN foram 3% melhores do que os obtidos pelos algoritmos canônicos, e realizaram em média 45% menos avaliações de Função Objetivo. Esta experimentação mostrou que a Grão Computacional é uma alternativa para a medição da qualidade da busca heurística. Desta forma, pode-se afirmar que os resultados obtidos pelo modelo atingiram as expectativas, indo ao encontro do objetivo da pesquisa. Como decorrência deste trabalho, obteve-se um sistema capaz de selecionar e parametrizar heurísticas de baixo nível, com capacidade de aprendizado dos movimentos heurísticos empregados pelo modelo.

Palavras-chave: Hiper-heurística. Meta-heurística. Metaespaço. Redes de Autômatos Estocásticos de Aprendizagem.

ABSTRACT

In the last decade, there has been a substantial increase in works about metaheuristics based on metaphors. Although these works have produced effective solutions within their proposals, the scientific community criticizes the validity of metaphorical bases, such as simple camouflage processes. In this sense, the research investigated the process of moving through the search space performed by heuristic mechanisms, free of abstractions employed in techniques. Or, in the conceptual terms proposed by the thesis, we developed a hyper-heuristic model based on stochastic learning automata networks, for choice and parametrization of low-level heuristics. To this end, the research started from an epistemic review finding model perspectives, with a catch on concepts and characteristics. After that, we evaluated eight metaheuristics in favor of the identification of heuristic mechanisms, that is, the processes leading to the search. This identification allied to the theory of stochastic learning automata networks guided the construction of their representations. These representations are consolidated in the figure of the meta-space, a part of the architecture of the Hyper-heuristic proposed in the work, named H2-SLAN model. Therefore, we present the process of experimentation and validation of the proposed model. The validation technique was based on solving instances of the traveling salesman problem, through the application of the canonical metaheuristic models. This process served to construct a comparative basis of results, considering the Computational Grain as a unit of measure of quantitative and qualitative performance. Having established the target patterns of execution, we apply the hyper-heuristic under the same context, aiming to obtain better performances. The results obtained by the H2-SLAN model were 3 % better than those obtained by the canonical algorithms and carried out on average 45% less objective function assessments. This experimentation showed that computational grain is an alternative for measuring the quality of the heuristic search. Thus, we can state that the results obtained by the model reached expectations, going to meet the goal of the research. As a result of this work, we obtained a system capable of selecting the parametrization of low-level heuristics, with the ability to learn the heuristics movements employed by the model.

Keywords: Hyper-heuristic. Metaheuristic. Meta-space. Stochastic Learning Automata Network.

LISTA DE FIGURAS

1	Metáforas adotadas por pesquisadores para projetar novas metaheurísticas	16
2	Espectro de diversificação e intensificação	30
3	Ortogonalidade entre a Gênese e a Completude	34
4	<i>Template</i> de alto-nível da meta-heurística baseada em uma solução	36
5	<i>Template</i> de alto-nível da meta-heurística populacional	37
6	<i>Template</i> de alto-nível para meta-heurísticas	38
7	Pseudocódigo do Algoritmo Genético	46
8	Pseudocódigo do Arrefecimento Simulado	48
9	Pseudocódigo da Busca Dispersa	50
10	Pseudocódigo da Busca em Vizinhança Variável	52
11	Pseudocódigo da Busca Local Iterada	54
12	Pseudocódigo da Busca Tabu	57
13	Pseudocódigo da GRASP	59
14	Otimização por Colônia de Formigas	61
15	<i>Framework</i> Hiper-heurístico	64
16	Classificação das abordagens hiper-heurísticas	65
17	Visão geral do Modelo H2-SLAN	73
18	Arquitetura do Agente de Aprendizagem	74
19	Autômato de Aprendizagem	77
20	Autômato $\mathcal{A}^{(30)}$ - Hiper-heurística	84
21	Arquitetura do Módulo Otimizador	85
22	Arquitetura do Bloco de Interface	86
23	Fluxo de operação da arquitetura do modelo H2-SLAN	90
24	Algoritmo H2-SLAN	91
25	Validação dos Experimentos	95
26	Espaço do Grão Computacional	100
27	Indicador de qualidade da solução	102
28	Cores dos espectros analíticos	102
29	Frequências dos mecanismos heurísticos	153
30	Autômato $\mathcal{A}^{(1)}$ - Sistema Meta-heurístico Geral	180
31	Autômato $\mathcal{A}^{(2)}$ - Critério de parada	181
32	Autômato $\mathcal{A}^{(3)}$ - Fase Geracional	182
33	Autômato $\mathcal{A}^{(4)}$ - Fase Substitutiva	183
34	Autômato $\mathcal{A}^{(5)}$ - Algoritmo Genético	184
35	Autômato $\mathcal{A}^{(6)}$ - Seleção	185
36	Autômato $\mathcal{A}^{(7)}$ - Roleta	186
37	Autômato $\mathcal{A}^{(8)}$ - Torneio	187
38	Autômato $\mathcal{A}^{(9)}$ - Cruzamento	188
39	Autômato $\mathcal{A}^{(10)}$ - Mutação	189
40	Autômato $\mathcal{A}^{(11)}$ - Swap	190
41	Autômato $\mathcal{A}^{(12)}$ - Inversão de Bit	191
42	Autômato $\mathcal{A}^{(13)}$ - Arrefecimento Simulado	192
43	Autômato $\mathcal{A}^{(14)}$ - Temperatura Inicial	193
44	Autômato $\mathcal{A}^{(15)}$ - Redução da Temperatura	194
45	Autômato $\mathcal{A}^{(16)}$ - Busca Dispersa	195
46	Autômato $\mathcal{A}^{(17)}$ - Tamanho Conjunto de Diversificação	196

47	Autômato $\mathcal{A}^{(18)}$ - Tamanho Conjunto de Referência	197
48	Autômato $\mathcal{A}^{(19)}$ - Busca Local	198
49	Autômato $\mathcal{A}^{(20)}$ - Busca em Vizinhança Variável	199
50	Autômato $\mathcal{A}^{(21)}$ - Vizinhanças	200
51	Autômato $\mathcal{A}^{(22)}$ - Busca Local Iterada	201
52	Autômato $\mathcal{A}^{(23)}$ - Busca Tabu	202
53	Autômato $\mathcal{A}^{(24)}$ - Tamanho Lista Tabu	203
54	Autômato $\mathcal{A}^{(25)}$ - GRASP	204
55	Autômato $\mathcal{A}^{(26)}$ - Construção Aleatória	205
56	Autômato $\mathcal{A}^{(27)}$ - Otimização por Colônia de Formigas	206
57	Autômato $\mathcal{A}^{(28)}$ - Atração do Feromônio	207
58	Autômato $\mathcal{A}^{(29)}$ - Evaporação do Feromônio	208
59	Autômato $\mathcal{A}^{(30)}$ - Hiper-heurística	209
60	Lente espectral da meta-heurística Algoritmos Genéticos	210
61	Lente espectral da meta-heurística Arrefecimento Simulado	211
62	Lente espectral da meta-heurística Busca Dispersa	212
63	Lente espectral da meta-heurística Busca em Vizinhança Variável	213
64	Lente espectral da meta-heurística Busca Local Iterada	214
65	Lente espectral da meta-heurística Busca Tabu	215
66	Lente espectral da meta-heurística GRASP	216
67	Lente espectral da meta-heurística Otimização por Colônia de Formigas	217

LISTA DE TABELAS

1	Publicações por ano	17
4	Composição do autômato $\mathcal{A}^{(30)}$ - Hiper-heurística	82
2	Delimitação do Alfabeto	83
3	Legenda de expressões regulares	84
5	Parâmetros e métodos aplicados aos experimentos	96
6	Níveis dos parâmetros e métodos aplicados aos experimentos	98
7	Modelo da tabela de experimentos	101
8	Resultados dos experimentos do AG para a instância berlin52	103
9	Resultados dos experimentos do SA para a instância berlin52	104
10	Resultados dos experimentos do SS para a instância berlin52	105
11	Resultados dos experimentos do VNS para a instância berlin52	106
12	Resultados dos experimentos do ILS para a instância berlin52	107
13	Resultados dos experimentos do BT para a instância berlin52	108
14	Resultados dos experimentos do GRASP para a instância berlin52	109
15	Resultados dos experimentos do ACO para a instância berlin52	110
16	Resultados dos experimentos consolidados para a instância berlin52	112
17	Resultados dos experimentos do AG para a instância eil101	113
18	Resultados dos experimentos do SA para a instância eil101	114
19	Resultados dos experimentos do SS para a instância eil101	115
20	Resultados dos experimentos do VNS para a instância eil101	116
21	Resultados dos experimentos do ILS para a instância eil101	117
22	Resultados dos experimentos do BT para a instância eil101	118
23	Resultados dos experimentos do GRASP para a instância eil101	119
24	Resultados dos experimentos do ACO para a instância eil101	120
25	Resultados dos experimentos consolidados para a instância eil101	122
26	Resultados dos experimentos do AG para a instância d493	123
27	Resultados dos experimentos do SA para a instância d493	124
28	Resultados dos experimentos do SS para a instância d493	125
29	Resultados dos experimentos do VNS para a instância d493	126
30	Resultados dos experimentos do ILS para a instância d493	127
31	Resultados dos experimentos do BT para a instância d493	128
32	Resultados dos experimentos do GRASP para a instância d493	129
33	Resultados dos experimentos do ACO para a instância d493	130
34	Resultados dos experimentos consolidados para a instância d493	132
35	Resultados dos experimentos do AG para a instância u1060	133
36	Resultados dos experimentos do SA para a instância u1060	134
37	Resultados dos experimentos do SS para a instância u1060	135
38	Resultados dos experimentos do VNS para a instância u1060	136
39	Resultados dos experimentos do ILS para a instância u1060	137
40	Resultados dos experimentos do BT para a instância u1060	138
41	Resultados dos experimentos do GRASP para a instância u1060	139
42	Resultados dos experimentos do ACO para a instância u1060	140
43	Resultados dos experimentos consolidados para a instância u1060	142
44	Resultados dos experimentos do AG para a instância rl1889	143
45	Resultados dos experimentos do SA para a instância rl1889	144
46	Resultados dos experimentos do SS para a instância rl1889	145

47	Resultados dos experimentos do VNS para a instância rl1889	146
48	Resultados dos experimentos do ILS para a instância rl1889	147
49	Resultados dos experimentos do BT para a instância rl1889	148
50	Resultados dos experimentos do GRASP para a instância rl1889	149
51	Resultados dos experimentos do ACO para a instância rl1889	150
52	Resultados dos experimentos consolidados para a instância rl1889	152
53	Melhores parametrizações	154
54	Composição do autômato $\mathcal{A}^{(1)}$ - Sistema Meta-heurístico Geral	180
55	Composição do autômato $\mathcal{A}^{(2)}$ - Critério de parada	181
56	Composição do autômato $\mathcal{A}^{(3)}$ - Fase Geracional	182
57	Composição do autômato $\mathcal{A}^{(4)}$ - Fase Substitutiva	183
58	Composição do autômato $\mathcal{A}^{(5)}$ - Algoritmo Genético	184
59	Composição do autômato $\mathcal{A}^{(6)}$ - Seleção	185
60	Composição do autômato $\mathcal{A}^{(7)}$ - Roleta	186
61	Composição do autômato $\mathcal{A}^{(8)}$ - Torneio	187
62	Composição do autômato $\mathcal{A}^{(9)}$ - Cruzamento	188
63	Composição do autômato $\mathcal{A}^{(10)}$ - Mutação	189
64	Composição do autômato $\mathcal{A}^{(11)}$ - Swap	190
65	Composição do autômato $\mathcal{A}^{(12)}$ - Inversão de Bit	191
66	Composição do autômato $\mathcal{A}^{(13)}$ - Arrefecimento Simulado	192
67	Composição do autômato $\mathcal{A}^{(14)}$ - Temperatura Inicial	193
68	Composição do autômato $\mathcal{A}^{(15)}$ - Redução da Temperatura	194
69	Composição do autômato $\mathcal{A}^{(16)}$ - Busca Dispersa	195
70	Composição do autômato $\mathcal{A}^{(17)}$ - Tamanho Conjunto de Diversificação	196
71	Composição do autômato $\mathcal{A}^{(18)}$ - Tamanho Conjunto de Referência	197
72	Composição do autômato $\mathcal{A}^{(19)}$ - Busca Local	198
73	Composição do autômato $\mathcal{A}^{(20)}$ - Busca em Vizinhança Variável	199
74	Composição do autômato $\mathcal{A}^{(21)}$ - Vizinhanças	200
75	Composição do autômato $\mathcal{A}^{(22)}$ - Busca Local Iterada	201
76	Composição do autômato $\mathcal{A}^{(23)}$ - Busca Tabu	202
77	Composição do autômato $\mathcal{A}^{(24)}$ - Tamanho Lista Tabu	203
78	Composição do autômato $\mathcal{A}^{(25)}$ - GRASP	204
79	Composição do autômato $\mathcal{A}^{(26)}$ - Construção Aleatória	205
80	Composição do autômato $\mathcal{A}^{(27)}$ - Otimização por Colônia de Formigas	206
81	Composição do autômato $\mathcal{A}^{(28)}$ - Atração do Feromônio	207
82	Composição do autômato $\mathcal{A}^{(29)}$ - Evaporação do Feromônio	208
83	Composição do autômato $\mathcal{A}^{(30)}$ - Hiper-heurística	209

LISTA DE SIGLAS

ACO	<i>Ant Colony Optimization</i>
AG	Algoritmo Genético
BKS	<i>Best Known Solution</i>
BT	Busca Tabu
DP	Desvio Padrão
FO	Função Objetivo
GC	Grão Computacional
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
ILS	<i>Iterated Local Search</i>
SA	<i>Simulated Annealing</i>
SAN	<i>Stochastic Automata Networks</i>
SS	<i>Scatter Search</i>
TSP	<i>Travelling Salesman Problem</i>
VNS	<i>Variable Neighborhood Search</i>

SUMÁRIO

1	INTRODUÇÃO	15
1.1	Objetivos	19
1.2	Declaração do Problema	20
1.3	Abordagem de Pesquisa e Limitações	20
1.4	Organização do trabalho	24
2	PERSPECTIVAS MODELARES	26
2.1	Princípios, conceitos e origens das heurísticas	27
2.2	Navegabilidade	31
2.3	Completeness	32
2.4	Gênese	34
2.5	Universalidade	35
2.6	Persistência	38
2.7	Contiguidade	40
2.8	Aptidão	41
2.9	Considerações sobre o capítulo	42
3	MECANISMOS HEURÍSTICOS	44
3.1	Algoritmos Genéticos	44
3.2	Arrefecimento Simulado	47
3.3	Busca Dispersa	49
3.4	Busca em Vizinhança Variável	52
3.5	Busca Local Iterada	53
3.6	Busca Tabu	55
3.7	<i>GRASP - Greedy Randomized Adaptive Search Procedure</i>	57
3.8	Otimização por Colônia de Formigas	60
3.9	Hiper-Heurísticas	62
3.9.1	Barreira de Domínio	64
3.9.2	Classificação das hiper-heurísticas	65
3.10	Considerações sobre o capítulo	69
4	MODELO	72
4.1	Visão Geral	73
4.2	Arquitetura do Agente de Aprendizagem	74
4.2.1	Sistema de aprendizado	76
4.2.2	Formalização do Metaespaço	79
4.2.3	Modelagem do Metaespaço	81
4.3	Arquitetura do módulo Otimizador	84
4.4	Arquitetura do bloco de interface	86
4.5	Retorno analítico	86
4.6	Considerações sobre o capítulo	89
5	PANORAMAS DE EXPERIMENTAÇÃO	93
5.1	Planejamento dos Experimentos	94
5.2	Validação dos experimentos	95
5.3	Análise dos Experimentos	103
5.3.1	Análise dos experimentos para a instância berlin52	103

5.3.2	Análise dos experimentos para a instância eil101	113
5.3.3	Análise dos experimentos para a instância d493	123
5.3.4	Análise dos experimentos para a instância u1060	133
5.3.5	Análise dos experimentos para a instância rl1889	143
5.4	Análise de Frequência	152
5.5	Considerações sobre o capítulo	155
6	CONSIDERAÇÕES FINAIS	157
6.1	Contribuições	158
6.2	Trabalhos Futuros	159
6.3	Epílogo	159
	REFERÊNCIAS	162
	APÊNDICE A – REDE DE AUTÔMATOS ESTOCÁSTICOS	180
	APÊNDICE B – LENTES ESPECTRAIS DE ANÁLISE DAS META-HEURÍSTICAS	210

1 INTRODUÇÃO

*“O homem sensato adapta-se ao mundo.
O homem insensato insiste em tentar adaptar o mundo a si.
Sendo assim, qualquer progresso depende do homem insensato.”*

George Bernard Shaw

O presente trabalho inicia-se com uma inquietação que parte do estudo sobre o contexto da Otimização Combinatória e do conjunto meta-heurístico que se relaciona, e talvez, mais amplamente, possa ser pensada no âmbito da Ciência Computacional como um todo. Tal ânsia origina-se do trabalho de Kenneth Sörensen (2015), intitulado *Metaheuristics – the metaphor exposed*, no qual, o autor questiona a originalidade das meta-heurísticas a partir de duas reflexões: em primeiro lugar, tenta compreender as fontes de inspiração para novas meta-heurísticas apresentando uma visão crítica sobre os métodos ditos “novos”; em segundo, acredita que esta enxurrada de abstrações inspiradoras, não passam de falácias na construção das “novas” meta-heurísticas. Diante desse diagnóstico proposto pelo autor, perguntamo-nos: o que torna o campo da Computação tão suscetível a este tipo de pesquisa? Fazendo-nos indagar sobre a contribuição científica deste tipo de pesquisa a que somos expostos.

Embora as meta-heurísticas baseadas em metáforas tenham produzido soluções notavelmente eficazes para problemas específicos, conjuntamente atraíram críticas da comunidade científica, pois acabam por esconder sua falta de eficácia ou novidade atrás de elaboradas metáforas (BROWNLEE; WOODWARD, 2015).

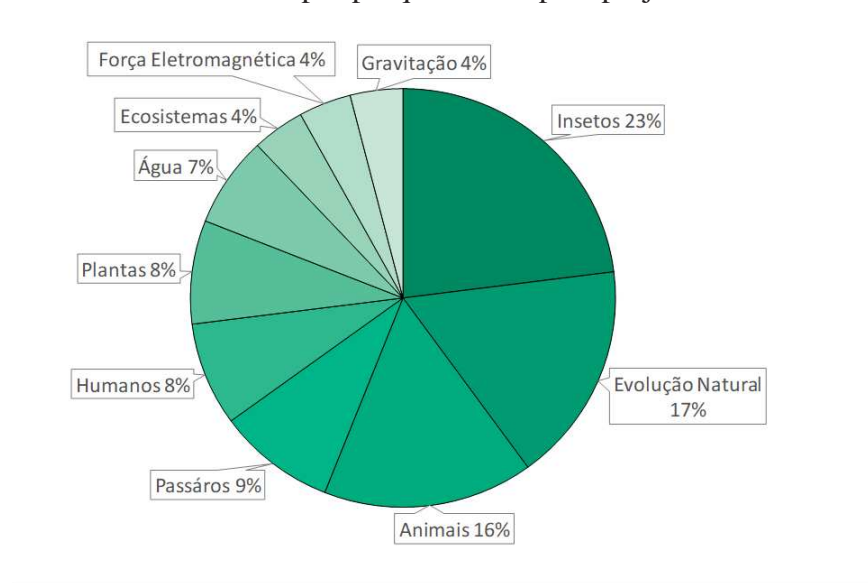
De acordo com Glover e Sorensen (2015, tradução nossa),¹

Um grande (e crescente) número de publicações se concentra no desenvolvimento de (supostas) novas estruturas meta-heurísticas baseadas em metáforas. [...] a novidade da metáfora subjacente não torna automaticamente a estrutura resultante "nova". Pelo contrário, há evidências crescentes de poucos dos métodos baseados em metáforas são novos em qualquer sentido interessante.

Corroborando a afirmação de Glover e Sorensen (2015), pode-se observar, na Figura 1, as principais fontes metafóricas utilizadas nas pesquisas dentre os anos de 1983 e 2016.

¹“A large (and increasing) number of publications focuses on the development of (supposedly) new metaheuristic frameworks based on metaphors. [...] the novelty of the underlying metaphor does not automatically render the resulting framework "novel". On the contrary, there is increasing evidence that very few of the metaphor-based methods are new in any interesting sense.”

Figura 1 – Metáforas adotadas por pesquisadores para projetar novas metaheurísticas



Fonte: Tradução nossa de (HUSSAIN et al., 2018, p. 28)

Recentemente, várias revistas, incluindo o Springer Journal of Heuristics, revisaram suas políticas editoriais para evitar métodos cuja única novidade seja a metáfora em que se baseiam na publicação. Em resposta ao artigo de Sorensen (2015), a revista Springer (2017, p. 1, tradução nossa)² concorda com o autor e atualizou sua política editorial para afirmar que:

A proposta de novos paradigmas só é aceitável se contiverem ideias básicas inovadoras, como as que estão incorporadas em estruturas clássicas, como algoritmos genéticos, busca tabu e arrefecimento simulado. O Journal of Heuristics evita a publicação de artigos que reencontram e incorporam ideias antigas em métodos que se afirmam baseados em metáforas de sistemas e processos naturais ou artificiais.

Deste ponderamento sobre as contribuições para o campo, dentro do âmbito da Otimização Combinatória, centramo-nos nas meta-heurísticas por três razões. Em primeiro lugar, a heterogeneidade do domínio oferece como desafio a oportunidade de generalização de suas funções. Em segundo lugar, esse território é conceitualmente amplo, bem definido e compreendido em teoria, mas não tem sido amplamente abordado a partir de um ponto de vista crítico das especificidades que conduzem a exploração do espaço de busca. Em terceiro lugar, como o uso das meta-heurísticas, em geral, requer a adaptação dos operadores de intensificação e diversificação. Estes aspectos tornam as meta-heurísticas um subdomínio provocativo, no sentido da reflexão sobre seus mecanismos, fundamentos e potencialidades de aplicação.

²“Proposing new paradigms is only acceptable if they contain innovative basic ideas, such as those that are embedded in classical frameworks like genetic algorithms, tabu search, and simulated annealing. The Journal of Heuristics avoids the publication of articles that repackage and embed old ideas in methods that are claimed to be based on metaphors of natural or manmade systems and processes.”.

Em um contexto histórico, Sorensen, Sevaux e Glover (2017, p. 3) dividem a história das meta-heurísticas em cinco períodos distintos. São eles:

- período pré-teórico (até século XIX), durante o qual são utilizadas heurísticas e até mesmo meta-heurísticas, mas não são formalmente estudadas.
- período inicial (1940 - 1980), durante o qual aparecem os primeiros estudos formais sobre heurísticas.
- período centrado no método (1980 - 2000), durante o qual o campo das meta-heurísticas realmente decola e muitos métodos diferentes são propostos.
- período centrado na estrutura (2000 - atualmente), durante o qual a percepção sobre as meta-heurísticas cresce e são descritas de forma mais útil como estruturas e não como métodos.
- período científico (o futuro), durante o qual o design das meta-heurísticas torna-se uma ciência em vez de uma arte.

Foram inúmeros os progressos feitos desde o período pré-teórico das meta-heurísticas até hoje. Hussain et al. (2018, p. 26), numa análise extensiva de 1222 publicações, entre os anos de 1983 e 2016, demonstram que a Computação é a terceira área com mais publicações sobre meta-heurísticas, sendo responsável por 11% das produções, ficando atrás da biologia com 54%, e a da física com 14%. Entretanto, os autores dizem que o campo das meta-heurísticas, especialmente quando comparado a outros campos de estudo como física, química e matemática, ainda carece de amadurecimento. Prova disso é a enxurrada de técnicas que vêm surgindo desde os anos 2000, período centrado na estrutura. A exemplo disso, Aranha e Campelo (2019) apresentam uma lista de 201 meta-heurísticas baseadas em computação evolutiva, das quais apenas 4 apareceram antes do ano 2000. Outro exemplo, mais geral, consolida na Wikipédia uma lista de 63 contribuições, das quais 41.26% surgiram após os anos 2000 (WIKIPEDIA, 2019). Esta abundância de produções margeiam-se fundamentadas em técnicas clássicas consolidadas, como vê-se na Tabela 1 a predominância de trabalhos baseados em Algoritmos Genéticos e Arrefecimento Simulado.

Tabela 1 – Publicações por ano

(continua)

Ano	Meta-heurística	Porcentagem	Publicações
1958	Evolução Diferencial	100%	1
1965	Comportamento das Abelhas	100%	2
1966	Arrefecimento Simulado	100%	1
1967	Comportamento das Abelhas	100%	1
1968	Arrefecimento Simulado	100%	1
1969	Comportamento das Abelhas	50%	2
1970	Evolução Diferencial	50%	2

(conclusão)

Ano	Meta-heurística	Porcentagem	Publicações
1971	Evolução Diferencial	100%	1
1972	Estrutura e dinâmica das galáxias	50%	2
1973	Comportamento das Abelhas	50%	2
1974	Evolução Diferencial	50%	4
1975	Arrefecimento Simulado	66%	3
1976	Evolução Diferencial	42%	7
1977	Evolução Diferencial	75%	8
1978	Evolução Diferencial	80%	5
1979	Comportamento das Abelhas	50%	4
1980	Evolução Diferencial	100%	9
1981	Algoritmo Genético	33%	9
1982	Evolução Diferencial	45%	11
1983	Arrefecimento Simulado	52%	17
1984	Arrefecimento Simulado	64%	25
1985	Arrefecimento Simulado	71%	77
1986	Arrefecimento Simulado	81%	96
1987	Arrefecimento Simulado	89%	133
1988	Arrefecimento Simulado	88%	322
1989	Arrefecimento Simulado	87%	488
1990	Arrefecimento Simulado	84%	722
1991	Arrefecimento Simulado	81%	886
1992	Arrefecimento Simulado	72%	1202
1993	Arrefecimento Simulado	64%	1497
1994	Arrefecimento Simulado	56%	1929
1995	Algoritmo Genético	53%	2046
1996	Algoritmo Genético	53%	2289
1997	Algoritmo Genético	57%	2484
1998	Algoritmo Genético	58%	2457
1999	Algoritmo Genético	64%	2724
2000	Algoritmo Genético	63%	2953
2001	Algoritmo Genético	64%	2842
2002	Algoritmo Genético	64%	3616
2003	Algoritmo Genético	64%	3913
2004	Algoritmo Genético	63%	4369
2005	Algoritmo Genético	62%	5358
2006	Algoritmo Genético	61%	7073
2007	Algoritmo Genético	62%	8011
2008	Algoritmo Genético	61%	9935
2009	Algoritmo Genético	60%	11938
2010	Algoritmo Genético	62%	13229
2011	Algoritmo Genético	58%	11264
2012	Algoritmo Genético	56%	11306
2013	Algoritmo Genético	54%	11931
2014	Algoritmo Genético	53%	13280
2015	Algoritmo Genético	53%	14974
2016	Algoritmo Genético	54%	17526
2017	Algoritmo Genético	48%	11987

Fonte: Elaborado pelo autor com base em (AISEARCH, 2019)

O panorama que tem se apresentado no campo da pesquisa computacional, de modo geral, mostra uma predominância de perspectivas modelares, com caminhos prontos e protegidos pelo saber hegemônico. São nesses locais onde, por vezes, perdemos na natureza da pesquisa e nos embrenhamos na burocracia do discurso, da economia, da política, enquanto o movimento de descoberta e de criação fica em um segundo plano. É daí que surgem contextos e momentos de certa estagnação científica, com: repetições de objetos, de problemas, de metodologias e de resultados de pesquisa.

1.1 Objetivos

A efetividade das meta-heurísticas está fortemente ligada ao conjunto de operadores de busca aplicados pelo usuário, limitando, assim, seu uso à formalização por um especialista. Por outro lado, as hiper-heurísticas formam uma tecnologia de busca emergente, fornecendo uma abordagem para superar o problema de tais dependências das meta-heurísticas, pois são assumidas como independentes de problemas e podem ser facilmente utilizadas por não especialistas (BOUSSAÏD; LEPAGNOT; SIARRY, 2013). Embora os resultados deste estudo possam ser aplicáveis a outras áreas, a abordagem tenta explorar as características particulares das técnicas e, por isso, neste trabalho, centramo-nos neste subdomínio bem definido.

O objetivo principal da tese é *desenvolver uma forma de resolução de problemas de Otimização Combinatória, capaz de codificar, armazenar e recuperar informações sobre os mecanismos heurísticos empregados no processo de otimização*. Argumenta-se que isso pode ser obtido pela construção de um modelo hiper-heurístico de otimização capaz de lidar com uma variedade de problemas e com o ajuste de suas operações de acordo com as estruturas encontradas nas instâncias dos problemas.

Pensa-se que a melhor forma de atingir essa intenção seja adequar as estratégias de intensificação e diversificação, gerenciando a multiplicidade inerente à heterogeneidade de componentes que conduzem a busca nas meta-heurísticas. Salienta-se que esse é o grande desafio da tese. Há dois meios para se alcançar tal objetivo: *i) uma maneira genérica de descrever as diferentes meta-heurísticas, e ii) aplicações em conjunturas reais que não apenas se justifiquem numericamente*. Em outras palavras, o interesse não está pautado exclusivamente em técnicas, nem se foca em uma análise qualitativa dos processos, mas sim na maneira como os mecanismos componentes das técnicas produzem resultados de ordem mais específica, percebendo a natureza de cada situação e de cada problema.

1.2 Declaração do Problema

São inúmeros os estudos sobre meta-heurísticas nas últimas duas décadas que se demonstram cada vez melhores em relação a seus concorrentes passados (AISEARCH, 2019). Este fato apresenta uma tendência exposta por Sörensen (2015) que reflete sobre a originalidade das novas pesquisas que apresentam algoritmos repaginados, pautados em novas metáforas inspiradoras. Esta lacuna científica fez com que fossem ressignificadas as políticas editoriais da área, afetando as pesquisas no sentido da demonstração qualitativa, que carece de compreensões mais profundas da sistemática das meta-heurísticas.

Neste sentido, esta pesquisa não intenciona puramente desenvolver um novo algoritmo, mas compreender os operadores heurísticos no conjunto da otimização, visando a resultados de ordem qualiquantitativa. Desta forma, um estudo que investigue as relações de natureza dos mecanismos heurísticos em prol dos resultados auxilia na compreensão mais apurada deste tema nos domínios da computação;

Assim sendo, o objetivo abstrato do nosso modelo hiper-heurístico pode ser declarado da seguinte forma:

Dado um problema de otimização e um conjunto de mecanismos heurísticos que compõe uma heurística, o modelo deve analisar o comportamento dos mecanismos heurísticos em prol da construção, recomendação e/ou parametrização das heurísticas de baixo nível, a fim de reduzir o grão computacional heurístico.

Para isso, mediremos a qualidade dos resultados obtidos e o tempo de computação na intenção de relacionar em cada um dos testes uma visão sobre qualidade e quantidade. Visando a abarcar a declaração acima, o modelo deve aprender sobre o processo de execução e diminuir as relações de domínio advindas do problema de otimização.

1.3 Abordagem de Pesquisa e Limitações

Diante do problema e do objetivo da tese, partiu-se de uma etapa de análise de como um sistema de otimização meta-heurístico pode ser afetado em função de seus mecanismos, como a formação das vizinhanças, armazenamento, interação das soluções e caminhos, definição de regiões viáveis, entre outras características. Desta análise, identificou-se como um passo importante a avaliação da organização do espaço de busca, detectando necessidades de implantação de restrições, definindo os recursos necessários para a construção de sistemas de inteligência computacional. Também, se fez necessário, nesta etapa, verificar a forma como a solução seria projetada, analisando os possíveis ganhos e perdas oriundos dessa abordagem.

A investigação de características e deficiências presentes em modelos e ferramentas atuais foi um procedimento exploratório utilizado para identificar quais propriedades heurísticas

são críticas na constituição do modelo proposto. Destarte, analisou-se as atuais modelagens das heurísticas para que fossem compreendidos os requisitos necessários para tal descrição, identificando características passíveis de ajustes para alcançar a construção do modelo e a delimitação da barreira de domínio. Além disso, para perceber como esse tema é trabalhado no campo, e quais são as abordagens comumente empregadas, a revisão bibliográfica apresentou-se como o passo inicial e constante no decorrer da pesquisa.

A trajetória evolutiva das pesquisas sobre meta-heurísticas no campo da Computação (no Brasil e em outros países) se solidifica num conjunto de contribuições concretas para o entendimento dos fenômenos computacionais que são de nosso interesse. Tais pesquisas conjuntam, muitas vezes, avanços importantes para o campo. Assim, a pesquisa da pesquisa torna-se, conseqüentemente, uma prática indispensável para gerar uma aproximação do estado da arte, a fim de que se contemple e se considere estes desenvolvimentos buscando-se avançar efetivamente com e a partir deles.

Tendo como base uma percepção conceitual sobre o campo a partir de uma pesquisa da pesquisa, chegou-se ao entendimento de que há uma certa bipolaridade com uma intensificação nos estudos que envolvem o tema meta-heurística, de modo que há uma preocupação maior com os resultados quantitativos dos nossos trabalhos, do que com sua real contribuição aos domínios da Computação. É entendível que, do ponto de vista do pesquisador, nas Ciências Computacionais, isso seja de extrema importância, mas não se deve confundir os quantificadores de nossos trabalhos em prol de uma suposta originalidade, operando, desta forma não somente sobre a quantificação e a qualificação, mas na constituição de estratégias de computabilidade. Sörensen (2015), entretanto, não elabora um alvo que seja a síntese do quantitativo ou do qualitativo; ele questiona a originalidade dos “novos” métodos que se permeiam marginalmente como variantes ocasionais de métodos já existentes.

Dessas leituras, revelou-se o horizonte de pesquisa. Mantendo o interesse pelo estudo das meta-heurísticas, mas sem restringir a problematização aos temas qualiquantitativos, interessamo-nos pelos processos condutores das buscas no espaço de soluções. Isso implica em fazer uma escolha equilibrada, não optando por perspectivas cristalizadas, logo, descarta-se trabalhar com a individualidade das técnicas meta-heurísticas, cujos trabalhos as transportam em esquemas puramente quantitativos; e também são inviáveis, os esquemas puramente qualitativos. Não é nosso objetivo apresentar uma história das meta-heurísticas, nem mesmo uma listagem cronológica, mas esclarecer o cenário que nos envolve, afastando-nos do período da estrutura em direção ao período científico, conforme classificação de Sorensen, Sevaux e Glover (2017, p. 3)

Diante dessas observações, de cunho histórico e epistemológico, o interesse de pesquisa se direciona para uma proposta de um diferente processo computacional, no qual as meta-heurísticas não são produzidas para usos individuais. Com inspirações cada vez mais metafóricas, o movimento teórico inicial foi pensar as meta-heurísticas como processos que se dão *entre* a esfera qualiquantitativa, em um plano de proximidade que conecta movimentos no espaço de

busca. Nesse limiar, encontrou-se um caminho produtivo para desdobrar a tese. Ao se levantar o estado da arte sobre meta-heurísticas, depara-se com diferentes enfoques de pesquisa. Nestes estudos, identificou-se que os algoritmos e suas variantes fazem diferentes usos dos mecanismos de exploração (diversificação e intensificação). Do ponto de vista do desenvolvimento, o modo de implementação contribui em algum grau para o desempenho. Assim, a validação e o teste da implementação de qualquer algoritmo são importantes (TALBI, 2009), fazendo-nos crer que os diferentes níveis e métodos exploratórios conduzem a maneiras eficientes de se distribuir pelo espaço de busca.

Acredita-se que cada caso solucionado por meio do emprego de meta-heurísticas demanda uma resposta adequada, não precisando ser, necessariamente, uma contrapartida hegemônica e utilizada em larga escala pela sua possível efetividade. Isto, deve-se ao fato de que diferentes problemas de otimização apresentam diferentes estruturas, bem como diferentes instâncias do mesmo problema, correspondem a diferentes estruturas (BOUSSAÏD; LEPAGNOT; SIARRY, 2013). Dessa forma, conseqüentemente, a compreensão dessas estruturas fornece um primeiro passo para o entendimento do comportamento dos diferentes componentes das meta-heurísticas, o que pode levar à concepção de um modelo de busca que incorpore mais conhecimento específico do problema.

Sabendo disso, que nenhum método é cabal, ratifica-se o *no-free lunch theorem* atestando que um desempenho superior obtido por um algoritmo em uma classe de problemas pode ter seu desempenho abatido quando aplicado a outras classes e/ou instâncias de problemas (WOLPERT; MACREADY, 1997). Em outro sentido, isso não impede que alguns algoritmos sejam melhores do que outros em classes específicas de problemas, ou seja, se calculada a média em todo o espaço de função possível, os algoritmos executarão, em média, igualmente bem. Sendo assim, não há algoritmos universalmente melhores, isto é, a principal tarefa é encontrar as melhores soluções para cada caso, e não a média de todos os possíveis espaços de função.

Apresenta-se, assim, uma perspectiva alternativa onde não se vê a necessidade de encontrar a média em todas as funções possíveis para um determinado problema de otimização, mas sim, investir na escolha do *mecanismo heurístico* mais adequados às características do cenário de dado problema. É nesta lacuna que a tese é desenvolvida.

Esta lacuna que aponto se abre diante do meu *eu pesquisador* como um abismo, em razão de uma fraqueza teórica/epistêmica do meu saber computacional. Minha perspectiva de aprendiz me fez observar que o presente tema ainda se encontra carente de uma reflexão no campo, pois este é povoado de práticas instrumentais de saber/fazer técnicos, grandemente justificado mediante explicações qualitativas. Neste momento, escrevo em primeira pessoa para registrar um tenro hiato entre o meu eu pesquisador, que sabe, do meu eu pesquisador que deseja saber sobre o que sabe.

Para sustentar e aprofundar a proposta, buscou-se caminhos teóricos que estão além do campo da Computação, como uma forma de introduzir os benefícios de uma problemática diversa e transdisciplinar. Estes novos caminhos que nos fazem migrar de pensadores/pesquisadores e profissionais de outros campos para compor o campo da Computação, nos fazem ver o conhecimento como algo que independe de fronteiras solidificadas, que não segue necessariamente uma ordem pré-estabelecida, ou ainda, que não possui sentido de causalidade, expandindo-se de forma múltipla. Este intercâmbio entre disciplinas e conhecimentos pode gerar inquietações logocêntricas que busquem defender parcelas de saberes como lugares basilares de estruturação (MALDONADO, 2003).

Desse modo, os caminhos foram inspirados por diferentes referências, dentre elas a metáfora das constelações do filósofo alemão Walter Benjamin (1984), que contorna a construção da problemática, dando-nos um lugar de fala para olhar o objeto de pesquisa deste trabalho.

As constelações, definidas como agrupamentos imaginários de estrelas, sempre fascinaram o homem que, intrigado com a esfera celeste, desde os tempos mais remotos, empenha-se, obsessivamente, em decifrar os enigmas do universo. Visíveis a olho nu ou com a ajuda de lentes e aparelhos inventados pela demanda investigatória desse olhar inquiridor, estrelas e constelações vieram habitar, por signos e narrativas, os mapas e os mitos. A alma do homem encontrou, nesses sinais luminosos vindos do céu, índices misteriosos que o instigaram à elaboração de histórias que intentavam desvelar os enigmas da cosmogonia, da origem e do destino final do universo, da vida e da humanidade. (OTTE; VOLPE, 2000, p. 35)

De acordo com Otte e Volpe (2000, p. 36), “a metáfora da constelação certamente foi uma das inspirações mais importantes de Walter Benjamin”, sendo um meio para mostrar reflexões abstratas – uma forma de relacionar ideias às coisas, assim como as constelações se relacionam com as estrelas.

Inspirados em Benjamin, apropria-se do termo constelação, utilizando-o como uma metáfora para expressar a heterogeneidade do objeto. Este objeto compõe o universo de pesquisa, o qual, observa-se uma infinidade de componentes que não se consegue, ou até mesmo não se pode compreender em totalidade. É destes pontos brilhantes do universo de pesquisa que são traçados alguns contornos. No caso desta pesquisa, a metáfora remete, então, a um procedimento que coloca o andar pelo espaço de busca, para detectar que certos mecanismos heurísticos brilham mais intensamente, permitindo traçar constelações, ressignificando as meta-heurísticas. Deste modo, se é possível compreender algo de um objeto heterogêneo, isto se dará pela cartografia de seus pontos mais brilhantes.

Nessa aplicação metafórica, os pontos brilhantes representam o *corpus* da pesquisa e as constelações as perspectivas com que são moldadas. O objetivo não é traçar fronteiras bem delimitadas, mas perceber que um ponto localizado em uma constelação pode pertencer também

a outras, como um ponto obscuro, ou até mesmo luminoso, o que nos permitirá observar e diferenciar as multiplicidades mais evidentes das mais sutis.

Constantemente usamos metáforas sem ter consciência disso. Benjamin nos mostra que não há necessidade de ir muito longe para detectar, dentro da nossa linguagem (**aqui refere-se à linguagem computacional**), metáforas muitas vezes bastante produtivas; não há necessidade de "inventar", pois as metáforas estão aí, basta identificá-las como tais. O problema é a instrumentalização das palavras que faz com que elas percam sua singularidade e os chamados sinônimos possam ser trocados uns pelos outros, pois o uso comunicativo da linguagem sempre visa uma finalidade fora dela. (grifo nosso) (OTTE; VOLPE, 2000, p. 37)

Todavia, a investigação não poderia ter Benjamin como eixo teórico central, pois não se daria suficiente profundidade para esquadrihar o estado da arte. Para se embrenhar nos estudos heurísticos, foi necessário que a problematização fosse conduzida também por modelos e teorias da área. Para isso, no segundo capítulo, intitulado *Perspectivas Modelares*, fundamenta-se a pesquisa inicialmente na reflexão da premissa metafórica que constitui as meta-heurísticas como um uso diferenciado dos mecanismos, delineando o *corpus*, explorando as peculiaridades em relação ao conjunto de técnicas analisadas no terceiro capítulo. A principal questão tratada nessa etapa do trabalho interroga sobre como os mecanismos heurísticos conduzem a navegação pelo espaço de busca. Esta questão remete ao procedimento de modelagem e experimentação que visa, por meio de um modelo hiper-heurístico, combinar os mecanismos heurísticos para obtenção de resultados de qualidade em menor tempo computacional.

1.4 Organização do trabalho

O presente trabalho está organizado em seis capítulos, que refletem as etapas do estudo desenvolvido para o tema que intitula este trabalho, conforme o desenvolvimento das etapas que a pesquisa seguiu. O primeiro capítulo consiste na introdução, que descreve a área de estudo, fornecendo ao leitor a apresentação do objeto de pesquisa e sua contextualização, de modo a enaltecer a relevância do tema, bem como os objetivos definidos para a pesquisa.

O segundo capítulo apresenta uma revisão epistêmica que versa a respeito das dimensões que os mecanismos heurísticos alcançam, apresentando pontos de vista sobre os conceitos que regem os processos de busca no espaço de soluções.

Em contraste com a grande quantidade de trabalhos sobre meta-heurísticas, esta pesquisa não visa a fornecer metodologias suficientes para a resolução da gama de problemas presentes no contexto da Otimização Combinatória. Tal trabalho seria resultado de uma pesquisa extensiva indo além dos domínios de uma única tese. Contudo, como inúmeras variantes foram resolvidas individualmente na literatura, descreve-se uma estratégia genérica para incluir as características destes métodos de solução heterogêneos no sistema desenvolvido.

Assim, o terceiro capítulo do texto, denominado *Mecanismos Heurísticos*, se apropria das perspectivas modelares elencadas no segundo capítulo como um espectro de análise que tem como função apresentar a composição do conjunto de técnicas meta-heurísticas que compõe a estrutura do modelo desenvolvido.

No quarto capítulo, passa-se do interesse centrado na relação dos mecanismos fundantes com as técnicas meta-heurísticas, os quais resultam na interface dos conteúdos. Neste capítulo, apresenta-se uma formalização das Redes de Autômatos Estocásticos de Aprendizagem para fundamentar o desenho das meta-heurísticas junto ao modelo desenvolvido. O modelo é apresentado à luz da linguagem computacional, ou seja, apresenta-se a arquitetura em seus níveis, bem como a modelagem das redes de autômatos estocásticos que regem o metaespaço.

A concepção de novos modelos traz consigo propriedades que, normalmente, são descobertas durante aplicações em conjunturas práticas. Desta forma, a prototipação é uma prática comum para validação de novas propostas. No caso de um modelo hiper-heurístico, torna-se interessante sua avaliação por meio de simulações medindo sua acurácia, explorando possibilidades que vão ao encontro de um problema clássico da área de Otimização Combinatória.

Desta forma, a fim de conseguir uma reutilização suficientemente extensiva, tenta-se capturar as extensões das variantes presentes na prática. As aplicações se dão em um conjunto de problemas e instâncias clássicas comumente encontrados na bibliografia sobre meta-heurísticas e limitam-se a cinco instâncias TSPLib³ para o Problema do Caixeiro Viajante (*Traveling Salesman Problem* - TSP), mas não se restringe quanto ao conjunto de possibilidades de aplicação do modelo proposto. Essas construções serão discutidas primeiro na forma como são abordadas na literatura e, em segundo lugar, no contexto da estrutura da modelagem proposta. Além disso, o processo de generalização permite o reaproveitamento da modelagem e da metodologia de solução, minimizando, conseqüentemente, os custos de produção de um novo mecanismo de otimização.

Após a etapa de modelagem, propomos uma reviravolta da problematização, colocando em xeque o caráter do modelo. O processo de validação e experimentação realizado para corroborar a proposta é apresentado no capítulo cinco. Este processo se dará por meio dos experimentos, um lugar de definição dos processos de experimentação e validação da tese. Este passo da pesquisa demonstra a compatibilidade da metodologia de solução na abordagem apresentada, confirmando-a como uma opção viável para abordar um conjunto heterogêneo de problemas ou variantes de problemas especialmente complexos.

Cada capítulo conta com uma seção de encerramento que apresenta considerações parciais, deixando as conclusões mais próximas de sua origem. Por sua vez, o sexto e último capítulo aborda as implicações e considerações finais, bem como dá indicações sobre as potencialidades do modelo H2-SLAN e os trabalhos futuros.

³Cf. TSPLIB, 2018

2 PERSPECTIVAS MODELARES

Nas últimas três décadas, as meta-heurísticas desempenharam uma conjunção entre a Pesquisa Operacional e a Inteligência Computacional devido à complexificação dos problemas que permeiam os dois campos. Embora este interfaceamento tenha aprofundado a compreensão científica acerca dos processos de busca em domínios complexos, uma investigação se faz necessária para compreender as características que conduzem à otimalidade desejada. Dessa maneira, neste capítulo ter-se-á como direção de pesquisa a exploração das estruturas algorítmicas de propósito geral, aqui nomeadas de perspectivas modelares, que auxiliam o processo de condução da busca em direção a regiões promissoras do espaço de soluções.

A proposta deste capítulo é refletir sobre a composição das constelações, numa intenção de conectar os pontos que refletem às meta-heurísticas, a partir da análise dos mecanismos que constituem nossa investigação. Para esta realização, foram delineadas perspectivas modelares que orientaram a reflexão acerca da condução da busca no espaço para, na sequência, qualificar e ponderar sobre o sentido de certas práticas constitutivas nos processos de busca. Especificamente, são objeto de exame as tenuidades que caracterizam e conduzem a busca pelo espaço de soluções.

O propósito dessa abordagem é aumentar o nível de compreensão acerca da generalidade que os sistemas de otimização podem alcançar por meio de novas conjunções, dirigindo-se a sistemas mais heterogêneos, capazes de atuar em uma gama de problemas. Dentro dessa proposta, encontram-se mecanismos que se comportam de maneira contrária às meta-heurísticas e que tendem a desenvolver necessidades mais específicas, da ordem de personalização de suas características. Essa visão sobre as especificidades dos mecanismos é de particular interesse neste trabalho, pois conduz a ideação do modelo.

Além desse estrato teórico, será apresentado neste capítulo um breve histórico emergente da área, que delineará os mecanismos de condução da busca pelo espaço, revisitando trabalhos correlatos e/ou que façam uso de tais maquinismos.

Mesmo que os diversos autores que permeiam este capítulo (e outros) não (re)signifiquem as perspectivas de modo a pensar suas influências conducentes da busca, há uma dureza, ainda que demasiada, para que estes mecanismos sejam produtivos em nossa abordagem, motivo pelo qual os pensamos como ponto de partida para reflexão. Nessa visão com devir epistemológico, entende-se as *perspectivas modelares* como algo que dirige a busca pelo espaço, agenciando movimentos pelos quais evolui o processo de busca.

Foi a partir destas perspectivas modelares que, numa visão ampliada sobre as meta-heurísticas, passa-se a pensá-las como formas que se movem no espaço de busca. O movimento no processo de busca por meio de um conjunto de mecanismos faz-nos atribuir os sentidos identitários da exploração do espaço de busca, nomeados de processos de intensificação e de diversificação, os quais, têm um importante papel nas meta-heurísticas. Neste momento, cabe situar o leitor sob o processo identitário que se tenta. Esta identidade meta-heurística que se

desvenda permeando as perspectivas, em relação aos conjuntos que as definem em associação, isto é, não seria apenas a própria referência identitária criada pelas perspectivas modelares que deve ser entendida no processo conducente da busca, mas também os próprios processos meta-heurísticos, ou dito de outra forma, não nos interessa apenas dizermos quais são, mas sim, como são.

2.1 Princípios, conceitos e origens das heurísticas

O termo heurística origina-se da palavra grega *heurískein*, e significa “descobrir”, “encontrar” referindo-se ao processo de obtenção de algum resultado desejado (AULETE, 2017). Ao ser utilizado no contexto computacional, esse verbete apresenta significados e modos de utilização, predominantemente relacionados ao processo de encontrar soluções.

Ao longo desta pesquisa, a intenção é que se faça uso do termo *heurística* como reflexo da nomenclatura computacional para algoritmos e métodos exploratórios que segundo Talbi (2009), são elementos que objetivam a resolução de problemas de otimização, cujas soluções são buscadas por meio de aproximações sucessivas, sem, no entanto, definir o quão próximas do ótimo global estão as soluções obtidas. Nesse processo, avalia-se continuamente os progressos alcançados até que uma solução¹ satisfatória seja encontrada e se possa dar o problema como “resolvido”, sem oferecer, no entanto, quaisquer garantias de qualidade em relação a soluções ótimas (SUCUPIRA, 2007).

O processo de encontrar ótimas soluções, por vezes demonstra-se intratável para uma gama de problemas de otimização. Essa não praticidade deve-se à extensão do espaço de busca ou à dificuldade em defini-lo. Como exemplo, podem haver restrições que tornam o espaço de soluções demasiadamente complexo. Na prática, devido à intratabilidade de tais problemas, é comum recorrer a abordagens heurísticas, onde são satisfatórias as soluções denominadas “boas”. Diferentemente dos algoritmos de otimização exata como a família Branch-X (*Branch and Bound* (LAWLER; WOOD, 1966), *Branch-and-Price* (BARNHART et al., 1998) e *Relax-and-Cut* (GOMORY, 2010)), Programação Linear (DANTZIG, 1945), Programação Dinâmica, as heurísticas não garantem a otimalidade das soluções obtidas, ou quaisquer garantias de qualidade da solução, mas tendem a gerar respostas rápidas e satisfatórias, com opções que venham a corresponder às exigências dadas para a qualidade da solução (PURANEN, 2011). No mesmo passo que Bandaru e Deb (2016), pode-se compreender as heurísticas no contexto da computação como métodos de resolução de problemas sem aplicação de procedimentos exaustivos. De outro modo é aquela que:

- Procura uma solução aproximada;
- Não necessita ter uma prova matemática de convergência; e

¹Cabe notar que, a utilização da palavra “solução”, no contexto deste trabalho, não faz referência restrita à otimalidade ou viabilidade, mas à validade e possibilidade.

- Não explora individualmente cada uma das possíveis soluções no espaço de busca antes de chegar a uma solução final, portanto, é computacionalmente eficiente.

Embora nos métodos heurísticos a exploração seja feita de forma algorítmica, o resultado é obtido puramente pela avaliação empírica do progresso, não assegurando as melhores soluções, mas sim, soluções aproximadas e válidas. Frequentemente, não é possível justificar a validade dos resultados dos algoritmos heurísticos em termos estritamente lógicos, mas sua importância advém do fato de que são desenvolvidos para lidar com problemas de otimização complexos e representam a única alternativa viável, em termos computacionais e temporais, nos casos em que ainda não existem estratégias exatas viáveis que apresentem garantias de otimalidade.

Em resumo, a importância das heurísticas procede do bom desempenho médio – detectado de forma predominantemente experimental – quando aplicadas ao tratamento de problemas para os quais não são conhecidos métodos eficientes que forneçam garantias de soluções ótimas em tempo computacional aceitável.

Para a contextualização sobre a complexificação das heurísticas, toma-se como ponto de partida o ano de 1945, onde Polya (1945) introduz o conceito de heurística na resolução de problemas de otimização. Com o passar do tempo, novas necessidades surgiram e o desempenho dos métodos heurísticos foi conduzido na direção da elaboração de estratégias genéricas, definindo estruturas algorítmicas para construção de heurísticas. A exemplo, criado por Dantzig em 1945, tem-se o algoritmo *simplex* para resolução de problemas de programação linear. Posteriormente, Edmonds (1971) apresenta a primeira heurística gulosa na literatura de Otimização Combinatória.

Todavia, a compreensão das características exitosas das heurísticas e a necessidade de métodos capazes de lidar de forma robusta com a profusão de problemas que surgem no mundo real, conduziram à construção de estratégias mais elaboradas denominadas meta-heurísticas. Tal conceito surgiu anteriormente à primeira definição do termo meta-heurística que foi, em 1986, introduzido por Glover (GLOVER, 1986). Entretanto, não há uma definição exata e consensual final acerca do termo, mas aqui será adotada a seguinte, apresentada por Dorigo e Stützle (2004, p. 33, tradução nossa) ²:

Uma meta-heurística é um conjunto de conceitos algorítmicos que pode ser usado para definir métodos heurísticos aplicáveis a uma ampla gama de problemas diferentes. Em outras palavras, uma meta-heurística pode ser vista como uma estrutura algorítmica de propósito geral projetada para guiar uma heurística específica para um problema em direção a regiões promissoras do

²“A metaheuristic is a set of algorithmic concepts that can be used to define heuristic methods applicable to a wide set of different problems. In other words, a metaheuristic can be seen as a general-purpose heuristic method designed to guide an underlying problem-specific heuristic toward promising regions of the search space containing high-quality solutions. A metaheuristic is therefore a general algorithmic framework which can be applied to different optimization problems with relatively few modifications to make them adapted to a specific problem.”

espaço de busca contendo soluções de alta qualidade. Uma meta-heurística é, portanto, uma estrutura algorítmica geral que pode ser aplicada a problemas de otimização diferentes com relativamente poucas modificações para torná-los adaptados a um problema específico.

Como mencionado, muitos problemas de otimização são complexos, ao ponto de não terem sua resolução factível em tempo hábil por meio de algoritmos exatos, devido ao tempo de processamento ou à quantidade de memória necessária para armazenar os dados e variáveis envolvidas.

Existem diversos problemas computáveis em nosso cotidiano, que crescem em número e complexidade na medida em que se acumula conhecimento e se deseja processá-los. Em todos os casos, tem-se algo em comum: a elevada complexidade destes problemas é um entrave na obtenção de resultados satisfatórios. A complexidade de um algoritmo refere-se à quantidade de tempo e espaço necessários para executá-lo (JOHNSONBAUGH, 1997). A análise de um algoritmo diz respeito ao processo de derivação de estimativas de tempo e espaço (memória) requeridos para executá-lo³.

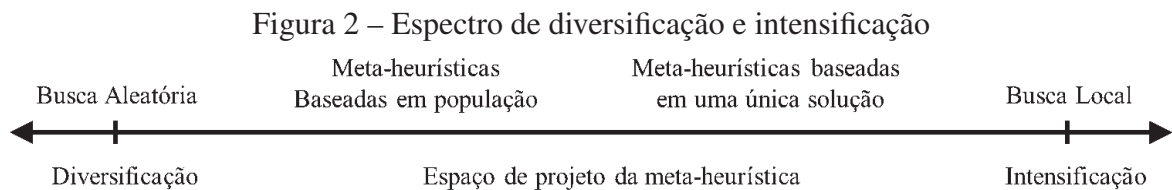
Partindo desse cenário, mas sem desconsiderar a relevância dos métodos clássicos, foram propostas as meta-heurísticas, a fim de obter soluções boas de maneira computacionalmente tratável. As meta-heurísticas têm sua relevância no contexto de otimização, pois descrevem métodos que usam uma ou mais heurísticas. Desta forma, Bandaru e Deb (2016, p. 4) definem as seguintes propriedades às meta-heurísticas:

- Procuram encontrar soluções quase ótimas, ao invés de tentar especificamente encontrar uma solução ótima;
- Geralmente não têm prova do processo de convergência para a solução ótima; e
- Normalmente o processo computacional é mais rápido que a busca exaustiva.

As meta-heurísticas têm como princípios o processo de orientação dos operadores de busca em um nível geral, fornecendo genericamente um modelo de caminamento no espaço de busca. Devido ao espaço de busca ser grande a ponto de seu percurso tornar-se exaustivo, apenas algumas regiões de interesse devem ser exploradas. Isto, instiga-nos a compreender a morfogênese das heurísticas, uma vez que grande parte do seu sucesso se dá pelo adequado balanço entre dois fatores, o primeiro deles a intensificação, ou seja, a exploração da experiência acumulada na busca (*exploitation*), e o segundo, a diversificação, ou seja, a exploração do espaço de busca para identificar regiões com soluções de alta qualidade (*exploration*). Em geral, as meta-heurísticas baseadas em uma solução são mais propensas à exploração para intensificação, enquanto que as meta-heurísticas baseadas em população são predispostas à exploração para diversificação, entretanto, isto não é uma regra (TALBI, 2009).

³Para melhor contextualização, recomenda-se os textos sobre complexidade computacional e temas relacionados Cf. PAPADIMITRIOU; STEIGLITZ, 1998 Cf. GAREY; JOHNSON, 1979

A principal diferença entre cada meta-heurística pode ser dada pelas particularidades com que a técnica tenta alcançar este balanço e como faz uso dos mecanismos heurísticos. Na Figura 2, pode-se compreender o contraponto que é posto em jogo quando se projeta uma meta-heurística. O desafio resume-se a manejar o balanço entre a exploração profunda das regiões promissoras, na esperança de encontrar melhores soluções (intensificação) e a exploração das regiões inexploradas do espaço de soluções, visando a uma uniformidade da busca, em prol de uma suposta completude (diversificação).



Fonte: Tradução nossa de (TALBI, 2009, p. 24)

Como mencionado anteriormente, ao longo dos anos, as (meta)heurísticas foram desenvolvidas para a obtenção de resultados experimentais baseados em uma classe específica de problemas para os quais ainda não havia sido, até então, adaptada uma implementação. Ainda que fosse claro que as heurísticas individualmente, por melhores desempenhos que apresentassem, poderiam ter peculiaridades e limitações. Os problemas de Otimização Combinatória não têm qualquer estrutura explorável, como formas de continuidade, diferenciação ou regularidade global ou local, podendo ser definidos apenas por uma tabela de busca completa (BURKE et al., 2003).

Wolpert e Macready (1997) corroboraram por meio do *No Free Lunch Theorem* a visão de que não existem algoritmos universais de otimização, contrariando argumentos que justificavam uma suposta superioridade absoluta de uma heurística sobre outra. Ainda que tais práticas perdurem na literatura, o teorema dirigiu atenções para a questão de quais algoritmos podem melhor solucionar determinados problemas. Destarte, essa dificuldade motivou pesquisas em prol de métodos mais genéricos que as atuais meta-heurísticas, trazendo como consequência dessa motivação o surgimento das hiper-heurísticas.

O termo hiper-heurística foi cunhado por Cowling, Kendall e Soubeiga (2001), numa tentativa de descrever *uma heurística que lidava diretamente com o problema por meio da escolha e do gerenciamento de outras heurísticas, dependendo das características da região do espaço de solução atualmente em exploração*. Ainda que ampla, essa definição faz referência a métodos que não fazem usos explícitos de informações sobre o problema. Nesse sentido, uma hiper-heurística é um algoritmo que pode adaptar-se a qualquer problema de otimização, fazendo-se assim importante a compreensão das características que a formam, uma vez que necessitam menores esforços para seu uso devido ao grau de automação quando comparada às meta-heurísticas.

2.2 Navegabilidade

A primeira relação que estabelece a perspectiva de onde parte esta pesquisa diz respeito a como as meta-heurísticas navegam pelo espaço de busca. Essa movimentação caracteriza uma suposta navegabilidade e presume que é do processo de oscilação pelo espaço de busca que se diferem as estratégias de movimento contínuo e descontínuo, sendo fortemente orientadas pela espacialidade alcançada, isto é, se percorrem ou não uma região vicinal do espaço de busca.

Essa perspectiva é composta pelos momentos das meta-heurísticas em que a busca se torna um meio de enunciação de si, resultado natural da união entre a solução unitária e a multiplicidade de uma população. A (des)continuidade dos movimentos pelo espaço de busca está intimamente relacionada à distinção entre os métodos baseados em uma população de soluções candidatas ou no uso de uma solução guia. Neste último caso, a cada iteração, apenas uma solução é manipulada, ou seja, começam com uma única solução inicial e se afastam dela, descrevendo uma trajetória no espaço de busca (BIRATTARI et al., 2001). Embora, por vezes, os algoritmos que operam por meio de soluções únicas sejam classificados homogeneamente como métodos de trajetória, abrangendo as meta-heurísticas baseadas em busca local, faz-se necessário desprender deste conceito o GRASP que efetua busca local operando sob uma solução, e que, no entanto, não gera trajetória.

Os métodos de trajetória têm em comum o compartilhamento da propriedade de descrever, claramente, suas trajetórias durante o processo de busca. Segundo Boussaïd, Lepagnot e Siarry (2013, p. 2), estes métodos podem ser vistos como extensões “inteligentes” dos algoritmos de buscas locais. Contrariamente, as meta-heurísticas baseadas em população efetuam processos de busca que delineiam a evolução de um conjunto de pontos no espaço de busca, de maneira não contínua.

A perspectiva modelar da Navegabilidade é o primeiro indício de uma composição que se move pelo espaço de busca. A característica que se apresenta define um dos agentes condutores do processo de exploração (*exploration* e *exploitation*) do espaço de busca, que outrossim se sobrepõe às demais perspectiva, em especial sobre a Contiguidade e a Universalidade.

Sob a ótica da Universalidade, então, observa-se basicamente os movimentos em que as soluções transitam no espaço de busca, ou ao menos parte delas. Como as meta-heurísticas, sejam baseadas em solução unitária ou populacional, formam esse conjunto que se espalha por diferentes estruturas condutivas, elas constituem desta forma esta perspectiva de movimentação. Entretanto, é importante salientar que, apensar dessa premissa de observação do movimento, os sentidos também são agenciados a partir dos parâmetros que orientam as capacidades e características de suas buscas.

Resumidamente, pode-se entender a perspectiva da Navegabilidade em diferentes níveis visto que os métodos assumem multiplicidades de movimento no fluxo de otimização que desenvolvem. Opta-se por dividi-los em três conjuntos em relação ao maquinismo que os aproxima: de um lado, os i) métodos baseados em população, que segundo a literatura não apresentam

quaisquer formas de trajetória, quais sejam, Algoritmos Genéticos, Busca Dispersa e Otimização por Colônia de Formigas; do outro lado, os ii) métodos baseados em somente uma solução e que geram trajetórias, sendo eles Arrefecimento Simulado, Busca Tabu, Busca em Vizinhança Variável, Busca Local Guiada, Busca Local Iterada e as suas variantes (BLUM; ROLI, 2003; BOUSSAÏD; LEPAGNOT; SIARRY, 2013) e a exceção os iii) métodos que fazem uso de uma solução apenas e não geram trajetória, a exemplo, o GRASP.

Essa interrelação com os demais objetos do *corpus* compõe uma tendência que sinaliza sobre as técnicas analisadas em um espectro de definição das características de navegação, ou seja, parte-se da crença de que a definição de uma trajetória, ou a sua suposta descontinuidade, é na verdade uma alteração do grau de navegabilidade. Esse pressuposto faz delinear um contorno sinuoso sobre os métodos descontínuos, observando-os com alto grau de navegabilidade, como instâncias capazes de atingir regiões distantes do espaço de busca, que por não se ter constituído mecanismos de observação qualificados, acaba-se por caracterizar sua descontinuidade. Por outro lado, a criação de trajetórias, num grau menor de navegabilidade, faz por observar a efemeridade da espacialidade do movimento, que no conjunto da universalidade da busca, em análise unitária, acaba-se por observar a composição de trajetos.

2.3 Completude

O processo de resolução de problemas de otimização traz consigo um aspecto significativo sobre a escolha dos métodos de busca que pondera acerca os níveis da otimalidade por estes alcançada. A perspectiva da Completude, neste sentido, diz respeito à “capacidade” de exploração do espaço de busca – aqui referimo-nos à capacidade não no sentido qualitativo do processo de busca, mas sobre a quantificação de sua exploração, observando os horizontes alcançados pelo mecanismo movente.

A habitual associação que se faz entre a completude e a não completude das heurísticas, ainda que situada e relacionada a um contexto, no caso da Otimização Combinatória, ainda é restrita e não dá conta de certas construções que estão enraizadas nos modos de fazer a exploração do espaço de busca. Por enquanto, essa divisão que se depara é mais da espacialidade do que da ordem da exploração puramente. Além disso, nos termos até aqui expostos, essa análise binária prende-se, ainda que não anseie, a uma análise discreta que não corresponde à natureza do espaço de busca.

Os métodos completos exploram o espaço de busca de maneira exaustiva, mantendo o controle das melhores soluções encontradas, garantindo que se existe uma solução, e se esta é ótima, ela será encontrada. É válido ressaltar que a exploração exaustiva do espaço de soluções não se resume a simples enumeração de todas as soluções. Uma técnica eficaz é aquela que tenta provar quais regiões específicas do espaço de busca não contêm melhores soluções para que assim se evite percorrê-las, promovendo uma poda do espaço de busca.

Ainda que os métodos completos sejam usuais, os problemas de maior interesse na li-

teratura em Otimização Combinatória são NP-completos, isto é, a busca exaustiva pelo espaço de soluções torna-se proibitiva devido a exponencialidade temporal relativa ao tamanho das entradas, levando-os à inviabilidade prática. Entre os métodos completos, pode-se citar a Programação Linear (DANTZIG, 1951), a Programação Lógica e a Programação por Restrições (ROSSI; BEEK; WALSH, 2006).

Ademais, em diversos domínios, comumente torna-se desnecessário corroborar a otimalidade, porque uma solução satisfatória é suficiente para o problema tratado, no entanto, sem garantias de qualidade. É neste cenário que se apresenta uma alternativa de desempenho, o uso de *Métodos Incompletos*, ou seja, que não oferecem garantias quanto à qualidade das soluções encontradas. Esses métodos exploram o espaço de busca de maneira não exaustiva – e, possivelmente, estocástica – o que os torna incapazes de detectar quando o espaço de busca foi completamente explorado, o que pode jamais ocorrer.

Os métodos incompletos trazem consigo algumas prerrogativas, que acabam por caracterizá-los. Uma vez que se movem pelo espaço de busca livremente, acabam não usando regras de exploração sistemática fixa, isto é, quanto maior o tempo de busca, maiores as chances de retornar uma solução melhor. Outra característica dos métodos incompletos é que não precisam ser exaustivos, o que os propicia explorar de antemão regiões promissoras do espaço de soluções em relação aos métodos completos (URLI, 2015).

Quando se observa as diferentes facetas da completude, aprofunda-se com maior intensidade em sua natureza. A essência desta perspectiva aponta que, ao menor indício da ausência de integralidade, haverá trânsito em direção à (in)completude.

A perspectiva da Completude diz respeito à composição e interrelação que as heurísticas enunciam sobre si, resultado natural da união entre a intensificação e a diversificação, ao invés da sistematização exaustiva realizada pelos métodos completos. O espaço de busca porta uma interioridade que permite aos métodos heurísticos esgueirarem-se, portanto, não exige ser conhecido e decifrado em totalidade. E as soluções, ao estabelecerem quaisquer conexões entre si, fornecem ao investigador/busgador do espaço uma visão não acabada, mas suficiente. Pode-se ainda dizer que a completude depende tanto de um olhar sobre o que caracteriza a busca quanto sobre a própria necessidade da busca.

Na perspectiva da Completude, então, observa-se basicamente que as meta-heurísticas utilizam a necessidade do meio/problema para contar sua “narrativa”, ou ao menos parte dela. Como as heurísticas partem desta premissa da incompletude da exploração do espaço de soluções, os métodos analisados formam esse conjunto apresentando-se em diferentes facetas de movimentação, constituindo a heterogeneidade do meio, e da busca em si. Entretanto, é importante adiantar que, apesar dessa premissa de observação da incompletude do movimento e da exploração, os movimentos também são agenciados a partir da qualidade de resultados que se deseja encontrar por meio da técnica empregada.

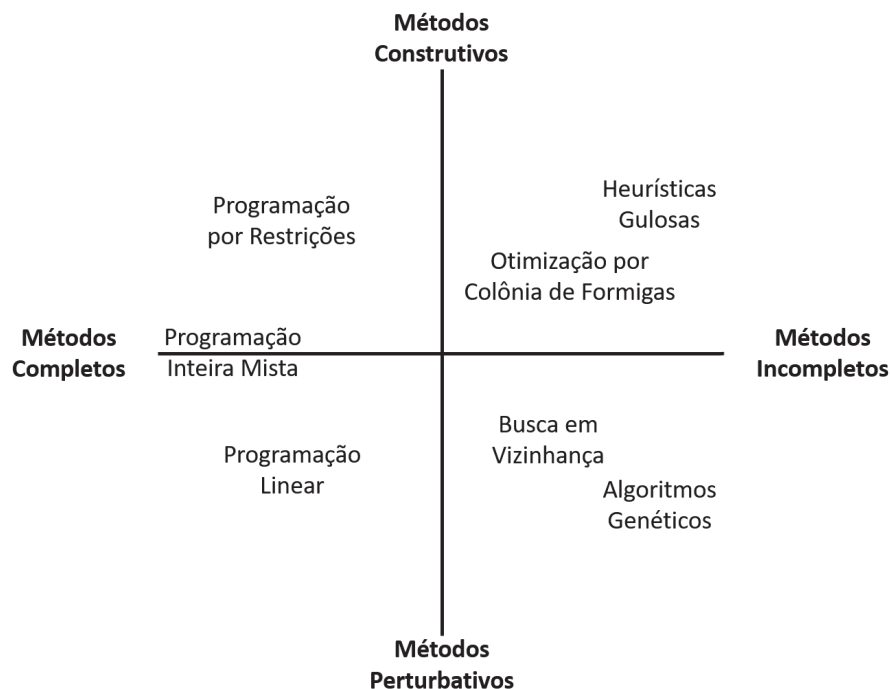
2.4 Gênese

Ao estabelecermos o olhar enquanto qualidade presente nas composições de soluções, principalmente o olhar direcionado aos mecanismos, para o método em sua estruturação, perguntamo-nos, que sentidos tal composição emprega à solução? Esse tipo de performance é tão comum nesse ambiente que nem mesmo é questionado, ou melhor, visto enquanto parte do processo. Em outras palavras, o sujeito/utilizador da heurística, por exemplo, não o faz por querer alcançar uma ideia diretamente sobre a estrutura, mas o faz porque essa é a forma de ser nesse meio de uso, como forma normatizada, aquela que se dirige ao usuário final, em prol unicamente do resultado.

As meta-heurísticas, enquanto processos, são uma das formas que se consegue concentrar de modo único as tendências sobre o elemento desconhecido na estrutura e, mais do que isso, conseguem produzir sentidos identitários próprios, de sua formação enquanto parte componente do espaço de soluções.

No domínio da classificação de métodos de busca, olhando-se pela perspectiva da composição das soluções. Isto é, faz-se necessário diferenciar como este desprendimento da estrutura e, ao mesmo tempo, operado por regras sistemáticas se apresenta. Desta forma, optou-se por dividir tais métodos em perturbativos e construtivos, em relação à recorrência com que são aproximados do conceito de Gênese. Por meio da Figura 3, pode-se entender os métodos da Gênese como opostos, estabelecendo ainda uma relação ortogonal à Completude, descrita anteriormente.

Figura 3 – Ortogonalidade entre a Gênese e a Completude



Nesta binarização, apresentam-se de um lado, os métodos perturbativos que fazem parte da rotina das heurísticas que usam a agitação de suas adjacências, podendo ser vinculados a um conjunto de heurísticas, a exemplo a mutação de soluções nos Algoritmos Genéticos, a geração de soluções vizinhas pela Busca em Vizinhança Variável, Busca Local Iterada e Busca Tabu; do outro lado, encontram-se os métodos construtivos, que operam apenas sobre soluções candidatas completas por meio da organização de seus componentes, isto é, se configuram enquanto um código comum, estrutural, a partir do qual, uma solução candidata é estendida iterativamente, definindo todas as variáveis de acordo com alguma política (BURKE et al., 2019). Em suma, contam com regras sistemáticas de combinação e seleção de componentes através de critérios e processos organizados, que manipulam os interesses metafóricos de cada meta-heurística, a exemplo, pode-se citar Otimização por Colônia de Formigas (DORIGO, 1992), GRASP (FEO; RESENDE, 1989) e Programação por Restrições (ROSSI; BEEK; WALSH, 2006).

Para Urli (2015, p. 16), os métodos perturbativos apresentam-se com uma vantagem, não importa o quanto a busca esteja estagnada, sempre fornecerão uma solução completa. No contexto projetual de uma hiper-heurística, onde a existência única de um mecanismo não dita seu comportamento, se faz necessário presumir que qualquer (des)vantagem acabe por ser compensada e/ou diminuída dentre a gama de dispositivos que compõem esse território permeável, permitindo trocas, experiências, promovendo processos de remediação e contágio entre as perspectivas, fazendo com que se encontre soluções sempre completas.

2.5 Universalidade

Nesta seção discorre-se sobre a dimensão técnica da busca no espaço, na apresentação e discussão desta nova perspectiva. A perspectiva que se apresenta aqui, sobre o uno e o múltiplo, tem forte relação com a distinção apresentada anteriormente entre os métodos de trajetória e os métodos descontínuos. A relação que existe entre eles diz respeito a suas operações no espaço de soluções, ou seja, quando se movimentam de maneira singular ou por meio de populações.

Como foi planejado metodologicamente, promove-se uma desnaturalização das estruturas meta-heurísticas: se desfaz a vínculo entre o uno e o múltiplo e observa-se apenas suas relações de grau. Com isso, apresenta-se as meta-heurísticas como se fossem planos autônomos, o que permite-nos alargar nossa compreensão sobre elas e superar o reconhecimento habitual sobre os métodos em trajetória e populacionais. Proceder-se, então, com uma leitura analítica desses planos, seguindo o objetivo de decifrar suas estruturas e reconstituir uma conexão em perspectiva conjunta, onde os meios se inter-relacionam.

No que diz respeito aos algoritmos baseados em solução única, como o Arrefecimento Simulado, Busca Tabu e a Busca Local Iterada, neles, a estrutura manipula e transforma, a cada iteração, apenas uma solução, isto é, o processo conducente da busca torna estes métodos fortemente orientados à exploração, com poder de intensificar a busca em regiões locais (TALBI,

2009).

Na Figura 4, observa-se *template* do algoritmo de alto nível para meta-heurísticas baseadas em uma solução, que consiste em um processo bifásico. O procedimento é iniciado com uma solução atual $s_t(t = 0)$ (linhas 1 e 2), onde na fase de geração (linha 4), um conjunto de soluções candidatas $C(s)$ é originado a partir da solução atual s_t . Na fase de substituição⁴ (linha 5), uma solução candidata é selecionada para substituir a solução atual, ou melhor, uma solução $s' \in C(s)$ é selecionada para ser a nova solução. Na linha 3, tem-se o condicionante do processo iterativo que se repete enquanto a condição de parada não for atingida. Por fim, na linha 7, tem-se como saída do processo uma solução s^* que se estima ser melhor que a solução de entrada s_t .

Figura 4 – *Template* de alto-nível da meta-heurística baseada em uma solução

```

1  $t = 0$ ;
2 Input:  $s_t$ 
3 while  $\neg$  CondiçãoDeParada () do
4   |   Geração (  $C(s_t)$  );
5   |    $s_t =$  Selecione (  $C(s_t)$  );
6   |    $t = t + 1$ ;
7 end
7 Output:  $s^*$ 

```

Fonte: Adaptado de (TALBI, 2009, p. 88)

A leitura sintética da Figura 4 nos permite notar como as meta-heurísticas baseadas em trajetória operam sobre o uno - em comum em seus dois mecanismos. Ambos são abstraídos em duas dimensões a partir das quais a solução é gerada e selecionada no espaço. A concretização destas dimensões através das meta-heurísticas produz enquadramentos, angulações, pontos de vista, etc. sobre os fluxos que se estabelecem sobre as paisagens de aptidão, orientando a busca e gerando trajetórias.

Na contramão dos algoritmos de solução única, tem-se os algoritmos populacionais, a exemplo: Algoritmos Genéticos (HOLLAND, 1992), Enxame de Partículas (KENNEDY; EBERHART, 1995), Otimização por Colônia de Formigas (DORIGO et al., 1991; DORIGO, 1992) e Busca Dispersa (GLOVER, 1997; GLOVER; LAGUNA; MARTI, 2003). Tais algoritmos operam sob a multiplicidade, em um conjunto de soluções a cada iteração. Cabe notar que, segundo Birattari et al. (2001, p. 3), os métodos baseados em população são tipicamente descontínuos, porém fornecem maneiras convenientes de exploração do espaço de busca, permitindo uma melhor diversificação. No entanto, seu desempenho final depende fortemente de como a população é manejada durante a busca.

O *template* do algoritmo de alto nível para meta-heurísticas populacionais, apresentado na Figura 5, também consiste de um processo bifásico. O procedimento começa com uma

⁴Também chamada de regra de transição, regra de pivoteamento ou estratégia de seleção.

população inicial $P_t(t = 0)$ (linhas 1 e 2) ⁵, na sequência, iterativamente, alternam-se as fases de geração e substituição. Na fase de geração (linha 4), uma nova população de soluções é criada, e na fase de substituição, uma nova população é selecionada a partir da população atual e da nova população $P_t \cup P'_t$ para ser a nova população. Esse processo permanece em iteração até que um critério de parada seja satisfeito (linha 3). Ao final, na linha 7, tem-se como saída do processo uma solução ou conjunto de soluções S^* que estima-se conter soluções melhores que as soluções de entrada P_t .

Figura 5 – *Template* de alto-nível da meta-heurística populacional

```

1  $t = 0$ 
2 Input:  $P_t$ 
3 while  $\neg$  CondiçãoDeParada() do
4   |   GeraçãoDaPopulação ( $P'_t$ )
5   |    $P_{t+1} =$  SelecionePopulação ( $P_t \cup P'_t$ )
6   |    $t = t + 1$ 
7 end
7 Output:  $S^*$ 

```

Fonte: Adaptado de (TALBI, 2009, p. 191)

Em nenhum dos algoritmos expostos acima foi retratada a presença de memória, ou seja, os procedimentos de geração e de seleção baseiam-se exclusivamente na população e na solução atual. A utilização de quaisquer estruturas de memória se daria pelo armazenamento de alguma forma de histórico da busca para auxiliar em sua condução. Cabe ressaltar que estes são algoritmos canônicos, ou seja, o uso de memória nesta apresentação os caracterizaria como extensões. Desta maneira, na seção 2.6 serão, em maiores detalhes, abordados os mecanismos de memória sob a perspectiva da Persistência.

Como visto acima, estes dois planos navegacionais, ou duas famílias de meta-heurísticas possuem características complementares, o que nos faz crer que uma classificação homogênea dos métodos baseados em população e dos métodos de busca em trajetória seja suficiente. Embora esta perspectiva apresente similaridades com a perspectiva da Navegabilidade, é tênue sua distinção.

Para tanto, propõe-se uma classificação homogênea. Assim seja, apresenta-se na Figura 6 uma generalização dos algoritmos acima apresentados, onde S é um conjunto de indivíduos, sendo $|S| \geq 1$, quer dizer que se permite a presença de conjuntos populacionais de um ou mais indivíduos, compreendendo assim de maneira genérica os algoritmos de busca baseado em uma solução, populações, trajetória, contínuos e descontínuos.

Até este momento, foi relatado o processo de refinamento de um *template* extensivo, pois para a concepção de uma hiper-heurística que opera em um espaço sem distinção sobre

⁵Cabe notar que a Otimização por Colônia de Formigas inicia o processo de solução com uma população vazia ou parcialmente definida.

Figura 6 – *Template* de alto-nível para meta-heurísticas

```

1  $t = 0$ 
2 Input:  $s_t$ 
3 while  $\neg$  CondiçãoDeParada() do
4   | FaseGeracional( $s'_t$ )
5   |  $s_{t+1} =$  FaseSubstitutiva( $s_t \cup s'_t$ )
6   |  $t = t + 1$ 
   end
7 Output:  $s^*$ 

```

Fonte: Elaborado pelo autor

o uno e o múltiplo, necessita-se de um algoritmo agnóstico/generalista, especialmente na concepção de grau de universalidade, representado pelo $|S|$. Essa decisão torna o processo mais homogêneo e representativo em relação à movimentação no espaço de soluções, contemplando buscas circunscritas aos limites característicos de cada forma de exploração.

2.6 Persistência

A discussão sobre a perspectiva da Persistência inicia pelo debate sobre o papel das características do passado da busca para a construção de seu futuro⁶. Durante a construção da pesquisa, verificou-se que os estudos voltados para o tema memória apresentam-se majoritariamente sob propostas de armazenamento e de permanência, entretanto encontramos sentidos de oposições, constituindo uma zona permeável de problematização, onde uma toca o outra, estabelecendo uma dialética.

Desta forma, inspirado pelos questionamentos de Birattari et al. (2001), e sabendo da dualidade com que se olha a persistência e/ou efemeridade, norteia-se essa seção. Ao olharmos para as meta-heurísticas, suas estruturas e processos internos apresentados no capítulo 3, pôde-se compreender os códigos que se formam e que enunciam os modos de agir da persistência e da efemeridade. Ao nos aproximarmos destes estudos, e de outros, encontramos impasses paradoxais comuns ao tema e que configuram dimensões duais, como recência e frequência, qualidade e influência. Num sentido mais amplo, estas dimensões agem em prol da influência conducente da busca.

Apropriando-se das quatro dimensões apresentadas por Martins e Ribeiro (2006, p. 108), sejam elas, recência, frequência, qualidade e influência, têm-se na perspectiva da Persistência um delinear do movimento que passa a ser regido pela ordem das estruturas de memória, pensando-as como o contexto, o meio e as condições moventes que originam o procedimento. Em termos gerais, recência/frequência e qualidade/influência podem ser vistas como dimensões complementares. Memórias baseadas em recência e frequência registram informações tempo-

⁶Aqui nos referimos ao uso da memória adaptativa, em contraste com a memória rígida presente, por exemplo no *Branch and Bound*

rais sobre o uso de elementos de memória específicos, enquanto a qualidade e influência classificam as soluções em termos de sua importância para representar características promissoras da solução (ou regiões no espaço da solução) e o impacto de certas escolhas na qualidade das soluções produzidas (REGO; GLOVER, 2006)

Esta binarização da lógica da memória, nativa do universo maquínico, demonstra um paradoxo sobre as compreensões de sentido, simplesmente por serem, por vezes, não representadas de maneira explícita. Na representação explícita de memória, as soluções são registradas em sua completude, consistindo tipicamente em soluções de elite visitadas durante a busca ou de maneira atributiva, registrando-se os atributos que mudam ao passar de uma solução para outra.

Deste lado, tem-se a Busca Tabu como exemplo de meta-heurística que faz uso explícito de estruturas de memória (MARTÍ; REY; CARLOS, 2009). Nela, a memória de curto prazo é usada para evitar ciclos, proibindo a revisão de soluções encontradas recentemente, enquanto a memória de longo prazo é usada para diversificação e intensificação. Ainda, que o uso de memória seja algo de longa data, tendo surgido em meados dos anos sessenta, o termo *programação de memória adaptativa* foi cunhado somente em 1998 por Taillard et al. como um aprimoramento da Busca Tabu para resolver o problema de roteamento de veículos, motivado pelos trabalhos de Glover sobre a Busca Tabu e a substituição de restrições (GLOVER, 1977; GLOVER, 1989; GLOVER, 1990). Outro exemplo, na Otimização por Colônia de Formigas têm-se o uso de memória, onde as soluções previamente visitadas são mantidas em uma matriz de trilhas de feromônio, que são utilizadas para influenciar a construção de novas soluções (BLUM; ROLI, 2003).

No outro extremo, existem os algoritmos sem memória, ou com memória implícita, ou seja, que não extraem informações durante a busca, mas que detém informações que de uma maneira ou outra conduzem-na. Como representantes dessa classe tem-se a Busca Local, o GRASP e o Arrefecimento Simulado que não usam estruturas de memória para influenciar a direção de busca futura e, portanto, são considerados algoritmos sem memória explícita.

Acionamos Birattari et al. (2001) como um exemplo do ponto de vista de pesquisa direcionado para a memória. Nesta visão, sobre a ausência de memória, a Busca Local Iterada pode ser tida como um algoritmo com memória adaptativa, ainda que sua experiência com memória seja pequena, acaba por fazer uso na escolha da melhor solução encontrada até o momento para a etapa de modificação. Num sentido mais amplo, também pode-se entender a população do Algoritmo Genético como uma forma de memória recente.

O princípio por trás da programação de memória adaptativa é que boas soluções podem ser construídas combinando diferentes componentes de outras boas soluções, ou seja, refere-se aos algoritmos que, de alguma forma, acabam por utilizar algum tipo de memória para identificar características pertinentes a sua busca. Em resumo, o emprego de estruturas de memória, sejam explícitas ou implícitas, caracteriza o uso da experiência da exploração durante o processo de busca.

2.7 Contiguidade

A contiguidade se aplica às relações de movimento e fluxos, circulação e conectividade sobre o espaço de busca. Entende-se estas relações como um aumento da porosidade, intensificando a conectividade e acessibilidade entre territórios do espaço de busca, correspondendo ao aumento potencial da exploração (*exploration* e *exploitation*). O que pode explicar algumas das variáveis identificadas empiricamente em relação à exploração dos potenciais pontos de interesse, ou seja, as diferenças entre espaços permitem que se estruture a busca em função da máxima exploração de oportunidades de cada região para a obtenção de melhores soluções.

A questão da contiguidade é considerada devido ao fato de corresponder a uma estrutura de integração espacial que equivale aos potenciais de obtenção de melhores soluções. O espaço de busca é formado por soluções situadas em diferentes regiões, embora possa ser contíguo, está sujeito a um “desinteresse” da busca, onde existem restrições que impõem uma descontinuidade de sua espacialidade. Desta forma, os aspectos morfológicos, sobretudo da estrutura do espaço soluções despertam interesse por identificar hierarquias, conectividades e potenciais de movimento. Da morfologia do espaço observam-se três aspectos, sendo dois os principais:

i) *aspecto intrarregional*, onde os fluxos de movimento intensificam a busca de soluções, comutando-as em uma mesma região de maneira endógena. Este aspecto é claramente observado em grande parte dos algoritmos meta-heurísticos canônicos que fazem uso apenas de uma única estrutura de vizinhança definindo os tipos de movimentos permitidos, a exemplo pode-se citar a Busca Tabu e o Arrefecimento Simulado. Em outras palavras, a topologia da paisagem de aptidão não é modificada durante a execução do algoritmo, pois não há troca de mecanismos de geração de vizinhança.

No caso da Busca Local Iterada que usa ao menos duas estruturas de vizinhança diferentes, o processo de busca é realizado até atingir um ótimo local, e em tal situação o movimento de troca de estrutura de vizinhança acaba por impulsionar a busca para outro ponto do espaço de soluções. Entretanto, aguardar a estagnação em um ótimo local acaba muitas vezes por não impulsionar adequada ou suficientemente a busca para uma nova região do espaço de soluções.

ii) *aspecto extrarregional* é visto onde os fluxos diversificam os movimentos no espaço de forma exógena. Este aspecto emerge de uma perspectiva onde torna-se relevante a elaboração de novas estruturas de vizinhança que possam ser aplicadas a uma diversidade de problemas, dessa forma Vaessens, Aarts e Lenstra (1998) conceberam o termo hipervizinhança. Nessa direção, Balas e Vazacopoulos (1998) propuseram o conceito de árvore de vizinhança, utilizado na Busca Local Guiada (*Guided Local Search*).

Na visão de Birattari et al. (2001, p. 4), os operadores de cruzamento e mutação do Algoritmo Genético podem ser entendidos como processos de mudança de vizinhança, ou seja, interpretados como formas de movimentação na hipervizinhança. Todavia, para aplicações específicas, a estrutura de vizinhança necessita ser mais complexa, como exemplo, as cadeias de ejeção, definidas e estudadas por Rego e Roucairol (1995) e Glover (1996), embora esses casos

sejam mais atípicos.

Destarte, pode-se inferir que a escolha entre diferentes estruturas de vizinhança acabe por impulsionar a busca para regiões de maior interesse. Fundamentados neste princípio, Mladenović e Hansen (1997) propuseram a Busca em Vizinhança Variável, que utiliza um conjunto de estruturas de vizinhança, possibilitando, assim, diversificar a busca pela troca entre diferentes paisagens de aptidão (BLUM; ROLI, 2003).

Fruto da fusão, ou hibridização dos dois aspectos da morfologia da espacialidade, emerge um terceiro, no qual a dualidade entre os movimentos internos e externos decorre um novo aspecto espacial que, alterando padrões de acessibilidade, conectividade e integração, acaba por estruturar novos movimentos, chamado de iii) *aspecto inter-regional*, onde se estabelecem interfaces de cooperação, ou de intensificação e diversificação simultaneamente entre regiões. Este aspecto da dimensão surgiu com o intuito de acrescer a capacidade de busca por todo o espaço de soluções a fim de permitir maior precisão aos algoritmos intrarregionais. Uma grande variedade pode ser encontrada na literatura, sob diversas denominações, como, Algoritmos Meméticos, Algoritmos Lamarckianos, Algoritmos Baldwinianos, Algoritmos Culturais, Busca Local Genética, as próprias hiper-heurísticas, entre outros (GOLDBERG; VOESSNER, 1999; DAVIS, 1991).

O aspecto inter-regional emerge como uma reorganização das interfaces tradicionais entre as diferentes formas de busca presentes nas meta-heurísticas. Este processo interfere diretamente no reordenamento do espaço, concentrando ou dispersando o uso de regiões e influenciando preferências locais.

2.8 Aptidão

Ao definirmos a perspectiva da Aptidão, retoma-se a base conceitual sobre as heurísticas, e sua característica aproximativa sobre os resultados, esta característica é levada em consideração pois a medida de aptidão das soluções é usualmente uma parte custosa do processo de busca. É neste fator sobre a relação quantitativa e qualitativa que se estabelece esta perspectiva por meio de duas facetas: a primeira retratada por Blum e Roli (2003) pelo modo como as meta-heurísticas utilizam a função objetivo, ou seja, se modificam ou não a avaliação dos estados de busca durante sua execução, como um objeto manipulado pela medida que se aplica sobre ele. A segunda faceta faz-se notar nas nuances tratadas pelo manuseio das soluções, formando dentro da avaliação um componente modificável, resultado do descontínuo processo de movimentos pelos quais a busca avança no espaço.

Esta segunda relação, mais complexa, entre o valor objetivo – interno à solução – e o processo do movimento, forma um todo dissociável, muitas vezes tão intrinsecamente ligado à estrutura das meta-heurísticas que só pode-se entendê-lo enquanto processo, numa tessitura espacial conjunta.

Essa perspectiva é, por si só, quantificadora e qualificadora, nela se tem permeado os

maquinismos mensuradores que auxiliam, também, o desenho das paisagens de aptidão. Como exemplo desta dinamicidade, tem-se a Busca Tabu, que é introduzida por Birattari et al. (2001) em seu trabalho como uma nova perspectiva de entendimento sobre função objetivo dinâmica. Tal entendimento se dá em função da Busca Tabu restringir alguns pontos de busca do espaço de soluções, correspondendo a valores de função objetivo elevados.

Sob a meta-heurística Algoritmo Genético, entende-se que as características pelas quais as soluções candidatas são escolhidas, isto é, por elitismo, proporcionalidade, entre outras características, determina a aptidão analisada e obtida pela busca, assim como o ponto de corte populacional.

Um outro exemplo singular é o método chamado Breakout, proposto por Morris (1993) para a solução de problemas de coloração de grafos e de satisfatibilidade. O conceito por trás do Breakout é penalizar a inclusão de características que venham a contribuir negativamente para o valor da função objetivo. Com base neste método, Voudouris, Tsang e Alsheddy (1995) propuseram a Busca Local Guiada cuja proposta é não ficar preso em ótimos locais.

De maneira sintética, pode-se entender a aptidão como fatores interno e externo à solução. Isto é, não se observa apenas como a função objetivo mede e modifica os resultados, mas como os resultados medidos modificam a busca, seja pela ordenação ou escolha de novas soluções, espaços, etc.

2.9 Considerações sobre o capítulo

Este capítulo consolida através de seis perspectiva modelares a compreensão dos processos de movimentação da busca agenciados pelas heurísticas no espaço de soluções. As perspectiva foram construídas como formas de entender o que está oculto em nosso tempo de trabalho – como extensões de nosso meio de pesquisa e operação – pois todo artefato que se constrói é uma extensão que modifica nosso ambiente e a nossa própria percepção sobre o objeto de estudo. De forma sintética, as perspectivas podem ser entendidas como:

- Navegabilidade – continuidade observável da busca, numa relação de grau, onde se estabelecem maiores ou menores níveis de trajetória;
- Completude – intensificação e diversificação como formas de exploração, determinando assim a verificação completa ou não do espaço de soluções;
- Gênese – características de construção e perturbação para a criação da solução;
- Universalidade – delimitando o tamanho do conjunto de soluções envolvidas no processo;
- Persistência – presença direta e/ou indireta de estruturas de memória, exercendo os papéis de recência/frequência e qualidade/influência.
- Contiguidade – estabelece relações de movimentos e fluxos, circulação e conectividade do espaço de busca por meio das estruturas de vizinhança;

- Aptidão – como as avaliações de quantidade e/ou qualidade influem na busca, seja pela ordenação ou escolha de novas soluções.

Finaliza-se este capítulo com a última perspectiva, referente à Aptidão. Considera-se essa a última, no sentido de algo encontrado, pois neste processo de conectar pontos não se pode definir os limites desse desenho investigativo que se constitui na tese. Sua importância advém de modo que o mais importante é conseguir delinear uma imagem deste espaço, mostrando seus modos de comportamento e de que forma eles nos ajudam a pensar o nosso objeto principal da tese.

À medida que as perspectivas evoluíram durante a pesquisa, se desprendendo e se reconectando ao mesmo tempo, especialmente quando se deixa as técnicas heurísticas e embrenha-se em meio às linhas de tensionamento que se constituíram através de seus mecanismos, chega-se a um ponto de análise profundo o suficiente para se observar como o processo de modificação de soluções emprega movimentos de espalhamento no espaço de busca. Esta constituição do movimento e da ocupação, embora preservem ou até mesmo definam as naturezas da busca, leva-nos ao encontro de uma grande quantidade de formadores de paisagens de aptidão.

Seria ingênuo pensar que qualquer passo dado, indiferente de direção não promova quaisquer modificações, e pensar que esses órgãos estruturais permanecem neutros à espera de nossas instruções sem modificar seu espaço. Essas extensões operam uma mudança no meio e na forma como realizamos quaisquer buscas no espaço de soluções.

Considerando este cenário, a revisão destes conceitos apontou que a representação dos mecanismos moventes no espaço constitui em um viés analítico desafiador para a compreensão das meta-heurísticas apresentadas no capítulo seguinte. Nessa perspectiva, o emprego desta abordagem epistêmica se destaca como uma estratégia, a qual busca compreender as diferentes perspectivas que compõem as meta-heurísticas a fim de identificar comportamentos em comum.

3 MECANISMOS HEURÍSTICOS

Este capítulo fornece uma visão sobre as áreas de trabalho estudadas na tese. Existem duas principais áreas relevantes que são as meta-heurísticas e as hiper-heurísticas. A primeira área diz respeito às meta-heurísticas analisadas na intenção de fornecer ao leitor uma compreensão acerca dos métodos utilizados no decorrer da pesquisa, e não detalhar o funcionamento particular de cada uma das técnicas empregadas na construção do texto.

O capítulo perfaz um conjunto de 8 técnicas que contemplam a gama de características das famílias de meta-heurísticas, divididas em algoritmos estocásticos, evolutivos, físicos e enxames. Essa delimitação do *corpus* de análise não foi projetada de maneira a inserir cada uma das técnicas no propósito do trabalho. Pelo contrário, como um trabalho exploratório, foi-se imergindo sobre os devires das técnicas que se fazem presentes no capítulo. Este processo de identificação é unido no Capítulo 4 no projeto dos autômatos de cada meta-heurística e consequentemente das expressões regulares que as definem.

A segunda área de trabalho contempla as hiper-heurísticas, que são definidas como heurísticas que buscam heurísticas no metaespaço, em vez do espaço de soluções diretamente. As hiper-heurísticas podem ser classificadas em dois tipos principais. O primeiro corresponde aos métodos construtivos, que criam heurísticas a partir de conjuntos de potenciais componentes. O segundo tipo seleciona uma heurística, ou sequência, de um conjunto pré-definido.

A contribuição deste capítulo é mostrar as possibilidades de construções hiper-heurísticas guiadas pelas concepções levantadas no Capítulo 2 sobre as perspectivas modelares que conduzem a busca. O objetivo deste capítulo é colocar esta contribuição em contexto e fornecer informações sobre outras abordagens, demonstrando os avanços na área e conduzindo os desenvolvimentos que estão condensados no Capítulo 4, por meio do modelo proposto.

Vendo os mecanismos heurísticos enquanto movimento, a ação em torno de sua caracterização é de tentar autenticar o seu processo constitutivo por meio da apreensão das meta-heurísticas, já que seu *modus operandi* revela-se também por meio de seu funcionamento. Para a realização desse processo de reconhecimento, fez-se uso de uma estratégia que norteia a construção desse capítulo e ajuda a compor a proposta metodológica da tese, o delinear das constelações – perspectivas modelares. A provocação é que, ao acionar essa visão, se crie um desenho investigativo que busque pensar as meta-heurísticas enquanto fundamento, e não apenas como experiências, permitindo ver assim sua multiplicidade enquanto proposta científica.

3.1 Algoritmos Genéticos

Os Algoritmos Genéticos pertencem a uma família de modelos computacionais inspirados na teoria da evolução, cuja estratégia adaptativa é uma técnica de otimização global. Este processo de otimização é fortemente embasado na genética populacional, sob os princípios da hereditariedade e das frequências genéticas, bem como a compreensão mendeliana de sua estru-

tura (cromossomos, genes e alelos) e dos mecanismos de cruzamento e mutação (BROWNLEE, 2011).

As primeiras pesquisas relacionadas à computação evolucionária surgiram em meados da década de 1950, porém, seu desenvolvimento efetivo ocorreu a partir dos anos 1970, pelas pesquisas de John Holland¹ – a quem é atribuído o título de criador da técnica (HOLLAND, 1992; GOLDBERG, 1989). Os Algoritmos Genéticos foram a primeira forma de algoritmos evolutivos a serem copiosamente utilizados em um conjunto diversificado de áreas, indo desde a Pesquisa Operacional à Inteligência Artificial, abrindo margem aos algoritmos evolutivos que são rotineiramente usados como ferramentas de busca e otimização para engenharias e aplicações científicas (WHITLEY; COLLINS, 2012).

O funcionamento do Algoritmo Genético é baseado em um procedimento iterativo que simula o processo evolutivo de uma população, por meio de estruturas sujeitas às forças competitivas prescritas no princípio Darwiniano de sobrevivência do melhor adaptado (DARWIN; BEER, 1998). De acordo com este princípio, os organismos de uma população que se adaptam melhor ao meio em que vivem possuem maiores chances de sobrevivência e reprodução, em contrapartida, os indivíduos menos adaptados acabam por ter menores chances de descendência. Do processo de (re)combinação de boas características, se produz, a cada geração, descendentes mais aptos e adaptados ao meio (GOLDBERG, 1989).

Esses algoritmos simulam o processo de evolução natural a partir de uma população de soluções iniciais. No processo de otimização, cada indivíduo/solução codificado, é projetado por estruturas de dados compostas por um conjunto de genes, chamado de cromossomo. Cada solução contribui com seu material genético (genótipo) proporcional à adequação de seu genoma (fenótipo) em seu ambiente. Pode-se, assim, ver cada cromossomo como um ponto no espaço de soluções, ou seja, uma possível representação de uma solução para um dado problema.

Do processo iterativo dos Algoritmos Genéticos, criam-se as próximas gerações por meio de um processo de acoplamento que envolve o cruzamento de indivíduos na população aliado ao processo de mutação, que se caracteriza pela introdução de erros de cópia aleatória. O processo de introdução de erros de cópia pode resultar em uma adaptação melhor entre os fenótipos dos indivíduos em uma população e seu ambiente. Após a troca de informação entre indivíduos, as melhores soluções e mais adaptadas são selecionadas, servindo de base para a próxima geração. O resultado desse processo será o aumento da adaptação global da população a cada nova geração.

Analogamente ao processo evolutivo biológico das espécies, os Algoritmos Genéticos nutrem informações relevantes sobre o meio em que se encontram, retendo-a durante o período de adaptação. Essas informações, posteriormente, minimizam o espaço de busca, restringindo-o, o que auxilia na geração de novas e melhores soluções (LOBO, 2005).

¹A formulação mais completa dos primeiros trabalhos de John Holland foi publicada em 1975, em seu livro *Adaptation in Natural and Artificial Systems*

Pode-se observar na Figura 7 o pseudocódigo do Algoritmo Genético, que consiste na geração da população inicial e a avaliação da aptidão de seus indivíduos. O critério de parada garantirá o fim do algoritmo, dada uma condição. Após esses passos, realiza-se a seleção dos indivíduos que serão submetidos às etapas de cruzamento e mutação. Dados esses processos, cada indivíduo será novamente avaliado e a nova população criada é atualizada.

Figura 7 – Pseudocódigo do Algoritmo Genético

```

Input:  $Populacao_{tamanho}$ ,  $Problema_{tamanho}$ ,  $P_{cruzamento}$ ,  $P_{mutacao}$ 
Output:  $S^*$ 
 $Populacao \leftarrow IniciePopulacao (Populacao_{tamanho}, Problema_{tamanho})$ ;
 $AvaliePopulacao (Populacao)$ ;
 $S^* \leftarrow MelhorSolucao (Populacao)$ ;
while  $\neg CondicaoParada ()$  do
     $Pais \leftarrow SelecionePais (Populacao, Populacao_{tamanho})$ ;
     $Filhos \leftarrow \emptyset$ ;
    foreach  $Pai_1, Pai_2 \in Pais$  do
         $Filho_1, Filho_2 \leftarrow Cruzamento (Pai_1, Pai_2, P_{cruzamento})$ ;
         $Filhos \leftarrow Mutacao (Filho_1, P_{mutacao})$ ;
         $Filhos \leftarrow Mutacao (Filho_2, P_{mutacao})$ ;
    end
     $AvaliePopulacao (Filhos)$ ;
     $S^* \leftarrow MelhorSolucao (Filhos)$ ;
     $Populacao \leftarrow Substitua (Populacao, Filhos)$ ;
end
return  $S^*$ ;

```

Fonte: Adaptado de (BROWNLEE, 2011, p. 93)

Na literatura, são encontrados alguns trabalhos que apresentam uma ampla revisão sobre os algoritmos genéticos, destacando-se o trabalho de Dao, Abhary e Marian (2017) e Lee (2018) que apresentam o estado da arte para os períodos até 2014 e de 2007 a 2017, respectivamente.

Whitley e Collins (2012) apresentam uma revisão sobre modelos e métodos dos Algoritmos Genéticos, analisando estruturas e características de operação, numa visão crítica sobre as vantagens e falhas que por vezes se apresentam nesta técnica. Já Coello, Segura e Miranda (2016), fornecem uma visão geral da Computação Evolutiva trazendo uma discussão sobre a teoria da evolução natural e as adoções realizadas por meio dos paradigmas computacionais que inspiram esta família de algoritmos. Widiartha, Anjarwani e Bimantoro (2017) introduzem uma versão do Algoritmo Genético multielementos aplicado ao TSP, na intenção de considerar características do problema durante o processo de avaliação, não fazendo da distância o único fator no cálculo da função objetivo.

Xueyuan (2018) fornece uma versão melhorada do Algoritmo Genético, também aplicada ao Problema do Caixeiro Viajante para prevenir o problema da convergência prematura, por meio do agrupamento de soluções em prol da diminuição da multiplicidade dos grupos. Seguindo no conjunto de melhoramentos ao algoritmo canônico, Gupta, Choubey e Choubey (2017) apresentam uma forma elitista de seleção da população. Na mesma linha Liu e Li (2018) introduzem uma forma de inicialização da população baseada em permutação gulosa, que fornece resultados melhores que a simples aleatorização característica do método. Já Akter et al. (2019) apresentam uma nova técnica de cruzamento aplicado ao TSP, que analisa os custos das estruturas dos cromossomos pais na geração de descendentes de melhor qualidade.

Outra pesquisa voltada à comparação de métodos aplicada ao TSP foi desenvolvida por Kureichik e Logunova (2018). A pesquisa analisa a degradação da população a cada geração quando submetidos a diferentes operadores de cruzamento. Como visto, não são raros os casos de modificações e hibridizações aplicadas aos Algoritmos Genéticos, como exemplo a Evolução Diferencial em Abidin (2018), e Busca Tabu Genética em Ge et al. (2019).

Noutros domínios, podemos encontrar em Gamarra, Mitre-Ortiz e Escalante (2019) a aplicação dos Algoritmos Genéticos em segmentação celular automática, já Tamimi et al. (2019) apresenta uma solução para melhoramento dos tempos das luzes de semáforo. Já em Morteza-nezhad e Daneshifar (2019) vê-se a aplicação no agrupamento em Big-Data. Em Yiqiu Yiqiu, Xia e Junwei (2019) a aplicação dos AG ao agendamento de tarefas em computação em nuvem.

3.2 Arrefecimento Simulado

Desenvolvido na década de 1980 por Kirkpatrick, Gelatt e Vecchi (1983) e de maneira independente por Černý (1985), a meta-heurística Arrefecimento Simulado (*Simulated Annealing* - SA) remonta os anos 1950 com o trabalho de Metropolis et al. (1953) sobre o processo de arrefecimento utilizado na metalurgia para obtenção de um estado sólido “bem-ordenado” de energia mínima (evitando as estruturas “metaestáveis”, características dos mínimos de energia) (KIRKPATRICK; GELATT; VECCHI, 1983; HENDERSON; JACOBSON; JOHNSON, 2003).

O processo fundamenta-se no lento esfriamento do metal, pois, o rápido esfriar acaba por não permitir que os átomos tenham a chance de se ajustar em uma rede e são congelados em uma configuração aleatória, resultando em um metal quebradiço (SUMAN; KUMAR, 2006). Ao diminuir a temperatura de maneira lenta e controlada, os átomos recebem tempo suficiente para se configurarem em um cristal mais forte, reduzindo seus defeitos (LUKE, 2013).

Analogamente, cada configuração de uma solução no espaço de busca representa um nível de energia interna do sistema. O aquecimento do sistema resulta em aumento da energia dos átomos, e o conseqüente relaxamento dos critérios de aceitação das amostras retiradas do espaço de busca, ou seja, permite-se que os átomos se movam livremente (SUMAN; KUMAR, 2006; HENDERSON; JACOBSON; JOHNSON, 2003). À medida que o sistema é arrefecido, o critério de aceitação das amostras é reduzido para se concentrar na melhoria dos movimentos,

permitindo com que uma nova configuração de baixa energia seja descoberta e explorada. Uma vez que o sistema esfriou, a configuração representará uma amostra próxima de um ótimo global (INGBER, 1993; BROWNLEE, 2011).

Na Figura 8, vê-se o pseudocódigo do Arrefecimento Simulado que, a partir de uma solução inicial, gera iterativamente um vizinho S_i com base na solução corrente S' . A cada geração, um novo vizinho tem sua função objetivo testada, isto é, $\Delta = f(S') - f(S_i)$. Se $\Delta < 0$, então o método aceita a solução S' como a nova solução atual, no caso de um problema de maximização. Caso contrário, se $\Delta \geq 0$, a solução vizinha candidata também poderá ser aceita, mas neste caso, com uma probabilidade de $\exp(-\Delta/T)$, onde T é um parâmetro da temperatura, responsável por regular a chance de aceitação de soluções com custo superior (GRANVILLE; RASSON; KRIVÁNEK, 1994). A temperatura, inicialmente elevada, é gradativamente diminuída por uma razão de resfriamento, fazendo com que o procedimento tenha maiores chances de escapar de ótimos locais e, à medida que a temperatura se aproxima de zero, o algoritmo comporta-se como o método da subida, pois, diminui a probabilidade de se aceitar movimentos de piora. O procedimento é finalizado quando a temperatura aproxima-se de zero e nenhuma solução que piore o valor da solução atual é mais aceita, isto é, quando o sistema está estável (INGBER, 1993). A solução obtida nesta situação evidencia a determinação de um ótimo local (KOULAMAS; ANTONY; JAEN, 1994).

Figura 8 – Pseudocódigo do Arrefecimento Simulado

```

Input: TamanhoSolucão, iteracaomax, temperaturamax
Output: S*
S' ← CriacaoDaSolucãoInicial (TamanhoSolucão);
S* ← S';
for i = 1 to iteracaomax do
    Si ← CriacaoDaSolucãoVizinha (S');
    temperaturaatual ← CalculoDaTemperatura (i, temperaturamax);
    if Custo (Si) ≤ Custo (S') then
        S' ← Si;
        if Custo (Si) ≤ Custo (S*) then
            S* ← Si;
        end
    else if Exp (  $\frac{\text{Custo}(S') - \text{Custo}(S_i)}{\text{temperatura}_{\text{atual}}}$  ) > Rand () then
        S' ← Si;
    end
end
return S*;

```

Fonte: Adaptado de (BROWNLEE, 2011, p. 170)

O algoritmo de Arrefecimento Simulado vem sendo ao longo dos anos melhorado por diversos pesquisadores em aplicações do TSP (DU; SWAMY, 2016). Zhao, Xiong e Shu (2015) usaram um aprimoramento do SA para resolver o problema do caixeiro viajante, cujos resulta-

dos indicaram que a proposta pode obter melhores soluções para a maioria dos benchmarks da TSPLib.

Zhan et al. (2016) apresentam um novo algoritmo chamado algoritmo de arrefecimento simulado baseado em lista, para resolver o problema do caixeiro viajante. Lin, Bian e Liu (2016) propuseram uma hibridização do arrefecimento simulado baseado em busca tabu. Em ambas as pesquisas, os algoritmos demonstraram melhor eficiência e precisão. Liu, Xiong e Liu (2009) introduziram o SA integrado à busca tabu para obter melhores soluções, onde a temperatura foi reduzida de forma adaptativa com uma função de controle de temperatura.

Baseado nas arestas das melhores soluções vizinhas, Li, Zhou e Zhang (2011) apresentam um modelo de vizinhança probabilística em conjunto com SA. Peisheng (2009) apresenta as vantagens de integração do SA com ACO para acelerar a convergência e obter as melhores soluções aproximadas. Mohsen (2016) propõe um algoritmo híbrido de otimização de colônias de formigas baseado em arrefecimento simulado. Já a abordagem de Ezugwu, Adewumi e Frîncu (2017) traz uma versão do SA baseado na busca de organismos simbióticos, cujos resultados fornecem vantagens em termos de convergência, tempo médio de execução e desvios percentuais.

3.3 Busca Dispersa

Os conceitos da Busca Dispersa (*Scatter Search*) e da Reconexão de Caminhos (*Path Relinking*), foram originalmente propostos por Glover no final da década de 1990 (GLOVER, 1997). Compete assinalar que os preceitos dos métodos foram apresentados pela primeira vez na década de 1970 em (GLOVER, 1977) e fundamentaram-se em formulações que remontam os anos 1960 (GLOVER, 1963), combinando restrições e regras de decisão.

A Busca Dispersa opera por meio de uma população, selecionando e combinando soluções “boas” do conjunto de referência para fornecer soluções iniciais para um procedimento de melhoria e para atualizar o conjunto de referência, incorporando soluções diversificadas e de alta qualidade (TALBI, 2009).

Da perspectiva de operacionalização por meio de uma população, pode-se inserir a Busca Dispersa na categoria dos Algoritmos Evolutivos (GLOVER; LAGUNA; MARTI, 2003). Todavia, em conjunto com a Reconexão de Caminhos difere-se de outros Algoritmos Evolutivos, fornecendo princípios unificadores para juntar soluções baseadas em construções de caminhos generalizados, por meio de modelos estratégicos (BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Tem-se na Figura 9 o pseudocódigo da Busca Dispersa, onde observa-se os cinco passos do algoritmo. São eles:

1. Método de Diversificação é responsável pela construção do conjunto de soluções iniciais para diversificar a busca selecionando soluções de alta qualidade.

2. Método de Melhoria serve para transformar uma solução em uma ou mais soluções aprimoradas, em geral, aplicando um procedimento de busca local.
3. Método de Atualização do Conjunto de Referência opera criando um conjunto de referência a partir das soluções melhores, mantendo soluções tanto diversas como de alta qualidade.
4. Método de Geração do Subconjunto opera sobre o conjunto de referência, produzindo um subconjunto de soluções base para combinação de soluções. Pode-se equivaler este método ao papel do operador de seleção nos Algoritmos Evolutivos, diferenciando-se pelo determinismo do operador, que nos Algoritmos Evolutivos é estocástico (TALBI, 2009).
5. Método de Combinação de Soluções transforma um subconjunto de soluções produzidas no método de geração do subconjunto em uma ou mais soluções combinadas. Pode-se entender o Método de Combinação de maneira análoga ao operador de cruzamento nos Algoritmos Evolutivos onde dois ou mais indivíduos são recombinados (TALBI, 2009).

Figura 9 – Pseudocódigo da Busca Dispersa

```

Input: ConjuntoDivtamanho, ConjuntoReftamanho
Output: ConjuntoRef
ConjuntoInicial ← ConstrucaoSolucaoInicial (ConjuntoDivtamanho);
ConjuntoRefinado ←  $\emptyset$ ;
for  $S_i \in$  ConjuntoInicial do
    | ConjuntoRefinado ← BuscaLocal ( $S_i$ );
end
ConjuntoRef ←
    SelecaoConjuntoDeRefInicial (ConjuntoReftamanho);
while  $\neg$  CondicaoDeParada () do
    | Subconjuntos ← SelecaoSubconjunto (ConjuntoRef);
    | ConjuntoCandidato ←  $\emptyset$ ;
    | for  $Subconjunto_i \in$  Subconjuntos do
    | | CandidatosRecombinados ← Recombinacao ( $Subconjunto_i$ );
    | | for  $S_i \in$  CandidatosRecombinados do
    | | | ConjuntoCandidato ← BuscaLocal ( $S_i$ );
    | | end
    | end
    | ConjuntoRef ← Selecao (ConjuntoRef, ConjuntoCandidato,
    | ConjuntoReftamanho);
end
return ConjuntoRef;

```

Fonte: Adaptado de (BROWNLEE, 2011, p. 67)

São diversas as aplicações da Busca Dispersa e da Reconexão de Caminhos encontradas na literatura, apresentando-se em diferentes problemas. Greistorfer (2003) e Russell e Chiang

(2006) aplicaram a Busca Dispersa ao problema padrão de roteamento de veículos, com objetivo de obter novas soluções a partir de soluções combinadas. Tang, Zhang e Pan (2010) apresentam a aplicação da Busca Dispersa no problema de roteamento do veículo considerando os custos de carregamento. Zhang, Chaovaitwongse e Zhang (2012) propuseram uma nova abordagem da técnica para o problema de roteamento de veículos com tempo de viagem, coletas e entregas simultâneas, incorporando um novo método de programação com restrições. No trabalho os autores desenvolvem paralelamente outra abordagem baseada em Algoritmo Genético como referência para comparação de desempenho. Em análise comparativa, a Busca Dispersa apresenta desempenho superior ao Algoritmo Genético.

Xu e Qu (2012) apresentaram uma hibridação da Busca Dispersa para o problema de roteamento com restrições de atraso. A abordagem desenvolvida faz uso de Busca Tabu e Busca em Vizinhança Variável para intensificar a busca. Os resultados obtidos são mais eficientes em comparação a outros algoritmos de descida. Belfiore e Yoshizaki (2006) trazem uma aplicação da Busca Dispersa em um problema real de roteamento de veículos com frota heterogênea, janelas de tempo e entregas fracionadas.

Pantrigo, Montemayor e Cabido (2005) propuseram um algoritmo chamado *Scatter Search Particle Filter* para concepção de rastreadores visuais de múltiplos objetos, cujos resultados foram obtidos em tempo real. Belfiore, Silva e Ronconi (2006) fazem uso da Busca Dispersa aplicada ao problema de programação de tarefas em uma única máquina com penalidades de adiantamento e atraso e data de entrega comum. Wang et al. (2012) apresentam uma abordagem para a seleção de características com base no conjunto aproximado usando a Busca Dispersa para melhorar o fluxo de caixa e as cobranças de crédito. No trabalho, faz-se uso de entropia condicional para buscar as soluções, o que apresentou resultados com maior precisão e menor custo computacional.

Outras correlações podem ser feitas com técnicas como a Otimização por de Enxames de Partículas apresentadas por Kennedy e Eberhart (1995). El-Sayed, EL-Wahed e Ismail (2008) apresentam uma hibridização da Busca Dispersa, que substituiu dois passos do algoritmo canônico (combinação e melhoria) com duas etapas do Algoritmo Genético (cruzamento e mutação). Esse algoritmo leva ao aumento da eficiência e da exploração do processo de busca. Sari et al. (2011) efetuam uma avaliação da Busca Dispersa e do Algoritmo Genético aplicados ao problema de programação de recursos restritos. Sagheer, Sadiq e Ibrahim (2012) apresentam um aprimoramento para a Busca Dispersa utilizando o Algoritmo das Abelhas, para intensificar regiões promissoras. Os experimentos são aplicados ao Problema do Caixeiro Viajante e fornecem melhores soluções que o algoritmo canônico.

Cabe citar que estas duas técnicas estão fortemente relacionadas com a Busca Tabu, pois, os avanços fornecidos pelos mecanismos de intensificação e diversificação se dão por meio da exploração da memória adaptativa.

Além dos problemas clássicos, pode-se encontrar aplicações em outros domínios como em Lorenzoni e Polycarpo (2007) com alocação de sondas em poços de petróleo, Liberatore e

Sechi (2009) com redes de distribuição de água, Maenhout e Vanhoucke (2010) com equipes de tripulação no setor aéreo, Duman e Ozcelik (2011) com detecção de fraude de cartão de crédito e Wang, Teng e Shi (2009) com design de layout de módulos de satélite, entre outros.

3.4 Busca em Vizinhaça Variável

A Busca em Vizinhaça Variável (*Variable Neighborhood Search - VNS*) é uma meta-heurística proposta por Hansen e Mladenovic (1999, 2002, 2005) para solução combinatorial de problemas de otimização global, cuja estratégia básica consiste em uma exploração sistemática da vizinhaça dentro da busca local, modificando-a dinamicamente (MLADENOVIC, 1995; MLADENOVIC; HANSEN, 1997).

A Figura 10 apresenta o pseudocódigo da meta-heurística VNS, onde inicialmente, define-se um conjunto de estruturas de vizinhaça e uma solução inicial que é gerada aleatoriamente. O processo iterativo é iniciado até que um determinado critério de parada seja atingido. Este ciclo consiste em três etapas: i) agitação, ii) busca local e iii) movimento. No processo de agitação, uma solução S' é selecionada aleatoriamente na n -ésima vizinhaça da solução atual S^* . Então, S' é usada como a solução inicial de um procedimento de busca local, para gerar uma nova solução S' . No final do processo de busca local, se S' for melhor que S^* , então S' substitui S^* e o ciclo recomeça. Caso contrário, o algoritmo se move para o próximo vizinho e uma nova fase de agitação se inicia. Cabe ressaltar que a eficiência da VNS se faz presente quando suas vizinhanças são complementares, ou seja, se um ótimo local para um vizinho de N_i não é um ótimo local para uma vizinhaça N_j (BOUSSAÏD; LEPAGNOT; SIARRY, 2013).

Figura 10 – Pseudocódigo da Busca em Vizinhaça Variável

Input: Vizinhaças

Output: S^*

```

 $S^* \leftarrow \text{SolucaoAleatoria}();$ 
while  $\neg \text{CondicaoDeParada}()$  do
  foreach  $Vizinho_i \in \text{Vizinhaças}$  do
     $Vizinho_{atual} \leftarrow \text{CalculaVizinhaça}(S^*, Vizinho_i);$ 
     $S' \leftarrow \text{SolucaoAleatoriaNaVizinhaça}(Vizinho_{atual});$ 
     $S' \leftarrow \text{BuscaLocal}(S');$ 
    if  $\text{Custo}(S') < \text{Custo}(S^*)$  then
       $S^* \leftarrow S';$ 
      Parada;
    end
  end
end
return  $S^*;$ 

```

Fonte: Adaptado de (BROWNLEE, 2011, p. 56)

Por ser uma heurística sem memória, a VNS torna-se uma heurística fácil de implementar e depende de poucos parâmetros: i) o critério de parada e ii) o número k_{max} de vizinhanças (MARTINS; RIBEIRO, 2006; DUARTE et al., 2018). Destaca-se que o uso de múltiplas estruturas de vizinhança não se restringe aos algoritmos rotulados como VNS (COTTA; SEVAUX; SÖRENSEN, 2008). A Busca Reativa introduz uma adaptação mais sofisticada dos vizinhos, em vez de se deslocar sobre um conjunto predefinido de vizinhanças (BATTITI; BRUNATO; MASCIA, 2009).

Brito, Expósito e Moreno (2016) apresentam o algoritmo de Busca em Vizinhança Variável para resolver o problema de roteamento de veículos com janela de tempo. Lan, Wang e Zhang (2016) apresentam um algoritmo melhorado de busca de vizinhança variável para resolver um problema robusto de *Job Shop Scheduling* com o *makespan* como critério de desempenho. Salehipour et al. (2011) se propõem resolver o problema reparador itinerante, onde apresentam uma comparação de hibridizações do GRASP com VNS, e do GRASP com *Variable Neighborhood Descent*, uma variante do VNS que seleciona a vizinhança de maneira determinística.

Atas, Tufan e Sevkli (2016) aplicam o VNS para identificação precoce de Alzheimer. Os autores introduzem um algoritmo bifásico, onde aliam a VNS para seleção de recursos mais preditivos, e *Linear Support Vector Machine* para classificar os recursos selecionados. Os resultados obtidos superam os das pesquisas anteriores em termos de precisão, sensibilidade e especificidade.

Wang, Lu e Zhao (2016) propuseram uma hibridização do Algoritmo do Morcego em conjunto com VNS. Os autores aliam a capacidade de busca global do algoritmo do morcego com a capacidade de busca local da VNS, demonstrando resultados superiores através da hibridização quando comparada ao algoritmo canônico.

Liu et al. (2016) apresentam um algoritmo híbrido entre Busca Tabu e VNS para resolver o problema de exploração de n veículos. A proposta apresenta resultados de qualidade incorporando três operadores de busca local do VNS (swap, inserção e inversão) na Busca Tabu. Wu Zixuan, Li Tieke e Wang Bailin (2016) propõem outra hibridização da VNS, neste caso aliada ao Recozimento Simulado, para programação de lotes de tubos de aço laminados a quente.

São diversas as aplicações e hibridizações propostas na literatura, nos trabalhos de Hansen, Mladenović e Pérez (2010), Mladenović et al. (2008) e Hansen et al. (2019) pode-se encontrar outras pesquisas recentes sobre a Busca em Vizinhança Variável e suas extensões.

3.5 Busca Local Iterada

A definição primária da Busca Local Iterada (*Iterated Local Search - ILS*) foi dada por Stützle (1998) em sua tese de doutoramento, entretanto, esta técnica passou por um processo de redescoberta por muitos autores, por meio de diferentes nomes dentre eles: Baxter (1981) que introduz o princípio, Baum (1986) com a descida iterativa, Johnson (1990) e Applegate, Cook

e Rohe (2003) com a técnica Iterated Lin-Kernighan; e Martin, Otto e Felten (1991), Martin e Otto (1996) com cadeias de Markov de passo largo e a otimização local encadeada².

A Busca Local Iterada é uma meta-heurística cuja simples ideia perpassa a repetida aplicação de busca local nas soluções iniciais. Seu procedimento consiste na perturbação do ótimo local da iteração atual, tornando este o mecanismo fundamental de sua operação. Por outro lado, uma perturbação demasiada tornaria o algoritmo semelhante a uma busca local *multistart*, gerando soluções iniciais aleatórias. Segundo Boussaïd, Lepagnot e Siarry (2013, p. 90), a premissa desta abordagem advém da expectativa de encontrar uma solução melhor localizada na bacia de atração de um ótimo local.

Observa-se na Figura 11 o pseudocódigo da Busca Local Iterada, onde o critério de aceitação define as condições para uma nova solução candidata otimizada substituir a solução ótima local. A combinação entre os mecanismos de perturbação e o critério de aceitação permite-nos balancear entre a intensificação e a diversificação da exploração do espaço busca. A exemplo, um critério de aceitação extrema em termos de diversificação é aceitar qualquer solução, independentemente da sua qualidade. Por outro lado, uma aceitação extrema em termos de intensificação é aceitar apenas soluções de melhora³ (STÜTZLE; RUIZ, 2018).

Figura 11 – Pseudocódigo da Busca Local Iterada

```

Output:  $S^*$ 
 $S^* \leftarrow \text{ConstrucaoDaSolucaoInicial}()$ ;
 $S^* \leftarrow \text{BuscaLocal}(S^*)$ ;
while  $\neg \text{CondicaoDeParada}()$  do
     $S' \leftarrow \text{Perturbacao}(S^*)$ ;
     $S' \leftarrow \text{BuscaLocal}(S')$ ;
    if  $\text{CriterioDeAceitacao}(S^*, S')$  then
         $S^* \leftarrow S'$ ;
    end
end
return  $S^*$ ;

```

Fonte: Adaptado de (STÜTZLE; RUIZ, 2018, p. 5)

Sabar, Turky e Song (2018) propõem um modelo híbrido de Algoritmo Genético baseado em ILS, aplicado ao problema de agendamento de projetos. Santos, Arroyo e Carvalho (2016) também apresentam um modelo híbrido entre a Busca Local Iterada e a *Random Variable Neighborhood Descent* aplicado em sequenciamento de máquinas paralelas. Já Moraes et al. (2018) apresentam uma hibridização tripla entre ILS, VNS e *Random Variable Neighborhood*

²Os leitores interessados nesses desenvolvimentos históricos podem consultar JOHNSON, D. S.; MCGEOCH, L. A. The traveling salesman problem: A case study in local optimization. **Local search in combinatorial optimization**, Chichester, UK, v. 1, p. 215–310, 1997. e LOURENÇO, H. R.; MARTIN, O. C.; STÜTZLE, T. Iterated local search: Framework and applications. In: _____. [S.l.]: Springer Nature, 2019. v. 272, p. 129–168..

³Pode-se encontrar na literatura, uma diversidade de critérios de aceitação que intentam balancear o equilíbrio entre a intensificação e a diversificação, sugere-se nesse caso a leitura da obra TALBI, E. G. **Metaheuristics: From Design to Implementation**. Hoboken, New Jersey, USA: JohnWiley & Sons, Inc., 2009.

Descent aplicada a uma variante do Problema do Caixeiro Viajante, chamada de Problemas de Carros de Aluguel. Os resultados dos estudos apresentam qualidade superior quando comparados com a literatura, demonstrando a efetividade na combinação das características de busca global e local das técnicas empregadas.

Chen et al. (2008) fazem uso conjunto do ILS e VNS para resolução do Problema de Roteamento de Veículos. Aquino e Arroyo (2014) aplicam o mesmo conjunto de técnicas para ordenação aleatória da vizinhança para melhoria do cruzamento de soluções não dominadas no Algoritmo Genéticos, para resolução do Problema de Roteamento de Veículos com janela de tempo.

Beirigo e Santos (2015) resolvem o problema de agendamento de viagens utilizando o ILS paralelo, com qualidade 30% superior em relação aos casos de teste. Gavalas et al. (2015) apresentam uma aplicação da ILS para planejamento de itinerário de planejamento de agentes móveis em redes de sensores sem fio. Bytyçi, Sylejmani e Dika (2016) propõe o uso de ILS no planejamento de sequências de partida e chegadas aeroportuárias, demonstrando além da qualidade dos resultados, a melhoria no tempo de entrega da solução. O trabalho de Li, Özcan e John (2017) aborda uma nova forma do problema de *flow-shop scheduling* com montagem distribuída, solucionada por meio da aplicação da Busca Local Iterada.

Como pode-se notar, são diversas as aplicações da Busca Local Iterada em conjunto a outras técnicas. Este amplo conjunto de aplicações e hibridizações que se fazem presentes na literatura, é fruto da facilidade de implementação da técnica base, e das possibilidades de integração a outros algoritmos.

3.6 Busca Tabu

A Busca Tabu foi proposta em 1986 por Fred Glover, que, com base em trabalhos anteriores, vistos na década de sessenta, formalizou a técnica em 1989 por meio de dois artigos intitulados *Tabu Search – Part I* e *Part II*, trazendo as principais definições da técnica conhecidas atualmente (GLOVER, 1986; GLOVER, 1989; GLOVER, 1990). São elas:

- uso de estruturas de dados em forma de lista para guardar o histórico evolutivo do processo de busca;
- mecanismo de controle para executar o balanço entre aceitação, ou não, de uma nova solução, baseado em informações registradas na lista tabu, referentes às restrições e aspirações desejadas; e
- incorporação de mecanismos de alternância entre estratégias de diversificação e intensificação.

A característica geral da Busca Tabu é ser um procedimento iterativo que aceita movimentos de piora para sair de ótimos locais (GLOVER, 1989). Entretanto, a Busca Tabu compre-

ende uma família de técnicas de otimização que faz uso de abordagens por meio de estruturas de memória para organização da busca (GLOVER, 1977).

Costa (1994, p. 99) afirma que as características da Busca Tabu a tornam um dos métodos de maior eficiência para o tratamento de problemas de otimização de grande porte. De acordo com Viana (1998, p. 111), a Busca Tabu foi criada para encontrar boas aproximações da solução ótima global em qualquer problema de otimização.

O princípio basal da Busca Tabu é prosseguir com o processo de busca sempre que encontrar um ótimo local, permitindo desta forma movimentos que não apresentem quaisquer melhoras na solução atual. Deste princípio, pelo uso da estrutura de memória, chamada de lista tabu, evita-se que o procedimento revisite soluções. Esta memória efetua o registro histórico recente da busca, uma “ideia-chave que pode ser ligada a conceitos de inteligência” (GEN-DREAU; POTVIN, 2019, p. 40).

É importante destacar que Busca Tabu não foi definida por Glover como heurística, mas sim como uma meta-heurística, levando à introdução do termo que sustenta uma das áreas deste trabalho. Para o autor, o conceito de *meta* destaca-se como uma estratégia geral para guiar e controlar heurísticas subordinadas.

O processo pelo qual a Busca Tabu procura a melhor solução ocorre por uma exploração agressiva selecionando a cada iteração o melhor movimento, independente se este movimento melhora ou não a solução atual (GLOVER, 1989; GLOVER; LAGUNA, 1997). Também pode ser definido como um método de busca em trajetória que tem por base a melhora da solução inicial, buscando novas soluções por meio da geração de vizinhanças (GLOVER; TAILLARD, 1993; GLOVER, 1990).

Na Figura 12, observa-se o pseudocódigo da Busca Tabu, onde o processo consiste na geração da solução inicial e na inicialização das funções e variáveis. O critério de parada garantirá o fim do algoritmo, dada uma condição, usualmente determinado pelo NB_{max} , que mede a quantidade de iterações que o algoritmo permanece sem melhoras. Após esses passos, realiza-se a geração de vizinhanças e se obtém a melhor solução. Na sequência, as variáveis e funções são atualizadas e, novamente, o critério de parada avalia a condição atual do algoritmo.

A Busca Tabu, por ser uma das meta-heurísticas mais consolidadas na academia, possui diversas aplicações e estudos envolvendo seus mecanismos. Amuthan e Thilak (2016) demonstram uma pesquisa sobre a variedade de aplicações da Busca Tabu.

Zhong et al. (2008) demonstram uma aplicação da Busca Tabu ao problema do Caixeiro Viajante, no trabalho os autores propõem o uso de novas estruturas de vizinhança com base nos movimentos passados. A pesquisa demonstrou que a variação destas estruturas fornece resultados melhores quando comparado as estruturas padrão. Kokubo e Fukuyama (2017) apresentam uma aplicação de Busca Tabu para agendamento de tripulações de trens. Andrade e Ferreira (2018) trazem um estudo comparativo da Busca Tabu e do Recozimento Simulado aplicado ao problema de reconfiguração de redes em sistemas de distribuição elétrica. Neste trabalho é corroborada a qualidade das duas técnicas, entretanto os autores ressaltam que a Busca Tabu obtém

Figura 12 – Pseudocódigo da Busca Tabu

```

Input:  $S_0, BTmax$ 
Output:  $S^*$ 
 $S^* \leftarrow S_0;$ 
 $Iter \leftarrow 0;$ 
 $MelhorIter \leftarrow 0;$ 
 $ListaTabu \leftarrow \emptyset;$ 
InicializeFunçãoDeAspiração( $A$ );
while ( $Iter - MelhorIter \leq BTmax$ ) do
     $Iter \leftarrow Iter + 1;$ 
     $ListaDeCandidatos \leftarrow \emptyset;$ 
    foreach  $S' \in S_{vizinhanca}^*$  do
        if MovimentoNãoÉTabu( $S', ListaTabu$ ) then
             $ListaDeCandidatos \leftarrow S';$ 
        end
    end
     $S' \leftarrow LocalizaMelhorCandidato(ListaDeCandidatos);$ 
    if Custo( $S'$ )  $\leq$  Custo( $S^*$ ) then
         $S^* \leftarrow S';$ 
         $MelhorIter \leftarrow Iter;$ 
    end
    AtualizeFunçãoDeAspiração( $A$ );
end
 $S \leftarrow S^*;$ 
return  $S^*;$ 

```

Fonte: Adaptado de (MARINHO, 2005, p. 25) e (BROWNLEE, 2011, p. 74)

soluções de qualidade em menor tempo, comparado ao Arrefecimento Simulado.

Xu, Yanpeng e Xuan (2013) apresentam uma hibridização da Busca Tabu em conjunto com Otimização por Enxame de Partículas, aplicado ao problema de *job shop scheduling*. Xue, Ren e Chen (2018) introduzem uma hibridização evolucionária da Busca Tabu, para obtenção de ontologias médicas. No algoritmo apresentado, os resultados apresentam qualidade temporal superior quando comparados à estratégia evolucionária sem hibridização. Chen et al. (2018) propõem o uso da Busca Tabu em conjunto com ACO para resolver o problema do Caixeiro Viajante. Os autores apresentam um novo algoritmo que reduz os tempos de operação da ACO quando aliado às estruturas de memória da Busca Tabu. Kefalas et al. (2019) apresentam um algoritmo memético combinando Busca Tabu e Algoritmos Genéticos aplicado ao problema de *job shop scheduling*.

3.7 GRASP - Greedy Randomized Adaptive Search Procedure

Proposto por Feo e Resende em 1989, o GRASP (*Greedy Randomized Adaptive Search Procedure*) é uma meta-heurística iterativa, gulosa, *multi-start* e sem memória (FEO; RE-

SENDE, 1989). O procedimento iterativo do GRASP consiste em duas etapas: construção e busca local (FEO; RESENDE, 1995). Na etapa de construção, cria-se uma solução viável por meio de uma heurística gulosa aleatória, ou seja, em cada iteração, um elemento é incorporado em uma solução parcial, até que uma solução completa seja construída. Uma abordagem similar foi proposta por Hart e Shogan em 1987, por meio de uma heurística semigulosa *multi-start*, mas sem o uso da busca local (HART; SHOGAN, 1987).

Segundo Talbi (2009, p. 164, tradução nossa)⁴:

O algoritmo guloso deve ser randomizado para poder gerar várias soluções. Caso contrário, o procedimento de busca local pode ser aplicado apenas uma vez.

Conclui-se assim que as iterações são completamente independentes, portanto, não há presença de memória na busca. O uso da aleatoriedade, assim como em outras heurísticas, é um mecanismo eficiente para se acessar diferentes regiões do espaço de soluções.

Na etapa seguinte, a solução criada é utilizada como ponto de partida para a busca local. Desta maneira, o algoritmo evolui até um determinado número de iterações, onde a melhor solução então é retornada, sendo mantida como resultado final (BOUSSAÏD; LEPAGNOT; SI-ARRAY, 2013).

Observa-se, na Figura 13, o pseudocódigo do GRASP, onde notam-se as etapas de construção e de busca local mencionadas anteriormente. Na etapa de construção, faz-se presente o parâmetro $\alpha \in [0, 1]$, responsável por definir a estratégia entre as características de exploração da busca, ou seja, os limites entre a intensificação, usando mais gula, ou a diversificação, usando mais aleatoriedade. Em resumo, quando α tende a 0 o algoritmo torna-se aleatório, aumentando a variância entre as soluções na fase de construção, em contrapartida, quando α tende a 1, passa a ser um algoritmo guloso e determinista.

Dado o comportamento do parâmetro α no desempenho da meta-heurística, pode-se afirmar que sua sensibilidade é determinante no processo de busca. Devido à criticidade na calibração deste parâmetro, diversas estratégias podem ser aplicadas para sua inicialização (GLOVER; LAGUNA; MARTI, 2003; TALBI, 2009):

- Estática: onde o valor de α é definido no início do procedimento e se mantém constante durante toda a busca;
- Dinâmico: a estratégia dinâmica inicializa aleatoriamente o valor de α a cada iteração; e
- Adaptável: o valor de α é auto ajustado no GRASP reativo, onde o seu valor é periodicamente atualizado de acordo com a qualidade das soluções obtidas (PARIS; RIBEIRO, 2000).

⁴“The greedy algorithm must be randomized to be able to generate various solutions. Otherwise, the local search procedure can be applied only once.”

Figura 13 – Pseudocódigo da GRASP

```

Input:  $\alpha$ 
Output:  $S^*$ 
 $S^* \leftarrow \text{ConstrucaoAleatoriaDaSolucao}()$ ;
while  $\neg \text{CondicaoDeParada}()$  do
     $S' \leftarrow \text{ConstrucaoAleatoriaGulosa}(\alpha)$ ;
     $S' \leftarrow \text{BuscaLocal}(S')$ ;
    if  $\text{Custo}(S') < \text{Custo}(S^*)$  then
         $S^* \leftarrow S'$ ;
    end
end
return  $S^*$ ;

```

Fonte: Adaptado de (BROWNLEE, 2011, p. 60)

O GRASP devido a sua qualidade e simplicidade de implementação apresenta diversas aplicações e hibridizações, sendo hibridizado de diferentes maneiras, por exemplo, substituindo a busca local por outra meta-heurística como Arrefecimento Simulado, Busca em Vizinhança Variável, Busca Tabu, Busca Local Iterada, entre outras (RESENDE, 2008; VILLEGAS et al., 2010; SALEHIPOUR et al., 2011).

Sohrabi, Ziarati e Keshtkaran (2017) aplicam o GRASP ao Problema de Orientação com Seleção de Hotéis, com resultados melhores em 174 de 205 instâncias com ótimo global conhecido. Woo, Kim e Kang (2017) usam o GRASP para classificação de padrões de layout. O trabalho apresenta resultados mais significativo em termos quantitativos e temporais, quando comparado as instâncias de *benchmark* da literatura.

Rezki e Aghezzaf (2017) propõem uma hibridização entre GRASP e ILS aplicada ao problema biobjetivo de orientação com restrição orçamentária. Festa et al. (2018) apresentam uma hibridização do GRASP com simulação de Monte Carlo para resolução do Problema de Roteamento de Veículos, com resultados de maior qualidade em comparação ao algoritmo canônico. Semlali, Riffi e Chebihi (2018) obtêm resultados de qualidade na resolução do Problema de Atribuição Quadrática por meio de um algoritmo de otimização por enxame de partículas baseado no Frango Híbrido integrando o GRASP como procedimento construtivo.

Ferone et al. (2016) trazem uma heurística de simulação que integra a simulação de Monte Carlo à estrutura do GRASP para resolver o Problema de Fluxo de Permutação com tempos de processamento aleatórios, onde são obtidos resultados de qualidade em baixo tempo computacional. Saad et al. (2018) apresentam um algoritmo híbrido de GRASP para geração da população inicial do Algoritmo Genético, aplicado no agendamento de tarefas de multiprocessadores. Os resultados obtidos foram superiores nas instâncias de teste, demonstrando-se uma alternativa viável de aplicação

Rodríguez-García, Duarte e Sánchez-Oro (2018) apresentam o GRASP aplicado na minimização de largura de banda 2D. Ban e Nguyen (2016) propõem uma hibridização do GRASP

com VND para resolver o Problema da k-Mínima Latência. Como resultados, os autores argumentam que o algoritmo apresenta qualidade e agilidade no fornecimento de soluções em testes de *benchmark*.

Ramos e Bazzan (2016) propõem o uso do GRASP para resolução do Problema de Atribuição de Tráfego, com resultados superiores aos obtidos por meio de algoritmos clássicos. Arias et al. (2016) introduzem o GRASP como uma forma de solução do problema de coordenação de carga de veículos elétricos. Santoso et al. (2017) apresentam um modelo de designação e agendamento de equipes de resgate usando Lógica *Fuzzy* e GRASP

3.8 Otimização por Colônia de Formigas

O entomologista francês Pierre-Paul Grassé observou que algumas espécies de insetos reagem ao que chamou de estímulos significativos (GRASSÉ, 1959; GRASSÉ, 1984). Destes estímulos de Grassé, parte-se do termo estigmergia como uma descrição particular de comunicação em que os indivíduos são estimulados pelo desempenho que alcançaram. Segundo Mohan e Baskaran (2012, p. 4618), o termo estigmergia atualmente é usado para descrever uma forma de comunicação indireta e não simbólica mediada por meio do ambiente.

Da estigmergia presente no comportamento de colônias de formigas forrageiras, Dorigo et al., em 1991, introduziram a meta-heurística Otimização por Colônia de Formigas (*Ant Colony Optimization - ACO*) (DORIGO et al., 1991; DORIGO, 1992; DORIGO; MANIEZZO; COLORNI, 1996). O comportamento observado por Dorigo et al. (1991) descreve o processo pelo qual as formigas buscam alimentos, onde elas caminham ao acaso explorando primeiramente a área que circunda seu ninho (DORIGO, 1992). Ao longo do tempo, o percurso entre o ninho e a fonte de alimentos é marcado pelas formigas com uma trilha química de feromônio. Este processo de demarcação do caminho mais favorável acaba por orientar outras formigas para a fonte de alimento (DORIGO; BLUM, 2005). Deste comportamento emerge o caminho mais curto entre o ninho e a fonte de alimento apresentando maior concentração de feromônio, portanto, acaba atraindo mais formigas. Em suma, outras formigas percebem a presença deste feromônio e tendem a seguir as rotas onde a concentração de feromônio é maior. Por meio deste mecanismo, as formigas são capazes de identificar e transportar alimentos para o seu ninho de forma eficaz.

As colônias de formigas artificiais exploraram esta característica das colônias de formigas reais de modo a construir soluções para um problema de otimização e trocar informações sobre a sua qualidade por meio de um esquema comunicativo, que é uma reminiscência da adotada pelas formigas reais (DORIGO; BIRATTARI; STÜTZLE, 2006).

Entretanto, por ser uma substância volátil, o feromônio diminui ao longo do tempo. No início do processo, as probabilidades de as formigas selecionarem as rotas são iguais, mas como os caminhos mais curtos precisam de menos tempo para viajar, eles emergirão com maiores taxas de seleção devido a maiores quantidades de feromônio. Este processo, suportado pelo

mecanismo de evaporação, acabará por encontrar o caminho mais curto. Em essência, as trilhas de feromônios representam a memória de longo prazo de todo o sistema com informações relacionadas ao processo (DORIGO; STÜTZLE, 2004).

Na Figura 14, vê-se o pseudocódigo da Otimização por Colônia de Formigas que é composto de duas fases: construção das soluções e atualização do feromônio. Na primeira fase, as soluções são construídas pela adição de componentes a soluções parciais, de acordo com a regra de transição probabilística. Ao usar essa transição probabilística, o algoritmo de construção passa a considerar a quantidade de feromônio e a informação heurística dependente do problema.

Figura 14 – Otimização por Colônia de Formigas

Input: $Problema_{tamanho}$, $Populacao_{tamanho}$, m , ρ , β , σ , q_0
Output: S^*
 $S^* \leftarrow CriarSolucaoHeuristica(Problema_{tamanho})$;
 $Feromonio_{inicializacao} \leftarrow \frac{1.0}{Problema_{tamanho} \times Custo(S^*)}$;
Feromônio $\leftarrow InicializacaoDoFeromônio(Feromonio_{inicializacao})$;
while \neg CondiçãoDeParada() **do**
 for $i = 1$ **to** $Populacao_{tamanho}$ **do**
 $S_i \leftarrow ConstruçãoSolução(Feromônio, Problema_{tamanho}, \beta, q_0)$;
 if $Custo(S_i) \leq Custo(S^*)$ **then**
 $S^* \leftarrow S_i$;
 end
 AtualizaçãoLocalDecaimentoFeromônio(**Feromônio**, S_i , σ);
 end
 AtualizaçãoGlobalDecaimentoFeromônio(**Feromônio**, S^* , ρ);
end
return S^* ;

Fonte: Adaptado de (BROWNLEE, 2011, p. 240)

A segunda fase consiste na atualização das quantidades de feromônio. Esta fase divide-se em duas etapas: i) etapa de evaporação e ii) de reforço. Na etapa de evaporação, a trilha de feromônios é reduzida por uma relação fixa para todos os componentes no espaço de decisão. Este processo de evaporação protege as formigas de uma convergência prematura para boas soluções e incentiva a diversificação da busca no espaço de soluções.

A parametrização da Otimização por Colônia de Formigas desenvolve papel crítico no processo de busca, pois, é necessário equilibrar o feromônio e a informação heurística, este balanceamento das características visa a equilibrar a intensificação e a diversificação da busca. Cabe notar que em uma situação de desequilíbrio onde a quantidade de feromônio tende a zero, o algoritmo atuará de maneira gulosa, caso contrário, somente as trilhas de feromônio guiarão a busca. Segundo Talbi (2009, p. 247), o número de formigas não é um parâmetro crítico, apenas dependente da capacidade computacional.

São diversas as aplicações onde se verifica o uso de ACO, em um domínio plural de

aplicações e hibridizações. Deng et al. (2017) apresentaram uma hibridização entre algoritmo genético e ACO, numa estratégia de multipopulação para melhorar a busca local. O processo se baseia em um limiar entre intensificação e diversificação adaptativo para distribuição uniforme do feromônio.

Chen e Chien (2011) apresentaram um método híbrido com uso de arrefecimento simulado genético, ACO e otimização de enxame de partículas. Cada algoritmo é responsável por uma tarefa específica, onde a ACO efetua a geração de soluções iniciais para o arrefecimento simulado genético buscar melhores soluções baseado nas soluções iniciais, que servem para atualizar as trilhas de feromônio. Ao final do processo, o algoritmo de otimização de enxame de partículas troca as informações de feromônio após um número predefinido de ciclos.

Eskandari et al. (2019) propuseram um aprimoramento da solução local para ACO baseado em operadores de mutação por troca, inserção e reversão. Hasan (2018) efetuam uma comparação entre os algoritmos ACO e cuckoo search para resolução do TSP.

Mavrovouniotis, Müller e Yang (2016) usaram operadores de busca local aliado a ACO, para resolução do TSP dinâmico. No trabalho, a melhor solução da ACO é passada aos operadores de busca locais para remover e inserir cidades gerando novas soluções.

Han et al. (2018) introduziram um aprimoramento para o ACO baseado em nichos, que aprimora o algoritmo base de duas maneiras: aplica uma estratégia de nicho e múltiplos depósitos de feromônio. Alves et al. (2018) propuseram uma adaptação da ACO baseada em interação social, chamada interação social e otimização de colônia de formigas. A interação social foi introduzida visando aumentar a deposição de feromônios. Pinteá, Pop e Chira (2017) propuseram um ACO baseado em agrupamento, onde as cidades são divididas em grupos e o ACO é usado para encontrar o custo mínimo para os agrupamentos. Xiao et al. (2018) desenvolveram um ACO multiestágio que reduz a concentração inicial de feromônio tendo por base o método do vizinho mais próximo.

Zhou, He e Qiu (2017) propuseram um algoritmo ACO para problemas de larga escala que utiliza placas de vídeo. Na mesma direção, Gülcü et al. (2018) apresentam um método cooperativo paralelo, que é uma hibridização do ACO e do algoritmo 3-Opt.

Grande parte dos trabalhos presentes na literatura apresentam melhorias em algoritmos existentes, sugerindo métodos para controlar os limiares entre a intensificação e a diversificação. Nesta direção, as pesquisas geralmente são realizadas através de hibridizações que mesclam as diferentes capacidades dos algoritmos. Em contraste, nosso objetivo neste trabalho foi melhorar o balanço entre a intensificação e a diversificação dos mecanismos heurísticos presentes na ACO por meio de uma hiper-heurística.

3.9 Hiper-Heurísticas

As meta-heurísticas vêm sendo utilizadas cada vez mais ao longo dos anos para a resolução de problemas de Otimização Combinatória em diferentes áreas, prova disso dá-se pela

profusão de trabalhos relacionados apresentados ao longo da tese. Esta abrangência acabou permitindo uma compreensão científica aprofundada sobre as características modelares das técnicas. Por outro lado, outros domínios afloraram, trazendo necessidades e, com isso, o desenvolvimento de novas meta-heurísticas. Estas criações trazem consigo implicadores de custo e manutenção, para que se compreenda as novas técnicas e as adapte às necessidades surgentes, levando a considerações apenas sobre heurísticas simples e que conduzem a soluções de baixa qualidade (SOGHIER, 2012).

A partir desta problemática, partiram pesquisas em prol da construção de sistemas que operassem em uma variedade de problemas visando assim aumentar o nível de generalidade em que podem atuar (BURKE; SOUBEIGA, 2003). Na prática, isso motivou a criação de sistemas capazes de operar em uma gama de problemas, sem a necessidade de sintonização de parâmetros e garantindo um nível mínimo de qualidade das soluções.

Do outro lado, fazendo um contraponto, tem-se o teorema *No Free Lunch*⁵ ⁶, como mencionado no Capítulo 1, afirmando que todos os algoritmos de busca apresentam o mesmo desempenho médio para todos os problemas em domínios finitos. Isso sugere que não é possível desenvolver uma metodologia de busca geral para todos os problemas de otimização. Em contrapartida, não indica que não seja possível o desenvolvimento de uma generalização para uma gama de problemas.

A ideia básica das hiper-heurísticas existe desde a década de 1960, quando Fisher (1963) e Crowston et al. (1963) hipotetizaram sobre a combinação de regras de *job-shop scheduling* e uma técnica de aprendizagem probabilística. Esta técnica de aprendizagem consiste em escolher entre duas heurísticas, portanto, essa abordagem pode ser classificada como uma das primeiras hiper-heurísticas. Destes estudos, chegaram-se às seguintes conclusões: i) uma combinação aleatória é melhor que qualquer regra de agendamento separada, e ii) aprendido é possível.

Ante as diversas apresentações que se fazem na literatura, foi somente em 2001 que Cowling, Kendall e Soubeiga cunharam propriamente o termo hiper-heurística como “heurísticas que selecionam heurísticas”. Para Ochoa, Qu e Burke (2009), o foco das hiper-heurísticas é encontrar de forma adaptativa um método de solução ao invés de produzir uma solução individual para um problema específico. Já para Ross (2005a), a diferença entre hiper-heurísticas e meta-heurísticas é que o espaço de busca das hiper-heurísticas não é o espaço de busca do problema. Dos conceitos de Cowling, Kendall e Soubeiga (2001), Ochoa, Qu e Burke (2009) e Ross (2005a), parte-se de um preceito próprio, de que as *hiper-heurísticas operam não sobre o espaço de busca, mas sim, no que se nomeou como metaespaço*⁷.

Para prosseguir o discurso sobre uma conceituação das hiper-heurísticas, cabe delinear como tal termo se fará presente no trabalho. Desta forma, apoia-se em Marshall (2015) que

⁵Cf. WOLPERT; MACREADY, 1997

⁶Cf. WHITLEY; WATSON, 2006

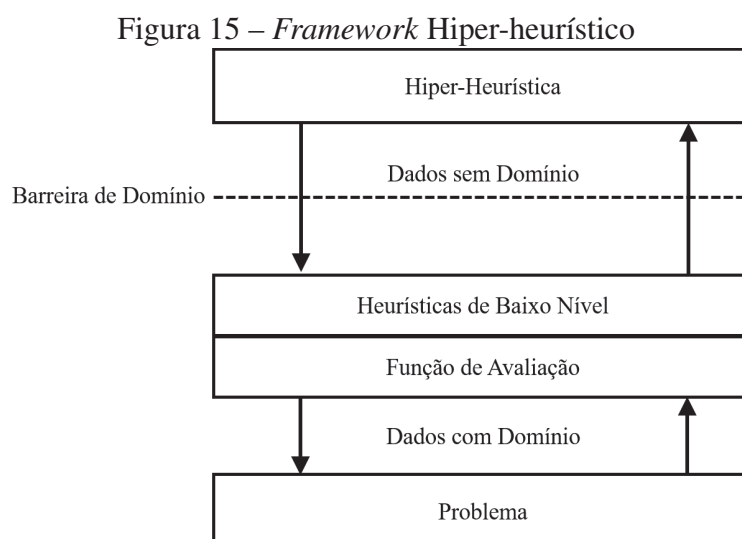
⁷Esta definição parte como um movimento de distanciamento do conceito de espaço de busca, a fim de esclarecermos nosso *modus operandi*, que é apresentado no Capítulo 4, junto de uma definição em maiores detalhes sobre o metaespaço.

efetua uma separação entre as heurísticas que se originam na *práxis* e as meta-heurísticas que empregam estratégias de exploração do espaço de busca. Esta delimitação se faz necessária uma vez que se confunde meta-heurísticas e hiper-heurísticas, pois, ambas são fundamentalmente heurísticas. Neste trabalho, tratam-se as heurísticas e as meta-heurísticas sob uma mesma perspectiva, e reserva-se o termo hiper-heurística para significar *heurísticas que operam no metaespaço selecionando heurísticas*.

3.9.1 Barreira de Domínio

Como já mencionado anteriormente, as meta-heurísticas normalmente operam diretamente sobre o espaço de busca, acessando diretamente o conhecimento do domínio. Em Burke et al. (2003, p. 467), é apresentada uma estrutura hiper-heurística que opera em um nível mais elevado de abstração, sem acesso às informações de domínio, apenas às heurísticas de baixo nível. Esta abordagem sugere que uma hiper-heurística pode ser aplicada a diferentes domínios de problemas, substituindo apenas a função objetivo e as heurísticas de baixo nível.

Na Figura 15, é apresentado um diagrama geral de um *framework* hiper-heurístico, onde a hiper-heurística têm acesso somente aos dados sem domínios devido à presença de uma barreira. Desta maneira, torna-se inespecífica, pois, não tem ciência do problema em questão. Sua forma de atuação condiz ao controle das heurísticas de baixo nível que possuem acesso aos dados do domínio, agindo diretamente no problema, conjuntamente com a função de avaliação.

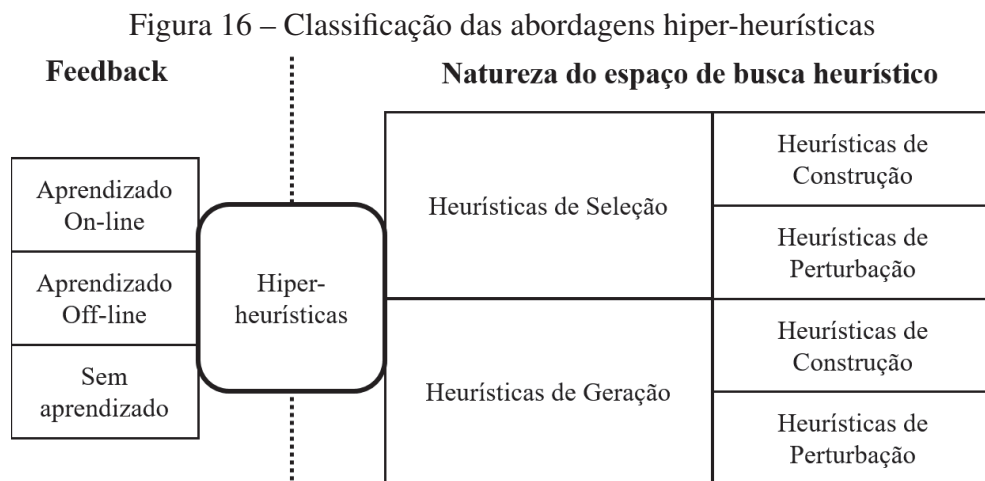


Fonte: Adaptado de (BURKE et al., 2003, p.467)

Burke et al. (2003, p.467) defendem que este interfaceamento padrão entre a hiper-heurística e as heurísticas, simplifica a implementação das heurísticas de baixo nível para outros domínios, pois favorece a comunicação através de uma interface única ao invés de utilizar interfaces individualizadas.

3.9.2 Classificação das hiper-heurísticas

As hiper-heurísticas podem, segundo Burke et al. (2013, p. 3), ser classificadas em duas grandes dimensões: i) sobre a natureza do espaço de busca heurístico e ii) sobre a fonte de *feedback* durante a aprendizagem (Figura 16). Estas dimensões oferecem um meio de automação do desenvolvimento de heurísticas para variações de domínios de problemas existentes ou até mesmo problemas em novos domínios.



Fonte: Tradução nossa de (BURKE et al., 2013, p. 3)

A partir desta definição, têm-se duas categorias i) as hiper-heurísticas de seleção onde heurísticas são escolhidas de acordo com algum critério; e ii) as hiper-heurísticas de geração que constroem novas heurísticas por meio da recombinação de operadores e componentes de uma ou mais heurísticas, compondo assim o primeiro nível da dimensão sobre a natureza do espaço de busca.

O segundo nível corresponde à distinção entre as i) heurísticas de construção que de maneira incremental, a partir de uma solução vazia ou parcial, criam uma heurística a partir de um conjunto de componentes potenciais; e as ii) heurísticas de perturbação, que operam em uma solução candidata completa gerada aleatoriamente ou utilizando heurísticas de construção simples e, em seguida, tentam melhorar a solução modificando alguns de seus elementos. Cabe notar que essa categorização diz respeito à natureza das heurísticas de baixo nível, usadas na estrutura das hiper-heurísticas (BURKE et al., 2019).

Outro viés analítico sobre as hiper-heurísticas relaciona-se à capacidade de aprendizado, dividindo-se em i) aprendizado *on-line*, onde durante a resolução do problema o algoritmo coleta dados e os incorpora a sua base de conhecimento, melhorando o processo durante sua execução; ii) o aprendizado *off-line* diz respeito aos algoritmos que primeiramente resolvem a instância do problema e somente após a execução, efetuam análise do processo para consolidar o aprendizado. O resultado desta forma de aprendizado é um algoritmo potencialmente mais apto do que qualquer instância para o qual já havia sido treinado. Esta categoria para iniciar

sua operação requer um conjunto prévio de treinamento, caso contrário a busca será iniciada de maneira aleatória, característica da última categoria as, iii) hiper-heurísticas sem aprendizado.

A seguir, são listadas algumas contribuições sobre hiper-heurísticas que se relacionam a esta pesquisa, ou demonstram avanços significativos na área. Cabe notar que os trabalhos encontrados abordam majoritariamente as hiper-heurísticas de seleção construtiva e perturbativa, no entanto, não as abordamos de maneira distinta.

Burke, Silva e Soubeiga (2005) usam uma hiper-heurística multiobjetivo para resolver o problema de agendamento de cursos. O algoritmo proposto faz uso da Busca Tabu para explorar o espaço de heurísticas de perturbação de baixo nível. Kendall e Hussin (2004, 2005) apresentam uma hiper-heurística também baseada em Busca Tabu. A hiper-heurística é usada para melhorar uma solução inicial criada usando uma heurística construtiva de baixo nível. No trabalho são apresentadas duas variações do algoritmo, uma aliada a *hill-climbing*, e outra com o algoritmo do grande dilúvio.

Ross e Marfn-Blazquez (2005b) avaliam o uso de uma hiper-heurística baseada em Algoritmo Genético, aplicado ao problema de horários de cursos universitários. Os resultados apresentaram desempenhos satisfatórios no conjunto de *benchmarks* aplicado. Pillay (2008, 2012) apresenta uma hiper-heurística evolucionária para buscar combinações heurísticas construtivas de baixo nível, avaliando o efeito da representação utilizada para as combinações heurísticas sobre o desempenho do algoritmo.

Chen, Kendall e Berghe (2007) introduzem uma hiper-heurística baseada em formigas para resolução do *Travelling Tournament Problem*. O algoritmo faz uso de cinco estruturas de vizinhança para obtenção de soluções viáveis de alta qualidade. Os resultados obtidos são ótimos para pequenas instâncias, e de boa qualidade para instâncias maiores.

Ersoy, Özcan e Uyar (2007) apresentam uma hiper-heurística aliada a um algoritmo memético para resolver o problema de agendamento de exames. A hiper-heurística é usada para selecionar um dos algoritmos de *hill-climbing* usado pelo algoritmo memético.

Bai et al. (2007) propõem uma hiper-heurística perturbativa de seleção, implementada com base em Arrefecimento Simulado para aceitação do movimento. Inicialmente, a seleção das três heurísticas de baixo nível é aleatória até que se obtenha um histórico de desempenho. A hiper-heurística é usada para melhorar uma solução inicial viável. Os resultados apresentaram melhor desempenho quando comparados a outras aplicações no mesmo conjunto de testes.

Qu e Burke (2009) apresentam uma hiper-heurística que faz uso de diferentes técnicas de busca, empregando VNS para explorar o espaço de combinações heurísticas. Os autores também apresentam uma variação da hiper-heurística empregando ILS para exploração do espaço de soluções em vários estágios do processo de construção. A hiper-heurística avalia o grau de saturação das heurísticas de baixo nível, cujas combinações apresentam bons desempenhos. Com base nisso, foi desenvolvido um mecanismo adaptativo na hiper-heurística para hibridizar o grau de saturação ponderado sobre as novas combinações heurísticas. Ambas as

hiper-heurística foram usadas para resolver problemas de *timetabling* de um conjunto de benchmarks, onde obtiveram resultados de qualidade.

Ozcan et al. (2009) apresentam uma hiper-heurística de perturbação cujo componente de aceitação de movimento emprega uma estratégia de aceitação tardia, ou seja, em vez de comparar a solução candidata atual com a obtida na iteração anterior, o componente de aceitação de movimentação compara essa solução a uma solução de n iterações anteriores. Outros métodos de seleção heurística foram testados pelos autores, eles incluem aprendizado por reforço aleatório, guloso, simples, aprendizado por reforço com Busca Tabu, sendo os melhores resultados obtidos pela estratégia de aceitação tardia.

Sabar e Ayob (2009) propõem uma hiper-heurística baseada em Busca Dispersa aplicada a problemas de *timetabling*. O algoritmo opera em alto nível com a busca dispersa, e no baixo nível seleciona entre dez estruturas de vizinhança. Já Pillay (2010) introduziu um algoritmo evolucionário hiper-heurístico para resolução do problema de horários escolares. Os autores incorporaram *hill-climbing* aos operadores genéticos e conseguiram aumentar o desempenho apresentado inicialmente, fornecendo resultados competitivos.

McClymont e Keedwell (2011a), McClymont e Keedwell (2011b) apresentam uma hiper-heurística de seleção *on-line* baseada em cadeias de Markov aplicada a problemas mono e multiobjetivo contínuos. A hiper-heurística baseia seu funcionamento em Estratégia Evolutiva e aplica aprendizado por reforço para atualizar os pesos das transições da cadeia de Markov. Os autores demonstram em instâncias de benchmark que o método foi capaz de aprender as boas combinações heurísticas empregadas, superando apenas abordagens baseadas em sintonia de parâmetros.

Raghavjee e Pillay (2012) apresentam a comparação de desempenho de um algoritmo genético e um algoritmo evolucionário hiper-heurístico aplicados ao problema de horários escolares. O algoritmo evolucionário explora o espaço de combinações através de quatro operadores de mutação e um operador de alocação e desalocação. A hiper-heurística fornece resultados que superam o Algoritmo Genético e outros métodos comparativos.

Özcan et al. (2012) e Sin e Kham (2012) implementaram uma hiper-heurística empregando aprendizado por reforço para seleção heurística e o algoritmo do grande dilúvio para aceitação de movimento. Soghier e Qu (2013) implementam uma hiper-heurística para seleção de salas e períodos, cuja abordagem emprega três fases. Durante a primeira fase, o conjunto de heurísticas de baixo nível é identificado e aplicado ao problema. Na segunda fase, uma sequência hibridizada de heurísticas é criada com base no desempenho apresentado pelas heurísticas na fase anterior. A última fase efetua um ajuste iterativo da hibridização das heurísticas de baixo nível empregadas na segunda fase. A hiper-heurística produziu bons resultados quando aplicada aos conjuntos de dados para agendamento de exames.

Anwar et al. (2013) empregam uma hiper-heurística perturbativa de seleção para resolver o problema de horários de aulas que faz uso de Busca Harmônica para otimizar o espaço de sequências heurísticas. Kalender et al. (2012), Kalender et al. (2013) apresentam uma hiper-

heurística que aplica o método do gradiente guloso combinado com Arrefecimento Simulado para aceitação do movimento. Os autores argumentam que o uso de uma heurística gulosa apresenta melhores resultados do que uma função aleatória simples.

Soria-Alcaraz et al. (2014) propõem uma hiper-heurística baseada em ILS para resolver o problema de agendamento de cursos universitários. A abordagem aplica um operador simples de perturbação aleatória que substitui uma variável selecionada aleatoriamente, após a fase de melhoria, o algoritmo faz uso de aprendizado *on-line*, na forma de probabilidades com base no desempenho anterior das heurísticas de baixo nível, para selecionar uma nova heurística para o próximo ciclo.

Sim e Hart (2016) apresentam uma hiper-heurística de geração e seleção aplicada ao problema de Roteamento de Veículos, já Espinoza-Nevárez et al. (2016) apresentam a mesma aplicação, mas com janelas de tempo. Os algoritmos propostos operam em duas etapas, inicialmente através de programação genética são construídas novas heurísticas de qualidade. Na segunda etapa, uma hiper-heurística perturbativa é aplicada às soluções candidatas criadas. Os autores demonstram que a combinação de geração e seleção fornecem resultados competitivos, com valores ótimos, superando aplicações recentes.

Li, Özcan e John (2017) trazem duas abordagens hiper-heurísticas baseadas em autômatos de aprendizagem para otimização multiobjetivo, aplicado ao problema de proteção à colisão veicular. São elas a hiper-heurística baseada em autômatos de aprendizagem e a hiper-heurística baseada em autômatos de aprendizagem com esquema de inicialização da classificação. Ambas as hiper-heurísticas combinam e controlam um conjunto de três algoritmos evolutivos multi-objetivo, são eles: algoritmo genético de classificação não dominada, algoritmo evolucionário de força de Pareto 2 e o algoritmo evolutivo baseado em indicadores. Os resultados experimentais mostram que os diferentes algoritmos evolutivos apresentam resultados de qualidade, entretanto a abordagem por meio de autômatos de aprendizagem com esquema de inicialização da classificação, oferece os melhores resultados.

Falcón-Cardona e Coello (2018) propõem uma hiper-heurística evolucionária multiobjetivo baseada em múltiplos indicadores baseado em estimadores de densidade. Neste modelo, os autores fazem uso de uma cadeia de Markov para armazenamento e atualização das probabilidades de transição entre indicadores. Os resultados demonstram significativa qualidade nas instâncias de teste adotadas.

Li (2018) apresenta um hiper-heurística de seleção multiobjetivo utilizando aprendizado por reforço para otimização de layout de parques eólicos. A hiper-heurística proposta apresenta memória *on-line*, adaptando as heurísticas de baixo nível baseadas em Algoritmos Evolutivos.

Doerr et al. (2018) apresentam uma hiper-heurística baseada no gradiente aleatório que ajusta os períodos de aprendizagem durante a execução. Os autores demonstram que a adequação dos intervalos é capaz de obter resultados em tempo ideal, em comparação com as abordagens de aprendizado contínuo.

Como a maioria das heurísticas de seleção existentes apenas recomenda uma heurística

para uma determinada instância, Rosales-Pérez et al. (2017) propuseram uma alternativa que fornecesse múltiplas abordagens, visto que são diversas as formas de resolução de problemas de otimização. Para isso, foi elaborada uma hiper-heurística coevolutiva cooperativa para escolha das heurísticas mais relevantes para o problema da satisfação de restrições. Os resultados experimentais sugerem que o método proposto supera os resultados apresentados pelos métodos de referência.

Turky et al. (2018) propõem um método hiper-heurístico que seleciona automaticamente o melhor algoritmo de busca local e efetua o ajuste de seus parâmetros. Os resultados mostram que a abordagem desenvolvida é eficaz e pode alcançar valores ótimos para os problemas de *bin packing* de múltiplas capacidades e de redistribuição de máquinas do Google.

Dufflo et al. (2019) apresentam uma hiper-heurística de geração baseada em Programação Genética aplicada ao Problema do Caixeiro Viajante. No trabalho, a programação genética é responsável pelo aprendizado sobre as estruturas geradas. Os resultados apontados superam as abordagens heurísticas tradicionais.

3.10 Considerações sobre o capítulo

Este capítulo não intencionou uma extensiva revisão sobre as meta-heurísticas e hiper-heurísticas, pois partiríamos de um conjunto com amplitude além dos interesses da pesquisa. A escolha foi pautada em algoritmos com características estocásticas, evolutivas, físicas e enxame de partículas, entre outros.

Percebeu-se ao longo desta etapa, que grande parte dos trabalhos sobre hiper-heurísticas concentram-se em abordagens de seleção perturbativa. O desempenho desta categoria de hiper-heurísticas é dependente da habilidade de seleção heurística e dos métodos de aceitação. Desde que o método de aceitação de movimento seja adequado, o método de seleção heurística torna-se eficaz mesmo em abordagens simples, ou seja, até mesmo a seleção aleatória é capaz de fornecer bons resultados.

As hiper-heurísticas de seleção construtiva são boas na criação de soluções iniciais de alta qualidade se as heurísticas construtivas de baixo nível forem adequadas ao domínio. Já, as hiper-heurísticas de geração construtiva apresentam potencial de automatização na criação de heurísticas construtivas de baixo nível, cuja generalização pode ser melhorada aliando etapas de treinamento. É interessante notar que os trabalhos que têm por base algum mecanismo de Algoritmos Genéticos exploram melhor as capacidades construtivas das hiper-heurísticas.

Por fim, as hiper-heurísticas de geração perturbativa oferecem uma maneira de se obter novas combinações heurísticas, pois apresentam a capacidade de identificações de bons componentes. O objetivo desta categoria é aumentar o desempenho das heurísticas perturbativas de baixo nível. Ainda que outras estratégias se façam presentes, a programação genética tem se mostrado a maneira mais eficaz, tanto para construção, quanto perturbação.

Os estudos levantados neste capítulo nos mostraram que tanto as meta-heurísticas, quanto

as hiper-heurísticas apresentam bons desempenhos em seus processos de seleção ou geração, de base construtiva ou perturbativa. Assim, retomamos o conjunto das oito meta-heurísticas abordadas neste capítulo, que invocam saberes canônicos, a fim de auxiliar a compreensão sobre as bases de suas operações no espaço de soluções, elencando seus principais mecanismos e parâmetros ⁸. Quais sejam:

1 Algoritmos Genéticos (Seção 3.1)

- Tamanho da população
- Probabilidade de cruzamento
- Probabilidade de mutação
- Função de seleção
- Função de cruzamento
- Função de mutação
- Atualização da população

2 Arrefecimento Simulado (Seção 3.2)

- Temperatura inicial
- Fator de redução de temperatura
- Criação da solução vizinha

3 Busca Dispersa (Seção 3.3)

- Tamanho do conjunto de diversificação
- Tamanho do conjunto de referência
- Seleção de um subconjunto
- Recombinação do subconjunto

4 Busca em Vizinhança Variável (Seção 3.4)

- Número de Vizinhanças
- Geração da Vizinhança
- Busca Local

5 Busca Local Iterada (Seção 3.5)

- Perturbação

⁸Por estarem presentes em todas as meta-heurísticas listadas, serão omitidos a Avaliação da Solução e o Critério de Parada

- Busca Local
- Critério de Aceitação

6 Busca Tabu (Seção 3.6)

- Tamanho da lista Tabu
- NBMax
- Geração da vizinhança
- Atualização da Lista Tabu
- Critério de aspiração

7 GRASP - Greedy Randomized Adaptive Search Procedure (Seção 3.7)

- Construção Aleatória
- Busca Local

8 Otimização por Colônia de Formigas (Seção 3.8)

- Construção da solução
- Atualização do Feromônio
- Atração do Feromônio
- Evaporação do Feromônio

Por fim, o capítulo se encerra trazendo um aporte sobre o núcleo do trabalho, reunindo características e conceitos sobre meta-heurísticas e hiper-heurísticas. Estas informações serão retomadas oportunamente no capítulo seguinte, durante a apresentação do modelo H2-SLAN e suas caracterizações.

4 MODELO

“A ciência é muito mais do que um corpo de conhecimento. É uma maneira de pensar. E isso é fundamental para o nosso sucesso. A ciência nos convida a aceitar os fatos, mesmo quando eles não estão de acordo com nossos preconceitos. Ela nos aconselha a levar hipóteses alternativas em nossas cabeças e ver quais são as que melhor correspondem aos fatos. Impõe-nos um equilíbrio perfeito entre a abertura sem obstáculos a novas ideias, por mais heréticas que sejam, e o mais rigoroso escrutínio cético de tudo estabelecendo novas ideias e sabedoria. Precisamos da ampla apreciação desse tipo de pensamento. Funciona. É uma ferramenta essencial para uma democracia em uma era de mudança. Nossa tarefa não é apenas treinar mais cientistas, mas também aprofundar a compreensão pública da ciência”

(SAGAN, 1989)

Neste capítulo, o modelo computacional aglutina as noções levantadas através dos capítulos anteriores. O capítulo 2 fundamentou um olhar epistemológico, servindo de base para a construção de uma visão apurada, observando mais profundamente o *corpus* para a articulação dos eixos teóricos da pesquisa, que convergem na questão dos movimentos no metaespaço. O propósito deste entendimento sobre as interações é obter uma melhora das capacidades de exploração no espaço de busca do problema, em prol de melhores resultados (qualitativos e quantitativos).

Assim, o capítulo 3 trouxe uma visão tecnicista que auxiliou a compreensão da ordem mecânica das meta-heurísticas, expondo as estruturas de cada uma das técnicas em análise. Desta maneira de pensar, pôde-se entender como os mecanismos se configuram em diferentes facetas, movimentando a busca no espaço de soluções, e desta forma, trazem consigo o desafio da concepção de alternativas que promovam interações entre os mecanismos.

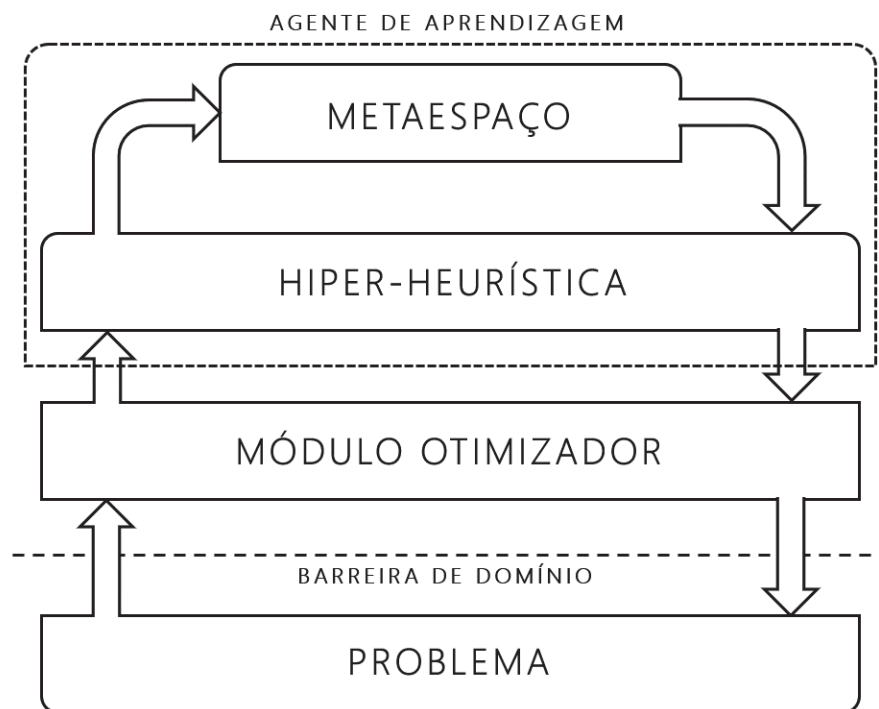
À vista disso, este capítulo apresenta o modelo desenvolvido, onde sob diferentes perspectivas são expostos seus níveis arquiteturais. Conforme ver-se-á no decorrer das seções, o modelo é conduzido a partir de conceitos presentes na literatura, onde o desenvolvimento se dá sob concepções basais de estruturação da hiper-heurística, consolidando o conceito de metaespaço sob a representação das Redes de Autômatos Estocásticos de Aprendizagem.

4.1 Visão Geral

A abordagem desenvolvida no modelo arquitetural propõe analisar o comportamento dos mecanismos heurísticos em prol da construção, recomendação e/ou parametrização das heurísticas de baixo nível. Desta forma, o capítulo culmina em um modelo Hiper-heurístico baseado em Redes de Autômatos Estocásticos de Aprendizagem, denominado no contexto deste trabalho como Modelo H2-SLAN (*Hyper-Heuristic model based on Stochastic Learning Automata Networks*). A arquitetura do H2-SLAN divide-se em três blocos principais, conforme visão geral apresentada na Figura 17.

Na figura, observa-se o bloco do metaespaço, caracterizando uma dimensão além do domínio do problema, interfaceando a hiper-heurística e os mecanismos heurísticos presentes no módulo otimizador. Este processo visa a construir uma base de conhecimento dos cenários dos problemas e dos algoritmos, reforçando as conexões da Rede de Autômatos Estocásticos, ou seja, aumentando as probabilidades de ocorrência de determinado mecanismo durante a busca. Por fim, a hiper-heurística seleciona o melhor nível, método, ou algoritmo para um novo cenário da mesma instância em outro momento, ou para uma nova instância. Após este processo, ocorrerá a incorporação do conhecimento através do aprendizado das novas instâncias resolvidas a fim de aumentar a qualidade da seleção e construção. O conhecimento armazenado é então melhorado com um novo conjunto de instâncias resolvidas e é usado novamente na fase de seleção, constituindo assim um ciclo.

Figura 17 – Visão geral do Modelo H2-SLAN

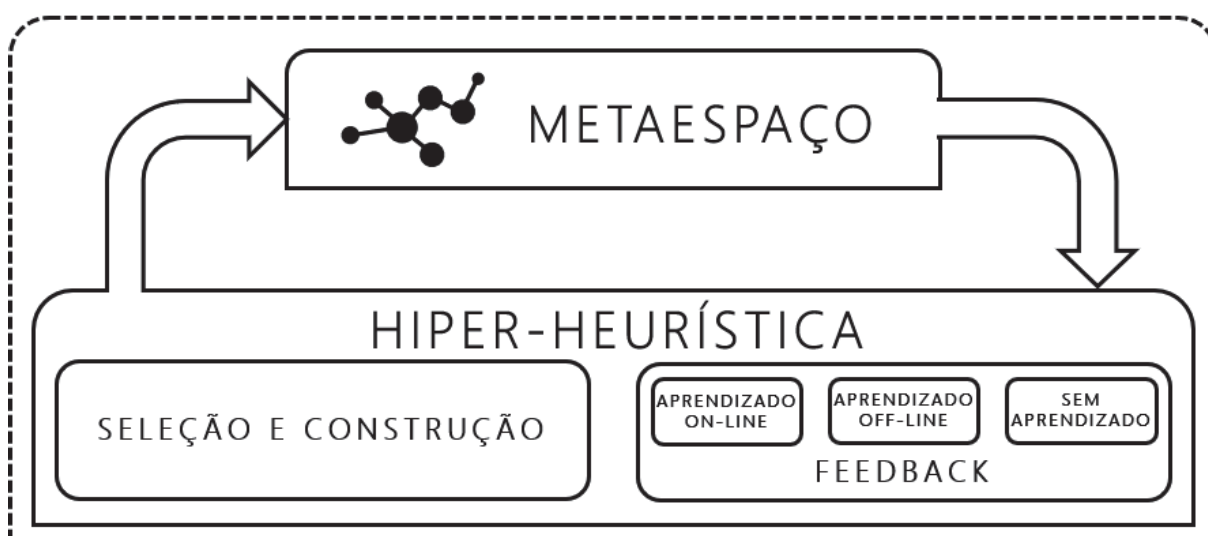


4.2 Arquitetura do Agente de Aprendizagem

O principal módulo desta arquitetura, o Agente de Aprendizagem, divide sua operação em duas etapas: i) a construção do conhecimento e ii) a seleção do algoritmo. Este processo relatado por Burke et al. (2003), descreve de maneira genérica uma estrutura hiper-heurística, sem no entanto consolidar quaisquer tecnologias de implementação. Para a construção deste módulo, foram relacionados os conceitos de Autômatos de Aprendizagem e SAN como mecanismos de construção e seleção, que consolidam estas funcionalidades através dos módulos de *Feedback* e de Seleção.

A arquitetura do módulo (Figura 18) visa, por meio da generalização inserida em sua estrutura, a desagregar qualquer dependência do domínio do problema por parte dos mecanismos heurísticos. Esta separação se dá por meio da identificação de componentes comuns presentes em um domínio abstrato, nomeado aqui de metaespaço. Esta dimensão se caracteriza como uma interface além do domínio do problema, conectando a hiper-heurística e os mecanismos heurísticos que operam ao nível da solução.

Figura 18 – Arquitetura do Agente de Aprendizagem
AGENTE DE APRENDIZAGEM



Fonte: Elaborado pelo autor

Esta interface, desenhada por meio do metaespaço na figura de uma rede de autômatos estocásticos de aprendizagem, quantificará as relações entre os mecanismos da busca na fase de aprendizado. Este processo visa a construir uma base de conhecimentos dos cenários, dados dos problemas e algoritmos, reforçando as conexões da SAN, ou seja, aumentando as probabilidades de ocorrência de determinado mecanismo durante a busca.

Com isso em mente, denomina-se o metaespaço como um modelo de nível intermediário, unindo os aspectos comportamentais do espaço de busca – advindo das heurísticas de baixo nível – apresentando assim independência do domínio. Desta maneira, propicia-se à

hiper-heurística um grau de generalidade, visando a atingir a não dependência do domínio do problema. O metaespaço não objetiva se estruturar por meio de *n-uplas*, atribuindo sentido e delimitando pares de condição-ação para o processo de construção e seleção realizado pela hiper-heurística, mas de outra forma, se conectando e se rompendo, sendo único e múltiplo ao mesmo tempo.

O processo de seleção visa a indicar os níveis, métodos, ou algoritmos mais adequados ao novo cenário da mesma instância em outro momento, ou para uma nova instância dada. Como saída do bloco de Seleção, é encaminhado ao Módulo Otimizador uma cadeia de caracteres que orienta o comportamento do bloco Executor.

Após este processo, poderá ocorrer, dada a caracterização do módulo de *Feedback*, a incorporação do aprendizado das novas instâncias/momentos resolvidos a fim de aumentar a qualidade da seleção futura. O fluxo de operação do bloco de *feedback* divide-se em três caracterizações:

- *on-line*, ao final de cada ciclo do módulo executor uma etapa de aprendizado é realizada;
- *off-line*, ao final de todo o processo do módulo executor é realizada uma única etapa de aprendizado; e
- *no line*, onde não é executado qualquer procedimento de aprendizado.

O bloco de *Feedback* atua na rede de autômatos estocásticos efetuando a aprendizagem, que pautará o desempenho das heurísticas de baixo nível e as características do problema. Desta maneira, o conhecimento armazenado é então melhorado com um novo conjunto de instâncias resolvidas e é usado novamente na fase de seleção, constituindo assim um ciclo.

Segundo Foster (2011, p. 31), um modelo de memória deve ser capaz de *codificar* (isto é, receber ou adquirir) informações, *armazenar* ou retê-las de maneira fiel e *recuperar* ou acessá-las onde estiverem armazenadas. Modelos computacionais de memória são limitados em diversos sentidos, pois, são inúmeras as variáveis e perspectivas de modelagem, sendo possível formulá-los em diversos níveis analíticos, desde o molecular ao mais abstrato. Se está longe de uma visão abrangente da base neural dos tipos de memória tipicamente estudados por psicólogos cognitivos. No entanto, alguns progressos foram feitos neste sentido em vários domínios.

Howard (2009) apresenta três vieses, a partir dos quais a modelagem de memória pode ser construída e assim o é feita observando-se i) a correspondência anatômica, no que diz respeito ao estudo do hipocampo; ii) a correspondência celular no que concerne ao sistema nervoso; e iii) a correspondência funcional, a qual é interesse deste trabalho, que considera as características da memória, como aprendizagem e organização.

É amplo o espectro de pesquisas em modelos de memória, além de abranger diversas áreas da Inteligência Computacional, fazendo com que sejam criadas alternativas, como novos algoritmos, técnicas híbridas e de redução do espaço de estados, etc. Neste sentido, a pesquisa colabora para o crescimento da área, unindo o formalismo das Redes de Autômatos Estocásticos

e os Autômatos de Aprendizagem, em um modelo de pensamento que auxiliou a construção e aplicação do metaespaço.

4.2.1 Sistema de aprendizado

Os autômatos são mecanismos de controle projetados para seguir automaticamente uma sequência predeterminada de operações ou responder a instruções codificadas. A estocasticidade do termo enaltece o caráter adaptativo do autômato por não seguir regras pré-determinadas, mas por se adaptar às mudanças em seu ambiente.

A clássica teoria de controle tem seu processo baseado na completude do conhecimento do sistema e/ou processo. Ou seja, o modelo matemático que o descreve é conhecido *a priori*. Durante a evolução desses sistemas, passou-se a considerar incertezas que se faziam presentes no sistema. A teoria do controle estocástico assume que algumas características das incertezas são conhecidas, mas podem, no entanto, não serem suficientes para o completo controle do sistema em caso de mudanças (UNSAI, 1997). Destarte, torna-se necessário observar o processo para obter-se mais conhecimento, adquirindo informações adicionais, uma vez que as características iniciais não são suficientes. Assim, uma possível abordagem para tais problemas, é vê-los como problemas de aprendizagem (NARENDRA; THATHACHAR, 1974).

Ainda que possuam bom desempenho, os sistemas baseados em regras possuem como desvantagem a necessidade de modificações no espaço de soluções, por não lidarem com o imprevisto. O conceito por trás de um sistema de aprendizagem é garantir a robustez de seu comportamento sem o total conhecimento do meio ou do sistema em questão. Assim, observa-se como vantagem do sistema de aprendizagem por reforço, em comparação a outras abordagens, a não necessidade de informações sobre o sistema e/ou meio, exceto o sinal de reforço (NARENDRA; THATHACHAR, 1989; GORDON; MARSH; WU, 1993).

Em um sistema de aprendizagem por reforço, ainda que seja mais lento do que outras abordagens, na maioria das aplicações, as ações precisam ser testadas um número significativo de vezes para que se tenha um desempenho satisfatório, em nosso caso, se dá pela reapresentação de um mesmo cenário ou de novos cenários de uso. Assim, o processo de aprendizagem deve ser claramente mais rápido do que as modificações aconteçam no ambiente, ou a aprendizagem por reforço deve ser combinada a um modelo adaptativo que antecipe as mudanças no ambiente (PENG; WILLIAMS, 1993).

Para Narendra (1974, p. 323), a aprendizagem é definida como qualquer alteração permanente no comportamento, resultante de uma experiência passada. Um sistema de aprendizagem deve possuir a capacidade de melhorar seu comportamento ao longo do tempo.

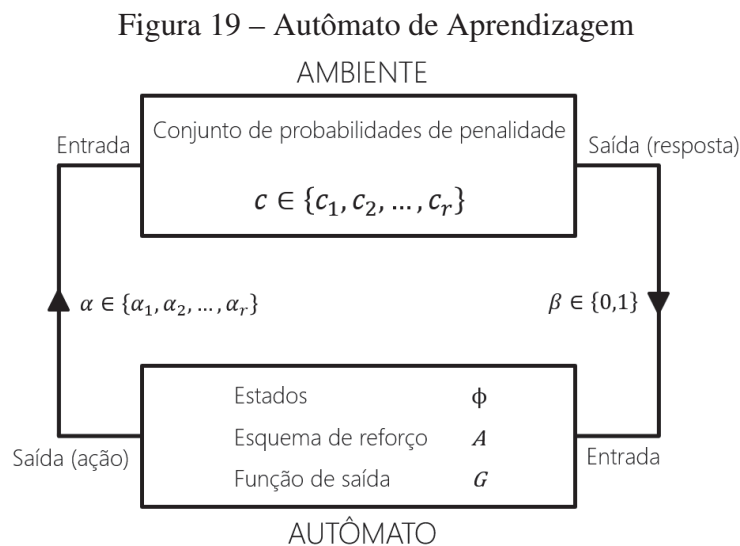
Quando não se possui qualquer informação sobre o sistema, o autômato estocástico pode ser uma solução. Inicialmente, todas as suas ações e probabilidades são iguais. Quer dizer, uma ação é selecionada aleatoriamente e, com base em sua resposta, as probabilidades são atualizadas. Os autômatos que visam a aprimorar seu desempenho são conhecidos como

autômatos estocásticos de aprendizagem.

4.2.1.1 Autômato de Aprendizagem

Os autômatos de aprendizagem podem ser considerados como sistemas que modificam suas estratégias de controle com base em sua experiência para alcançar boas performances de controle, apesar das imprevisões das mudanças no ambiente em que está operando. Em outras palavras, os autômatos de aprendizado por meio da coleta e processamento de informações sobre o ambiente, devem ser capazes de mudar sua estrutura ao longo do tempo para assim alcançarem o objetivo a que se propõem. A necessidade de adaptação nos sistemas de aprendizagem deve-se principalmente ao fato de que o ambiente muda com o tempo e o esquema de aprendizagem deve ter flexibilidade suficiente para rastrear as melhores ações (UNSAL, 1997).

Na Figura 19, pode-se observar a conexão de realimentação do autômato estocástico e o ambiente. Nesse caso, as ações do autômato formam as entradas para o ambiente, e as respostas do ambiente, por sua vez, são as entradas do autômato, influenciando a atualização das probabilidades de ação. Como as respostas são aleatórias, o vetor de probabilidades de ação, denotado por $p(n)$, também é aleatório.



Fonte: Adaptado de (UNSAL, 1997, p. 32) e (NARENDRA; THATHACHAR, 1974, p. 325)

De acordo com Narendra e Thathachar (1974, p. 325), sob a perspectiva da psicologia, a experiência de aprendizagem de um organismo é dita quando se aumenta a probabilidade de respostas corretas resultantes de sua interação com o ambiente. De forma similar, o autômato estocástico aqui abordado possui comportamento semelhante. Pode-se assim referir a esse como um autômato de aprendizagem. Em suma, um autômato de aprendizagem é um autômato estocástico que opera em um ambiente e atualiza suas probabilidades de ação, de acordo com as entradas que recebe do ambiente, de modo a melhorar seu desempenho na seleção das heurísticas de baixo nível.

4.2.1.2 Ambiente

Os autômatos de aprendizagem podem ser demonstrados por um número finito de ações que podem ser realizadas de forma randômica. Ao realizar uma ação específica, o ambiente provê uma resposta aleatória que pode ser favorável ou não. A proposta do autômato é determinar como a escolha de uma ação em qualquer momento deve ser guiada por respostas e ações anteriores. É interessante notar que as decisões devem ser tomadas com pouco conhecimento sobre a “natureza” do ambiente. Além disso, a incerteza é devida ao fato de que a saída do ambiente é influenciada pela ação de outros agentes desconhecidos para o decisor (NARENDRA; THATHACHAR, 1974).

O ambiente onde o autômato “vive” responde às suas ações, por meio de respostas que pertencem a um conjunto de respostas admissíveis e que são probabilisticamente relacionadas com suas ações. É de difícil definição o termo ambiente no contexto de autômatos de aprendizagem, pois, tal definição contempla um ambiente aleatório desconhecido, onde o autômato deve operar. Matematicamente, um ambiente é representado por uma tripla α, c, β , onde α representa um recurso finito do conjunto de ações/saída, β representa um grupo (binário) de resposta/entrada do conjunto, e c é um conjunto de probabilidades de penalização, onde cada elemento c_i está relacionado a um α_i do conjunto de ação α (UNSAL, 1997).

A ação/saída $\alpha(n)$ do autômato pertence a um conjunto α , e é aplicado ao ambiente no instante $t = n$. A entrada $\beta(n)$ do ambiente é um elemento do conjunto β que pode assumir os valores β_1 e β_2 . Em um caso mais simples, os valores de β_i podem ser 0 ou 1, onde 0 está associado com a resposta de sucesso/não penalidade e 1 com a resposta de falha/penalidade. Os elementos de c são definidos conforme a Equação (4.1):

$$Prob \{ \beta(n) = 1 | \alpha(n) = \alpha_i \} = c_i, \quad (i = 1, \dots, r) \quad (4.1)$$

Portanto, c_i é a probabilidade da ação α_i resultar em penalidade de entrada do ambiente. Quando as probabilidades de penalidades c_i são constantes, ou seja, não dependem de n , o ambiente é definido como um *ambiente estacionário* (POZNYAK; NAJIM, 1997).

4.2.1.3 Esquema de reforço

O esquema de reforço é a base do processo de aprendizagem dos autômatos de aprendizagem. Os esquemas são classificados com base em sua linearidade. Narendra e Thathachar (1989) apresentam uma variedade de sistemas lineares, não lineares e híbridos existentes para as várias estruturas de autômatos de aprendizagem. A característica de linearidade de um esquema de reforço é definida pela linearidade das funções de atualização g_k e h_k .

As primeiras pesquisas em esquemas de reforço centraram-se nos sistemas lineares por razões de simplicidade analítica. Eventualmente, a necessidade de esquemas de reforço mais eficientes e complexos elevaram as pesquisas para sistemas não lineares e híbridos (UNSAL,

1997).

A ideia básica por trás do esquema de reforço é simples. Se selecionada a ação α_i no instante n ocorre uma entrada de não penalidade, a probabilidade de ação $p_i(n)$ é aumentada e todos os demais componentes de $p(n)$ são diminuídos. No caso contrário, para uma entrada de penalidade, $p_i(n)$ é diminuído e os outros componentes são aumentados. As alterações em $p_i(n)$ são conhecidas como recompensas ou penalidades, respectivamente.

Neste trabalho, será utilizado um esquema de reforço linear, pois, não se terá variação na estrutura/topologia do autômato a cada interação sua com o ambiente. A definição geral do esquema de reforço linear pode ser obtida pelas funções na Equação (4.2) como segue:

$$\begin{aligned} g_k(p(n)) &= \alpha \cdot p_k(n); \\ h_k(p(n)) &= \frac{\beta}{|p|-1} - \beta \cdot p_k(n); \\ 0 < \alpha, \beta &< 1. \end{aligned} \quad (4.2)$$

Portanto, o regime linear é generalizado como:

Se $\alpha(n) = \alpha_i$,

$$\text{quando } \beta = 0 \quad \begin{cases} p_j(n+1) = (1 - \alpha) \cdot p_j(n) & \forall j \neq i \\ p_i(n+1) = p_i(n) + \alpha \cdot [1 - p_i(n)] \end{cases} \quad (4.3)$$

$$\text{quando } \beta = 1 \quad \begin{cases} p_j(n+1) = \frac{\beta}{|p|-1} + (1 - \beta) \cdot p_j(n) & \forall j \neq i \\ p_i(n+1) = (1 - \beta) \cdot p_i(n) \end{cases} \quad (4.4)$$

Pode-se ver na definição que o parâmetro α está associado com a resposta de recompensa e o parâmetro β com resposta de penalidade. Se os parâmetros α e β são iguais, o sistema é denominado de sistema linear de recompensa e punição, denotado por L_{R-P} (BUSH; MOSTELLER, 1955).

4.2.2 Formalização do Metaespaço

Para a representação do metaespaço, tomamos por base o formalismo das Redes de Autômatos Estocásticos (*Stochastic Automata Networks* - SAN), proposto inicialmente por Planteau em 1984. A adoção das SAN serve para a definição e solução de sistemas complexos por meio da modularização, onde se possui um grande espaço de estados. Basicamente, as SAN tratam-se de uma alternativa para a descrição e modelagem de processos Markovianos com espaço de estados discretos, ou seja, qualquer processo especificado como uma cadeia de Markov pode ser modelado por SAN (ESPINDOLA, 2011).

O formalismo da SAN descreve um sistema completo por meio de um conjunto de subsistemas com comportamentos independentes (eventos locais) e com ocasional interdependên-

cia (taxas funcionais e eventos de sincronização). Tais subsistemas são descritos por autômatos estocásticos, ou seja, um autômato que possui transições modeladas como um processo estocástico de tempo discreto ou contínuo (SALES, 2004). Em resumo, estes subsistemas são estados com regras e funções de probabilidade que dirigem os movimentos entre os estados do autômato.

Cabe notar que toda SAN pode ser representada por um único autômato estocástico que contenha todos os possíveis estados do sistema, chamado de estado global, que é definido pela combinação dos estados locais de cada autômato (BRENNER; FERNANDES; SALES, 2005). Esse autômato único, equivalente à cadeia de Markov, em um modelo em SAN, entretanto, reduz o problema da explosão do espaço de estados (DELAMARE, 2008).

Uma SAN \mathcal{A} é um modelo composto de N autômatos e E eventos, é definido por cada um dos autômatos $\mathcal{A}^{(i)}$ ($i \in [1, \dots, N]$), cada um dos eventos $e \in \mathcal{E}$ e a função de atingibilidade \mathcal{F} . O autômato $\mathcal{A}^{(i)}$ tem como parâmetros a união dos parâmetros de todos os elementos funcionais contidos nos seus rótulos de transição. Descreve-se a seguir formalmente a conceituação da SAN, em que se propõe sua modelagem como uma quadrupla $SAN = (\mathcal{A}, \mathcal{E}, \mathcal{T}, \mathcal{F})$.

Sejam

\mathcal{A}	conjunto de autômatos que compreende N autômatos, tal que $ \mathcal{A} = N$, nomeados $\mathcal{A}^{(i)}$, com $i \in [1, \dots, N]$;
\mathcal{S}	espaço de estados produto do modelo em SAN definido pelo produto cartesiano dos espaços de estados dos autômatos da rede, como $\prod_{i=1}^N \mathcal{S}^{(i)}$;
$\mathcal{S}^{(i)}$	conjunto de estados do autômato $\mathcal{A}^{(i)}$.

Os Eventos são entidades do modelo responsáveis pela ocorrência de uma transição que modifica o estado global do modelo. Aos eventos devem ser associadas probabilidades, que podem ter valores constantes ou funcionais de acordo com os estados dos demais autômatos do modelo. Dentro do formalismo de SAN, podem-se modelar dois tipos de eventos, classificados como (BRENNER; FERNANDES; ALEGRETTI, 2002):

- **Evento Local:** altera o estado local de um único autômato sem ocasionar mudança alguma a outro autômato do modelo;
- **Evento Sincronizante:** altera o estado local de mais de um autômato simultaneamente, fazendo com que haja interação entre eles.

Sejam

\mathcal{E}	conjunto de eventos do modelo compostos por E eventos, tal que $ \mathcal{E} = E$, nomeados e_j , onde $j \in [1, \dots, E]$;
e	evento do modelo, tal que $e \in \mathcal{E}$.

Taxas e Probabilidades Funcionais permitem associar diferentes valores a um mesmo evento de acordo com o estado global do modelo, expressando assim mais de uma alternativa de transição. Dessa maneira, probabilidades são utilizadas para indicar em que proporções o evento seguirá por uma transição ou outra. As taxas e probabilidades funcionais são expressas por funções que levam em consideração os estados atuais dos autômatos do modelo, assim, variando seus valores dado os estados que os autômatos envolvidos na função se encontram (FARINA, 2003). O conjunto \mathcal{T} contém todas as tuplas de transição (e, f) , sendo ao menos uma para cada evento e no conjunto de eventos \mathcal{E} , onde:

Sejam

e	identificador do evento;
f	elemento funcional definido de $\mathcal{S} \rightarrow [0, 1]$, o qual define a probabilidade de uma transição quando da ocorrência do evento e ;
\mathcal{T}	conjunto de tuplas de transição local de um modelo SAN, tal que $\mathcal{T}^* = \mathcal{T} \cup \{\}$.

Os elementos funcionais definem taxas de probabilidades funcionais, mesmo aqueles que possuem valor constante. Obviamente, tal definição não representa uma restrição, visto que elementos constantes podem ser vistos como funções constantes. Dessa maneira, todos os elementos de um modelo em SAN podem ser considerados como funções de $\prod_{i=1}^N \mathcal{S}^{(i)} \rightarrow R^+$.

Função de Atingibilidade define quais estados podem ser atingíveis ou alcançados em um modelo de SAN. Essa função se faz necessária, pois o autômato global constitui-se modularmente pela combinação de todos os autômatos do modelo (BRENNER, 2004)

Seja

\mathcal{F}	função de atingibilidade é um elemento funcional que associa aos estados globais de $\prod_{i=1}^N \mathcal{S}^{(i)}$, definido de $\mathcal{S} \rightarrow [1, \dots, 0]$, onde o $dom(\mathcal{S})$ representa todas as possibilidades de estados globais do modelo, e o $codom([0, \dots, 1])$ fornece quais estados podem ser atingíveis em um modelo SAN.
---------------	--

4.2.3 Modelagem do Metaespaço

A abordagem realizada nesta seção é de longe final, nem intenciona sobre tal, pois, a heterogeneidade das meta-heurísticas é tanta que são descritas neste trabalho apenas 8 técnicas. Esta formalização dirige-se para, por meio das SAN, constituir um desenho inicial das meta-heurísticas apresentadas no capítulo anterior.

O desenho parte do mapeamento das meta-heurísticas através da identificação de seus mecanismos, que posteriormente definem os eventos. A representação dos eventos se dá por

meio de um alfabeto, que rege as expressões regulares dos autômatos. A delimitação do alfabeto/eventos e consequentemente das expressões regulares, facilita a visualização das definições das funções de probabilidade, e consequentemente a validação das entradas para os autômatos.

Na Tabela 2, têm-se uma versão reduzida das definições mencionadas para a construção dos autômatos, contemplando apenas os parâmetros e métodos das meta-heurísticas. Foram inseridos para fins de ilustração um conjunto limitado de eventos, na intenção de representar os principais mecanismos heurísticos presentes nas técnicas levantadas no capítulo anterior¹.

A construção destes autômatos isolados são partes componentes da rede de autômatos estocásticos que caracteriza no modelo proposto uma representação do metaespaço. A elaboração das SAN de cada meta-heurística propiciou com que fossem identificadas intersecções entre as meta-heurísticas. Tais intersecções fazem-se ver na presença de elementos comuns entre técnicas, a exemplo vê-se na Tabela 2 a repetição de mecanismos através do alfabeto criado para tal descrição.

4.2.3.1 Exemplo de expressão regular

Para entender as expressões regulares apresentadas na Tabela 2 e Apêndice A, adicionamos uma legenda dos símbolos usados, apresentada na Tabela 3.

A seguir, apresentamos um exemplo de como as expressões regulares foram construídas, explicando o autômato $A^{(30)}$ descrito na Tabela 4, o qual é composto por três partes:

Tabela 4 – Composição do autômato $A^{(30)}$ - Hiper-heurística

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Hiper-heurística			$A+ (B+ C+ A+)^*$

Fonte: Elaborado pelo autor

- Parte I: $A+$ avalia um ou mais critérios de parada. Como A é um evento de sincronização, irá executar a SAN ($A1 | A2 | A3$) com diferentes critérios de parada.
- Parte II: O próximo elemento é o conjunto de parênteses ($B+ C+ A+$), representando um grupo de execuções que ocorre sequencialmente.
 - $B+$ realiza uma ou mais vezes a fase Geracional. Como B é um evento de sincronização, ele irá executar a sub-rede ($D | E | F | J | K | M | O | S | T$) uma ou mais vezes.
 - $C+$ realiza uma ou mais vezes a fase Substitutiva. Como C é um evento de sincronização, ele vai executar a sub-rede ($G | H | I | L | N | Q | R | U$) uma ou mais vezes.

¹As representações completas, de todos os 30 autômatos e suas expressões regulares, estão descritos sobre todos os níveis no Apêndice A.

Tabela 2 – Delimitação do Alfabeto

Autômato	Evento (Mecanismo Heurístico)		Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
$A^{(1)}$	Sistema Meta-heurístico Geral				$A(BCA)^*$
$A^{(2)}$	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
$A^{(5)}$	Fase Geracional	$e^{(5)}$	Sincronizante	B	$(D E F J K M O S T)$
$A^{(4)}$	Fase Substitutiva	$e^{(6)}$	Sincronizante	C	$(G H I L N Q R U)$
$A^{(5)}$	Algoritmo Genético				$A(DEFGA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
$A^{(6)}$	Seleção	$e^{(7)}$	Sincronizante	D	$(D1 D2)$
$A^{(9)}$	Cruzamento	$e^{(22)}$	Sincronizante	E	$(E1 E2 E3 E4 E5 E6)$
$A^{(10)}$	Mutação	$e^{(29)}$	Sincronizante	F	$(F1 F2)$
	Atualização da População	$e^{(44)}$	Local	G	
$A^{(13)}$	Arrefecimento Simulado				$A(FHIA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
	Criação solução vizinha	$e^{(29)}$	Sincronizante	F	$(F1 F2)$
$A^{(14)}$	Temperatura Inicial	$e^{(45)}$	Sincronizante	H	$(H1 H2 H3 H4 H5 H6)$
$A^{(15)}$	Redução da Temperatura	$e^{(52)}$	Sincronizante	I	$(I1 I2 I3 I4 I5 I6)$
$A^{(16)}$	Busca Dispersa				$A(JDEKL^*A)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
$A^{(17)}$	Tamanho Conjunto de Diversificação	$e^{(59)}$	Sincronizante	J	$(J1 J2 J3 J4 J5 J6)$
	Seleção de um subconjunto	$e^{(7)}$	Sincronizante	D	$(D1 D2)$
	Recombinação do subconjunto	$e^{(22)}$	Sincronizante	E	$(E1 E2 E3 E4 E5 E6)$
$A^{(18)}$	Tamanho Conjunto de Referência	$e^{(66)}$	Sincronizante	K	$(K1 K2 K3 K4 K5 K6)$
$A^{(19)}$	Busca Local	$e^{(73)}$	Sincronizante	L	$(L1 L2)$
$A^{(20)}$	Busca em Vizinhança Variável				$A(MFLA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
$A^{(21)}$	Vizinhanças	$e^{(76)}$	Sincronizante	M	$(M1 M2 M3 M4 M5 M6)$
	Geração da vizinhança	$e^{(29)}$	Sincronizante	F	$(F1 F2)$
	Busca Local	$e^{(73)}$	Sincronizante	L	$(L1 L2)$
$A^{(22)}$	Busca Local Iterada				$A(FLNA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
	Perturbação	$e^{(29)}$	Sincronizante	F	$(F1 F2)$
	Busca Local	$e^{(73)}$	Sincronizante	L	$(L1 L2)$
	Critério de aceitação	$e^{(83)}$	Local	N	
$A^{(23)}$	Busca Tabu				$A(OMFPA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
$A^{(24)}$	Tamanho Lista Tabu	$e^{(84)}$	Sincronizante	O	$(O1 O2 O3 O4 O5 O6)$
	NBMax (Vizinhanças)	$e^{(76)}$	Sincronizante	M	$(M1 M2 M3 M4 M5 M6)$
	Geração da vizinhança	$e^{(29)}$	Sincronizante	F	$(F1 F2)$
	Atualização da Lista Tabu	$e^{(91)}$	Local	P	
	Critério de aspiração	$e^{(92)}$	Local	Q	
$A^{(25)}$	GRASP				$A(RLA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
$A^{(26)}$	Construção Aleatória	$e^{(93)}$	Sincronizante	R	$(R1 R2 R3 R4 R5 R6)$
	Busca Local	$e^{(73)}$	Sincronizante	L	$(L1 L2)$
$A^{(27)}$	Otimização por Colônia de Formigas				$A((FS)+TA)^*$
	Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
	Construção da Solução	$e^{(29)}$	Sincronizante	F	$(F1 F2)$
$A^{(28)}$	Atração do Feromônio	$e^{(100)}$	Sincronizante	S	$(S1 S2 S3 S4 S5 S6)$
$A^{(29)}$	Evaporação do Feromônio	$e^{(107)}$	Sincronizante	T	$(T1 T2 T3 T4 T5 T6)$
$A^{(30)}$	Hiper-heurística				$A+(B+C+A)^*$

Fonte: Elaborado pelo autor

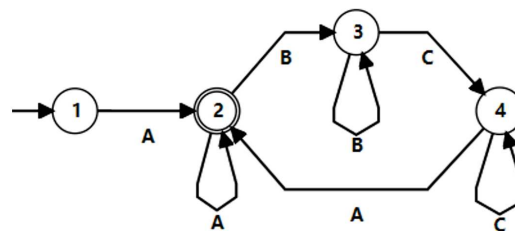
Tabela 3 – Legenda de expressões regulares

Caractere	Legenda
+	Um ou mais
(...)	Grupo de captura
	Operador lógico OU
*	Zero ou mais vezes

Fonte: Elaborado pelo autor

- A+ avalia novamente, um ou mais critérios de parada, como mencionado na Parte I.
- Parte III: O caractere final *, expressa que o grupo na Parte II pode ser executado zero ou mais vezes, considerando que a avaliação do primeiro critério de parada (Parte I) também pode ser alcançada no início da operação.

Para ilustração, apresentamos na Figura 20 o autômato $A(30)$

Figura 20 – Autômato $\mathcal{A}^{(30)}$ - Hiper-heurística

Fonte: Elaborado pelo autor

Por fim, o caractere |, apresentado na Parte I no autômato $(A1 | A2 | A3)$, possui um grupo formado por eventos locais ($A1$ OU $A2$ OU $A3$), que representam diferentes critérios de parada. Por exemplo, ao executar uma meta-heurística, um ou mais critérios de parada podem ser acionados pelo Bloco de Seleção e Construção, quando esse bloco gera a sequência de execução. No caso de $A1$ representa o número de interações sem melhora, delimitado como 100 interações, $A2$ representa o número de repetições, definido como 500, e $A3$, o número de avaliações, definido como 1000.

4.3 Arquitetura do módulo Otimizador

O módulo Otimizador pode ser considerado o núcleo de interesse desta arquitetura, uma vez que é o responsável pelo principal objetivo, resolver instâncias de problemas de otimização. O funcionamento do módulo (21) baseia-se no acionamento do bloco executor, responsável

por sequenciar a execução do conjunto de heurísticas de baixo nível a fim de otimizar um problema. Na arquitetura, o fluxo deste módulo recebe da hiper-heurística uma cadeia de caracteres contendo as instruções de execução. Esta cadeia de caracteres acaba por determinar quais mecanismos serão acionados, bem como a ordem de execução.

Ao receber o conjunto de caracteres que delimitam sua operação, o executor aciona e coordena o fluxo de entrada e saída de cada um dos mecanismos heurísticos descritos pelo ciclo da hiper-heurística. Cabe notar que, o processo de avaliação das soluções é coordenado pelo módulo executor, mas é realizado no bloco de interface.

Esta separação de comportamento faz com que haja um isolamento lógico entre a execução das heurísticas e a avaliação do problema, que no contexto desta arquitetura se dá através da barreira de domínio. Destaca-se que, diferentemente da arquitetura proposta por Burke (2003) (Vide seção 3.9.1), onde a função de avaliação se encontra no mesmo nível das heurísticas, operando com dados do domínio, no modelo desenvolvido efetua-se um isolamento, estando livre de quaisquer informações do problema.

Ao final do ciclo de operações do bloco executor, calcula-se o Grão Computacional, que é informado para a hiper-heurística a fim de que esta coordene o próximo ciclo de ações. Enquanto o módulo Otimizador continuar recebendo cadeias de caracteres da hiper-heurística, o processo de otimização perdurará. Já, se a cadeia de caracteres recebida no ciclo seguinte for vazia, finda-se o processo de otimização e o valor e a solução de saída são informados ao bloco de interface.

Figura 21 – Arquitetura do Módulo Otimizador



Fonte: Elaborado pelo autor

4.4 Arquitetura do bloco de interface

Para a construção do bloco de interface, foram caracterizadas as condições necessárias à otimização de um problema. Buscando vislumbrar as modificações que possam ser aplicadas para o atendimento de outros padrões de problemas, foram delimitados alguns parâmetros de entrada no modelo. De forma geral, foram delimitadas as partes componentes de um problema de otimização, sendo elas: função de objetivo, variáveis de decisão, limites e restrições.

Na Figura 22, observa-se o bloco de interface, que foi desenvolvido como uma interface de conexão com o restante da arquitetura. Optou-se pelo desenvolvimento de uma interface, visto que, a cada necessidade possa ser estendida/reconfigurada sob a demanda de novos problemas. Sua composição é delimitada por três entradas e duas saídas externas à arquitetura.

As entradas visam à representação das características de um problema de otimização, como descrito acima, sendo a função objetivo representada na primeira entrada, e o Leitor da Instância corresponde as variáveis de decisão, limites e restrições. Já o bloco Construtor da Solução é uma representação programática da solução, ou seja, será composto pelas variáveis de decisão, e avaliado pela função objetivo dentro dos limites e restrições impostos pelo problema. Cabe ressaltar que, as três entradas possuem relação direta entre si, no entanto foram fragmentadas por questões de facilitação da inserção do problema na arquitetura.

As saídas do bloco são duas, a solução final obtida durante o processo de otimização e seu valor alcançado. Nota-se que a solução de saída possui relação com o construtor da solução, e conseqüentemente o valor de saída tem dependência da função de avaliação.

Figura 22 – Arquitetura do Bloco de Interface



Fonte: Elaborado pelo autor

4.5 Retorno analítico

A importância do retorno analítico está no fato das perspectivas que foram traçadas no início da pesquisa serem parte da intenção que desenvolvemos ao longo do trabalho, reunir os atravessamentos construídos ao longo da investigação, conseguindo sintetizar esse caminho de linhas divergentes que saíram de um primeiro objeto de estudo, que conduz a movimentação no espaço de soluções. Nesse processo se buscou entender como os mecanismos heurísticos se comportam e agenciam os movimentos em suas estruturas heurísticas. Esse extrato teórico nos fez compreender como tais mecanismos criam formas de movimentação no espaço de busca,

atribuindo sentidos identitários aos processos de exploração (*exploration/exploitation*) que nos são de interesse.

Após essa etapa, com as perspectivas de diferenciação do objeto sendo lançadas, o processo encontrou na consolidação do modelo seu ápice. Pode-se agora observar com o amadurecimento da pesquisa o aparecimento das perspectivas no escopo do modelo desenvolvido.

Quando foi apresentada a perspectiva da Navegabilidade, lançamos os olhares para a compreensão dos mecanismos heurísticos capazes ou não de gerar trajetórias em seu processo de movimento e oscilação pelo espaço de busca. Dessa perspectiva, conseguiu-se abordar dentro do conjunto de técnicas aplicadas ao modelo, as três modalidades identificadas. Neste momento, nos diferenciamos da maioria dos trabalhos que tivemos contato, pois a hiper-heurística foi fundamentada em um conjunto amplo de técnicas não encontrado na literatura.

A segunda perspectiva fala sobre a completude da busca e analisa a capacidade de exploração do espaço de soluções, não se refere aqui à capacidade no sentido qualitativo da busca, mas sobre a quantificação de sua exploração. Nesta perspectiva, foram dissociados os métodos completos dos métodos incompletos, que são frutos do interesse de pesquisa. Ainda que não tenham sido abordados, os métodos completos fazem parte de um conjunto que tratamos como possibilidades futuras na expansão e renovação dos horizontes produtivos para o modelo proposto.

A terceira perspectiva dialoga sobre a Gênese das soluções, em como os métodos geram soluções dentro de seus processos. O uso de métodos com bases perturbativas e construtivas, permitiu a contemplação das características da natureza do espaço de busca heurístico classificado por Burke et al. (2003). Diante dessa perspectiva e de sua ordem gerativa, cabe explorar as dimensões alcançáveis pelo modelo, ainda que não tenham sido exploradas em suas totalidades.

O modelo H2-SLAN demonstrou-se capaz de operar como uma heurística de seleção construtiva e perturbativa, e de geração construtiva e perturbativa, cabe ressaltar que estas características foram exploradas de maneiras distintas. As heurísticas de seleção serviram para a condução dos experimentos das meta-heurísticas, pois delimitando as expressões regulares de cada experimento, o modelo operou em sua forma mais simples, apenas selecionando os mecanismos de construção ou perturbação próprios de cada meta-heurística em seus parâmetros e níveis delimitados. Do outro lado, explorou-se as heurísticas de geração construtiva e perturbativa de maneira geral, trazendo à tona a natureza hiper-heurística do modelo. Nesta forma de operação, a SAN é responsável por gerar uma nova heurística, independentemente de suas características construtivas ou perturbativas. Essas características foram exploradas de maneira conjunta, pois as meta-heurísticas base contemplam ambas. Ainda que não se tenha explorado as duas possibilidades hiper-heurísticas individualmente, elas estão presentes no modelo através da possibilidade de delimitação de expressões regulares específicas.

Com base na classificação hiper-heurística proposta por (DRAKE et al., 2019), o modelo apresenta uma hiper-heurística mono-objetivo, com três tipos de feedback on-line, off-line e sem aprendizado, deixando a escolha do modelo determinado pelos testes. Quanto às heurísti-

cas de baixo nível, o modelo apresenta heurísticas de baseadas em uma solução e populacionais, sendo caracterizado como misto. Note-se que a escolha de heurísticas de baixo nível que serão executadas é delimitada por uma expressão regular que engloba as três possibilidades. O critério de aceitação do movimento é estocástico, uma vez que é fornecido através da rede de autômatos estocásticos usados na construção da cadeia de execução. Os parâmetros de configuração são definidos de acordo com a expressão regular. Eles podem ser estáticos, como em testes individuais com meta-heurísticas e adaptativos no caso de hiper-heurísticas, que, de acordo com as características obtidas em cada ciclo operacional, adaptam os parâmetros para a execução do próximo ciclo.

Outra perspectiva que foi desenvolvida dentro do contexto do modelo diz respeito à Universalidade, que comumente não é tratada de maneira adequada na literatura, onde são apresentadas hiper-heurísticas baseadas em algoritmos populacionais ou em trajetória. Por vezes tratamos esta caracterização em outras perspectivas, mas aqui identificamos claramente a necessidade de um algoritmo agnóstico, ou seja, que não exercesse quaisquer dependências das estruturas heurísticas originais, sendo generalista na forma de tratamento das soluções. Este passo serviu para a criação do autômato heurístico base de toda a operação do modelo.

A perspectiva da Persistência, traz consigo uma complexidade onde não olhamos apenas a estrutura das meta-heurísticas na intenção de identificar a presença de memórias diretas, mas visamos a memória indireta, onde acaba-se por observar a qualidade das soluções influenciando a busca. Ao abordar a qualidade e a influência da memória, acabamos classificando as soluções nos termos da importância das propriedades que movem a busca para regiões promissoras do espaço, e não na representação das características das soluções. Quando nos referimos à memória direta, tratamos da recência e da frequência com que se efetua o registro de informações que influenciam de maneira objetiva a busca. No modelo, pode-se encontrar apenas na Busca Tabu a presença de memória direta, nas demais meta-heurísticas e no meta-espaço temos a presença da memória indireta influenciando os ciclos futuros da busca.

Ao abordarmos a influência conducente da memória indireta, traz-se outro viés das hiper-heurísticas, referente às fontes de *feedback*, que se dividem em três atuações (*on-line*, *off-line* e sem aprendizado). Nos trabalhos de McClymont e Keedwell (2011a, 2011b) e Falcón-Cardona e Coello (2018), encontramos uma abordagem que faz uso de cadeias de Markov para representação do aprendizado. Em Li, Özcan e John (2017) e Li (2018), é introduzida uma abordagem baseada similar, mas com o uso de autômatos de aprendizagem. Estas foram as principais aproximações do modelo H2-SLAN encontradas na literatura.

O modelo desenvolvido difere-se pela presença de todas as possibilidades de *feedback*, pois este processo é fruto apenas de sua parametrização. No entanto, os experimentos hiper-heurísticos conduzidos, apresentados no capítulo seguinte, são operados somente com aprendizado *on-line*, pois intenciona-se demonstrar a potencialidade do metaespaço e da aprendizagem conduzida sobre ele.

No tocante à Contiguidade da busca, observa-se as relações de movimentos e fluxos, cir-

culação e conectividade que são empregadas pelos mecanismos na exploração (intensificação e diversificação) do espaço de soluções. As meta-heurísticas aplicadas no modelo apresentam aspectos de base intrarregionais e extrarregionais. Entretanto, pela combinação de mecanismos agenciada pela hiper-heurística, promove-se hibridizações, surgindo o terceiro aspecto da Contiguidade, o inter-regionalismo.

Por fim, a perspectiva da Aptidão se divide no que chamamos de aptidão interna, como uma via de avaliação dada pela função objetivo do problema, que é presente em todas as meta-heurísticas. No outro extremo, a perspectiva mais complexa, que é referente à aptidão externa, ou seja, sobre a qualificação das soluções, dada pelo Grão Computacional, como um meio de conduzir a busca.

4.6 Considerações sobre o capítulo

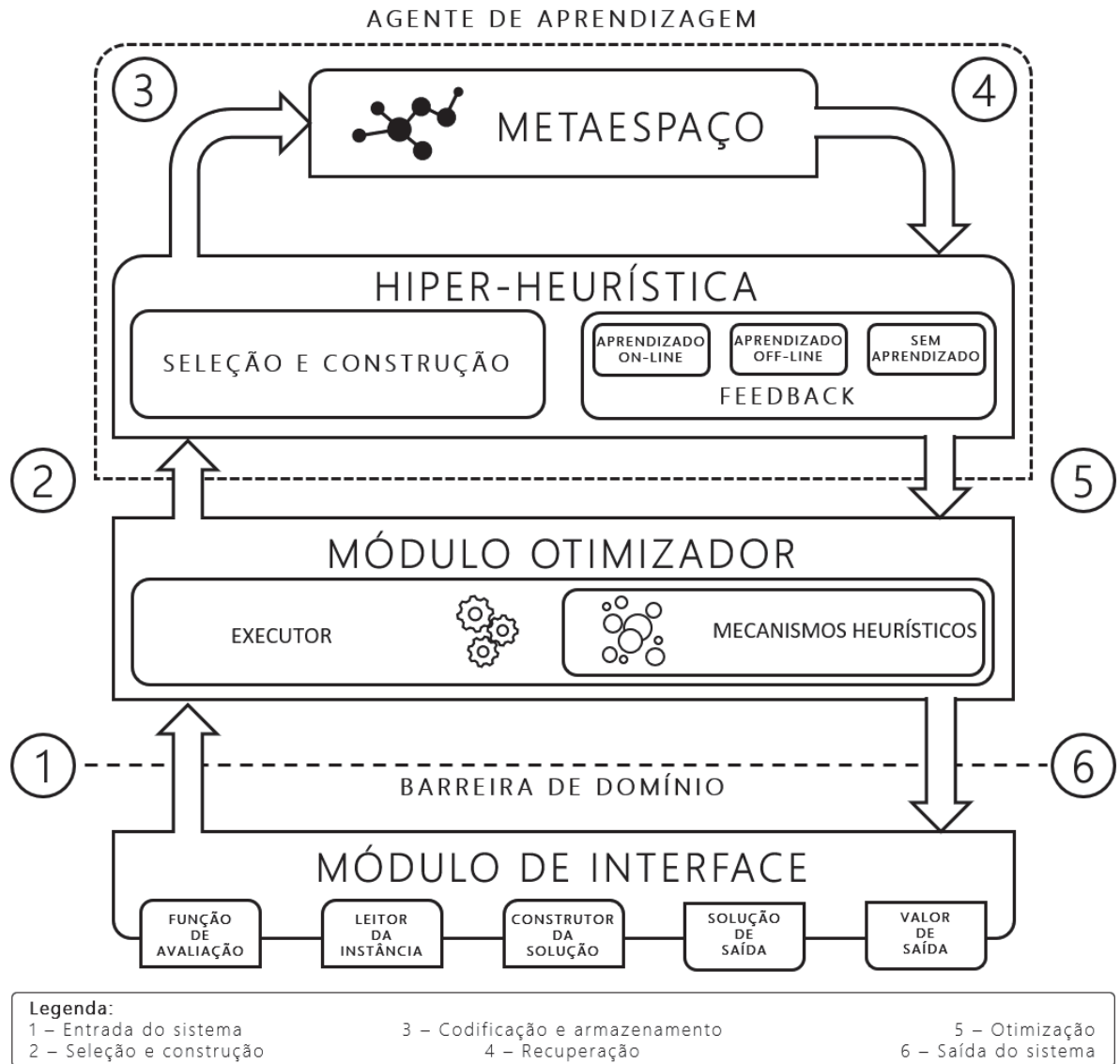
Este capítulo apresentou a arquitetura do H2-SLAN, um modelo Hiper-heurístico baseado em Redes de Autômatos Estocásticos de Aprendizagem. O modelo divide-se em três blocos conforme apresentado no decorrer do capítulo. De acordo com a visão geral apresentada na Figura 23, o modelo hiper-heurístico proposto baseado em redes de autômatos estocásticos com aprendizado permite o controle e a integração de vários mecanismos heurísticos e opera como o algoritmo ilustrado na Figura 24.

As entradas do sistema atravessam a barreira do domínio, fornecendo as instruções para a execução do algoritmo; são o tamanho da população que será operada pelo modelo (linha 1), expressão regular que condiciona a ordem e quais heurísticas podem ser definidas (linha 2), construtor de soluções (linha 3), função de avaliação (linha 4), e a rede de autômatos estocásticos (linha 5). A operação começa com o Módulo Executor que inicializa a população base delimitada como 100 indivíduos (linha 6) e armazena a melhor solução dessa população inicial (linha 7), para calcular posteriormente o grão computacional.

A consulta ao metaespaço para obter uma sequência de execução é realizada pelo bloco Seleção e Construção (linha 8), que constrói a sequência consultando a SAN com base na expressão regular de entrada. Essa expressão regular define quais mecanismos e a ordem que serão acionados. Esta etapa fornece o desenho de um mapa de possibilidades de interconexões dos mecanismos definidos no metaespaço, levando-nos a um decalque instantâneo de necessidades momentâneas, permitindo a adaptação às mudanças introjetadas pelos novos domínios de aplicação que estarão presentes.

Devemos observar que o metaespaço é representado apenas pela SAN, como uma maneira de reter as informações para a operação hiper-heurística. Essa modularização ocorre porque pensamos a fase de aprendizado isolada de seu uso; isso significa que o processo de seleção de heurísticas é independente da construção do conhecimento. Essa estrutura permite um acoplamento fraco se houver necessidade de trocar tecnologias de codificação, armazenamento e recuperação.

Figura 23 – Fluxo de operação da arquitetura do modelo H2-SLAN



Fonte: Elaborado pelo autor

Figura 24 – Algoritmo H2-SLAN

```

popSize : tamanho da população;
regEx : expressão regular;
solBuilder : construção da solução;
OF : função de avaliação;
SAN : rede de autômato estocásticos;
Pop ← InicializarPopulação (popSize, solBuilder);
S* ← ObterMelhorSolução (Pop);
execString ← SeleçãoConstrução (regEx);
while ¬CondiçãoDeParada (execString [1]) do
    for h ← 2 to | execString | do
        aplicarMeta-Heuristica (execString [h ], Pop, solBuilder,
            leitorDaInstncia, OF);
    end
    CG ← obterGrãoComputacional (Pop, S*);
    S* ← ObterMelhorSolução (Pop);
    atualizarSAN (SAN, CG, execString);
    execString ← SeleçãoConstrução (regEx);
end
return S*;

```

O módulo Otimizador recebe uma sequência contendo a ordem de execução das instruções correspondentes aos mecanismos heurísticos. Assim, o ciclo principal do algoritmo será executado se não atender à condição de parada (linha 9), representada pela primeira posição da cadeia de execução, que sempre será uma condição de parada de acordo com a estrutura dos autômatos $A^{(2)}$. Aplicamos três critérios diferentes de parada na hiper-heurística, representada pelos eventos $e^{(2)}$, $e^{(3)}$ e $e^{(4)}$, representados respectivamente pelos eventos funcionais $A1$, $A2$ e $A3$. Onde $A1$ representa o número de interações sem melhoria, delimitadas como 100 interações; $A2$ representa o número de repetições, definido como 500; e $A3$, o número de avaliações, definido como 1000.

O algoritmo executa cada elemento da sequência de execução, do segundo elemento até o final da sequência (linha 10), aplicando o mecanismo heurístico h sobre toda a população ou parte dela, dependendo das características da meta-heurística (linha 11). Quando uma população se aplica a um algoritmo de solução única, apenas o melhor indivíduo dessa população é processado pela meta-heurística. Caso contrário, quando um algoritmo de solução única retorna a um algoritmo de população, ele se torna parte de sua população. Assim, temos uma população constante entre transições de métodos. Após aplicar todos os mecanismos heurísticos descritos pela sequência de execução, calcula-se o grão computacional do ciclo atual (linha 13). Então, a

melhor solução para o ciclo atual é obtida para calcular o grão computacional para o próximo ciclo (linha 14).

Quando o bloco Feedback recebe o Grão Computacional, de acordo com o formato de aprendizado (online, offline e sem aprendizado), ele aplica as regras de aprendizado no metaespaço, atualizando as probabilidades da rede de autômatos estocásticos (linha 15). No final do ciclo, o módulo Executor solicita novamente ao bloco de Seleção e Construção uma nova sequência de execução para o próximo ciclo (linha 16), que retorna à linha 9 e avalia a condição de parada. Se não satisfizer a condição, o ciclo é repetido (linhas 10 a 16); caso contrário, se for satisfeito, o bloco Executor termina a operação e retorna a melhor solução obtida (S^*) (linha 18).

Como já mencionado, o modelo visa a navegação no metaespaço, construído sobre a perspectiva de rizoma, por meio da modelagem sistêmica da interação entre os mecanismos heurísticos. É nesta correspondência funcional que através das SAN será efetuado o armazenamento, a codificação e a recuperação do conhecimento acerca das interações entre os mecanismos heurísticos.

5 PANORAMAS DE EXPERIMENTAÇÃO

*“O que eu ouço, eu esqueço.
O que eu vejo, eu lembro.
O que eu faço, eu entendo.”*

Confúcio

Nos capítulos anteriores, foram introduzidas as áreas que conduziram esta pesquisa até este momento. Neste caminho, construímos definições epistêmicas e técnicas, chegando ao modelo hiper-heurístico que a partir deste capítulo será validado sob os aspectos de funcionalidade e desempenho. O compilado das construções se constitui como uma das contribuições mais importantes da tese para o campo da Computação por serem sugestivas e lançarem uma visão ressignificada sobre estes temas.

Observar o funcionamento de um sistema é uma parte importante do processo de aprendizagem, e é parte integrante da compreensão sobre como o sistema e seus processos funcionam. Isso significa que, para entender realmente as relações de causa e efeito em um sistema, é necessário modificar suas variáveis para observarmos as alterações produzidas (MONTGOMERY, 2013). Em outras palavras, é preciso realizar experimentos no sistema.

Assim, na continuidade deste capítulo, a seção 5.1 apresenta os aspectos do planejamento dos experimentos, abordando os conceitos utilizados. Para analisarmos um experimento, é preciso inicialmente delimitar claramente o objeto de estudo, como os dados devem ser coletados e, pelo menos, uma compreensão qualitativa de como esses dados devem ser analisados.

A forma adotada para este processo foi o planejamento experimental, apresentado na seção 5.2, que contempla a metodologia de validação e os dados aplicados aos 241 experimentos, onde realiza-se uma série de ajustes no processo, fazendo mudanças nas entradas e observando as mudanças correspondentes nas saídas, o que produzirá informação que pode levar à melhoria do processo. Nela, realiza-se uma série de ajustes no processo, de modo a oferecer para quem a aplica a visualização de um processo de modificações, já que as mudanças nas entradas e as mudanças correspondentes nas saídas podem ser acompanhadas, de fato. Essa característica pode ser vista enquanto positiva, já que produzirá informações que podem levar à melhoria do processo.

Já na seção 5.3, são apresentados os experimentos realizados para validação dos aspectos funcionais e de desempenho. A seção divide-se em cinco grandes grupos, correspondentes às instâncias da TSPLib (*berlin52*, *eil101*, *d493*, *u1060* e *rl1889*), que se dividem em 9 subseções cada, correspondentes às oito meta-heurísticas analisadas e a hiper-heurística desenvolvida. Por fim, na seção 5.4, são apresentadas considerações sobre os resultados obtidos pela hiper-heurística, analisando a frequência de execução dos mecanismos heurísticos.

5.1 Planejamento dos Experimentos

Antes de adentrar ao projeto dos experimentos, cabe introduzir alguns termos. Particularmente, faz-se necessário associarmos nosso objeto de pesquisa com os termos utilizados na concepção dos experimentos. Ao fazer uso do termo *fatores*, neste contexto refere-se aos mecanismos das meta-heurísticas levantados no Capítulo 3. Entretanto, pretende-se estudar tais fatores aplicados a diferentes instâncias do mesmo problema, onde cada experimento será definido pela combinação de características e *níveis* dos fatores (RARDIN; UZSOY, 2001).

Por *níveis*, entende-se como as possibilidades que um determinado fator pode assumir. Os níveis podem representar parâmetros (e.g. comprimento da lista de tabu, tamanho da população), métodos (e.g. tipo de perturbação, operador de mutação), ou até mesmo a combinação destes.

Levar-se-á também em consideração a randomização e o bloqueio como aspectos relevantes. Compreende-se randomização por fatores que não serão explicitamente controlados, devendo assim ser definidos para evitar vieses sistemáticos (DEAN; VOSS, 1999). O objetivo fundamental da randomização é a certeza de que cada tratamento tem a mesma probabilidade de ser atribuído a qualquer unidade experimental. Uma desvantagem dessa abordagem é o aumento na variabilidade observada.

Para redução da variabilidade, far-se-á uso da técnica de bloqueio em fatores que não são de interesse, ou seja, os fatores que geram perturbação. O bloqueio consiste em segmentar as observações em blocos de maneira que cada observação se encontre em condições semelhantes (e.g. faixas de valores de mutação, comprimentos de lista tabu, tamanho de população/vizinhança) (DEAN; VOSS, 1999).

Segundo Birattari et al. (2001, p. 6, tradução nossa)¹,

No projeto experimental de meta-heurísticas, podemos simplesmente corrigir o fator de incômodo durante o experimento, (embora possa limitar as conclusões em determinados níveis), ou melhor, podemos implementar o procedimento de bloqueio por meio da definição de níveis para o fator de incômodo e executar um conjunto de experiências para cada nível.

Entretanto, na análise fatorial deve-se considerar alguns ajustes. Primeiramente, a determinação da importância dos fatores, selecionando os mais relevantes, pois o projeto fatorial requer a combinação de todos os níveis e fatores, acarretando com isso o crescimento explosivo do número de experimentos, requerendo assim um processo de triagem no projeto fatorial (MC-GEOCH, 1996; RARDIN; UZSOY, 2001). Alternativas podem ser consideradas neste caso, reduzindo as combinações e mantendo a ortogonalidade da matriz de experiências (BIRATTARI

¹“In experimental design of metaheuristics we can simply fix the nuisance factor during the experiment, (although it could limit the conclusions to that particular chosen level), or rather we can implement the blocking procedure by defining levels for the nuisance factor and running a set of experiments for each level.”

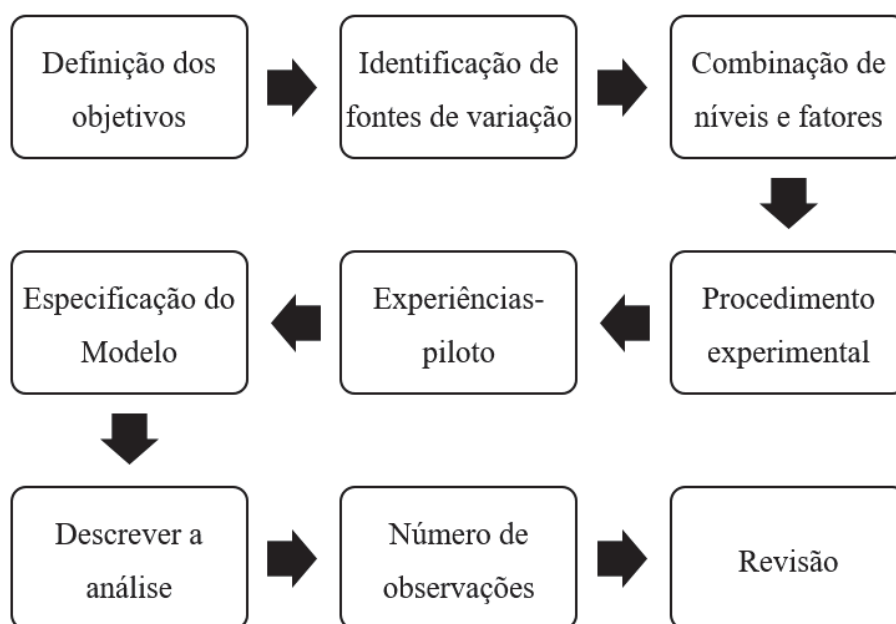
et al., 2001).

Após a obtenção de resultados, visa-se a analisar os dados por meio de ferramentas estatísticas. O processo de análise dos resultados médios e do desvio padrão auxilia a delimitar os fatores de significativo impacto na construção e seleção de heurísticas de baixo nível (RARDIN; UZSOY, 2001).

5.2 Validação dos experimentos

A validação dos experimentos foi dividida em nove etapas (Figura 25), tomando por base a metodologia de Dean e Voss (1999) para projetar experimentos computacionais a fim de compreender o comportamento das meta-heurísticas em problemas de Otimização Combinatória.

Figura 25 – Validação dos Experimentos



Fonte: Elaborado pelo autor

Iniciou-se pela *definição do objetivo* por meio de uma pergunta a ser abordada no experimento. No que diz respeito à pesquisa, intenta-se *estudar qual o efeito dos diferentes mecanismos heurísticos e seu comportamento frente à construção e a parametrização das heurísticas de baixo nível, compreendendo as ações de intensificação e diversificação aplicadas durante o processo de busca*. Não obstante, fez-se a triagem paramétrica, com base nos autômatos construídos para cada meta-heurística, a fim de encontrar o melhor conjunto de parâmetros e métodos por meio de testes estatísticos e do design experimental (XU; CHIU; GLOVER, 1998). Para cada meta-heurística foram aplicados os parâmetros e métodos exibidos na Tabela 5:

Tabela 5 – Parâmetros e métodos aplicados aos experimentos

Algoritmo Genético		Arrefecimento Simulado	
Taxa de Cruzamento	Taxa de Mutação	Temperatura Inicial	Redução da Temperatura
Busca Dispersa		Busca em Vizinhança Variável	
Conjunto de Referência	Conjunto de Diversificação	Vizinhanças	Perturbação
Busca Local Iterada		Busca Tabu	
Busca Local	Perturbação	NBMax	Lista Tabu
GRASP		Otimização por Colônia de Formigas	
Busca Local	Lista de Candidatos α	Atração do Feromônio	Evaporação do Feromônio

Fonte: Elaborado pelo autor

Na sequência, fez-se a *identificação das fontes de variação, métodos e níveis correspondentes*, bem como a escolha das unidades experimentais que foram utilizadas nos testes. Nesta definição, optou-se por instâncias de teste onde o ótimo global fosse conhecido, assim conseguimos mensurar a proximidade dos resultados obtidos em relação ao ótimo global e, consequentemente, se estabelece uma medida de qualidade. Para este propósito, optou-se por resolver o Problema do Caixeiro Viajante, onde as unidades experimentais escolhidas são provenientes da TSBLib, sendo elas: berlin52, eil101, d493, u1060 e rl1889, cujos valores ótimos conhecidos são respectivamente 7542, 629, 35002, 224094 e 316536. As unidades experimentais foram resolvidas pela combinação dos níveis dos fatores existentes, representando uma parte do universo da classe de problemas, com instâncias de pequeno a grande porte. Selecionamos apenas cinco instâncias com diversos tamanhos, porque os parâmetros foram avaliados com base no delineamento experimental de Taguchi. A quantidade de combinações de configurações (240 por instância), aplicada a cada instância de teste, não é possível testar considerando a imensa quantidade de recursos computacionais necessários.

A escolha do TSP como estudo de caso, deve-se a este ser um dos problemas de otimização combinatória mais intensamente estudados e provou ser NP-difícil, ou seja, nenhum algoritmo pode resolvê-lo em tempo polinomial. O TSP é frequentemente usado como um indicador de desempenho para métodos de otimização e, por isso, também é um dos problemas de otimização mais estudados (MATAI; SINGH; MITTAL, 2010). Sua aplicação vai além da informática teórica, pesquisa operacional e otimização combinatória, sendo aplicada a vários cenários, como planejamento e logística (DAVENDRA, 2010).

O problema pode ser descrito como encontrar um caminho mais curto, considerando um conjunto de cidades e os custos de viagem (ou distância) entre cada par possível, começando de uma cidade e terminando na mesma cidade, visitando cada cidade apenas uma vez. O TSP padrão pode ser matematicamente formulado da seguinte maneira (MAGNANTI; ORLIN, 1993):

$$\text{Minimizar } FO = \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \quad (5.1)$$

Sujeito a:

$$\sum_{i=1}^n x_{ij} = 1 \quad j = 1, 2, \dots, n \quad (5.2)$$

$$\sum_{j=1}^n x_{ij} = 1 \quad i = 1, 2, \dots, n \quad (5.3)$$

$$x_{ij} \in 0, 1 \quad i, j = 1, 2, \dots, n \quad (5.4)$$

$$\tilde{x} \quad \text{forma um ciclo Hamiltoniano} \quad (5.5)$$

Onde c_{ij} e x_{ij} são, respectivamente, o custo e a variável de decisão associada à atribuição do elemento i à posição j . Quando $x_{ij} = 1$, o elemento i é atribuído à posição j . \tilde{x} é a solução para o TSP.

A Equação (5.1) representa a função objetivo que calcula o custo total a ser minimizado. A restrição (5.2) garante que cada posição j seja ocupada por apenas uma cidade, enquanto a restrição (5.3) garante que cada cidade i seja atribuída exatamente a uma posição. A restrição (5.4) representa as restrições de integralidade $(0, 1)$ à variável x_{ij} . A restrição (5.5) garante que cada cidade seja visitada uma vez na rota final e que nenhuma sub-rota seja formada.

A escolha das combinações de níveis dos fatores do projeto fatorial foi fracionária, uma vez que para sua completude seria necessário um alto poder de processamento. Como não há indicações na literatura de níveis para os fatores, optou-se pela definição empírica dos valores. Cabe ressaltar que esta etapa ajudou a moldar a construção dos autômatos das meta-heurísticas com níveis e fatores. Na Tabela 6, são apresentados os fatores e respectivos níveis utilizados nos autômatos e testes.

Tabela 6 – Níveis dos parâmetros e métodos aplicados aos experimentos

Algoritmo Genético		Arrefecimento Simulado	
Taxa de Cruzamento	Taxa de Mutação	Temperatura Inicial	Redução da Temperatura
2%	1%	100	0,001
5%	2%	500	0,010
10%	5%	1000	0,050
25%	10%	2500	0,100
50%	15%	5000	0,500
75%	20%	10000	1,000
Busca Dispersa		Busca em Vizinhança Variável	
Conjunto de Referência	Conjunto de Diversificação	Vizinhanças	Perturbação
2	25	5	1%
5	50	10	2%
10	75	25	5%
15	100	50	10%
20	150	75	15%
25	200	100	20%
Busca Local Iterada		Busca Tabu	
Busca Local	Perturbação	NBMax	Lista Tabu
2-opt	1%	5	10
3-opt	2%	10	25
	5%	25	50
	10%	50	75
	15%	75	100
	20%	100	200
GRASP		Otimização por Colônia de Formigas	
Busca Local	Lista de Candidatos α	Atração do Feromônio	Evaporação do Feromônio
2-opt	0,1	0,5	0,01
3-opt	0,2	1,0	0,02
	0,3	2,0	0,03
	0,5	3,0	0,05
	0,7	4,0	0,10
	0,9	5,0	0,15

Fonte: Elaborado pelo autor

Como a *especificação do procedimento experimental* visa à precisão, deve-se ter em mente uma unidade que reflita a formalização do problema de otimização. Esta observação se faz importante pois o cenário hiper-heurístico lida com uma gama de problemas de diferentes naturezas. Visto isso, identificou-se o processo de avaliação como um denominador comum necessário a todas as técnicas e a todos os problemas independente de suas naturezas. Outro fator de impacto é a qualidade obtida pelas soluções encontradas (BARR et al., 1995).

Deste ponderamento, foi elaborado o conceito de granularidade (tamanho do grão computacional) capaz de medir a eficiência das heurísticas. Esta granularidade perfaz a relação entre a qualidade obtida pelas soluções e o tempo computacional despendido para a obtenção de tal resultado (Veja Equação 5.6).

$$GC = \frac{t}{|FO(S_0) - FO(S^*)|} \quad (5.6)$$

Sejam

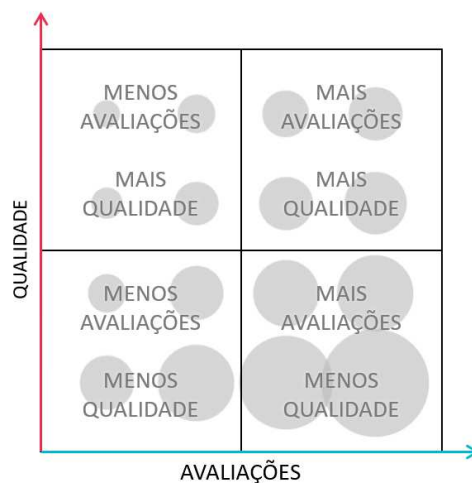
t	número de avaliações da Função Objetivo
S_0	valor da melhor solução inicial
S^*	valor da melhor solução obtida até o momento

A primeira medida de desempenho de um algoritmo estocástico é a qualidade da solução encontrada, pois esse é o objetivo do processo de otimização. Essas medidas incluem o valor da melhor solução encontrada S^* e o valor da primeira solução aceitável (viável) S_0 . Segundo (YOUNES; ELKAMEL; AREIBI, 2010), se o algoritmo for testado em problemas de *benchmark* com soluções ótimas conhecidas, esses valores poderão ser usados como medidas de desempenho. Adicionamos a Equação 5.6, que apresenta o Grão Computacional (CG). Onde t denota o número de chamadas de funções objetivas realizadas no ciclo de operação do modelo. O parâmetro $FO(S_0)$ é o valor da primeira solução aceitável entre as soluções geradas inicialmente. O parâmetro $FO(S^*)$ é o melhor valor de saída do atual ciclo de execução, o valor da melhor solução obtida até o momento.

O tempo de execução, geralmente relatado como tempo da CPU, é um indicador primário da eficiência de um algoritmo. No entanto, quando usado como base para a comparação de algoritmos, o tempo da CPU se torna uma fonte não realista, pois vários fatores o influenciam, incluindo a diferença nas habilidades de codificação dos programadores que desenvolvem os programas e nas máquinas que os executam. Para isso, o número de avaliações da função objetivo é frequentemente usado na literatura como uma alternativa ao tempo da CPU, pois é independente do hardware usado e menos dependente das habilidades de programação. Em geral, quanto mais tempo é aplicado ao algoritmo, melhor pode ser a qualidade da solução final, portanto, nossa medida de desempenho considera a relação entre o tempo de execução indicado pelas avaliações da função objetivo (t) e a diferença de qualidade da solução ($|FO(S_0) \sim FO(S^*)|$).

Para melhor compreendermos a relação do grão computacional, na perspectiva da quantidade de avaliações realizadas e a otimização causada pelos mecanismos heurísticos, apresenta-se na Figura 26 a ilustração destes movimentos, onde as bolhas ao fundo indicam o valor do Grão Computacional. Por questões de maior qualidade de resultados, aliado ao menor esforço computacional, intenciona-se que os resultados percorram os quadrantes em ordem crescente.

Figura 26 – Espaço do Grão Computacional



Fonte: Elaborado pelo autor

Como já mencionado, um dos fatores críticos na construção de meta-heurísticas é o ajuste de seus parâmetros. Por outro lado, em uma hiper-heurística este processo deve ocorrer de maneira automatizada, fazendo parte do processo de busca no metaespaço. Assim, pode-se relacionar a importância do tempo computacional na delimitação dos critérios de parada das heurísticas internas. Esta medida reflete o número de avaliações efetuadas pela função objetivo.

A execução de *experiências piloto* visou a definição de parte do processo final de experimentação, uma vez que forneceu indícios sobre possíveis problemas e soluções esperadas, como o processo de coleta de dados, os limites de tempo de execução, entre outros (RARDIN; UZSOY, 2001).

É interessante mencionar que as primeiras experiências piloto realizadas foram de cunho empírico no sentido da definição do ferramental a ser usado nos testes. Tais experiências se deram ao nível de ferramentas e linguagens, baseados em C++, C# e Python. Dos testes, optou-se pela linguagem Python devido aos seguintes fatores: i) facilidade de implementação das estruturas heurísticas, ii) propicia a construção de um ambiente controlado para reprodutibilidade dos experimentos, iii) é válida e aceita pela literatura, pois se faz presente em diversas publicações encontradas durante a pesquisa, e iv) capaz de fornecer gráficos, testes estatísticos e comparativos entre experimentos. Ainda que testes de *benchmark* entre linguagens não a coloquem no topo do ranking, nossa intenção relaciona-se ao tempo computacional, medido aqui pelo número de vezes em que se avalia a função objetivo do problema (BENCHMARKSGAME, 2017).

A etapa de *especificação do modelo* visou a identificação da relação entre as variáveis de resposta e as fontes de variação. Como exemplo de exposição dos experimentos, foi adotado o padrão da Tabela 7, em que as linhas e colunas representarão os fatores e suas segmentações serão os níveis de valores bloqueados.

Tabela 7 – Modelo da tabela de experimentos

			Fator X					
			Nível X1	Nível X2	Nível X3	Nível X4	Nível X5	Nível X6
Fator Y	Nível Y1	FO	1º Quadrante (Q1)			2º Quadrante (Q2)		
		DP						
		GC						
	Nível Y2	FO						
		DP						
		GC						
	Nível Y3	FO						
		DP						
		GC						
	Nível Y4	FO	3º Quadrante (Q3)			4º Quadrante (Q4)		
		DP						
		GC						
	Nível Y5	FO						
		DP						
		GC						
	Nível Y6	FO						
		DP						
		GC						

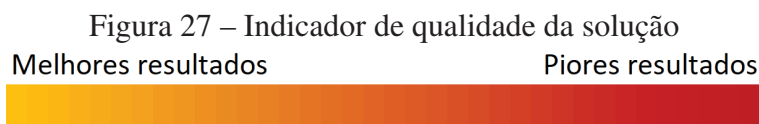
FO - Função Objetivo
 DP - Desvio Padrão
 GC - Grão Computacional

Fonte: Elaborado pelo autor

O processo de análise das tabelas se concentrará em quatro formas de observação. São elas:

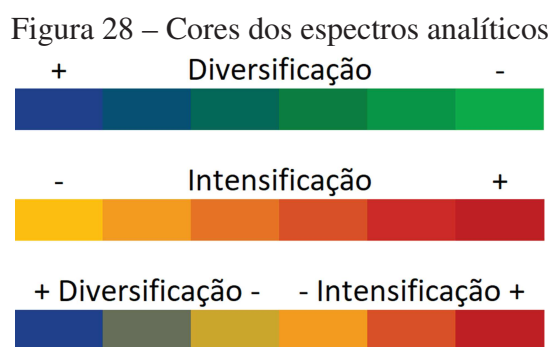
- i) valores numéricos de Função Objetivo (FO), Desvio Padrão (DP) e Grão Computacional (GC);
- ii) apresentação dos 5 melhores resultados em negrito, com destaque ao melhor resultado sublinhado;
- iii) coloração dos resultados conforme qualidade, variando dos melhores resultados em amarelo para os piores em vermelho (Veja Figura 27); e

- iv) observação espectral dos resultados com base em lentes analíticas que demonstram a intensificação e diversificação introduzidas pelos mecanismos analisados.



Fonte: Elaborado pelo autor

As lentes espectrais foram desenvolvidas para a análise dos experimentos de cada uma das meta-heurísticas abordadas no trabalho. Para sua elaboração, consideraram-se os parâmetros, métodos e níveis mencionados nas Tabela 5 e 6, sob a estrutura da Tabela 7. Cada uma das lentes divide-se em três espectros de cores (diversificação, intensificação e geral), conforme Figura 28, na intenção de facilitar a compreensão dos movimentos de diversificação e intensificação realizados pelos mecanismos de busca, indicando por meio de cores quentes uma maior intensificação e pelas cores frias maior diversificação².



Fonte: Elaborado pelo autor

A delimitação do número de observações necessárias faz-se primordial pois se está considerando algoritmos estocásticos, ou seja, produzem diferentes resultados a cada execução. O número de observações deve ser suficientemente grande para que se possa reduzir a variabilidade dos dados. Sendo assim, estabeleceu-se o número mínimo de 100 observações, onde nos apoiamos no Teorema Central do Limite e na Lei dos Grandes Números para definição da normalidade dos processos avaliados (MOOD; GRAYBILL; BOES, 1974).

Como uma alteração percentual é uma maneira de expressar uma alteração em uma variável, representando a mudança relativa entre dois valores, apresentamos a diferença percentual obtida pelas meta-heurísticas em relação às hiper-heurísticas, calculada de acordo com a Equação 5.7.

$$GAP = \frac{MH - HH}{MH} \cdot 100 \quad (5.7)$$

²As oito lentes desenvolvidas no trabalho estão em suas três formas apresentadas no Apêndice B

Por fim, a última etapa visou a *revisão e análise das decisões tomadas* para que se atinja a acurácia desejada, uma vez que uma decisão errada no projeto dos experimentos poderia causar atrasos e resultados imprevisíveis.

5.3 Análise dos Experimentos

5.3.1 Análise dos experimentos para a instância berlin52

5.3.1.1 Análise dos experimentos com Algoritmo Genético para a instância berlin52

Na Tabela 8, vê-se os resultados do Algoritmo Genético, onde os cantos com a presença de tons vermelhos mais quentes indicam que valores extremos - demasiadamente baixos ou elevados - não contribuem para a adequada condução da busca, priorizando assim valores intermediários. Observa-se que taxas de mutação intermediárias apresentam o comportamento desejado de redução do valor da função objetivo, salientando a Taxa de Mutação de 0.1 com a presença de 4 dos melhores resultados para o experimento, com destaque ao quadrante Q4, onde encontramos a solução de melhor qualidade. Desta forma, entende-se que a paisagem de aptidão formada pela atuação dos mecanismos do Algoritmo Genético é melhorada em condições de maior diversificação, ou seja, com valores mais elevados para as taxas dos mecanismos.

Tabela 8 – Resultados dos experimentos do AG para a instância berlin52

			Taxa de Mutação					
			0,01	0,02	0,05	0,10	0,15	0,20
Taxa de Cruzamento	0,02	FO	8775,884	8384,064	8303,056	7831,150	8505,157	8900,456
		DP	112,833	131,750	130,477	134,248	145,803	152,579
		GC	1,942	2,501	2,301	2,130	2,377	1,272
	0,05	FO	8550,146	8117,132	8056,344	7746,525	8184,318	8353,884
		DP	109,930	92,767	92,073	99,598	128,611	95,473
		GC	1,197	1,870	1,342	1,839	1,344	1,456
	0,10	FO	8221,407	7946,486	7986,398	7751,505	8053,423	8294,258
		DP	93,959	113,521	125,501	121,809	126,554	130,338
		GC	1,430	1,395	2,279	2,601	1,270	2,558
	0,25	FO	8146,700	7885,015	7800,800	7664,876	7955,425	8158,578
		DP	139,658	101,379	133,728	98,548	113,649	128,206
		GC	1,548	1,597	1,475	1,868	2,439	1,838
	0,50	FO	8237,159	7975,690	7873,219	7731,651	8109,066	8316,814
		DP	105,906	136,726	123,722	99,407	115,844	142,574
		GC	1,388	1,244	2,645	2,412	1,301	2,101
	0,75	FO	8358,537	8219,784	8096,450	7829,580	8243,486	8566,403
		DP	143,289	93,940	138,796	111,851	94,211	134,615
		GC	1,294	1,567	1,367	1,261	1,586	1,278

Fonte: Elaborado pelo autor

5.3.1.2 Análise dos experimentos com Arrefecimento Simulado para a instância berlin52

Na Tabela 9, observa-se que os melhores resultados se concentram nos quadrantes inferiores, em especial no quadrante Q3. Nota-se que temperaturas mais elevadas e um lento resfriamento fornecem melhores soluções, visto que a região apresenta predominância de tons menos quentes, sugerindo que para instância berlin52 torna-se interessante uma maior diversificação do processo realizada pela interação entre os mecanismos. Observa-se pela predominância de tons vermelhos, que maiores fatores de redução de temperatura apresentam comportamento final de maior instabilidade do material, ou seja, com valores elevados de função objetivo, corroborando assim que a lenta redução da temperatura conduz a materiais com maior metaestabilidade.

Tabela 9 – Resultados dos experimentos do SA para a instância berlin52

			Redução da Temperatura					
			0,001	0,01	0,05	0,1	0,5	1
Temperatura Inicial	100	FO	8259,636	7993,725	8214,448	8509,798	8651,308	9069,878
		DP	106,195	125,616	117,349	109,412	135,949	116,613
		GC	1,558	2,980	2,895	2,777	1,744	1,469
	500	FO	8271,302	7961,418	8099,458	8240,598	8352,588	8620,398
		DP	141,794	113,735	127,277	94,178	119,323	98,519
		GC	1,912	2,846	2,219	2,970	3,015	1,571
	1000	FO	7953,888	7872,828	7971,880	8088,039	8450,336	8371,549
		DP	136,352	112,469	91,107	115,543	132,791	107,634
		GC	1,830	2,390	1,685	2,344	1,651	1,544
	2500	FO	7824,388	7683,703	7752,262	7793,153	7946,093	7937,067
		DP	100,599	120,744	110,747	111,331	113,516	113,387
		GC	2,603	2,130	1,837	3,265	1,636	1,957
	5000	FO	7917,422	7736,940	7800,182	7940,028	8246,474	8151,981
		DP	113,106	110,528	122,574	136,115	141,368	128,103
		GC	2,413	3,095	1,892	1,684	2,452	2,093
	10000	FO	7996,309	7890,213	7947,721	8122,357	8404,356	8293,413
		DP	137,080	101,446	124,893	104,430	144,075	142,173
		GC	1,680	2,317	2,012	1,666	2,533	1,704

Fonte: Elaborado pelo autor

5.3.1.3 Análise dos experimentos com Busca Dispersa para a instância berlin52

Neste experimento, verifica-se, na Tabela 10, que a Busca Dispersa concentrou as soluções de maior qualidade nos quadrantes inferiores, visto por meio da incidência de tons amarelos. Nesta região, em particular, constata-se também a menor média do grão computacional dentre os demais experimentos. Este comportamento se deve à maior intensificação da busca pelos valores elevados de Conjunto de Referência. Contrariamente, verifica-se que conjuntos de diversificação maiores acabam por aumentar demasiadamente a diversificação, tornando a busca mais lenta, acarretando assim em valores maiores de grão computacional. Já na presença de conjuntos de diversificação menores, nota-se que estes não propiciam suficiente heterogeneidade das estruturas internas da instância de teste, visto pelo aumento dos valores médios da FO.

Tabela 10 – Resultados dos experimentos do SS para a instância berlin52

			Conjunto de Diversificação					
			25	50	75	100	150	200
Conjunto de Referência	2	FO	8972,717	8674,861	8184,502	8092,249	8562,409	8978,027
		DP	153,818	136,319	128,614	92,483	122,320	102,606
		GC	1,497	1,199	1,717	1,053	1,246	1,873
	5	FO	8475,593	8382,370	8128,420	7934,962	8431,857	8427,475
		DP	133,188	143,698	139,344	124,692	132,501	120,392
		GC	1,091	1,713	1,991	2,120	1,598	1,467
	10	FO	8200,610	8119,662	7910,400	7827,906	8101,562	8036,860
		DP	93,721	139,194	113,006	123,010	92,589	91,850
		GC	1,221	1,256	1,221	1,723	1,159	2,087
	15	FO	7806,694	7814,872	7781,992	7689,390	7837,056	7841,551
		DP	111,524	100,477	111,171	109,848	123,154	100,820
		GC	1,537	1,225	1,397	1,101	1,351	1,796
	20	FO	8032,048	8080,111	7976,567	7796,187	8074,887	8202,982
		DP	91,795	115,430	136,741	133,649	138,427	140,623
		GC	1,463	1,928	1,340	1,283	1,122	1,467
	25	FO	8595,470	8314,222	8243,340	7993,721	8384,694	8469,891
		DP	135,072	118,775	105,986	91,357	143,738	108,899
		GC	0,978	1,267	1,397	1,532	1,026	1,689

Fonte: Elaborado pelo autor

5.3.1.4 Análise dos experimentos com Busca em Vizinhança Variável para a instância berlin52

Neste experimento com a Busca em Vizinhança Variável, vê-se, na Tabela 11, a tendência de concentração de soluções de melhor qualidade nos quadrantes esquerdos, em destaque o quadrante Q1, onde os operadores introduzem maior intensificação. Apesar deste quadrante não apresentar o menor índice médio de grão computacional, nele encontram-se os menores valores de desvio padrão, o que leva a crer que mais estruturas de vizinhanças, produzem resultados mais dispersos, demonstrado por maiores valores de FO e desvios padrões. Constatase, que maiores valores de aleatoriedade fornecidos pela perturbação das estruturas de vizinhança, produzem resultados de qualidade inferior, bem como o aumento dos desvios padrões. Esta observação se dá também pelo efeito de borda da tabela, apresentando tons vermelhos, onde ocorre um aumento da diversificação, mostrando um aumento médio dos resultados, nos quadrantes Q2, Q3 e Q4.

Tabela 11 – Resultados dos experimentos do VNS para a instância berlin52

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Vizinhanças	5	FO	8228,159	8108,757	7921,684	8107,974	8177,312	8568,895
		DP	141,054	104,255	90,534	104,245	105,137	134,654
		GC	1,676	1,565	1,453	1,461	1,264	1,211
	10	FO	7944,480	7916,178	7873,395	7892,018	8234,150	8287,888
		DP	90,794	113,088	134,972	101,469	141,157	94,719
		GC	1,703	1,349	2,104	1,768	2,120	1,253
	25	FO	7846,670	7813,514	7804,143	7825,489	7991,392	8043,812
		DP	112,095	111,622	89,190	111,793	102,746	126,403
		GC	1,707	1,185	1,284	1,924	1,821	1,195
	50	FO	7970,383	7951,407	7846,894	7943,823	8227,709	8442,216
		DP	136,635	124,951	100,889	136,180	94,031	96,482
		GC	1,256	1,367	1,325	1,286	1,255	1,535
	75	FO	8238,402	8073,143	7896,235	7969,281	8410,808	8522,894
		DP	94,153	103,798	135,364	91,077	120,154	97,405
		GC	1,459	1,706	1,592	1,458	1,248	1,727
	100	FO	8234,715	8246,299	7937,249	8229,959	8442,314	8938,444
		DP	117,639	129,585	90,711	141,085	108,544	153,230
		GC	1,441	1,215	1,302	1,369	1,689	1,467

Fonte: Elaborado pelo autor

5.3.1.5 Análise dos experimentos com Busca Local Iterada para a instância berlin52

O experimento apresentado na Tabela 12, concentra os melhores resultados no quadrante Q1, dado pelo operador de busca local 2-opt quando aliado a baixas taxas de perturbação, indicado pela presença de tons menos quentes em uma região de maior intensificação. Pode-se notar certa uniformidade entre os operadores de busca local quando submetidos a baixas perturbações. Entretanto, quando se elevam as taxas de perturbação, a diversificação é aumentada e assim nota-se que o operador 3-opt é mais afetado, uma vez que efetua mais trocas de posições dentro da estrutura da solução em relação ao operador 2-opt.

Tabela 12 – Resultados dos experimentos do ILS para a instância berlin52

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Busca Local	2-opt	FO	<u>7627,555</u>	7772,395	7934,679	7979,786	8140,459	8287,551
		DP	<u>83,863</u>	88,827	124,688	136,796	116,292	142,072
		GC	<u>1,789</u>	1,193	1,777	1,193	1,584	1,713
	3-opt	FO	7649,887	7812,702	7963,282	8197,326	8428,700	8603,111
		DP	98,356	122,771	91,009	105,394	132,451	98,321
		GC	1,568	1,298	1,316	1,683	1,082	1,369

Fonte: Elaborado pelo autor

5.3.1.6 Análise dos experimentos com Busca Tabu para a instância berlin52

Analisando a Tabela 13, percebe-se que as melhores soluções estão concentradas nos quadrantes do lado direito, especialmente no quadrante Q4. Desta percepção, onde os operadores apresentam maior diversificação, ocasionada por valores elevados de NBmax e de Lista Tabu, aumentou-se o grão computacional, prova disso, é o quadrante onde se obtém a menor média de FO, também apresentando o maior grão computacional médio. Pela presença de tons menos quentes, podemos afirmar que valores elevados de Lista Tabu propiciam melhores valores médios para os experimentos, observado pelo tangenciamento de valores a partir das Listas Tabu de tamanho 75. Esta demonstração de comportamentos atesta que a relação entre os valores de NBmax e os tamanhos de Lista Tabu exercem uma influência mais harmônica um sobre o outro nos experimentos do quadrante Q4, onde apresentam maior diversificação. De maneira oposta, observa-se o aumento dos desvios padrões, em especial no quadrante Q1, onde têm-se maior intensificação ocasionada pelos menores ciclos e memórias menos restritivas.

Tabela 13 – Resultados dos experimentos do BT para a instância berlin52

			Lista Tabu					
			10	25	50	75	100	200
NBmax	10	FO	8701,381	8499,270	8241,945	7940,326	8021,635	8090,246
		DP	136,736	133,560	105,968	124,777	137,514	115,575
		GC	1,597	1,053	1,746	1,031	1,203	1,030
	25	FO	8483,504	8211,014	8062,280	7865,419	7898,111	8141,126
		DP	145,431	140,760	115,175	123,599	135,396	127,932
		GC	1,836	1,131	1,659	1,671	1,380	0,956
	50	FO	8345,445	8094,603	7882,411	7761,016	7791,387	7852,247
		DP	95,377	138,765	135,127	121,959	111,306	112,175
		GC	1,065	1,523	1,385	1,796	1,522	1,705
	75	FO	7963,307	7852,715	7781,261	7672,470	7743,353	7789,444
		DP	125,138	112,182	111,161	109,607	99,557	111,278
		GC	1,442	1,567	0,975	1,919	1,919	1,725
	100	FO	8380,718	8156,650	7948,146	7799,398	7787,451	7970,083
		DP	107,752	93,219	102,190	111,420	133,499	136,630
		GC	0,951	1,051	0,971	1,144	1,166	1,091
	200	FO	8487,301	8172,629	7981,615	7860,478	7904,804	8116,910
		DP	133,372	105,077	125,425	101,063	90,341	127,551
		GC	1,083	0,972	1,486	1,455	1,587	1,333

Fonte: Elaborado pelo autor

5.3.1.7 Análise dos experimentos com GRASP para a instância berlin52

O experimento apresentado na Tabela 14, concentra os melhores resultados nos quadrantes do lado direito, dado pelas maiores listas de candidatos, onde observa-se tons menos quentes, demonstrando uma tendência de diversificação na produção de resultados de maior qualidade. Nota-se pelo degradê da tabela, que a maioria dos experimentos apresenta decréscimo dos valores de função objetivo quando se aumenta o tamanho da lista de candidatos, independentemente do operador de Busca Local, fruto da inserção de variabilidade estrutural nas soluções. Este comportamento de decaimento deve-se à diversificação das soluções oriundas deste mecanismo. Cabe ressaltar que além da diversificação gerada pelo operador, obteve-se maiores valores de grão computacional nos quadrantes do lado direito se comparado com as demais parametrizações.

Tabela 14 – Resultados dos experimentos do GRASP para a instância berlin52

			Lista de Candidatos α					
			0,1	0,2	0,3	0,5	0,7	0,9
Busca Local	2-opt	FO	8065,665	7983,348	7914,490	7762,613	7715,469	<u>7642,829</u>
		DP	126,746	114,048	113,064	110,894	88,177	<u>98,412</u>
		GC	1,671	2,817	1,721	3,395	2,729	<u>1,937</u>
	3-opt	FO	8104,709	8253,528	7967,819	7979,704	7761,150	7659,325
		DP	92,625	106,117	102,443	125,395	121,961	97,705
		GC	2,344	2,127	1,865	1,916	2,733	2,692

Fonte: Elaborado pelo autor

5.3.1.8 Análise dos experimentos com Otimização por Colônia de Formigas para a instância berlin52

Vê-se, na Tabela 15, a incidência da alta atração do feromônio na qualidade das soluções, através da redução dos desvios padrões e grão computacionais, concentrando-as na parte esquerda da tabela, em especial no quadrante Q3, onde a incidência da intensificação dos operadores é aumentada. Pode-se perceber claramente uma tendência de melhores soluções com taxas de evaporação do feromônio levemente menores. Por outro lado, quando muito pequena, acaba por piorar os resultados, aumentando significativamente o valor do grão computacional, pois a quantidade de soluções geradas torna-se maior em comparação à evolução dos valores médios da função objetivo.

Tabela 15 – Resultados dos experimentos do ACO para a instância berlin52

			Evaporação do Feromônio					
			0,01	0,02	0,03	0,05	0,10	0,15
Atração do Feromônio	0,5	FO	8716,832	8170,270	7983,256	8218,829	8658,619	8658,140
		DP	136,979	140,062	102,642	129,153	148,433	98,950
		GC	1,479	2,901	1,879	1,644	1,467	2,366
	1,0	FO	8446,577	8129,785	7824,372	8232,569	8243,551	8278,436
		DP	96,532	116,140	100,599	94,086	105,989	141,916
		GC	2,857	1,959	2,773	2,536	1,560	3,009
	2,0	FO	8268,349	7898,437	7785,250	7914,013	8152,561	8328,662
		DP	118,119	90,268	100,096	124,363	93,172	95,185
		GC	2,461	1,686	1,819	2,251	3,153	2,157
	3,0	FO	8018,764	7763,796	7694,346	7804,767	7945,011	8057,088
		DP	103,098	99,820	87,935	133,796	90,800	126,611
		GC	1,649	1,743	1,649	1,621	1,723	2,526
	4,0	FO	8388,426	7885,016	7764,542	8062,223	8239,900	8211,898
		DP	95,868	123,907	88,738	92,140	94,170	129,044
		GC	1,661	2,808	2,768	1,700	2,245	1,790
	5,0	FO	9023,029	8162,443	7871,461	8169,247	8502,568	8661,099
		DP	128,900	139,928	89,960	116,704	97,172	111,357
		GC	2,064	1,720	1,779	2,083	2,274	1,977

Fonte: Elaborado pelo autor

5.3.1.9 Análise dos experimentos consolidados para a instância berlin52

A seguir, apresentam-se as consolidações das médias e dos melhores resultados dos experimentos realizados para a instância berlin52. Os experimentos apresentados na Tabela 16 dividem-se em duas seções. Na primeira seção, apresenta-se a consolidação da melhor média de 100 repetições dos experimentos, dentre os 36 conjuntos de parâmetros testados para cada meta-heurística, bem como a melhor execução média da hiper-heurística. Para cada melhor valor obtido pelas meta-heurísticas e hiper-heurística, faz-se o cálculo do Gap percentual em relação a hiper-heurística e ao BKS (*Best Known Solution*³).

Na segunda seção, observa-se a consolidação da média de 100 repetições do experimento nos 36 conjuntos de parâmetros testados para cada meta-heurística, além da melhor execução média da hiper-heurística. Em cada valor médio obtido pela meta-heurística, calcula-se também o Gap percentual em relação à hiper-heurística e à melhor solução conhecida.

Para a instância berlin52, a meta-heurística Busca Local Iterada apresentou o melhor resultado dentre às 288 parametrizações testadas, distribuídas em 36 conjuntos de parâmetros em cada uma das 8 meta-heurísticas. Já o melhor resultado médio, fruto dos 36 parâmetros testados, foi obtido pela GRASP, dentre as 8 meta-heurísticas testadas.

O resultado obtido pela hiper-heurística sobre a Busca Local Iterada apresenta uma melhora de 0.101% em relação a FO, 15% melhor em comparação ao DP e 42,774% no que concerne ao GC. Por outro lado, o resultado obtido pela hiper-heurística em contraste com a GRASP, fornece resultados 3,557% melhores, sendo o DP 34,078% melhor e o GC 56,034% menor.

Já quando efetuamos uma análise sobre a média dos melhores resultados, obtém-se com a hiper-heurística uma melhora de 0,842% nos resultados da FO, uma redução de 27,535% do desvio padrão e 37,350% sobre o GC. Numa perspectiva mais geral, a hiper-heurística obteve sobre a média das médias das meta-heurísticas, uma melhoria de 5,762% sobre a FO, um desvio padrão 37,974% menor e o GC mostrou-se 39,727% melhor.

Ao compararmos a hiper-heurística em relação ao BKS, obteve-se em média resultados 1,032% superiores, contudo, destaca-se o valor de GC obtido pela hiper-heurística é o melhor dentre todos os testes efetuados com as meta-heurísticas.

³Tradução livre: Melhor Solução Conhecida

Tabela 16 – Resultados dos experimentos consolidados para a instância berlin52

BKS	7542	Melhor	GAP HH	GAP BKS	Média	GAP HH	GAP BKS
AG	FO	7.664,876	0,587%	1,629%	8.142,523	6,419%	7,962%
	DP	98,548	27,667%	-	119,274	40,236%	-
	GC	1,868	45,177%	-	1,759	41,780%	-
SA	FO	7.683,703	0,831%	1,879%	8.120,476	6,165%	7,670%
	DP	120,744	40,963%	-	118,835	40,015%	-
	GC	2,130	51,932%	-	2,177	52,965%	-
SS	FO	7.689,390	0,904%	1,954%	8.177,170	6,815%	8,422%
	DP	109,848	35,107%	-	119,467	40,332%	-
	GC	1,101	7,000%	-	1,448	29,292%	-
VNS	FO	7.804,143	2,361%	3,476%	8.114,113	6,091%	7,586%
	DP	89,190	20,077%	-	113,384	37,131%	-
	GC	1,284	20,225%	-	1,493	31,406%	-
ILS	FO	7.627,555	0,101%	1,134%	8.033,120	5,145%	6,512%
	DP	83,863	15,000%	-	111,737	36,204%	-
	GC	1,789	42,774%	-	1,464	30,049%	-
BT	FO	7.672,470	0,686%	1,730%	8.034,781	5,164%	6,534%
	DP	109,607	34,964%	-	119,266	40,232%	-
	GC	1,919	46,644%	-	1,365	24,964%	-
GRASP	FO	7.642,829	0,301%	1,337%	7.900,887	3,557%	4,759%
	DP	98,412	27,566%	-	108,132	34,078%	-
	GC	1,937	47,146%	-	2,329	56,034%	-
ACO	FO	7.694,346	0,968%	2,020%	8.170,393	6,738%	8,332%
	DP	87,935	18,937%	-	110,629	35,565%	-
	GC	1,649	37,901%	-	2,104	51,330%	-
MÉDIA	FO	7.684,914	0,842%	1,895%	8.086,683	5,762%	7,222%
	DP	99,768	27,535%	-	115,091	37,974%	-
	GC	1,710	37,350%	-	1,767	39,727%	-
HH	FO	7.619,855	-	1,032%	7.619,855	-	1,032%
	DP	71,283	-	-	71,283	-	-
	GC	1,024	-	-	1,024	-	-

Fonte: Elaborado pelo autor

5.3.2 Análise dos experimentos para a instância eil101

5.3.2.1 Análise dos experimentos com Algoritmo Genético para a instância eil101

Observa-se na Tabela 17, a concentração de soluções de maior qualidade nos quadrantes inferiores, sugerindo que a elevação da diversificação por meio do aumento das taxas de cruzamento propicia melhores soluções. Em contrapartida, ocorre uma compensação por meio da baixa diversificação ocasionada pelos pequenos valores de taxas de mutação, cuja diversificação introduzida em maiores proporções, nos quadrantes do lado direito, incidem negativamente, como se observa pela presença de tons mais quentes.

Tabela 17 – Resultados dos experimentos do AG para a instância eil101

			Taxa de Mutação					
			0,01	0,02	0,05	0,10	0,15	0,20
Taxa de Cruzamento	0,02	FO	744,544	696,997	663,651	728,967	736,096	790,197
		DP	8,509	8,961	11,377	8,331	11,567	11,289
		GC	1,305	1,334	1,710	1,718	1,282	1,513
	0,05	FO	697,880	678,915	663,690	700,179	716,226	719,339
		DP	8,973	8,729	10,429	8,002	11,255	9,249
		GC	1,085	2,195	1,076	1,288	1,258	1,758
	0,10	FO	675,931	671,535	650,958	669,757	698,557	724,858
		DP	10,622	11,512	11,159	7,654	11,975	12,426
		GC	1,292	1,886	1,148	1,275	1,977	1,869
	0,25	FO	666,701	659,944	644,976	660,916	686,022	685,843
		DP	8,572	9,428	10,135	7,553	11,760	7,838
		GC	1,269	1,588	2,130	1,753	1,943	1,091
	0,50	FO	657,149	646,155	641,173	656,878	657,086	673,163
		DP	7,510	7,385	8,244	10,322	8,448	8,655
		GC	1,378	1,123	1,251	1,424	1,177	1,249
	0,75	FO	678,525	674,568	657,015	684,354	687,996	708,728
		DP	10,663	7,709	11,263	8,799	10,811	8,100
		GC	1,219	1,269	1,273	1,219	1,817	1,255

Fonte: Elaborado pelo autor

5.3.2.2 Análise dos experimentos com Arrefecimento Simulado para a instância eil101

Observa-se na Tabela 18 a concentração de soluções de qualidade no quadrante Q3, uma região de análise onde predomina a diversificação, dada por valores elevados de temperatura combinados com o lento resfriamento. É notória a depreciação dos resultados nos quadrantes do lado direito, indicados pela predominância de tons vermelhos, onde se reduz a temperatura de maneira acelerada. Percebe-se que os diversos valores de temperatura inicial acabam apresentando pequenas variações médias nos valores obtidos para o experimento, dando a entender que grande parte da contribuição para a qualidade dos resultados obtidos se dá pela maior diversificação propiciada pelo mecanismo de redução de temperatura.

Tabela 18 – Resultados dos experimentos do SA para a instância eil101

			Redução da Temperatura					
			0,001	0,01	0,05	0,1	0,5	1
Temperatura Inicial	100	FO	681,840	655,988	669,922	696,083	719,805	734,452
		DP	7,792	11,246	7,656	9,944	8,226	8,394
		GC	1,927	2,307	2,607	3,235	2,349	3,206
	500	FO	671,034	652,329	661,659	688,260	717,389	702,071
		DP	7,669	7,455	11,343	10,816	10,248	11,033
		GC	3,296	2,345	1,905	2,444	2,762	3,259
	1000	FO	666,317	642,684	648,614	679,345	695,080	695,751
		DP	10,471	10,099	8,339	8,734	10,923	8,945
		GC	2,889	1,712	2,075	1,903	1,869	1,717
	2500	FO	654,762	641,179	647,637	670,993	678,906	680,813
		DP	11,224	7,305	11,102	11,503	9,699	9,726
		GC	1,880	3,154	1,883	1,889	2,465	1,808
	5000	FO	648,041	639,868	641,653	648,221	657,752	663,063
		DP	9,258	9,992	11,000	9,260	10,336	8,525
		GC	2,174	1,932	3,302	1,940	2,014	2,177
	10000	FO	661,721	645,130	651,681	678,578	684,851	691,788
		DP	9,453	11,059	9,310	10,663	10,762	11,859
		GC	3,069	3,429	2,086	2,606	2,742	2,979

Fonte: Elaborado pelo autor

5.3.2.3 Análise dos experimentos com Busca Dispersa para a instância eil101

Através dos resultados de maior qualidade presentes no quadrante Q4 da Tabela 19, vê-se que os Conjuntos de Referência e Diversificação maiores elevam a qualidade dos resultados, entretanto não fornecem os melhores valores de Grão Computacional, que são aprimorados na presença de maior intensificação como a vista no quadrante Q3. Por outro lado, pode-se entender que os diferentes tamanhos de Conjunto de Referência apresentam certa constância que é alterada pontualmente pelas interações com os Conjuntos de Diversificação de tamanho médio alto que trazem maior diversificação para o processo de otimização.

Tabela 19 – Resultados dos experimentos do SS para a instância eil101

			Conjunto de Diversificação					
			25	50	75	100	150	200
Conjunto de Referência	2	FO	711,339	712,400	685,012	668,733	692,216	701,354
		DP	10,162	8,142	7,829	11,464	9,889	8,015
		GC	2,039	1,645	1,776	2,518	2,141	1,563
	5	FO	712,368	685,166	677,627	668,770	686,129	684,357
		DP	10,177	9,788	11,616	10,509	7,841	10,754
		GC	1,243	1,277	1,994	1,747	1,287	1,486
	10	FO	679,963	675,756	657,347	650,008	664,316	661,531
		DP	7,771	8,688	8,452	10,214	8,541	7,560
		GC	1,776	2,343	2,642	2,137	2,153	1,591
	15	FO	651,941	649,662	642,939	642,207	645,891	648,035
		DP	11,176	9,281	11,022	11,009	9,227	8,332
		GC	1,726	1,606	1,577	2,168	1,989	2,481
	20	FO	676,607	670,248	659,604	655,826	659,499	668,556
		DP	10,632	9,575	8,481	7,495	7,537	10,506
		GC	1,689	2,059	1,514	1,504	2,660	1,351
	25	FO	704,068	677,863	669,204	663,938	670,700	692,167
		DP	10,058	10,652	7,648	11,382	8,623	9,888
		GC	1,654	1,578	1,540	1,389	2,657	1,692

Fonte: Elaborado pelo autor

5.3.2.4 Análise dos experimentos com Busca em Vizinhaça Variável para a instância eil101

Observa-se na tabela 20, a concentração de soluções de qualidade e de desvios padrões no quadrante (Q1), onde se apresentam menores vizinhanças e taxas de perturbação, originando maior intensificação gerada pelos valores dos parâmetros deste quadrante. Em contrapartida, os quadrantes do lado direito, com cores mais quentes, fornecem valores reduzidos de grão computacional, indicando o menor esforço computacional em prol da qualidade obtida nesta região.

Tabela 20 – Resultados dos experimentos do VNS para a instância eil101

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Vizinhanças	5	FO	680,360	672,176	656,280	668,364	694,933	695,672
		DP	9,719	8,642	8,438	7,638	9,928	11,926
		GC	2,732	2,753	3,774	3,182	2,308	2,169
	10	FO	670,733	664,981	654,490	663,329	681,781	684,427
		DP	10,540	8,550	8,415	11,371	10,714	10,755
		GC	2,308	3,720	4,019	2,275	2,258	2,101
	25	FO	659,907	651,061	646,460	656,619	671,678	681,323
		DP	11,313	9,301	8,312	10,318	7,676	7,787
		GC	3,195	2,040	4,330	2,423	3,065	3,607
	50	FO	673,454	665,990	650,752	662,816	677,506	685,166
		DP	10,583	11,417	10,226	8,522	11,614	9,788
		GC	2,896	3,917	3,049	4,104	4,220	2,185
	75	FO	677,974	671,743	664,170	670,942	686,194	715,144
		DP	11,622	9,596	7,591	11,502	10,783	12,260
		GC	2,292	2,592	2,388	3,642	3,212	2,053
	100	FO	679,085	678,209	663,406	675,709	708,595	710,323
		DP	8,731	10,658	8,530	9,653	9,111	8,118
		GC	2,517	3,318	4,180	2,186	2,224	2,631

Fonte: Elaborado pelo autor

5.3.2.5 Análise dos experimentos com Busca Local Iterada para a instância eil101

Os experimentos apresentados na tabela 21, concentram os melhores resultados nos quadrantes do lado esquerdo, onde tem-se maior presença de intensificação, da mesma forma que apresentam-se aos menores valores de desvio padrão. O operador de busca local 3-opt, apresenta no quadrante Q3, além do melhor resultado médio para os experimentos, um valor de grão computacional superior, sugerindo que a falta de diversidade é um fator implicante neste aumento. Verifica-se que o aumento da diversificação propiciado pelo aumento dos valores do mecanismo de perturbação introduz maiores níveis de aleatoriedade em ambos os mecanismos de busca local, fruto disso vê-se o aumento médio dos desvios padrões e FO nos quadrantes do lado direito, destacados pela presença de tons mais quentes.

Tabela 21 – Resultados dos experimentos do ILS para a instância eil101

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Busca Local	2-opt	FO	636,998	653,575	674,503	676,184	681,491	689,948
		DP	7,280	7,469	8,672	10,626	9,736	9,856
		GC	1,600	1,216	1,898	1,576	1,539	1,272
	3-opt	FO	636,797	648,531	655,563	671,367	686,590	691,822
		DP	6,991	8,338	10,302	9,591	11,770	9,883
		GC	2,232	1,251	1,578	1,151	1,234	1,283

Fonte: Elaborado pelo autor

5.3.2.6 Análise dos experimentos com Busca Tabu para a instância eil101

Observa-se na tabela 22, a concentração de melhores soluções no quadrante Q4, pela presença de tons menos quentes, provenientes da alta diversificação introduzida pelos dois mecanismos da Busca Tabu, fruto do prolongamento do tempo de busca dado pelo alto valor de NBMax e da memória de longo prazo. Por outro lado, também no quadrante Q4, encontram-se a menor variabilidade dos desvios-padrões médios e os melhores grãos computacionais. Pode-se notar que a contribuição do mecanismo de memória de maior duração, visto pelos valores de lista tabu elevados, aumenta a diversificação da busca, estendendo o tempo de ciclagem, o que contribui para a obtenção de melhores resultados.

Tabela 22 – Resultados dos experimentos do BT para a instância eil101

			Lista Tabu					
			10	25	50	75	100	200
NBMax	10	FO	784,609	778,095	705,484	694,864	678,820	710,599
		DP	10,088	12,227	12,094	10,919	8,728	11,167
		GC	2,055	1,626	1,288	1,170	1,169	1,881
	25	FO	797,221	743,385	707,531	680,615	679,289	705,425
		DP	13,667	8,496	11,118	9,723	11,645	8,062
		GC	1,326	1,133	1,508	2,109	1,580	1,413
	50	FO	753,658	712,813	688,293	673,444	661,309	670,269
		DP	9,690	9,165	7,866	10,583	7,558	11,490
		GC	1,278	1,315	2,128	1,646	2,181	2,013
	75	FO	693,998	666,993	657,584	648,350	644,928	652,182
		DP	11,897	7,623	10,333	11,115	9,213	7,454
		GC	1,955	2,168	2,131	1,325	1,184	1,397
	100	FO	666,499	651,911	649,959	645,755	643,113	644,215
		DP	10,474	10,244	9,285	7,380	10,106	10,123
		GC	1,773	1,905	1,414	1,909	1,345	1,725
	200	FO	647,729	645,719	637,055	637,381	636,097	636,613
		DP	9,253	9,225	7,281	8,105	6,764	6,264
		GC	1,863	1,680	1,282	1,662	1,512	1,450

Fonte: Elaborado pelo autor

5.3.2.7 Análise dos experimentos com GRASP para a instância eil101

Na Tabela 23, são apresentados os resultados de maior qualidade nos quadrantes inferiores, especialmente reduzidos quando uma maior diversidade é introduzida, vista pelo aumento da Lista de Candidatos, fenômeno este especialmente observado pelos tons amarelos do lado direito da tabela. Cabe também notar que o operador 3-opt apresenta maior estabilidade na evolução das soluções de qualidade se comparado ao operador 2-opt que tem uma instabilidade pontual na lista de candidatos com valor 0.3.

Tabela 23 – Resultados dos experimentos do GRASP para a instância eil101

			Lista de Candidatos α					
			0,1	0,2	0,3	0,5	0,7	0,9
Busca Local	2-opt	FO	720,670	699,429	708,108	674,804	656,234	637,577
		DP	9,266	8,993	10,116	8,676	8,437	8,197
		GC	2,226	1,369	1,939	1,356	2,709	2,198
	3-opt	FO	667,564	658,811	652,802	648,711	642,256	636,113
		DP	10,490	11,294	10,258	10,194	10,093	6,846
		GC	1,610	2,184	2,153	1,824	1,590	2,315

Fonte: Elaborado pelo autor

5.3.2.8 Análise dos experimentos com Otimização por Colônia de Formigas para a instância eil101

Na tabela 24, observa-se no quadrante Q3, os melhores resultados para os parâmetros testados, onde encontram-se menores valores médios de função objetivo, desvios-padrões e grão computacional. Ao mesmo tempo, pode-se notar que os valores médios de Evaporação do Feromônio, independentes da atratividade conduzem todos os experimentos a melhores resultados. De outra maneira, se pode concluir que um processo de busca mais intenso por parte dos dois mecanismos perfaz melhores resultados.

Tabela 24 – Resultados dos experimentos do ACO para a instância eil101

			Evaporação do Feromônio					
			0,01	0,02	0,03	0,05	0,10	0,15
Atração do Feromônio	0,5	FO	775,920	690,335	653,459	668,897	683,380	757,280
		DP	13,301	9,862	7,468	9,556	10,739	12,982
		GC	1,564	3,077	3,243	3,114	2,997	2,257
	1,0	FO	753,242	697,899	651,475	671,733	683,895	748,769
		DP	11,837	7,976	8,376	10,556	11,724	8,557
		GC	2,083	2,840	1,638	2,812	1,729	1,787
	2,0	FO	711,895	682,831	650,673	660,170	661,784	715,822
		DP	9,153	9,755	9,295	7,545	7,563	11,249
		GC	2,280	1,920	2,764	2,371	1,702	2,068
	3,0	FO	691,165	657,241	642,406	644,863	654,512	676,351
		DP	11,849	8,450	7,342	7,370	10,285	8,696
		GC	2,184	2,334	2,432	2,020	2,786	1,996
	4,0	FO	668,649	652,229	637,412	641,076	647,322	670,480
		DP	8,597	10,249	7,262	8,242	11,097	7,663
		GC	2,317	1,589	2,820	1,903	2,519	1,771
	5,0	FO	719,337	671,084	645,730	651,337	660,003	689,861
		DP	12,331	7,670	11,070	9,305	10,371	7,884
		GC	1,744	2,170	2,290	2,225	2,662	2,351

Fonte: Elaborado pelo autor

5.3.2.9 Análise dos experimentos consolidados para a instância eil101

Os experimentos apresentados na Tabela 25, consolidam as médias e os melhores resultados dos experimentos realizados para a instância eil101. Os dados estão segmentados em duas seções. Primeiro é apresentada a consolidação da melhor média de 100 repetições dos experimentos, dentre os 36 conjuntos de parâmetros testados para cada meta-heurística, além da melhor execução média da hiper-heurística. Em cada melhor valor obtido pelas meta-heurísticas e hiper-heurística, calcula-se o Gap percentual em relação ao valor obtido pela hiper-heurística e ao valor da melhor solução conhecida.

Já na segunda seção, têm-se a compilação média de 36 conjuntos de parâmetros testados para cada meta-heurística, que foram repetidos 100 vezes cada, bem como a melhor execução média da hiper-heurística. Para cada valor médio fornecido pela meta-heurística, tem-se o Gap percentual relativo à hiper-heurística e ao BKS.

Nos testes, a meta-heurística Busca Tabu fornece o melhor valor dentre as parametrizações testadas, nas 8 meta-heurísticas em seus 36 conjuntos de parâmetros. Por outro lado, a GRASP obteve o melhor resultado médio, nos 36 parâmetros testados em cada uma das 8 meta-heurísticas.

A hiper-heurística apresenta um resultado 0,1% melhor em relação a Busca Tabu, já sobre o DP é 15% melhor e 29,665% no que diz respeito ao GC. Já o resultado obtido pela hiper-heurística sobre a meta-heurística GRASP, os resultados da FO são 4,718% melhores, e o DP 38,865% menor, e por fim o GC 45,643% melhor.

Na análise sobre a média dos melhores resultados, a hiper-heurística apresenta uma melhora de FO de 0,632%, um desvio padrão 27,612% menor e o GC 48,093% inferior. Ao observarmos a hiper-heurística em relação à média das médias das meta-heurísticas, ela obteve valores 5,805% menores sobre a FO, um desvio padrão 39,719% menor e o GC 43,912% inferior.

No exame da relação entre a hiper-heurística e o BKS, os resultados médios são 1,027% superiores, entretanto, cabe notar que o valor de GC obtido pela hiper-heurística é o melhor dentre os testes efetuados com as meta-heurísticas.

Tabela 25 – Resultados dos experimentos consolidados para a instância eil101

BKS	629	Melhor	GAP HH	GAP BKS	Média	GAP HH	GAP BKS
AG	FO	641,173	0,891%	1,935%	684,874	7,215%	8,883%
	DP	8,244	30,253%	-	9,589	40,040%	-
	GC	1,251	15,000%	-	1,455	26,944%	-
SA	FO	639,868	0,689%	1,728%	671,257	5,333%	6,718%
	DP	9,992	42,458%	-	9,760	41,091%	-
	GC	1,932	44,965%	-	2,426	56,172%	-
SS	FO	642,207	1,051%	2,100%	672,871	5,560%	6,975%
	DP	11,009	47,774%	-	9,443	39,109%	-
	GC	2,168	50,946%	-	1,839	42,174%	-
VNS	FO	646,460	1,702%	2,776%	674,215	5,748%	7,188%
	DP	8,312	30,823%	-	9,768	41,137%	-
	GC	4,330	75,444%	-	2,941	63,843%	-
ILS	FO	636,797	0,210%	1,240%	666,947	4,721%	6,033%
	DP	6,991	17,756%	-	9,210	37,568%	-
	GC	2,232	52,357%	-	1,486	28,435%	-
BT	FO	636,097	0,100%	1,128%	681,439	6,748%	8,337%
	DP	6,764	15,000%	-	9,623	40,250%	-
	GC	1,512	29,665%	-	1,625	34,564%	-
GRASP	FO	636,113	0,103%	1,131%	666,923	4,718%	6,029%
	DP	6,846	16,011%	-	9,405	38,865%	-
	GC	2,315	54,066%	-	1,956	45,643%	-
ACO	FO	637,412	0,306%	1,337%	678,903	6,399%	7,934%
	DP	7,262	20,823%	-	9,534	39,693%	-
	GC	2,820	62,301%	-	2,288	53,523%	-
MÉDIA	FO	639,516	0,632%	1,672%	674,679	5,805%	7,262%
	DP	8,177	27,612%	-	9,541	39,719%	-
	GC	2,320	48,093%	-	2,002	43,912%	-
HH	FO	635,459	-	1,027%	635,459	-	1,027%
	DP	5,750	-	-	5,750	-	-
	GC	1,063	-	-	1,063	-	-

Fonte: Elaborado pelo autor

5.3.3 Análise dos experimentos para a instância d493

5.3.3.1 Análise dos experimentos com Algoritmo Genético para a instância d493

Concentrados no quadrante Q3 da Tabela 26, os resultados sugerem que uma atuação balanceada entre a intensificação dada pela Taxa de Mutação e a diversificação fornecida pela Taxa de Cruzamento, contribuem positivamente para a determinação de soluções de qualidade. Sob a perspectiva da taxa de mutação, os melhores resultados são encontrados na presença de menor diversificação, onde se equalizam positivamente com a maior diversificação introduzida pelo mecanismo de cruzamento, como se observa no quadrante Q3. A partir dos tons mais quentes nos quadrantes do lado direito, infere-se que o comportamento de elevação da diversificação presente nesta região pelo mecanismo de mutação, fornece menores valores de grão computacional como frutos do encurtamento da busca. Este encurtamento deve-se o uso demasiado de aleatoriedade que limita a convergência da busca ativando o critério de parada pela não obtenção de resultados de qualidade.

Tabela 26 – Resultados dos experimentos do AG para a instância d493

			Taxa de Mutação					
			0,01	0,02	0,05	0,10	0,15	0,20
Taxa de Cruzamento	0,02	FO	93638,213	72692,468	53184,739	84947,707	90131,110	92667,307
		DP	1337,689	1246,157	911,738	970,831	1287,587	1059,055
		GC	1,393	2,376	2,810	1,428	1,127	1,051
	0,05	FO	73771,427	60306,208	47084,284	62879,050	80003,996	81976,875
		DP	843,102	775,366	538,106	718,618	1257,206	936,879
		GC	1,925	2,429	2,032	2,668	2,012	2,043
	0,10	FO	57491,228	52048,166	47522,164	54323,927	64567,066	77210,148
		DP	903,434	892,254	610,999	931,267	737,909	1213,302
		GC	1,420	1,921	2,515	2,160	1,919	1,143
	0,25	FO	55532,614	52969,315	43084,497	50977,389	63347,574	60881,905
		DP	713,991	605,364	615,493	873,898	904,965	869,742
		GC	1,481	2,066	2,669	2,384	1,630	1,693
	0,50	FO	62091,980	54526,256	46596,820	56246,058	72608,573	80385,200
		DP	709,623	934,736	532,535	803,515	1140,992	1033,524
		GC	1,917	2,955	2,765	3,132	1,245	1,155
	0,75	FO	66226,057	58506,725	47121,702	64022,905	69507,716	93106,980
		DP	756,869	1002,972	740,484	1006,074	992,967	1330,100
		GC	2,240	2,123	2,412	1,589	2,141	1,681

Fonte: Elaborado pelo autor

5.3.3.2 Análise dos experimentos com Arrefecimento Simulado para a instância d493

Observa-se na Tabela 27, a concentração de soluções no quadrante inferior esquerdo (Q3), propondo que quanto maior a temperatura melhor são os resultados obtidos em conjunto com o lento resfriamento. Percebe-se que os experimentos apresentam aumento médio dos valores de função objetivo quando o resfriamento é acelerado, dando a entender que grande parte da contribuição para a qualidade dos resultados se dá pela menor intensificação ocasionada pelo mecanismo de redução de temperatura, principalmente em temperaturas mais elevadas.

Tabela 27 – Resultados dos experimentos do SA para a instância d493

			Redução da Temperatura					
			0,001	0,01	0,05	0,1	0,5	1
Temperatura Inicial	100	FO	78373,577	59554,960	53110,032	78003,849	78987,875	82663,107
		DP	895,698	935,864	758,715	891,473	902,719	944,721
		GC	1,395	2,603	1,558	1,139	2,379	1,123
	500	FO	74092,041	56814,668	49210,671	68533,630	70327,644	80368,768
		DP	1164,303	649,310	562,408	881,147	1004,681	918,500
		GC	1,302	1,342	2,349	2,398	2,228	1,580
	1000	FO	66969,594	55455,488	46301,493	54294,333	60642,995	67827,380
		DP	956,708	950,666	661,450	775,633	693,063	1065,859
		GC	2,211	1,911	1,887	1,528	1,794	1,692
	2500	FO	57023,213	51864,643	42583,970	53663,781	56694,794	67375,348
		DP	977,541	889,108	608,342	613,300	647,940	962,505
		GC	1,462	2,410	1,773	1,628	1,460	1,965
	5000	FO	47278,082	43272,016	41182,290	48791,372	52732,026	56244,099
		DP	810,481	618,172	647,150	557,616	677,983	723,138
		GC	2,235	1,592	1,571	1,453	1,387	2,697
	10000	FO	65589,385	57205,828	45118,283	60877,902	70522,854	65021,489
		DP	1030,690	817,226	644,547	869,684	805,975	1021,766
		GC	1,761	1,389	2,939	1,510	1,427	1,763

Fonte: Elaborado pelo autor

5.3.3.3 Análise dos experimentos com Busca Dispersa para a instância d493

Vê-se na Tabela 28 que os conjuntos de diversificação e de referência maiores, representados no quadrante Q4, onde se realiza uma busca diversa, concebem melhores resultados médios de FO e desvio padrão. Outra perspectiva analítica demonstra que uma busca intensa, como a presente no quadrante Q1, obtém uma redução dos valores de grão computacional. Pode-se assim entender que os diferentes tamanhos de conjunto de diversificação são mais sensibilizados quando submetidos a tamanhos de conjuntos de referência extremos, ou seja, valores intermediários apresentam maior constância e qualidade nos resultados dos experimentos.

Tabela 28 – Resultados dos experimentos do SS para a instância d493

			Conjunto de Diversificação					
			25	50	75	100	150	200
Conjunto de Referência	2	FO	83349,527	71214,614	65644,105	56291,155	66156,120	70977,222
		DP	1190,708	813,881	1031,550	804,159	850,579	912,564
		GC	0,914	0,997	1,142	1,111	1,718	1,019
	5	FO	71886,268	64310,563	58043,622	51281,499	60807,715	73318,654
		DP	821,557	918,722	995,034	732,593	781,813	1152,150
		GC	1,047	1,085	2,161	2,134	1,283	1,168
	10	FO	62380,850	52183,836	53574,458	48152,175	53731,618	54218,435
		DP	802,039	670,935	841,884	550,311	921,113	697,094
		GC	1,105	1,161	1,352	1,705	1,490	2,253
	15	FO	50282,344	52144,663	46009,900	44239,450	49877,746	51496,921
		DP	861,983	819,416	657,284	631,992	570,031	882,804
		GC	1,551	1,297	1,617	2,294	1,186	1,403
	20	FO	47595,076	45616,514	43863,899	41717,899	44906,570	46340,204
		DP	747,923	521,332	751,953	476,776	641,522	728,203
		GC	1,154	1,522	1,417	1,305	2,209	2,204
	25	FO	58340,085	56097,636	51157,890	45408,623	55618,644	61268,776
		DP	1000,116	801,395	803,910	583,825	794,552	787,741
		GC	1,985	2,218	2,023	1,544	2,155	1,130

Fonte: Elaborado pelo autor

5.3.3.4 Análise dos experimentos com Busca em Vizinhaça Variável para a instância d493

Os experimentos demonstrados na Tabela 29, aglutinam no quadrante Q3 os melhores resultados, consequências de uma busca equilibrada, efetuada entre os dois mecanismos, onde menores índices de perturbação intensificam a busca, contrabalanceando a elevada diversificação trazida por vizinhanças maiores. Sob a ótica do grão computacional, o quadrante Q3 obteve os piores valores médios, dando a entender que o equilíbrio entre os mecanismos de busca consome mais recursos computacionais.

Tabela 29 – Resultados dos experimentos do VNS para a instância d493

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Vizinhanças	5	FO	64198,570	57681,062	52092,223	51674,710	65524,330	70338,500
		DP	1008,835	741,614	744,175	590,568	1123,274	1105,319
		GC	1,783	1,282	1,368	1,493	1,270	1,007
	10	FO	57289,954	50002,802	45414,352	52434,619	63010,331	68630,582
		DP	818,428	714,326	778,532	898,879	900,148	882,393
		GC	1,242	1,324	1,589	1,519	1,300	1,088
	25	FO	49689,624	46296,905	42893,232	46895,236	55985,093	62425,809
		DP	851,822	595,246	674,037	736,925	799,787	802,618
		GC	1,616	2,206	2,560	1,572	1,113	1,196
	50	FO	45854,634	41682,042	41268,266	42839,552	45110,157	53858,061
		DP	720,573	714,549	471,637	550,794	515,545	846,341
		GC	2,393	1,528	2,288	1,831	1,526	1,444
	75	FO	50226,399	45443,158	43277,258	47295,032	55298,312	60132,438
		DP	574,016	519,350	618,247	743,208	710,978	773,131
		GC	1,704	1,638	1,729	1,286	1,586	1,103
	100	FO	52796,152	52144,808	45746,970	50564,042	57027,810	65152,960
		DP	603,385	744,926	784,234	794,578	733,215	744,605
		GC	1,289	1,765	1,360	2,377	1,803	1,370

Fonte: Elaborado pelo autor

5.3.3.5 Análise dos experimentos com Busca Local Iterada para a instância d493

Pode-se observar na Tabela 30, os resultados de melhor qualidade dispostos no quadrante Q3, gerados pelo operador de busca local 3-opt em conjunto com a baixa diversidade operada pelas reduzidas taxas de perturbação. Pode-se observar um tangenciamento de resultados entre os operadores de Busca Local, que é diminuído com o aumento da diversificação, elevando os valores de desvios padrão e grão computacional.

Tabela 30 – Resultados dos experimentos do ILS para a instância d493

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Busca Local	2-opt	FO	43781,178	38391,833	47741,716	57963,586	66628,162	65674,733
		DP	500,356	438,764	750,227	745,246	1142,197	1125,853
		GC	1,892	1,936	1,493	1,857	1,466	2,104
	3-opt	FO	41424,467	38050,734	43363,926	52789,806	59001,644	55237,926
		DP	710,134	597,940	743,382	603,312	842,881	868,025
		GC	1,902	2,400	1,590	2,326	2,352	1,535

Fonte: Elaborado pelo autor

5.3.3.6 Análise dos experimentos com Busca Tabu para a instância d493

Na tabela 31, é possível ver os melhores resultados concentrados no quadrante Q4, tal efeito é fruto da alta diversificação introduzida pelos dois mecanismos da Busca Tabu, consequência da extensão do tempo de busca dado pelo alto valor de NBMax e da memória de longo prazo. Também se observa neste quadrante a menor variabilidade dos desvios-padrões médios e a pior concentração dos grãos computacionais médios. Nota-se a contribuição do mecanismo de memória cuja maior duração, visto pelos maiores valores de lista tabu, aumenta a diversificação na busca de soluções. Também se percebe que o aumento das estruturas de memória afeta majoritariamente buscas mais curtas, onde é significativamente menor o processo de intensificação.

Tabela 31 – Resultados dos experimentos do BT para a instância d493

			Lista Tabu					
			10	25	50	75	100	200
NBMax	10	FO	134611,417	98748,909	87757,269	71799,252	63646,130	71461,108
		DP	1538,416	1269,629	1128,308	1128,274	1000,153	1122,960
		GC	0,396	1,025	0,821	1,050	2,182	1,059
	25	FO	123006,200	100923,794	70605,064	65477,642	58098,906	61970,608
		DP	1405,785	1441,768	907,779	1028,934	663,988	708,236
		GC	0,367	0,839	1,363	1,803	1,607	1,577
	50	FO	94390,636	84421,968	69754,081	63363,398	50048,502	58627,182
		DP	1618,125	964,822	1195,784	995,711	786,476	753,778
		GC	0,755	1,203	1,292	1,425	1,925	1,920
	75	FO	68298,316	56220,429	52686,414	44993,332	42887,521	45343,182
		DP	878,121	722,834	827,929	707,038	490,143	712,536
		GC	1,128	1,920	2,118	1,359	1,653	1,508
	100	FO	51615,368	50383,855	44473,640	42727,996	41069,980	44201,458
		DP	663,626	575,815	762,405	549,360	528,043	505,160
		GC	2,082	2,544	1,740	2,502	1,458	1,316
	200	FO	59499,907	52777,412	48489,501	48341,360	45377,790	48921,892
		DP	764,999	753,963	761,978	621,532	518,603	698,884
		GC	1,176	1,907	1,517	1,944	1,456	1,335

Fonte: Elaborado pelo autor

5.3.3.7 Análise dos experimentos com GRASP para a instância d493

Na tabela 32, apresentam-se os resultados de melhor qualidade nos quadrantes do lado direito, especialmente reduzidos por meio da maior diversidade introduzida pelo aumento da Lista de Candidatos. Cabe notar que o operador de Busca Local 2-opt apresenta a maior variabilidade nos diferentes tamanhos de Lista de Candidatos, por outro lado, o operador 3-opt mostra uma redução dos resultados mais estável, elucidando o papel da diversificação introduzida pela Lista de Candidatos.

Tabela 32 – Resultados dos experimentos do GRASP para a instância d493

			Lista de Candidatos α					
			0,1	0,2	0,3	0,5	0,7	0,9
Busca Local	2-opt	FO	71564,680	82296,036	75586,135	61868,443	39919,626	47575,819
		DP	1124,588	1175,658	1295,762	795,451	456,224	815,585
		GC	2,839	1,483	1,981	1,789	3,171	2,410
	3-opt	FO	55719,291	52308,658	48486,996	44231,281	37120,492	40940,276
		DP	795,990	597,813	623,404	695,063	477,263	467,889
		GC	2,857	1,979	2,550	3,035	2,128	2,065

Fonte: Elaborado pelo autor

5.3.3.8 Análise dos experimentos com Otimização por Colônia de Formigas para a instância d493

A Tabela 33 revela a ocorrência da alta atração do feromônio na qualidade das soluções, concentrando-as na parte inferior da tabela, com os melhores resultados apresentados no quadrante Q4. Sob outra ótica, é nítida a tendência que dispõe as melhores soluções com taxas de evaporação do feromônio médias, isto é, presentes na região central das taxas analisadas, em contrapartida nesta mesma região, numa análise do grão computacional médio, percebe-se sua elevação. Noutra perspectiva, quando observamos o grão computacional médio, percebe-se o aumento nos quadrantes inferiores, onde majoritariamente concentram-se os melhores resultados deste experimento, frutos da alta intensificação.

Tabela 33 – Resultados dos experimentos do ACO para a instância d493

			Evaporação do Feromônio					
			0,01	0,02	0,03	0,05	0,10	0,15
Atração do Feromônio	0,5	FO	104205,498	74672,383	59819,545	47683,003	63440,248	88766,175
		DP	1786,380	853,399	683,652	749,304	815,660	1014,471
		GC	0,867	1,029	1,263	1,716	1,430	1,381
	1,0	FO	100253,878	67186,834	55593,815	47715,491	63377,582	92558,591
		DP	1432,198	863,831	873,617	817,980	724,315	1190,039
		GC	0,835	1,036	2,177	1,756	1,042	0,849
	2,0	FO	77603,905	64774,073	58431,898	43543,501	55962,872	75921,355
		DP	1330,353	832,810	751,267	497,640	959,364	1084,591
		GC	1,339	1,285	1,443	1,619	2,246	1,722
	3,0	FO	70299,727	51283,353	46886,040	41233,025	47554,208	64026,773
		DP	903,854	586,095	602,821	589,043	611,411	914,668
		GC	1,032	1,519	1,739	1,573	2,079	2,127
	4,0	FO	53431,936	46840,377	42636,231	39532,214	42625,831	49725,674
		DP	763,313	669,148	730,907	621,221	730,729	568,293
		GC	2,475	1,616	1,476	1,735	2,653	1,599
	5,0	FO	67388,284	55159,270	51468,814	43805,091	49008,738	74531,732
		DP	770,152	787,990	808,796	688,366	560,100	1171,213
		GC	1,286	1,217	1,642	1,884	1,729	0,998

Fonte: Elaborado pelo autor

5.3.3.9 Análise dos experimentos consolidados para a instância d493

Na Tabela 34, pode-se observar a consolidação dos melhores resultados dos experimentos e de suas médias, para a instância d493, divididos em duas partes. A primeira parte traz a consolidação da melhor média de 100 repetições dos experimentos, dentre as parametrizações testadas em cada uma das 8 meta-heurísticas, incluindo a melhor execução média obtida pela hiper-heurística. Em cada cenário exposto, calcula-se o Gap percentual relativo ao valor obtido pela hiper-heurística e ao BKS.

Já na segunda parte, têm-se os valores médios testados para cada meta-heurística, incluindo a melhor execução média da hiper-heurística. Para cada um dos valores médios encontrado pela meta-heurística, mede-se o Gap percentual em relação a hiper-heurística e ao BKS.

Neste experimento a meta-heurística GRASP forneceu dentre as parametrizações analisadas o melhor resultado, onde apresenta uma melhora de 0,457% em relação a FO, 13,037% melhor em comparação ao DP e 45,415% no que concerne ao GC. Em contrapartida, a Busca em Vizinhança Variável apresenta o melhor resultado médio, fornecendo resultados de FO 29,921% melhores, um DP 44,518% inferior e o GC 26,065% melhor.

Já sobre a média dos melhores resultados, a hiper-heurística obteve melhora de 8,288% nos resultados da FO, uma diminuição de 23,95% sobre o desvio padrão e 36,653% no GC. No geral, a hiper-heurística obteve sobre a média das médias da FO das meta-heurísticas uma melhora de 35,869%, sobre o desvio padrão uma melhora de 48,903% e um GC 32,891% inferior.

Por fim, ao colocarmos lado-a-lado a hiper-heurística e o Ótimo Global, foi obtido um resultado 5,568% superior. Vale destacar que o GC obtido pela hiper-heurística é o melhor dentre os testes efetuados com as outras técnicas.

Tabela 34 – Resultados dos experimentos consolidados para a instância d493

BKS	35002	Melhor	GAP HH	GAP BKS	Média	GAP HH	GAP BKS
AG	FO	43.084,497	14,236%	23,092%	65.116,287	43,254%	86,036%
	DP	615,493	32,568%	-	909,426	54,362%	-
	GC	2,669	56,482%	-	1,990	41,649%	-
SA	FO	41.182,290	10,275%	17,657%	60.127,041	38,545%	71,782%
	DP	647,150	35,866%	-	820,447	49,413%	-
	GC	1,571	26,085%	-	1,801	35,524%	-
SS	FO	41.717,899	11,426%	19,187%	55.819,591	33,803%	59,475%
	DP	476,776	12,948%	-	793,096	47,668%	-
	GC	1,305	11,000%	-	1,529	24,067%	-
VNS	FO	41.268,266	10,461%	17,903%	52.727,666	29,921%	50,642%
	DP	471,637	12,000%	-	748,062	44,518%	-
	GC	2,288	49,236%	-	1,571	26,065%	-
ILS	FO	38.050,734	2,890%	8,710%	50.837,476	27,315%	45,242%
	DP	597,940	30,588%	-	755,693	45,078%	-
	GC	2,400	51,618%	-	1,905	39,024%	-
BT	FO	41.069,980	10,029%	17,336%	64.361,706	42,589%	83,880%
	DP	528,043	21,400%	-	880,608	52,869%	-
	GC	1,458	20,332%	-	1,480	21,516%	-
GRASP	FO	37.120,492	0,457%	6,052%	54.801,478	32,573%	56,567%
	DP	477,263	13,037%	-	776,724	46,565%	-
	GC	2,128	45,415%	-	2,357	50,732%	-
ACO	FO	39.532,214	6,529%	12,943%	60.526,332	38,951%	72,922%
	DP	621,221	33,189%	-	842,750	50,752%	-
	GC	1,735	33,055%	-	1,539	24,554%	-
MÉDIA	FO	40.378,297	8,288%	15,360%	58.039,697	35,869%	65,818%
	DP	554,440	23,950%	-	815,851	48,903%	-
	GC	1,944	36,653%	-	1,772	32,891%	-
HH	FO	36.951,012	-	5,568%	36.951,012	-	5,568%
	DP	415,041	-	-	415,041	-	-
	GC	1,161	-	-	1,161	-	-

Fonte: Elaborado pelo autor

5.3.4 Análise dos experimentos para a instância u1060

5.3.4.1 Análise dos experimentos com Algoritmo Genético para a instância u1060

Observa-se na Tabela 35, em especial no quadrante Q4, a concentração de soluções de melhor qualidade como consequência da elevada diversificação introduzida pelos dois mecanismos do Algoritmo Genético. Conjuntamente obtém-se nesta mesma região de análise a menor média de grão computacional, reforçando que tal conjunto conduz à otimalidade desejada. Pode-se notar que dentre as interações paramétricas testadas, os valores de Taxa de Mutação provenientes do lado esquerdo, interagem de maneira desfavorável com as diferentes Taxas de Cruzamento, ilustrado pela presença de cores quentes, além de elevar os valores médios de FO para cada experimento.

Tabela 35 – Resultados dos experimentos do AG para a instância u1060

			Taxa de Mutação					
			0,01	0,02	0,05	0,10	0,15	0,20
Taxa de Cruzamento	0,02	FO	311691,976	318106,405	305402,528	267803,005	273476,374	287148,432
		DP	4452,743	5453,253	3926,604	3060,606	3516,125	4512,333
		GC	2,938	2,466	2,095	2,720	1,675	2,970
	0,05	FO	287052,483	292715,808	274554,015	251662,797	258435,520	266829,320
		DP	4100,750	3345,324	4314,420	4314,219	2953,549	4193,032
		GC	1,744	2,547	1,606	1,777	3,075	2,043
	0,10	FO	290558,194	267324,283	271755,980	244589,770	260432,394	269598,620
		DP	4150,831	3818,918	3882,228	3494,140	4464,555	3081,127
		GC	1,915	2,130	2,532	2,351	1,925	3,418
	0,25	FO	262918,913	253624,174	254234,544	240815,298	244733,726	258676,113
		DP	3380,386	4347,843	3995,114	3784,240	3845,816	4064,910
		GC	2,817	3,309	1,774	1,700	2,579	1,627
	0,50	FO	289666,612	263760,941	261912,800	248181,276	260255,571	261446,992
		DP	3724,285	3768,013	3741,611	3545,447	3346,143	3734,957
		GC	2,509	1,765	2,626	2,903	2,371	1,613
	0,75	FO	301245,242	290638,716	280127,870	253521,087	268075,028	277093,175
		DP	3873,153	4567,180	4802,192	3621,730	4595,572	3958,474
		GC	1,567	2,880	2,707	1,953	1,942	2,109

Fonte: Elaborado pelo autor

5.3.4.2 Análise dos experimentos com Arrefecimento Simulado para a instância u1060

Concentrados no quadrante Q4, da Tabela 36, os resultados de melhor qualidade indicam maior intensificação por parte da Redução de Temperatura, bem como uma alta diversificação introduzida pela elevada Temperatura Inicial, fruto disso, no quadrante Q4, encontra-se uma interação equalizada entre os dois mecanismos. Verifica-se que o resfriamento muito lento, ou muito rápido, não contribui qualitativamente para os resultados médios da FO, por outra perspectiva, nota-se que, quanto menores as Temperaturas Iniciais, independentes dos fatores de resfriamento, os resultados acabam sendo piorados, como observa-se pela presença de tons mais quentes nos quadrantes superiores.

Tabela 36 – Resultados dos experimentos do SA para a instância u1060

			Redução da Temperatura					
			0,001	0,01	0,05	0,1	0,5	1
Temperatura Inicial	100	FO	294719,654	285343,399	272929,290	254031,560	277630,925	289784,986
		DP	5052,337	3668,701	4678,788	3629,022	3172,925	4967,743
		GC	2,125	1,270	1,416	1,412	1,703	1,815
	500	FO	297249,768	294126,732	274034,785	248230,935	261353,512	273878,748
		DP	3397,140	5042,173	3523,304	2836,925	3360,259	3521,298
		GC	1,413	2,078	1,217	1,565	1,509	1,983
	1000	FO	276649,233	273806,870	261064,608	249174,118	260575,709	268193,678
		DP	4347,345	4302,679	3356,545	4271,556	2978,008	3831,338
		GC	1,193	1,891	1,689	2,289	1,095	1,386
	2500	FO	258049,824	255451,526	256916,430	240309,919	252188,561	257935,292
		DP	4423,711	2919,446	2936,188	2746,399	4323,232	4421,748
		GC	1,764	1,181	1,418	1,163	1,651	2,255
	5000	FO	251274,129	247683,818	246982,060	234576,726	239497,630	243324,761
		DP	3589,630	2830,672	3528,315	3015,986	3421,395	4171,282
		GC	2,120	1,643	2,009	1,670	1,355	1,124
	10000	FO	235790,021	238329,546	235229,136	229281,673	231744,886	238022,963
		DP	4042,115	3404,708	3696,458	2620,362	3641,705	4080,394
		GC	2,174	2,297	1,168	1,173	1,567	1,133

Fonte: Elaborado pelo autor

5.3.4.3 Análise dos experimentos com Busca Dispersa para a instância u1060

Encontra-se na tabela 37, no quadrante Q4, os resultados de melhor qualidade, fornecidos pelos Conjuntos de Referência e Conjuntos de Diversificação de tamanho superior, estes parâmetros fornecem respectivamente maior intensificação e diversificação, nivelando assim suas características e produzindo soluções de qualidade. Nota-se no quadrante Q2, onde tem-se mais diversidade e no quadrante Q3, onde há a presença de mais intensificação, que os valores médios de grão computacional são elevados, concluindo-se assim que movimentos extremos de intensificação e diversificação geram buscas mais longas e/ou de menor qualidade final.

Tabela 37 – Resultados dos experimentos do SS para a instância u1060

			Conjunto de Diversificação					
			25	50	75	100	150	200
Conjunto de Referência	2	FO	327993,600	296286,200	292262,218	273127,450	259064,352	294809,488
		DP	4685,623	3809,394	3757,657	3511,639	2960,735	4632,721
		GC	1,006	0,584	0,670	0,830	1,038	0,834
	5	FO	307175,218	276950,634	264363,701	256297,246	248458,162	271732,752
		DP	3949,396	3956,438	3021,299	2929,111	2839,522	3493,707
		GC	1,180	0,905	0,751	0,907	0,700	1,027
	10	FO	272607,763	263060,636	254222,045	250364,532	238771,878	253153,450
		DP	3115,517	3006,407	3994,918	3576,636	4093,232	3978,126
		GC	0,812	0,757	0,645	0,893	0,839	0,821
	15	FO	265678,723	257652,240	243740,357	241468,927	236881,831	249653,534
		DP	3415,869	3680,746	3830,206	3104,600	3722,429	3566,479
		GC	0,796	0,712	0,830	0,733	0,856	0,786
	20	FO	247061,282	240777,164	242580,286	234746,563	232966,467	239882,314
		DP	3529,447	3439,674	3811,976	3018,170	3660,902	3426,890
		GC	0,772	0,742	0,943	1,271	0,712	0,700
	25	FO	273702,595	267683,056	251993,366	250181,786	242440,091	251196,442
		DP	4692,044	3059,235	2879,924	3574,026	3463,430	2870,816
		GC	0,701	1,270	1,093	0,746	1,060	0,834

Fonte: Elaborado pelo autor

5.3.4.4 Análise dos experimentos com Busca em Vizinhaça Variável para a instância u1060

Analisando a Tabela 38, percebe-se no lado direito a maior concentração de soluções de qualidade, sobretudo no quadrante Q4. Nota-se também que a diversificação fruto dos dois mecanismos, acaba por elevar o grão computacional médio deste quadrante. Através da presença de cores mais quentes nos quadrantes superiores, nota-se que vizinhanças menores, independentes das perturbações geradas, estão mais propensas a ruídos, uma vez que se constata a maior média dos desvios padrões na parte superior da tabela, com destaque para o quadrante Q1.

Tabela 38 – Resultados dos experimentos do VNS para a instância u1060

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Vizinhanças	5	FO	304974,716	273031,980	271718,396	247339,900	267615,576	281085,693
		DP	3485,425	3900,457	4658,030	3886,770	3058,464	4015,510
		GC	0,993	1,593	1,303	1,449	1,213	1,195
	10	FO	294662,034	279425,342	254795,041	251996,633	267162,114	276922,723
		DP	3367,566	3592,612	3639,929	3959,947	3053,281	4351,643
		GC	1,327	1,964	1,561	1,967	1,560	1,974
	25	FO	277534,945	268375,668	257542,647	244498,870	251524,044	257805,888
		DP	4361,263	4600,726	4047,099	4191,409	2874,561	4419,530
		GC	1,200	1,644	1,292	1,364	1,238	1,413
	50	FO	258499,264	260540,480	252099,310	237650,250	251533,085	255062,038
		DP	4062,131	3722,007	2881,135	2716,003	3593,330	4372,492
		GC	1,781	1,117	1,479	1,177	1,233	1,221
	75	FO	233918,347	230560,916	229606,726	227346,488	229673,718	231539,563
		DP	3341,691	2634,982	2624,077	3247,807	3937,264	3638,479
		GC	1,120	1,788	1,250	1,836	2,250	1,431
	100	FO	246629,746	241044,040	234441,272	232499,368	236241,358	241374,856
		DP	3170,954	3099,138	3349,161	2989,278	3712,364	2758,570
		GC	2,100	1,144	2,117	2,232	1,854	1,092

Fonte: Elaborado pelo autor

5.3.4.5 Análise dos experimentos com Busca Local Iterada para a instância u1060

Os experimentos apresentados na tabela 39, concentram os melhores resultados nos quadrantes do lado esquerdo, conjuntamente aos menores valores de desvio padrão, dados pelo operador de busca local 3-opt. Cabe notar que o operador 3-opt traz consigo a inserção de maior diversidade nas soluções geradas em comparação com o operador 2-opt. É sabido que o aumento da diversificação criado pela elevação das taxas de perturbação introduz maiores níveis de aleatoriedade em ambos os mecanismos de busca local. Isto é observado pelo aumento médio dos desvios padrões nos quadrantes do lado direito e a deterioração dos resultados médios da FO, visto pelas cores mais quentes.

Tabela 39 – Resultados dos experimentos do ILS para a instância u1060

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Busca Local	2-opt	FO	259516,864	248276,556	238799,411	264778,332	280647,797	269739,908
		DP	3336,645	3546,808	3411,420	4539,057	4811,105	4238,770
		GC	1,575	1,731	1,434	1,913	0,994	1,096
	3-opt	FO	245257,205	244423,065	235033,932	248997,609	262408,526	262887,443
		DP	4204,409	3491,758	2686,102	3912,820	3748,693	4131,088
		GC	1,704	1,836	1,033	1,520	1,509	1,953

Fonte: Elaborado pelo autor

5.3.4.6 Análise dos experimentos com Busca Tabu para a instância u1060

Analisando a Tabela 40, percebe-se que as melhores soluções estão concentradas nos quadrantes do lado direito, especialmente no quadrante Q4. Tal efeito é fruto da alta diversificação introduzida pelos dois mecanismos da Busca Tabu, decorrentes da extensão do tempo de busca dado pelo alto valor de NBMax e da memória de longo prazo. Observa-se que o uso do mecanismo de memória de maior duração, favorece o decaimento dos valores médios dos experimentos. Também se nota que buscas mais curtas e com menor memória, onde é expressivamente maior o processo de intensificação, geram grãos computacionais menores e consequentemente, resultados de qualidade inferior, como se percebe pela presença de tons mais quentes.

Tabela 40 – Resultados dos experimentos do BT para a instância u1060

			Lista Tabu					
			10	25	50	75	100	200
NBMax	10	FO	384791,178	332765,396	301014,426	296100,631	281175,364	257167,765
		DP	5497,017	4278,412	5160,247	4230,009	3213,433	3306,443
		GC	0,606	0,785	0,943	0,859	0,723	1,369
	25	FO	384218,736	353048,739	310498,352	292557,181	276472,066	252281,891
		DP	6037,723	4034,843	3548,553	3343,511	3949,601	3243,624
		GC	0,993	0,809	0,828	0,693	1,321	1,493
	50	FO	325644,948	312291,564	282080,754	276717,682	263233,434	246563,537
		DP	5117,278	4015,177	3223,780	3162,488	4136,525	4226,803
		GC	0,751	0,690	0,742	0,911	1,074	1,172
	75	FO	292607,694	261664,029	253103,778	251757,202	245219,234	236903,224
		DP	4180,110	4111,863	4338,922	4315,838	3503,132	2707,465
		GC	1,193	0,760	1,080	0,823	1,068	0,899
	100	FO	260891,104	251204,398	240673,870	239157,755	234421,100	232090,714
		DP	4099,717	4306,361	2750,559	4099,847	3348,873	3647,140
		GC	1,092	0,778	1,220	1,175	0,861	0,832
	200	FO	246607,616	239642,999	234194,999	232585,420	231271,900	227721,903
		DP	4227,559	3423,471	2676,514	2990,384	3303,884	3578,487
		GC	1,289	0,783	0,830	1,391	1,226	0,929

Fonte: Elaborado pelo autor

5.3.4.7 Análise dos experimentos com GRASP para a instância u1060

Na tabela 41, encontram-se os resultados de melhor qualidade no quadrante do lado direito, onde são notadamente menores devido à elevada diversificação. Esta observação é notada pelo decaimento dos valores médios, a partir das Listas de Candidatos de valor 0.5. Particularmente neste experimento, as Listas de Candidatos de tamanho médio, fornecem os melhores resultados, sendo estes localizado em uma região de maior equilíbrio entre os mecanismos de operação, destacando-se o quadrante Q4 com os melhores resultados.

Tabela 41 – Resultados dos experimentos do GRASP para a instância u1060

			Lista de Candidatos α					
			0,1	0,2	0,3	0,5	0,7	0,9
Busca Local	2-opt	FO	312006,126	306500,904	306258,516	246773,622	259652,556	271999,525
		DP	4457,230	4378,584	4812,634	3525,337	2967,458	3108,566
		GC	1,750	1,253	1,391	1,489	1,428	2,242
	3-opt	FO	257337,908	258464,301	249673,641	233188,740	235899,838	240959,496
		DP	2941,005	3692,347	2853,413	3997,521	4043,997	4130,734
		GC	1,598	1,274	2,406	1,736	1,345	1,329

Fonte: Elaborado pelo autor

5.3.4.8 Análise dos experimentos com Otimização por Colônia de Formigas para a instância u1060

Concentrando-se no quadrante Q4, da tabela 42, vê-se a presença da diversificação por meio da ocorrência de maiores valores de evaporação do feromônio na qualidade das soluções, juntamente com a alta intensificação trazida pela atração do feromônio. Em contrapartida, obtém-se uma elevação do grão computacional médio nesta região. Já o quadrante oposto (Q1) que obtém os piores resultados, notado pela maior incidência de cores quentes, traz uma redução dos valores médios de grão computacional.

Tabela 42 – Resultados dos experimentos do ACO para a instância u1060

			Evaporação do Feromônio					
			0,01	0,02	0,03	0,05	0,10	0,15
Atração do Feromônio	0,5	FO	304603,950	279247,661	263623,699	251270,965	265697,457	279554,810
		DP	5221,782	4388,178	4519,263	4307,502	3036,542	3194,912
		GC	1,285	1,901	1,064	1,557	1,769	0,966
	1,0	FO	297997,486	266480,208	262697,005	246971,234	265504,217	263649,888
		DP	5108,528	3426,174	4503,377	3880,977	4172,209	3013,142
		GC	0,928	1,509	1,461	1,012	1,071	0,975
	2,0	FO	291304,220	262309,032	249818,001	246253,852	255100,584	260773,272
		DP	4161,489	2997,818	4282,594	3166,121	4373,153	2980,266
		GC	1,662	1,000	1,361	1,739	1,191	2,077
	3,0	FO	289511,996	250227,379	252292,812	241917,523	251880,768	260794,902
		DP	4963,063	4289,612	3243,765	2764,772	3598,297	4098,206
		GC	0,955	1,115	1,129	1,193	1,076	1,445
	4,0	FO	258540,497	246850,157	243227,953	236232,742	239937,969	245842,464
		DP	3693,436	3526,431	3474,685	3374,753	4113,222	2809,628
		GC	1,756	1,710	1,857	1,037	2,086	1,304
	5,0	FO	253393,169	240832,200	235843,998	233489,505	235805,356	240443,878
		DP	2895,922	3096,414	4043,040	3335,564	3031,783	2747,930
		GC	1,187	1,395	1,132	1,615	1,324	1,889

Fonte: Elaborado pelo autor

5.3.4.9 Análise dos experimentos consolidados para a instância u1060

Na Tabela 43 podemos observar a reunião dos resultados dos experimentos realizados com a instância u1060, que estão divididos em duas seções. Inicialmente têm-se a consolidação da melhor média das repetições dos experimentos, para os 36 conjuntos de parâmetros testados em cada meta-heurística, incluindo a melhor execução média da hiper-heurística. Após, faz-se o cálculo do Gap percentual das meta-heurísticas e da hiper-heurística em relação ao BKS.

A segunda seção traz a média das repetições nos 36 conjuntos de parâmetros testados para cada meta-heurística, incluindo a melhor execução média da hiper-heurística. Em cada valor médio obtido pela meta-heurística, apresenta-se o Gap percentual em relação a hiper-heurística e ao BKS.

Em ambos os casos para a instância u1060, a meta-heurística Busca em Vizinhança Variável apresentou o melhor resultado dentre as 288 parametrizações testadas, e o melhor resultado médio dos 36 parâmetros testados. O resultado obtido pela hiper-heurística sobre o melhor resultado da Busca em Vizinhança Variável apresenta uma melhora de 0,143% em relação a FO, 29,808% melhor em comparação ao DP e 64,305% no que concerne ao GC. Por outro lado, o resultado obtido pela hiper-heurística em contraste com a média da Busca em Vizinhança Variável, fornece resultados 10,761% melhores, sendo o DP 36,535% melhor e o GC 56,699% menor.

Analisando a média dos melhores resultados, obtém-se através da hiper-heurística uma melhora dos resultados em 2,318%, uma diminuição do desvio padrão em 30,793% e um CG 45,711% menor. Numa perspectiva mais geral, a hiper-heurística obteve sobre a média das médias das meta-heurísticas, uma melhora de 13,268% sobre a FO, um desvio padrão 39,081% menor e o GC mostrou-se 51,512% melhor. Já na comparação da hiper-heurística com o valor ótimo global conhecido, os resultados são em média 1,306% superiores, contudo, vale notar que, assim como os demais experimentos com a hiper-heurística o valor de GC obtido é o melhor dentre todos os experimentos efetuados com as meta-heurísticas.

Tabela 43 – Resultados dos experimentos consolidados para a instância u1060

BKS	224094	Melhor	GAP HH	GAP BKS	Média	GAP HH	GAP BKS
AG	FO	240.815,298	5,728%	7,462%	271.390,722	16,349%	21,106%
	DP	3.784,240	39,758%	-	3.936,995	42,095%	-
	GC	1,700	61,459%	-	2,297	71,471%	-
SA	FO	229.281,673	0,986%	2,315%	258.482,428	12,171%	15,346%
	DP	2.620,362	13,000%	-	3.715,329	38,640%	-
	GC	1,173	44,149%	-	1,609	59,271%	-
SS	FO	232.966,467	2,552%	3,959%	260.305,232	12,787%	16,159%
	DP	3.660,902	37,728%	-	3.557,193	35,913%	-
	GC	0,712	8,000%	-	0,854	23,315%	-
VNS	FO	227.346,488	0,143%	1,451%	254.396,473	10,761%	13,522%
	DP	3.247,807	29,808%	-	3.592,086	36,535%	-
	GC	1,836	64,305%	-	1,513	56,699%	-
ILS	FO	235.033,932	3,409%	4,882%	255.063,887	10,994%	13,820%
	DP	2.686,102	15,129%	-	3.838,223	40,605%	-
	GC	1,033	36,583%	-	1,525	57,030%	-
BT	FO	227.721,903	0,308%	1,619%	273.342,850	16,946%	21,977%
	DP	3.578,487	36,294%	-	3.870,433	41,099%	-
	GC	0,929	29,498%	-	0,972	32,598%	-
GRASP	FO	233.188,740	2,645%	4,058%	264.892,931	14,297%	18,206%
	DP	3.997,521	42,972%	-	3.742,402	39,084%	-
	GC	1,736	62,266%	-	1,603	59,134%	-
ACO	FO	233.489,505	2,770%	4,193%	257.497,856	11,836%	14,906%
	DP	3.335,564	31,654%	-	3.717,626	38,678%	-
	GC	1,615	59,432%	-	1,382	52,575%	-
MÉDIA	FO	232.480,501	2,318%	3,742%	261.921,547	13,268%	16,880%
	DP	3.363,873	30,793%	-	3.746,286	39,081%	-
	GC	1,342	45,711%	-	1,469	51,512%	-
HH	FO	227.021,240	-	1,306%	227.021,240	-	1,306%
	DP	2.279,715	-	-	2.279,715	-	-
	GC	0,655	-	-	0,655	-	-

Fonte: Elaborado pelo autor

5.3.5 Análise dos experimentos para a instância rl1889

5.3.5.1 Análise dos experimentos com Algoritmo Genético para a instância rl1889

Observa-se o acúmulo de soluções de melhor qualidade no quadrante Q4 da Tabela 44, como fruto da alta diversificação dada pelos mecanismos em análise, ao mesmo tempo, têm-se nesta região, a menor média de Grão Computacional, fortalecendo a otimalidade almejada. Dentre as interações paramétricas testadas, nota-se que os valores de Taxa de Mutação, com a presença de tons mais quentes, no lado esquerdo, interagem de maneira danosa na qualidade dos resultados com as diferentes Taxas de Cruzamento, ou seja, a alta intensificação da busca contribui negativamente para a obtenção de bons resultados.

Tabela 44 – Resultados dos experimentos do AG para a instância rl1889

			Taxa de Mutação					
			0,01	0,02	0,05	0,10	0,15	0,20
Taxa de Cruzamento	0,02	FO	417735,367	389200,763	376642,711	371789,520	352126,254	380546,811
		DP	5370,883	5004,010	5918,671	5311,279	4527,338	6523,660
		GC	1,723	1,253	2,123	1,485	2,337	2,035
	0,05	FO	379653,537	386254,740	358106,630	360692,292	348224,682	352021,208
		DP	5423,622	4966,132	4092,647	4637,472	4477,174	5531,762
		GC	1,188	1,688	1,577	1,635	2,037	1,538
	0,10	FO	377312,548	359627,453	364206,348	342328,602	344430,554	346471,024
		DP	5390,179	4110,028	4162,358	3912,327	3936,349	4949,586
		GC	2,467	1,488	2,553	1,328	1,612	1,437
	0,25	FO	352054,447	351822,509	341560,239	340538,066	<u>331050,518</u>	338553,823
		DP	4526,414	6031,243	4391,489	3891,864	<u>5675,152</u>	5320,132
		GC	1,472	1,609	1,994	1,544	<u>1,914</u>	1,427
	0,50	FO	376710,778	374933,576	364083,230	349429,960	340754,055	354645,222
		DP	6457,899	5356,194	5201,189	5990,228	5354,707	6079,632
		GC	1,344	1,769	2,185	1,886	1,472	1,293
	0,75	FO	383554,091	372713,352	359684,370	360427,072	352054,971	354729,836
		DP	6027,279	4259,581	5652,183	5663,854	5029,357	4560,812
		GC	1,315	1,487	1,227	1,267	1,646	1,235

Fonte: Elaborado pelo autor

5.3.5.2 Análise dos experimentos com Arrefecimento Simulado para a instância rl1889

No quadrante Q3 da Tabela 45 estão concentrados os resultados de maior qualidade e os menores desvios padrões, demonstrando que a elevação da temperatura inicial e a baixa redução de temperatura viabilizam melhores resultados, por meio do aumento da diversidade das soluções. Esta observação sobre as temperaturas iniciais pode ser claramente notada na predominância de tons amarelados nos quadrantes inferiores. Cabe ressaltar que se obteve os melhores valores de Grão Computacional no quadrante Q2, onde é realizada uma busca mais intensa, apresentando um movimento contrário ao da obtenção de soluções de qualidade.

Tabela 45 – Resultados dos experimentos do SA para a instância rl1889

			Redução da Temperatura					
			0,001	0,01	0,05	0,1	0,5	1
Temperatura Inicial	100	FO	394786,628	377267,616	355952,044	384254,045	412883,165	431637,594
		DP	6767,771	4311,630	5085,029	4940,409	5308,498	5549,626
		GC	1,628	1,422	1,394	1,753	2,282	1,296
	500	FO	397007,118	358005,550	349744,959	374397,618	406539,976	424102,488
		DP	5671,530	5625,802	5495,992	4813,684	6969,257	6058,607
		GC	1,732	2,374	1,572	2,160	1,358	1,295
	1000	FO	369073,638	356374,472	336222,970	362083,930	389042,886	393854,654
		DP	4217,984	5600,170	3842,548	4655,365	5557,756	5063,846
		GC	1,728	1,567	2,251	1,291	1,363	1,750
	2500	FO	353860,589	345313,602	334022,278	350405,675	373855,901	362213,984
		DP	6066,182	3946,441	5248,922	6006,954	4272,639	6209,383
		GC	2,613	2,590	2,444	1,408	1,667	1,389
	5000	FO	345834,480	336133,672	326673,582	344252,803	345640,877	351502,163
		DP	4446,443	5762,292	5133,442	4917,897	4937,727	5021,459
		GC	2,218	1,853	1,925	1,951	1,578	1,437
	10000	FO	333107,874	326556,236	323091,493	327550,636	329396,112	334506,446
		DP	3806,947	4665,089	5077,152	4211,365	5646,790	5734,396
		GC	1,913	2,303	1,422	2,318	1,461	1,679

Fonte: Elaborado pelo autor

5.3.5.3 Análise dos experimentos com Busca Dispersa para a instância rl1889

Encontram-se na Tabela 46, representados no quadrante Q4, os melhores resultados médios de FO e Desvio Padrão, originados pelo equilíbrio entre a alta diversificação e intensificação, como contribuições respectivas dos Conjuntos de Diversificação e Conjuntos de Referência de tamanhos maiores. Sob outra perspectiva, pode-se entender que os diferentes experimentos com Conjuntos de Referência trazem certa estabilidade, que é alterada pontualmente nos Conjuntos de Diversificação de tamanho 1000, acarretando uma elevação na diversificação das soluções geradas, notadamente pelos tons amarelados nos quadrantes direitos.

Tabela 46 – Resultados dos experimentos do SS para a instância rl1889

			Conjunto de Diversificação					
			25	50	75	100	150	200
Conjunto de Referência	2	FO	430408,819	393871,320	375534,470	355635,754	342302,048	375784,589
		DP	7378,437	6752,080	5901,256	5588,562	4890,029	5905,186
		GC	2,127	3,067	2,490	2,077	1,650	2,150
	5	FO	381646,198	383141,682	371364,514	358234,807	344708,028	361295,544
		DP	5452,089	4926,107	4244,166	5117,640	4431,960	6193,638
		GC	2,473	1,805	1,550	1,660	2,573	1,691
	10	FO	364576,109	362043,664	343002,147	335842,418	335498,402	351550,128
		DP	5208,230	5172,052	4900,031	4797,749	5751,401	4519,930
		GC	2,400	1,617	1,934	1,616	1,848	1,723
	15	FO	341056,958	333819,464	330403,498	329448,101	325241,946	330255,490
		DP	3897,794	5245,734	5192,055	3765,121	3717,051	5661,523
		GC	1,861	3,149	1,682	1,625	1,663	3,052
	20	FO	355543,094	350140,724	352336,720	337291,046	331281,493	350137,194
		DP	5587,106	5502,211	4026,705	5782,132	4259,333	4001,568
		GC	1,795	1,564	1,931	2,682	1,835	2,798
	25	FO	402448,013	368503,038	368974,685	349476,360	344582,311	370552,152
		DP	6324,183	5264,329	5271,067	4992,519	4922,604	4764,242
		GC	1,796	2,158	2,230	1,816	2,324	2,833

Fonte: Elaborado pelo autor

5.3.5.4 Análise dos experimentos com Busca em Vizinhaça Variável para a instância rl1889

Ao analisarmos a Tabela 47, percebe-se a presença de tons menos quentes nos quadrantes inferiores, concentrando as soluções de qualidade, em especial no quadrante Q4, uma região privilegiada pela maior diversificação. Ao observarmos a interferência das taxas de Perturbação de 10%, entende-se que independentes dos tamanhos de vizinhaça, as perturbações extremas, ou seja, demasiadamente pequenas não efetuam a movimentação necessária, assim como valores elevados efetuam uma movimentação descomedida.

Tabela 47 – Resultados dos experimentos do VNS para a instância rl1889

			Perturbação					
			0,01	0,02	0,05	0,10	0,15	0,20
Vizinhanças	5	FO	383884,288	371079,110	360209,616	339894,910	364786,952	371917,222
		DP	6032,467	5301,130	4116,681	5826,770	4168,994	5313,103
		GC	2,310	2,157	1,621	1,617	1,599	1,780
	10	FO	372785,922	369141,341	349613,430	343499,042	355066,253	362800,555
		DP	4260,411	4218,758	4495,030	5397,842	4565,138	5182,865
		GC	1,909	1,445	2,291	1,429	1,494	2,430
	25	FO	364131,955	351822,520	345642,849	333761,030	342367,827	351883,988
		DP	4161,508	4020,829	4937,755	5721,618	5869,163	4524,223
		GC	1,478	2,227	1,568	2,398	2,375	2,640
	50	FO	355767,768	347300,688	340811,414	332226,270	339502,001	354932,170
		DP	6098,876	4465,295	3894,988	5695,307	4365,026	4056,368
		GC	2,144	1,396	1,734	1,804	1,802	1,417
	75	FO	349254,957	343414,817	334491,862	329692,873	333391,925	342902,784
		DP	4490,421	4905,926	5734,146	4709,898	4762,742	5388,472
		GC	1,485	1,932	1,859	1,994	1,462	1,826
	100	FO	334156,732	330669,160	328124,969	323149,334	328985,255	332851,024
		DP	3818,934	4723,845	5624,999	4154,777	5169,768	5230,516
		GC	1,485	1,579	1,987	1,620	2,312	2,846

Fonte: Elaborado pelo autor

5.3.5.5 Análise dos experimentos com Busca Local Iterada para a instância rl1889

Na tabela 48, vê-se concentrados no quadrante Q3 os resultados de melhor qualidade. Conjuntamente, observa-se os menores valores de desvio padrão, que entretanto apresentam os maiores valores médios de Grão Computacional. Pode-se notar que a diversificação provocada pelo aumento da perturbação contribui negativamente para a obtenção de melhores resultados. Sob outra ótica, a maior diversificação do operador 3-opt fornece um equilíbrio quando aliado a baixas perturbações.

Tabela 48 – Resultados dos experimentos do ILS para a instância rl1889

			Perturbação					
			1%	2%	5%	10%	15%	20%
Busca Local	2-opt	FO	348550,544	336414,554	332638,898	361496,876	366611,438	372280,002
		DP	4481,364	4325,330	4751,984	4131,393	4713,576	5850,114
		GC	1,911	1,774	1,926	2,021	2,416	2,530
	3-opt	FO	340567,349	334487,551	329517,134	344898,618	346660,282	366894,575
		DP	4378,723	5256,233	3765,910	5912,548	4457,061	4193,081
		GC	2,306	3,218	2,611	2,483	2,039	1,755

Fonte: Elaborado pelo autor

5.3.5.6 Análise dos experimentos com Busca Tabu para a instância r11889

Na observação do quadrante Q4 da tabela 49, constata-se o papel da diversificação dos mecanismos na geração de soluções de melhor qualidade e dos valores reduzidos para os desvios padrões neste quadrante. Observa-se que o uso do mecanismo de memória de maior duração beneficia o decaimento dos valores médios dos experimentos, especialmente em buscas mais longas, isto é, onde o processo de intensificação é claramente menor.

Tabela 49 – Resultados dos experimentos do BT para a instância r11889

			Lista Tabu					
			10	25	50	75	100	200
NBMax	10	FO	511699,794	457295,829	390216,138	391839,619	376270,541	345893,521
		DP	8771,996	5226,238	5017,065	5037,938	5912,823	5435,470
		GC	2,597	1,614	2,207	1,519	1,813	2,326
	25	FO	466720,288	406460,184	387749,360	392117,651	372795,811	343406,013
		DP	7334,176	5225,917	6647,132	6722,017	6390,785	5886,960
		GC	2,093	2,464	1,574	1,863	2,635	1,800
	50	FO	413509,152	398235,912	381909,618	366630,370	347667,547	336167,664
		DP	7088,728	6257,993	6001,437	5761,334	3973,343	3841,916
		GC	1,636	1,893	2,804	2,401	2,007	2,245
	75	FO	384948,600	359894,293	349283,061	345128,911	332622,978	328349,434
		DP	4949,339	4113,078	4989,758	4437,372	4276,581	3752,565
		GC	2,759	1,955	1,971	2,541	1,724	2,199
	100	FO	359197,844	342067,125	338990,420	330682,760	326457,002	324375,777
		DP	4105,118	4886,673	4842,720	4251,635	4663,671	5097,334
		GC	2,103	2,541	3,253	1,717	2,267	2,242
	200	FO	337235,804	331911,552	327955,771	324895,634	323148,398	320802,917
		DP	4335,889	4267,434	4685,082	5105,503	5078,046	3654,890
		GC	2,067	2,497	2,950	1,893	2,406	2,532

Fonte: Elaborado pelo autor

5.3.5.7 Análise dos experimentos com GRASP para a instância rl1889

Encontram-se na tabela 50, os resultados de melhor qualidade e os menores Grãos Computacionais no quadrante Q4, que se devem à diversificação introduzida pelos mecanismos. O decaimento notado de valores de FO a partir dos valores médios, diga-se, nos quadrantes do lado direito, em especial para o operador 2-opt, demonstra que a diversificação introduzida pelas Listas de Candidatos maiores beneficia a menor diversificação ocasionada por este operador.

Tabela 50 – Resultados dos experimentos do GRASP para a instância rl1889

			Lista de Candidatos α					
			0,1	0,2	0,3	0,5	0,7	0,9
Busca Local	2-opt	FO	415928,157	397767,208	382060,991	334134,270	352132,644	358166,520
		DP	6536,014	5682,389	6549,617	4296,012	4024,373	5628,331
		GC	1,136	2,367	1,359	2,036	1,361	2,172
	3-opt	FO	345743,646	342670,470	346232,212	324870,034	329063,406	334744,360
		DP	4939,195	3916,234	5935,409	4176,900	5641,087	4782,062
		GC	2,084	1,380	1,649	1,251	1,503	1,635

Fonte: Elaborado pelo autor

5.3.5.8 Análise dos experimentos com Otimização por Colônia de Formigas para a instância rl1889

No quadrante inferior direito (Q4) da tabela 51, encontram-se os melhores resultados médios de FO e de desvios padrões, indicando uma maior intensificação por parte da Atração do Feromônio, aliado a uma maior diversificação dada pelas elevadas taxas de Evaporação do Feromônio. Noutra ótica, entende-se que a alta atração do feromônio realiza um prolongamento da busca, pela intensificação, depreciando os valores médios de Grão Computacional visto nos quadrantes inferiores.

Tabela 51 – Resultados dos experimentos do ACO para a instância rl1889

			Evaporação do Feromônio					
			0,01	0,02	0,03	0,05	0,10	0,15
Atração do Feromônio	0,5	FO	428098,396	364819,906	360534,902	339426,078	329554,558	358607,761
		DP	6115,691	4169,370	5150,499	5333,838	5178,714	6147,562
		GC	2,176	2,009	1,910	1,260	2,176	1,371
	1,0	FO	399971,503	371041,963	355051,662	343990,985	328384,708	359292,930
		DP	6285,266	5830,659	5072,167	5405,573	5629,452	4106,205
		GC	1,224	1,413	1,449	1,808	1,311	2,134
	2,0	FO	393847,353	350615,552	343913,082	338890,264	325227,994	350538,516
		DP	6189,030	4507,914	5404,348	5809,547	5575,337	5007,693
		GC	2,245	1,675	1,887	1,455	1,377	1,830
	3,0	FO	374878,396	349140,688	344549,722	332155,221	323464,843	336131,942
		DP	6426,487	3990,179	5414,353	3796,060	4620,926	4801,885
		GC	1,878	1,358	1,565	1,752	1,425	1,986
	4,0	FO	356254,480	342114,072	333923,478	328275,065	322784,369	331436,330
		DP	4071,480	4887,344	5724,402	3751,715	5072,326	5681,766
		GC	1,726	1,697	1,911	1,332	2,254	2,295
	5,0	FO	347768,634	331765,106	329120,736	323200,224	320482,350	326689,978
		DP	4968,123	3791,601	5642,070	3693,717	3662,655	5600,400
		GC	1,735	2,079	2,474	1,417	1,865	1,891

Fonte: Elaborado pelo autor

5.3.5.9 Análise dos experimentos consolidados para a instância rl1889

Ao observarmos as duas seções da Tabela 52, têm-se na primeira a consolidação da melhor média de 100 repetições dos experimentos, dentre as parametrizações analisadas, incluindo a melhor execução média da hiper-heurística. Em cada caso apresentado, calcula-se o Gap percentual relativo à hiper-heurística e à melhor solução conhecida para o problema.

Na segunda parte da tabela, temos acesso à consolidação da média das repetições dos experimentos para as 36 parametrizações testadas por cada meta-heurística, além da melhor execução média da hiper-heurística. Em cada valor médio obtido pela meta-heurística, calcula-se também o Gap percentual em relação a hiper-heurística e ao BKS.

Para a instância rl1889, a meta-heurística Otimização por Colônia de Formigas apresentou melhor resultado nos dois casos, sobre as 288 parametrizações testadas, distribuídas nos conjuntos paramétricos aplicados a cada uma das 8 meta-heurísticas, e no resultado médio dos 36 parâmetros.

O resultado alcançado pela hiper-heurística sobre o melhor valor obtido pela Otimização por Colônia de Formigas apresenta uma melhora de 0,123% em relação a FO, 12,187% melhor se comparado ao DP e 40,971% inferior sobre o GC. Analisando os mesmos valores sobre o resultado médio da Otimização por Colônia de Formigas, a hiper-heurística fornece resultados 7,785% melhores, sendo o DP 36,561% melhor e o GC 37,436% menor.

Numa análise sobre a média dos melhores resultados, a hiper-heurística obteve uma melhora percentual de 1,432% nos resultados da FO, com redução de 22,227% do desvio padrão e 37,275% no GC. Já sobre a média das médias, a hiper-heurística obteve uma melhoria de 9,927% sobre a FO, 36,328% sobre o desvio padrão e uma melhora de 41,574% sobre o GC.

No comparativo da hiper-heurística sobre o BKS, a média dos resultados é 1,122% superior, não obstante, salienta-se o valor de GC obtido pela hiper-heurística, que assim como se demonstrou nas demais consolidações dos experimentos anteriores, apresenta o melhor GC em comparação aos obtidos pelas meta-heurísticas.

Tabela 52 – Resultados dos experimentos consolidados para a instância rl1889

BKS	316536	Melhor	GAP HH	GAP BKS	Média	GAP HH	GAP BKS
AG	FO	331.050,518	3,312%	4,585%	361.296,421	11,406%	14,141%
	DP	5.675,152	43,327%	-	5.103,186	36,975%	-
	GC	1,914	42,488%	-	1,655	33,488%	-
SA	FO	323.091,493	0,930%	2,071%	361.587,493	11,477%	14,233%
	DP	5.077,152	36,651%	-	5.184,640	37,965%	-
	GC	1,422	22,558%	-	1,788	38,441%	-
SS	FO	325.241,946	1,585%	2,750%	356.609,248	10,241%	12,660%
	DP	3.717,051	13,472%	-	5.147,440	37,516%	-
	GC	1,663	33,801%	-	2,090	47,326%	-
VNS	FO	323.149,334	0,947%	2,089%	347.664,300	7,932%	9,834%
	DP	4.154,777	22,588%	-	4.872,350	33,989%	-
	GC	1,620	32,019%	-	1,874	41,237%	-
ILS	FO	329.517,134	2,862%	4,101%	348.418,152	8,131%	10,072%
	DP	3.765,910	14,594%	-	4.684,776	31,346%	-
	GC	2,611	57,834%	-	2,249	51,051%	-
BT	FO	320.802,917	0,223%	1,348%	365.959,258	12,535%	15,614%
	DP	3.654,890	12,000%	-	5.222,943	38,420%	-
	GC	2,532	56,525%	-	2,198	49,899%	-
GRASP	FO	324.870,034	1,472%	2,633%	355.292,827	9,909%	12,244%
	DP	4.176,900	22,998%	-	5.175,635	37,857%	-
	GC	1,251	12,000%	-	1,661	33,716%	-
ACO	FO	320.482,350	0,123%	1,247%	347.109,549	7,785%	9,659%
	DP	3.662,655	12,187%	-	5.069,899	36,561%	-
	GC	1,865	40,971%	-	1,760	37,436%	-
MÉDIA	FO	324.775,716	1,432%	2,603%	355.492,156	9,927%	12,307%
	DP	4.235,561	22,227%	-	5.057,609	36,328%	-
	GC	1,860	37,275%	-	1,909	41,574%	-
HH	FO	320.087,715	-	1,122%	320.087,715	-	1,122%
	DP	3.216,304	-	-	3.216,304	-	-
	GC	1,101	-	-	1,101	-	-

Fonte: Elaborado pelo autor

5.4 Análise de Frequência

As frequências relativas das meta-heurísticas foram avaliadas por cada componente heurístico, com o objetivo de comparar o desempenho de cada um desses componentes na hiper-heurística. Assim, pode-se comparar, lado a lado, a frequência relativa com a qual cada mecanismo gerou boas soluções em meta-heurísticas e hiper-heurísticas. A figura 29 mostra as frequências relativas com as quais os mecanismos heurísticos produziram soluções de qualidade nas meta-heurísticas e hiper-heurísticas, enquanto a tabela 53 resume a melhor parametrização mostrada no histograma com cores mais claras, indicando os mecanismos heurísticos mais frequentemente usados pelo modelo. A partir dessa observação, podemos inferir que esses mecanismos representam o parâmetro que gera os melhores desempenhos, além de refletir a

matriz de transições obtida pela hiper-heurística.

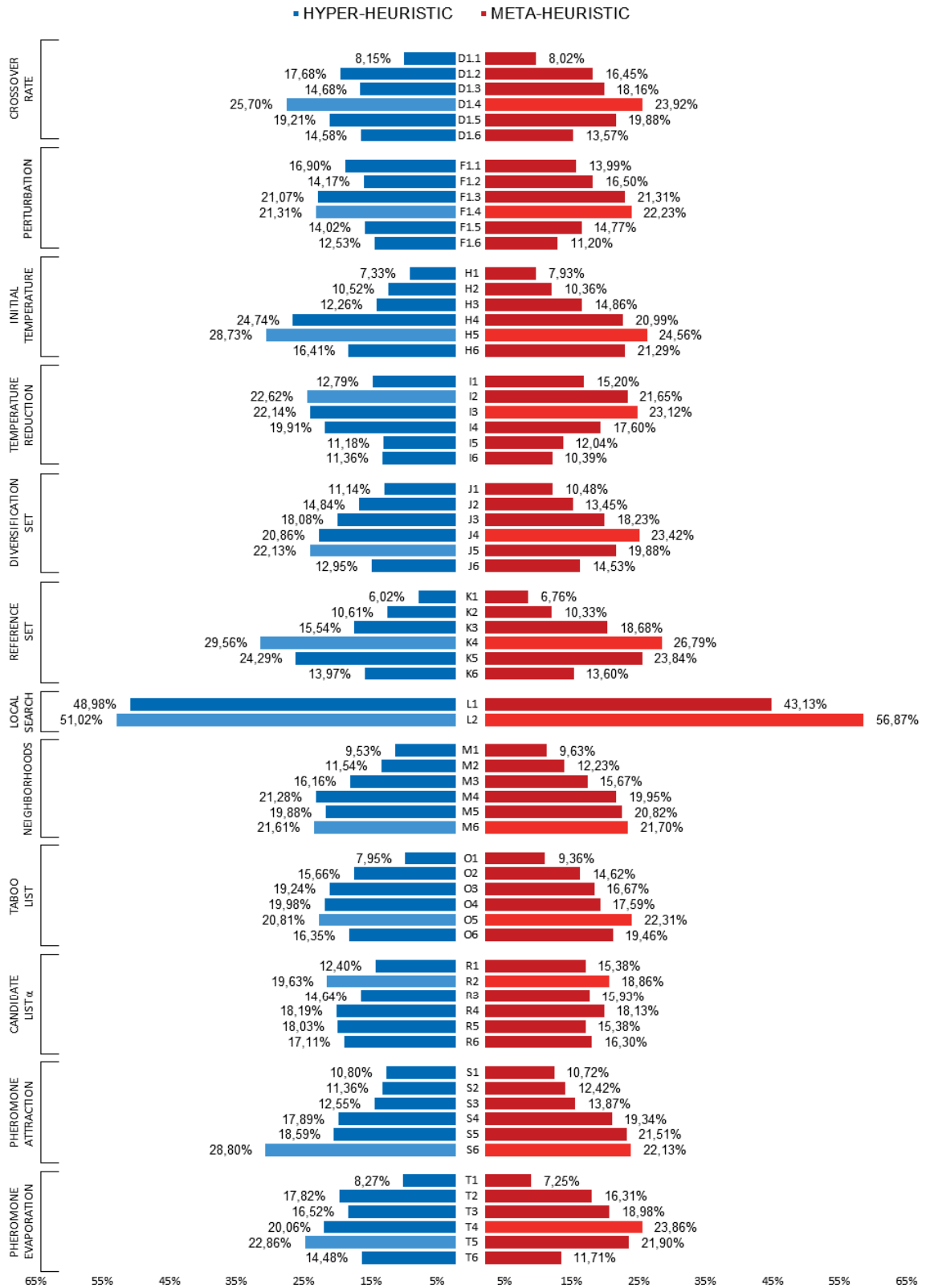


Figura 29 – Frequências dos mecanismos heurísticos

Tabela 53 – Melhores parametrizações

Algoritmo Genético				Arrefecimento Simulado			
Taxa de cruzamento		Taxa de mutação		Temperatura Inicial		Redução da Temperatura	
HH	MH	HH	MH	HH	MH	HH	MH
25%	25%	10%	10%	5000	5000	0,010	0,050
Busca Dispersa				Busca em Vizinhança Variável			
Conjunto de Referência		Conjunto de Diversificação		Vizinhanças		Perturbação	
HH	MH	HH	MH	HH	MH	HH	MH
15	15	150	100	100	100	10%	10%
Busca Local Iterada				Busca Tabu			
Busca Local		Perturbação		NBMax		Lista Tabu	
HH	MH	HH	MH	HH	MH	HH	MH
3-opt	3-opt	10%	10%	100	100	100	100
GRASP				Otimização por Colônia de Formigas			
Busca Local		Lista de Candidatos		Atração do Feromônio		Evaporação do Feromônio	
HH	MH	HH	MH	HH	MH	HH	MH
3-opt	3-opt	0,2	0,2	5	5	0,10	0,05

Para obter resultados mais conclusivos, foi realizado um teste dos postos sinalizados de Wilcoxon emparelhado, com nível de significância de $\alpha = 0,01$, para verificar se as diferenças no uso de mecanismos heurísticos por meta-heurística e hiper-heurísticas são estatisticamente significantes. Na avaliação, obtivemos um *valor-p* de 0,8695. Como o *valor-p* calculado foi maior que o nível de significância ($p \geq \alpha$), a hipótese nula H_0 não foi rejeitada. Esse valor p indica que a diferença na frequência de uso de cada mecanismo heurístico não é estatisticamente significativa. Isso corrobora empiricamente que o uso de mecanismos heurísticos por meta-heurística e hiper-heurística é semelhante, e podemos entender que as hiper-heurísticas aprenderam o comportamento apresentado pelas meta-heurísticas.

Esse entendimento da frequência de ação exibe o comportamento dos mecanismos heurísticos em suas frequências operacionais no processo de otimização. Observando o contexto sobre as frequências aplicadas às meta-heurísticas, observou-se que o desempenho desses mecanismos está fortemente relacionado aos experimentos individuais realizados com cada meta-heurística, independentemente da instância do teste.

Outro ponto que cabe observar é a ausência de sinergia entre os algoritmos que levou a hiper-heurística a gastar tempo de execução para aprender. Nossa conclusão sobre por que a hiper-heurística realiza menos avaliações, em média, é porque a cadeia de operações gerada pela SAN contém várias operações individuais aplicadas às soluções antes de executar a avaliação final da solução, o que alteraria o grão computacional.

5.5 Considerações sobre o capítulo

Durante o trabalho, foram alcançados alguns resultados que contribuíram para chegarmos até este ponto da pesquisa. Desta forma, este capítulo apresentou o projeto dos experimentos e as avaliações do modelo H2-SLAN usando cinco instâncias do Problema do Caixeiro Viajante. A avaliação engloba a análise de três diferentes perspectivas:

i) execução do modelo com as meta-heurísticas

Dada a ausência de dados da literatura para corroboração e validação do grão computacional, e conseqüentemente do funcionamento dos algoritmos implementados, tomou-se por base a padronização das instâncias de teste já consolidadas na literatura.

Nosso processo tem início por meio da resolução das instâncias com o uso das meta-heurísticas canônicas, apresentadas no Capítulo 3. Este processo verificou a viabilidade de operação das meta-heurísticas no problema selecionado, atingindo resultados quantitativos satisfatórios, na ordem de 5% acima da melhor solução conhecida, representado por um Grão Computacional médio de 1,783.

Por meio destes experimentos, pode-se compreender os processos agenciados pelas meta-heurísticas, conduzindo uma análise sobre os espectros dos movimentos de intensificação e diversificação realizados por cada um dos mecanismos nelas operantes.

ii) execução do modelo com a hiper-heurística

Após validar o modelo em termos quantitativos e de funcionalidade, abrimos nosso leque de operações para a hiper-heurística atuar sob todo o conjunto de mecanismos heurísticos mapeados, em prol da obtenção de melhores resultados quantitativos e qualitativos. Assim, ao final de cada experimento com uma instância da TSPLib verificou-se o conjunto de resultados da hiper-heurística em relação ao ótimo global, na ordem de 2% acima da melhor solução, representado por um Grão Computacional médio de 0,982.

Não obstante, se considerarmos a qualidade média, a hiper-heurística obteve resultados 3% melhores quando comparada às meta-heurísticas. Já em observação ao Grão Computacional, a hiper-heurística apresentou um esforço computacional 45% menor.

iii) análise de frequência dos mecanismos heurísticos

Essa compreensão pode ser vista na análise conjunta sobre a frequência com que são acionados os mecanismos heurísticos pelo metaespaço. Através desse processo, foi possível visualizar o comportamento dos mecanismos heurísticos observando suas frequências de atuação, que representam a quantidade de vezes que cada mecanismo foi acionado durante o processo de otimização. Ao observar o contexto sobre as frequências aplicadas as meta-heurísticas podemos notar que a maior presença destes mecanismos se relaciona fortemente aos experimentos expostos neste capítulo, com os melhores resultados.

Este processo serviu de base para capturar a essência das meta-heurísticas através do metaespaço e validar o funcionamento do sistema como um todo. Como segundo passo, iniciamos a execução da hiper-heurística que demonstrou frequência similar aos comportamentos apresentados pelas meta-heurísticas. Desta observação, pode-se afirmar que o comportamento esperado pelo modelo, quando da atuação da hiper-heurística, vai de encontro aos comportamentos efetivados pelas meta-heurísticas. Essa observação nos dá indícios de que o modelo está capturando a essência das características do problema para a composição do metaespaço, e conseqüentemente para a obtenção de melhores resultados.

Considerando estas três perspectivas, podemos tirar as seguintes conclusões em poucas palavras: i) as meta-heurísticas obtiveram resultados de qualidade; ii) a hiper-heurística melhora o desempenho da busca em termos qualitativos, resultando em melhores soluções, e quantitativos apresentando valores menores de grão computacional quando comparados à meta-heurística; iii) o H2-LAN apresentou análise de frequência similar às meta-heurísticas, demonstrando sua capacidade de obtenção, armazenamento e recuperação do conhecimento acerca das operações dos mecanismos heurísticos.

6 CONSIDERAÇÕES FINAIS

*“Não importa a incompreensão.
Cada criatura vê o horizonte que os próprios olhos podem abranger”*

Chico Xavier

Desde o início do processo de doutoramento, este projeto se reconfigurou de diversas formas, rerepresentando objetos e horizontes, e se encontra agora na etapa mais consolidada de todas, por conta do esforço em estabelecer textualmente todos os movimentos que nos trouxeram até aqui. Ao longo dos capítulos anteriores, a investigação se debruçou em como poderiam os mecanismos heurísticos adequarem suas estruturas para a resolução de problemas de Otimização Combinatória. Em diferentes dimensões, foram delineadas sete perspectivas modelares que auxiliaram a visualização dos mecanismos heurísticos, são elas: *navegabilidade*, *completude*, *gênese*, *universalidade*, *persistência*, *contiguidade* e *aptidão*. Estas perspectivas serviram para compreender e mapear as estruturas que norteiam o funcionamento dos mecanismos heurísticos, para assim modelá-los sob a vista de uma arquitetura, estabelecendo o modelo H2-SLAN.

Com isso, alcançou-se o objetivo principal e finaliza-se a pesquisa. O texto aqui não toma mais a forma de sua estrutura, mas busca sintetizar os aprendizados contidos no texto. Em cada capítulo, já foram apresentadas considerações com um movimento nesse sentido, de sistematização dos seus resultados. Por fim, neste momento, cruza-se as dimensões abertas no decorrer do trabalho, na exposição dos pensamentos que, em conjunto, formam a tese.

Apesar dos méritos dos muitos algoritmos mencionados durante o trabalho, sob quaisquer que sejam suas estratégias de diversificação e intensificação, poderíamos ter nos perguntado se precisávamos de outro algoritmo novo. Já sabendo da existência do teorema *No Free Lunch*, que logicamente prova que não há algoritmo para resolver todos os problemas de otimização. No escopo deste trabalho, isso pode ser declarado como nenhum mecanismo heurístico, ou sua combinação, é capaz de resolver todos os problemas de otimização combinatória. Em outras palavras, há sempre espaço para as melhorias das técnicas atuais para resolver quantitativamente e qualitativamente a corrente de novos problemas surgentes. Isso foi a tentativa, não de propor outro algoritmo, mas de empreender uma nova forma de uso dos mecanismos heurísticos pré-existentes.

6.1 Contribuições

A importância do retorno analítico da tese está no fato de apresentar como parte da sua intenção, reunir os atravessamentos construídos ao longo da investigação, conseguindo sintetizar o caminho das linhas divergentes que saíram do nosso objeto. Esta seção apresenta as considerações de passagem pela pesquisa, e se constitui a partir da retomada dos aspectos de maior importância construídos e expostos até aqui, aspectos resultantes dos processos técnicos e analíticos. Desse modo, chegando ao fim do trabalho, munido dos processos alcançados com a realização de todas as fases que a tese abrigou, aproveita-se para elencá-las.

Os desenvolvimentos deste estudo contribuirão em benefício da sociedade, considerando que melhores resultados em Pesquisa Operacional desempenham um importante papel não somente nas ciências e nas tecnologias atuais, mas na forma como a sociedade consome e produz. Para o campo da Computação, o estudo ajuda na descoberta de pontos sensíveis e talvez carentes de análises epistêmicas, num terreno onde muitos não se propõem explorar. Dessa forma, novos olhares podem ser lançados sobre a Computação, visando novas teorias e modelos que discutam os temas de maneira mais profunda e façam interface com outros campos, estabelecendo um diálogo dentro da própria ciência. A maior demanda por soluções computacionais com embasamentos científicos justifica a necessidade de abordagens mais profundas e que fazem a ciência e a sociedade evoluírem qualitativamente.

De maneira mais objetiva, podemos citar como contribuições à Otimização Combinatória, alguns resultados da tese:

1. Representação homogênea de meta-heurísticas por meio de um *template* de alto-nível (Veja seção 2.5);
2. Definição de metaespaço como um espaço de busca de mecanismos heurísticos, livre das definições impostas pelo domínio do problema (Veja seção 3.9);
3. Ressignificação da arquitetura hiper-heurística, apresentando o modelo H2-SLAN como uma alternativa livre das características do problema, através do conceito de metaespaço (Veja seção 4.1);
4. Representação dos mecanismos heurísticos através de autômatos, delimitando um alfabeto de otimização heurística (Veja seção 4.2.3);
5. Mensuração de desempenho através do conceito de Grão Computacional heurístico (Veja seção 5.2);
6. Processo de análise constituído através de lentes espectrais, para compreensão dos processos de intensificação e diversificação (Veja seção 5.2).
7. Resultados 3% melhores com 45% menos esforço computacional em comparação às meta-heurísticas (Veja seção 5.5).

6.2 Trabalhos Futuros

Concluo aqui uma tese que deixa questões em aberto e que provavelmente serão assunto de pesquisas que ocuparão ainda vários anos de minha vida e de minha atividade acadêmica. Estou ciente de que a abordagem, com seus resultados, foi mais do que suficiente para a constituição da tese de doutorado. Contudo, corri o risco da exploração de um meio que está em plena construção, quiçá dando um passo rumo ao período científico das meta-heurísticas, como descrevem Sorensen, Savaux e Glover (2017). A própria formulação das perspectivas modelares, mecanismos heurísticos e do próprio modelo, dessa forma, só foram possíveis por uma abordagem epistêmica e tecnicista, mas que dão continuidade em trabalhos nas seguintes direções:

- Como o trabalho considerou apenas o tempo computacional, dado sob a quantidade de avaliações da Função Objetivo, pode-se considerar como um ponto negativo o tempo de execução da hiper-heurística que é superior em comparação às meta-heurísticas. Isso dá margem para a medição e conseqüente redução destes tempos;
- Por serem as meta-heurísticas um domínio heterogêneo, cabe a ampliação do conjunto de algoritmos e o conseqüente aprofundamento e análise considerando valores fracionários menores, o que permitirá a expansão dos limites do metaespaço inserindo novas estruturas de movimentação, e conseqüentemente melhores resultados;
- Como o modelo consolida o metaespaço através das redes de autômatos estocásticos de aprendizagem, pode-se analisar a viabilidade de outras formas para representação do metaespaço, que sejam computacionalmente melhores ou contribuam para uma visão mais apurada sobre a busca heurística;
- Aplicação do modelo H2-SLAN em novas classes de problemas, para demonstrar abrangências e limitações da abordagem desenvolvida;
- Estabelecer uma gramática do processo de otimização, objetivando melhorar ainda mais os resultados quantitativos, qualitativos e temporais;
- Consolidar uma análise epistêmica mais profunda, observando as contribuições individuais de cada mecanismo heurístico ao processo de busca. O que pode contribuir para a redução do uso demasiado de novas meta-heurísticas, cujo único propósito é obter melhores resultados sem uma profunda compreensão dos impactos gerados.

6.3 Epílogo

Durante o trabalho, foram obtidos resultados satisfatórios para a conclusão da proposta lançada no início desse percurso, sobre o desenvolvimento de uma forma de resolução de problemas de Otimização Combinatória adequando os mecanismos de busca durante o processo

de solução. Sob a ótica quantitativa e qualitativa, os números cumpriram seu papel, mostrando uma perspectiva já consolidada no campo da Computação. Para exemplificar o pensamento, acionamos Fisher (1963) e Crowston et al. (1963), corroborando suas duas conclusões, de que uma combinação aleatória de mecanismos heurísticos é melhor do que qualquer regra de agendamento separada e que o aprendizado sobre eles é possível.

Esta tese nos aproximou de uma compreensão sobre as metáforas aplicadas aos algoritmos computacionais. Em vez de resumir, concentrar-se-á aqui em méritos e limitações, delineando alguns tópicos que não são conclusões, mas observações deste processo de passagem que a tese abrigou.

O trabalho iniciou de uma inquietação que incitada pelo trabalho de Sörensen (2015), guiou o olhar sobre o que era desenvolvido ao longo destes quatro anos de curso. Não foi necessária uma extensa revisão, mas o uso de trabalhos fundamentais para a área, invocou saberes canônicos que guiaram a seleção empírica das meta-heurísticas aplicadas durante a pesquisa. Neste período, flinando por diversos terrenos, percebeu-se que a grande maioria dos trabalhos reiteravam usos que encerravam um potencial instigante, mostrando um grau de ociosidade analítica. Como resultante deste flinar, procurou-se entender muito mais do que o funcionamento do meio, chegando à compreensão do modo em que eram produzidos sentidos de movimentação.

Este flinar permitiu uma expansão, fazendo com que fossem percorridos outros caminhos teóricos além do campo da Computação. Retoma-se aqui o início da pesquisa, onde nos perguntamos: o que tornaria o campo da Computação tão suscetível a este tipo de pesquisa? Entendeu-se neste percurso, que esta ociosidade analítica pode se originar em duas vertentes. A primeira delas, versa sobre a possível incompreensão das técnicas atuais, fazendo com que sejam criadas novas formas de solução pelo simples fato de apresentarem bons resultados, mas que carecem de diálogo com métodos consolidados e discussões epistêmicas. Em segundo, pensa-se que esta incompreensão seja o *zeitgeist*¹ onde estas pesquisas ocorrem, ou seja, um espaço onde é de maior interesse a presença de hipóteses corroboradas do que submetidas a testes de falseamento.

Neste momento de reflexão, retomo o hiato sobre meu saber computacional, agora com certa tessitura. As analogias e metáforas empregadas em meta-heurísticas nos convidaram a refletir sobre a questão chave dos processos de otimização por elas empregados. Entende-se assim que as metáforas têm seu lugar para estimular ideias nas fases iniciais de uma investigação, mas cabe o cuidado para que não apenas vistam as descrições algorítmicas de maneira alegórica. Como diz Laguna (2000, p. 16), cabe à prudência determinar quando o simbolismo obscurece em vez de iluminar o caminho para uma melhor compreensão. No âmbito do design meta-heurístico, ainda há muito o que aprender. A questão que merece uma reflexão mais profunda se dá sobre saber quando analogias limitam ou aprimoraram o acesso a novas descobertas

¹Termo alemão que significa 'espírito do tempo'; o espírito de uma época; o nível cultural, o sentimento, o pensamento, a atmosfera (fig.) ou tudo aquilo que caracteriza um dado período (AULETE, 2020).

reais.

REFERÊNCIAS

- ABIDIN, D. A hybrid genetic - differential evolution algorithm (HybGADE) for a constrained sequencing problem. In: **2018 International Conference on Artificial Intelligence and Data Processing (IDAP)**. [S.l.]: IEEE, 2018.
- AISEARCH. **Metaheuristic Publications Per year**. 2019. Disponível em: <<https://aisearch.github.io>>. Acesso em: 08 de Julho de 2019.
- AKTER, S. et al. A new crossover technique to improve genetic algorithm and its application to TSP. In: **2019 International Conference on Electrical, Computer and Communication Engineering (ECCE)**. [S.l.]: IEEE, 2019.
- ALVES, D. R. de S. et al. Siaco: a novel algorithm based on ant colony optimization and game theory for travelling salesman problem. In: ACM. **Proceedings of the 2nd International Conference on Machine Learning and Soft Computing**. [S.l.], 2018. p. 62–66.
- AMUTHAN, A.; THILAK, K. D. Survey on tabu search meta-heuristic optimization. In: IEEE. **2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPE5)**. [S.l.], 2016. p. 1539–1543.
- ANDRADE, B. A. R. de; FERREIRA, N. R. Simulated annealing and tabu search applied on network reconfiguration in distribution systems. In: IEEE. **2018 Simposio Brasileiro de Sistemas Eletricos (SBSE)**. [S.l.], 2018. p. 1–6.
- ANWAR, K. et al. Harmony search-based hyper-heuristic for examination timetabling. In: IEEE. **2013 IEEE 9th International Colloquium on Signal Processing and its Applications**. [S.l.], 2013. p. 176–181.
- APPLEGATE, D.; COOK, W.; ROHE, A. Chained lin-kernighan for large traveling salesman problems. **INFORMS Journal on Computing**, INFORMS, v. 15, n. 1, p. 82–92, 2003.
- AQUINO, R. de F.; ARROYO, J. E. C. A hybrid multi-objective iterated local search heuristic for vehicle routing problem with time windows. In: IEEE. **2014 14th International Conference on Hybrid Intelligent Systems**. [S.l.], 2014. p. 117–122.
- ARANHA, C.; CAMPELO, F. **Evolutionary Computation Bestiary**. 2019. Disponível em: <<https://github.com/fcampelo/EC-Bestiary>>. Acesso em: 08 de Julho de 2019.
- ARIAS, N. B. et al. Grasp algorithm for charging coordination of plug-in electric vehicles in electrical distribution systems. In: IEEE. **2016 Ieee Pes Transmission & Distribution Conference And Exposition-latin America (pes T&d-la)**. [S.l.], 2016. p. 1–6.
- Atas, P. K.; Tufan, K.; Sevkli, A. Z. A variable neighborhood search based feature selection model for early prediction of the alzheimer's disease. In: **2016 Electric Electronics, Computer Science, Biomedical Engineerings' Meeting (EBBT)**. [S.l.: s.n.], 2016. p. 1–4.
- AULETE, C. **Dicionário Eletrônico Aulete Digital**. 2017. Disponível em: <<http://www.aulete.com.br/heur%C3%ADstica>>. Acesso em: 02 de Fevereiro de 2017.
- AULETE, C. **Dicionário Eletrônico Aulete Digital**. 2020. Disponível em: <<http://www.aulete.com.br/zeitgeist>>. Acesso em: 07 de Junho de 2020.

- BAI, R. et al. **A simulated annealing hyper-heuristic methodology for flexible decision support.** School of CSiT, University of Nottingham. [S.l.], 2007.
- BALAS, E.; VAZACOPOULOS, A. Guided Local Search with Shifting Bottleneck for Job Shop Scheduling. **Management Science**, v. 44, n. 2, p. 262–275, 1998.
- BAN, H. B.; NGUYEN, D. N. An effective grasp+ vnd metaheuristic for the k-minimum latency problem. In: IEEE. **2016 IEEE RIVF International Conference on Computing & Communication Technologies, Research, Innovation, and Vision for the Future (RIVF).** [S.l.], 2016. p. 31–36.
- BANDARU, S.; DEB, K. Metaheuristic Techniques. **Decision Sciences: Theory and Practice**, p. 693–750, 2016.
- BARNHART, C. et al. Branch-and-price: Column generation for solving huge integer programs. **Operations research, INFORMS**, v. 46, n. 3, p. 316–329, 1998.
- BARR, R. S. et al. Designing and reporting on computational experiments with heuristic methods. **Journal of heuristics**, Springer, v. 1, n. 1, p. 9–32, 1995.
- BATTITI, R.; BRUNATO, M.; MASCIA, F. **Reactive Search and Intelligent Optimization.** 1. ed. [S.l.]: Springer US, 2009.
- BAUM, E. B. Iterated descent: A better algorithm for local search in combinatorial optimization problems. **Manuscript**, 1986.
- BAXTER, J. Local optima avoidance in depot location. **Journal of the Operational Research Society**, JSTOR, p. 815–819, 1981.
- BEIRIGO, B. A.; SANTOS, A. G. dos. A parallel heuristic for the travel planning problem. In: IEEE. **2015 15th International Conference on Intelligent Systems Design and Applications (ISDA).** [S.l.], 2015. p. 283–288.
- BELFIORE, P. P.; SILVA, V. F. d.; RONCONI, D. P. Scatter search para programação de tarefas em uma máquina com penalidades de adiamento e atraso e data de entrega comum. In: **Encontro Nacional de Engenharia de Produção, 26.** [S.l.]: ABEPRO, 2006.
- BELFIORE, P. P.; YOSHIZAKI, H. T. Y. Scatter search para problemas de roteirização de veículos com frota heterogênea, janelas de tempo e entregas fracionadas. **Production**, SciELO Brasil, v. 16, n. 3, p. 455–469, 2006.
- BENCHMARKSGAME. **The Computer Language Benchmarks Game.** 2017. Disponível em: <<http://benchmarksgame.alioth.debian.org/>>. Acesso em: 3 de Dezembro de 2017.
- BENJAMIN, W. et al. **Origem do drama barroco alemão.** [S.l.]: Brasiliense São Paulo, 1984.
- BIRATTARI, M. et al. **Classification of Metaheuristics and Design of Experiments for the Analysis of Components.** Darmstadt, 2001. 1–12 p.
- BLUM, C.; ROLI, A. Metaheuristics in combinatorial optimization: Overview and conceptual comparison. **ACM Computing Surveys**, ACM, New York, NY, USA, v. 35, n. 3, p. 268–308, 2003.

BOUSSAÏD, I.; LEPAGNOT, J.; SIARRY, P. A survey on optimization metaheuristics. **Information Sciences**, v. 237, p. 82–117, 7 2013.

BRENNER, L. Agregação de redes de autômatos estocásticos. 2004.

BRENNER, L.; FERNANDES, P.; ALEGRETTI, C. G. P. **Redes de Autômatos Estocásticos: Um formalismo para avaliação de desempenho e confiabilidade de sistemas**. Porto Alegre, RS, Brasil, 2002.

BRENNER, L.; FERNANDES, P.; SALES, A. The need for and the advantages of generalized tensor algebra for kronecker structured representations. **International Journal of Simulation: Systems, Science & Technology**, Nottingham, v. 6, n. 3-4, p. 52–60, 2005.

Brito, J.; Expósito, A.; Moreno, J. A. Variable neighbourhood search for closeopen vehicle routing problem with time windows. **IMA Journal of Management Mathematics**, v. 27, n. 1, p. 25–38, Jan 2016. ISSN 1471-678X.

BROWNLEE, A.; WOODWARD, J. R. **Why we fell out of love with algorithms inspired by nature**. 2015. Disponível em: <<http://theconversation.com/why-we-fell-out-of-love-with-algorithms-inspired-by-nature-42718>>. Acesso em: 3 de Novembro de 2017.

BROWNLEE, J. **Clever Algorithms: Nature-inspired Programming Recipes**. [S.l.]: Lulu Enterprises Incorporated, 2011.

BURKE, E.; SOUBEIGA, E. Scheduling nurses using a tabu-search hyperheuristic. In: **Proceedings of the 1st Multidisciplinary International Conference on Scheduling: Theory and Applications (MISTA 2003), Nottingham, UK**. [S.l.: s.n.], 2003. p. 180–197.

BURKE, E. K. et al. Hyper-heuristics: a survey of the state of the art. **Journal of the Operational Research Society**, v. 64, n. 12, p. 1695–1724, 1 2013.

BURKE, E. K. et al. Hyper-heuristics: An emerging direction in modern search technology. In: GLOVER, F.; KOCHENBERGER, G. A. (Ed.). **Handbook of Metaheuristics**. New York: Kluwer Academic Publishers, 2003. v. 57, n. July, p. 457–474.

BURKE, E. K. et al. A classification of hyper-heuristic approaches: Revisited. In: **Handbook of Metaheuristics**. [S.l.]: Springer, 2019. p. 453–477.

BURKE, E. K.; SILVA, J. D. L.; SOUBEIGA, E. Multi-objective hyper-heuristic approaches for space allocation and timetabling. In: **Metaheuristics: Progress as Real Problem Solvers**. [S.l.]: Springer, 2005. p. 129–158.

BUSH, R. R.; MOSTELLER, F. **Stochastic models for learning**. [S.l.]: Wiley, 1955. (Wiley publications in statistics).

BYTYÇI, E.; SYLEJMANI, K.; DIKA, A. Iterated local search algorithm for planning the sequence of arrivals and departures at airport runways. In: IEEE. **2016 39th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)**. [S.l.], 2016. p. 1254–1258.

ČERNÝ, V. Thermodynamical approach to the traveling salesman problem: An efficient simulation algorithm. **Journal of optimization theory and applications**, Springer, v. 45, n. 1, p. 41–51, 1985.

- CHEN, H. et al. Ant colony optimization with tabu table to solve tsp problem. In: IEEE. **2018 37th Chinese Control Conference (CCC)**. [S.l.], 2018. p. 2523–2527.
- CHEN, P. et al. A new hybrid iterated local search for the open vehicle routing problem. In: IEEE. **2008 IEEE Pacific-Asia Workshop on Computational Intelligence and Industrial Application**. [S.l.], 2008. v. 1, p. 891–895.
- CHEN, P.-C.; KENDALL, G.; BERGHE, G. V. An ant based hyper-heuristic for the travelling tournament problem. In: IEEE. **2007 IEEE Symposium on Computational Intelligence in Scheduling**. [S.l.], 2007. p. 19–26.
- CHEN, S.-M.; CHIEN, C.-Y. Solving the traveling salesman problem based on the genetic simulated annealing ant colony system with particle swarm optimization techniques. **Expert Systems with Applications**, Elsevier, v. 38, n. 12, p. 14439–14450, 2011.
- COELLO, C. A. C.; SEGURA, C.; MIRANDA, G. History and philosophy of evolutionary computation. In: **Handbook on Computational Intelligence**. [S.l.]: WORLD SCIENTIFIC, 2016. p. 509–545.
- COSTA, D. A tabu search algorithm for computing an operational timetable. **European Journal of Operational Research**, v. 76, p. 98–110, 1994.
- COTTA, C.; SEVAUX, M.; SÖRENSEN, K. **Adaptive and Multilevel Metaheuristics**. [S.l.]: Springer, 2008. (Studies in Computational Intelligence).
- COWLING, P.; KENDALL, G.; SOUBEIGA, E. A hyperheuristic approach to scheduling a sales summit. **Practice and Theory of Automated Timetabling III**, p. 176–190, 2001.
- CROWSTON, W. B. et al. **Probabilistic and parametric learning combinations of local job shop scheduling rules**. [S.l.], 1963.
- DANTZIG, G. B. **A History of Scientific Computing**. [S.l.], 1945.
- DANTZIG, G. B. **Maximization of a linear function of variables subject to linear inequalities**. [S.l.]: Wiley, 1951. 339–347 p.
- DAO, S. D.; ABHARY, K.; MARIAN, R. A bibliometric analysis of genetic algorithms throughout the history. **Computers & Industrial Engineering**, Elsevier BV, v. 110, p. 395–403, 2017.
- DARWIN, C.; BEER, G. **The Origin of Species**. [S.l.]: Oxford University Press, Incorporated, 1998. (Oxford world's classics).
- DAVENDRA, D. **Traveling Salesman Problem: Theory and Applications**. [S.l.]: BoD–Books on Demand, 2010.
- DAVIS, L. **Handbook of Genetic Algorithms**. [S.l.]: Van Nostrand Reinhold, 1991. (VNR computer library). ISBN 9780442001735.
- DEAN, A. M.; VOSS, D. **Design and analysis of experiments**. [S.l.]: Springer, 1999. v. 1.
- DELAMARE, F. L. **Análise de padrões de mobilidade utilizando redes de autômatos estocásticos**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brasil, 2008.

- DENG, W. et al. A novel collaborative optimization algorithm in solving complex optimization problems. **Soft Computing**, Springer, v. 21, n. 15, p. 4387–4398, 2017.
- DOERR, B. et al. On the runtime analysis of selection hyper-heuristics with adaptive learning periods. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference**. [S.l.], 2018. p. 1015–1022.
- DORIGO, M. **Optimization, learning and natural algorithms**. 1992.
- DORIGO, M.; BIRATTARI, M.; STÜTZLE, T. Ant colony optimization: Artificial ants as a computational intelligence technique. **IEEE Computational Intelligence Magazine**, v. 1, p. 28–39, 1 2006.
- DORIGO, M.; BLUM, C. Ant colony optimization theory: A survey. **Theoretical computer science**, Elsevier, v. 344, n. 2-3, p. 243–278, 2005.
- DORIGO, M.; MANIEZZO, V.; COLORNI, A. The ant system: Optimization by a colony of cooperative agents. **IEEE TRANSACTIONS ON SYSTEMS, MAN AND CYBERNETICS-PART B: CYBERNETICS**, p. 29–41, 1996.
- DORIGO, M. et al. **Positive Feedback as a Search Strategy**. [S.l.], 1991.
- DORIGO, M.; STÜTZLE, T. **Ant Colony Optimization**. [S.l.]: BRADFORD BOOK, 2004. (A Bradford book).
- DRAKE, J. H. et al. Recent advances in selection hyper-heuristics. **European Journal of Operational Research**, Elsevier, 2019.
- DU, K.-L.; SWAMY, M. Simulated annealing. In: **Search and Optimization by Metaheuristics**. [S.l.]: Springer, 2016. p. 29–36.
- DUARTE, A. et al. Variable neighborhood descent. **Handbook of Heuristics**, Springer, p. 341–367, 2018.
- DUFLO, G. et al. A gp hyper-heuristic approach for generating tsp heuristics. In: **33rd IEEE International Parallel & Distributed Processing Symposium (IPDPS 2019)**. [S.l.: s.n.], 2019.
- DUMAN, E.; OZCELIK, M. H. Detecting credit card fraud by genetic algorithm and scatter search. **Expert Systems with Applications**, Elsevier, v. 38, n. 10, p. 13057–13063, 2011.
- EDMONDS, J. Matroids and the greedy algorithm. **Mathematical Programming**, v. 1, n. 1, p. 127–136, 1971.
- EL-SAYED, S. M.; EL-WAHED, W. F. A.; ISMAIL, N. A. A hybrid genetic scatter search algorithm for solving optimization problems. **Constraints**, v. 500, p. 1, 2008.
- ERSOY, E.; ÖZCAN, E.; UYAR, Ş. Memetic algorithms and hyperhill-climbers. In: **Proc. of the 3rd Multidisciplinary Int. conf. on scheduling: theory and applications**, P. Baptiste, G. Kendall, AM Kordon and F. Sourd, Eds. [S.l.: s.n.], 2007. p. 159–166.
- ESKANDARI, L. et al. A modified and enhanced ant colony optimization algorithm for traveling salesman problem. In: **Mathematical Methods in Engineering**. [S.l.]: Springer, 2019. p. 257–265.

- ESPINDOLA, L. da S. **Proposta de uma Representação Tensorial para Modelos Markovianos Ocultos**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brasil, 2011.
- ESPINOZA-NEVÁREZ, D. et al. Selection and generation hyper-heuristics for solving the vehicle routing problem with time windows. In: ACM. **Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2016. p. 139–140.
- EZUGWU, A. E.-S.; ADEWUMI, A. O.; FRÎNCU, M. E. Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem. **Expert Systems with Applications**, Elsevier, v. 77, p. 189–210, 2017.
- FALCÓN-CARDONA, J. G.; COELLO, C. A. C. A multi-objective evolutionary hyper-heuristic based on multiple indicator-based density estimators. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference**. [S.l.], 2018. p. 633–640.
- FARINA, A. G. **Aplicação de Redes de Autômatos Estocásticos no Teste Estatístico de Software**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), Porto Alegre, RS, Brasil, 2003.
- FEO, T. A.; RESENDE, M. G. C. A probabilistic heuristic for a computationally difficult set covering problem. **Operations Research Letters**, v. 8, n. 2, p. 67–71, 1989.
- FEO, T. A.; RESENDE, M. G. C. Greedy Randomized Adaptive Search Procedures. **Journal of Global Optimization**, v. 6, n. 2, p. 109–133, 1995.
- FERONE, D. et al. Combining simulation with a grasp metaheuristic for solving the permutation flow-shop problem with stochastic processing times. In: IEEE. **2016 Winter Simulation Conference (WSC)**. [S.l.], 2016. p. 2205–2215.
- FESTA, P. et al. Integrating biased-randomized grasp with monte carlo simulation for solving the vehicle routing problem with stochastic demands. In: IEEE. **2018 Winter Simulation Conference (WSC)**. [S.l.], 2018. p. 2989–3000.
- FISHER, H. Probabilistic learning combinations of local job-shop scheduling rules. **Industrial scheduling**, Prentice-Hall, p. 225–251, 1963.
- FOSTER, J. K. **Memory**. [S.l.]: L&PM, 2011. v. 977.
- GAMARRA, M.; MITRE-ORTIZ, A.; ESCALANTE, H. Automatic cell image segmentation using genetic algorithms. In: **2019 XXII Symposium on Image, Signal Processing and Artificial Vision (STSIVA)**. [S.l.]: IEEE, 2019.
- GAREY, M. R.; JOHNSON, D. S. A guide to the theory of np-completeness. **WH Freeman, New York**, v. 70, 1979.
- GAVALAS, D. et al. An iterated local search approach for multiple itinerary planning in mobile agent-based sensor fusion. In: IEEE. **2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)**. [S.l.], 2015. p. 1–7.
- GE, Y. et al. A tabu-genetic hybrid search algorithm for job-shop scheduling problem. **E3S Web of Conferences**, EDP Sciences, v. 95, p. 04007, 2019.

GENDREAU, M.; POTVIN, J.-Y. Tabu search. In: _____. **Handbook of Metaheuristics**. [S.l.]: Springer International Publishing, 2019. p. 37–55.

GLOVER, F. Parametric combinations of local job shop rules. **ONR Research memorandum**, n. 117, 1963.

GLOVER, F. Heuristics for integer programming using surrogate constraints. **Decision Sciences**, Wiley Online Library, v. 8, n. 1, p. 156–166, 1977.

GLOVER, F. Future paths for integer programming and links to artificial intelligence. **Computer Operations Research**, Elsevier Science Ltd., Oxford, UK, UK, v. 13, n. 5, p. 533–549, 1986.

GLOVER, F. Tabu search – part i. **Operations Research Society of America Journal of Computing**, v. 1, n. 3, p. 190–206, 1989.

GLOVER, F. Tabu search – part ii. **Operations Research Society of America Journal of Computing**, v. 2, n. 1, p. 4–32, 1990.

GLOVER, F. Ejection chains, reference structures and alternating path methods for traveling salesman problems. **Discrete Applied Mathematics**, v. 65, n. 1-3, p. 223–253, 1996.

GLOVER, F. A template for scatter search and path relinking. In: SPRINGER. **European Conference on Artificial Evolution**. Berlin, 1997. p. 1–51.

GLOVER, F.; LAGUNA, M. **Tabu Search**. Norwell, MA, EUA: Kluwer Academic Publishers, 1997.

GLOVER, F.; LAGUNA, M.; MARTI, R. Scatter search and path relinking: Advances and applications. In: **Handbook of metaheuristics**. New York: Springer, 2003. p. 1–35.

GLOVER, F.; SORENSEN, K. **Metaheuristics**. 2015. Disponível em: <<http://www.scholarpedia.org/article/Metaheuristics>>. Acesso em: 5 de Fevereiro de 2019.

GLOVER, F.; TAILLARD, E. A user's guide to tabu search. **Annals of Operations Research**, Baltzer Science Publishers, Baarn/Kluwer Academic Publishers, v. 41, n. 1, p. 1–28, 1993.

GOLDBERG, D. E. **Genetic Algorithms in Search, Optimization and Machine Learning**. 1. ed. Boston, MA, EUA: Addison-Wesley Longman Publishing Co., Inc., 1989. Hardcover.

GOLDBERG, D. E.; VOESSNER, S. Optimizing global-local search hybrids. In: **PROCEEDINGS OF THE GENETIC AND EVOLUTIONARY COMPUTATION CONFERENCE, 220228**. (ALSO ILLIGAL. [S.l.]: Morgan Kaufmann, 1999. p. 220–228.

GOMORY, R. E. Outline of an algorithm for integer solutions to linear programs and an algorithm for the mixed integer problem. **50 Years of Integer Programming 1958-2008**, Springer, p. 77–103, 2010.

GORDON, T. J.; MARSH, C.; WU, Q. H. Stochastic optimal control of active vehicle suspensions using learning automata. **Proceedings of the Institution of Mechanical Engineers, Part I: Journal of Systems and Control Engineering**, SAGE Publications Sage UK: London, England, v. 207, n. 3, p. 143–152, 1993.

GRANVILLE, V.; RASSON, J. P.; KRIVÁNEK, M. Simulated Annealing: A Proof of Convergence. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, v. 16, n. 6, p. 652–656, 1994.

GRASSÉ, P. La reconstruction du nid et les coordinations interindividuelles chez *bellicositermes natalensis* et *cubitermes* sp. la théorie de la stigmergie: Essai d'interprétation du comportement des termites constructeurs. **Insectes sociaux**, Springer, v. 6, n. 1, p. 41–80, 1959.

GRASSÉ, P. **Termitologia, fondation des sociétés. Construction**. [S.l.]: Paris, Masson, 1984.

GREISTORFER, P. A tabu scatter search metaheuristic for the arc routing problem. **Computers & Industrial Engineering**, Elsevier, v. 44, n. 2, p. 249–266, 2003.

GÜLCÜ, Ş. et al. A parallel cooperative hybrid method based on ant colony optimization and 3-opt algorithm for solving traveling salesman problem. **Soft Computing**, Springer, v. 22, n. 5, p. 1669–1685, 2018.

GUPTA, I. K.; CHOUBEY, A.; CHOUBEY, S. Randomized bias genetic algorithm to solve traveling salesman problem. In: **2017 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT)**. [S.l.]: IEEE, 2017.

HAN, X.-C. et al. Multimodal optimization of traveling salesman problem: A niching ant colony system. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2018. p. 87–88.

HANSEN, P.; MLADENOVIĆ, N. An introduction to variable neighborhood search. In: _____. **Meta-Heuristics: Advances and Trends in Local Search Paradigms for Optimization**. Boston, MA: Springer US, 1999. cap. 30, p. 433–458.

HANSEN, P.; MLADENOVIĆ, N. Developments of variable neighborhood search. In: _____. **Essays and Surveys in Metaheuristics**. Boston, MA: Springer US, 2002. cap. 19, p. 415–439.

HANSEN, P.; MLADENOVIĆ, N. Variable neighborhood search. In: _____. **Search Methodologies: Introductory Tutorials in Optimization and Decision Support Techniques**. Boston, MA: Springer US, 2005. p. 211–238.

HANSEN, P. et al. Variable neighborhood search. In: _____. **Handbook of Metaheuristics**. Cham: Springer International Publishing, 2019. p. 57–97. ISBN 978-3-319-91086-4.

HANSEN, P.; MLADENOVIĆ, N.; PÉREZ, J. A. M. **Variable neighbourhood search: Methods and applications**. [S.l.]: Springer, 2010. v. 175. 367–407 p.

HART, J. P.; SHOGAN, A. W. SEMI-GREEDY HEURISTICS: AN EMPIRICAL STUDY. **OPERATIONS RESEARCH LETTERS**, v. 6, n. 3, p. 107–114, 1987.

HASAN, L. S. Solving traveling salesman problem using cuckoo search and ant colony algorithms. **Journal of Al-Qadisyah for computer science and mathematics**, v. 10, n. 2, p. Page–59, 2018.

HENDERSON, D.; JACOBSON, S. H.; JOHNSON, A. W. The Theory and Practice of Simulated Annealing. In: **Handbook Metaheuristics**. [S.l.]: Kluwer Academic Publishers, 2003. p. 287–319.

HOLLAND, J. H. **Adaptation in natural and artificial systems**. Cambridge, MA, EUA: MIT Press, 1992.

HOWARD, M. W. **Memory: Computational models**. [S.l.]: Oxford: Academic Press., 2009. v. 5.

HUSSAIN, K. et al. Metaheuristic research: a comprehensive survey. **Artificial Intelligence Review**, Springer, p. 1–43, 2018.

INGBER, L. Simulated annealing: Practice versus theory. **Mathematical and Computer Modelling**, v. 18, n. 11, p. 29–57, 1993.

JOHNSON, D. Local optimization and the traveling salesman problem. **Automata, languages and programming**, Springer, p. 446–461, 1990.

JOHNSON, D. S.; MCGEOCH, L. A. The traveling salesman problem: A case study in local optimization. **Local search in combinatorial optimization**, Chichester, UK, v. 1, p. 215–310, 1997.

JOHNSONBAUGH, R. Discrete mathematics (fourth edition). **Upper Saddle River. NJ: Prentics Hall. international1**, p. 257–280, 1997.

KALENDER, M. et al. A greedy gradient-simulated annealing hyper-heuristic for a curriculum-based course timetabling problem. In: IEEE. **2012 12th UK Workshop on Computational Intelligence (UKCI)**. [S.l.], 2012. p. 1–8.

KALENDER, M. et al. A greedy gradient-simulated annealing selection hyper-heuristic. **Soft Computing**, Springer, v. 17, n. 12, p. 2279–2292, 2013.

KEFALAS, M. et al. A tabu search-based memetic algorithm for the multi-objective flexible job shop scheduling problem. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2019. p. 1254–1262.

KENDALL, G.; HUSSIN, N. M. A tabu search hyper-heuristic approach to the examination timetabling problem at the mara university of technology. In: SPRINGER. **International Conference on the Practice and Theory of Automated Timetabling**. [S.l.], 2004. p. 270–293.

KENDALL, G.; HUSSIN, N. M. An investigation of a tabu-search-based hyper-heuristic for examination timetabling. In: **Multidisciplinary Scheduling: Theory and Applications**. [S.l.]: Springer, 2005. p. 309–328.

KENNEDY, J.; EBERHART, R. Particle swarm optimization. In: IEEE. **IEEE International Conference on Neural Networks**. [S.l.], 1995. v. 4, p. 1942–1948.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. **science**, Washington, v. 220, n. 4598, p. 671–680, 1983.

KOKUBO, T.; FUKUYAMA, Y. Train crew scheduling using tabu search. In: IEEE. **2017 56th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)**. [S.l.], 2017. p. 63–66.

KOULAMAS, C.; ANTONY, S.; JAEN, R. A survey of Simulated Annealing application to operations- research problems. **Management Science**, v. 22, n. 1, p. 41–56, 1994.

- KUREICHIK, V. M.; LOGUNOVA, J. Some of the new indicators in genetic algorithms for the traveling salesman problem. In: **2018 IEEE East-West Design & Test Symposium (EWDTS)**. [S.l.]: IEEE, 2018.
- LAGUNA, M. **Global optimization and meta-heuristics**. 2000.
- Lan, F.; Wang, B.; Zhang, X. Improved variable neighbourhood search algorithm for robust job shop scheduling problems. In: **2016 8th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)**. [S.l.: s.n.], 2016. v. 02, p. 592–595.
- LAWLER, E. L.; WOOD, D. E. Branch-and-bound methods: A survey. **Operations research, INFORMS**, v. 14, n. 4, p. 699–719, 1966.
- LEE, C. A review of applications of genetic algorithms in operations management. **Engineering Applications of Artificial Intelligence**, Elsevier BV, v. 76, p. 1–12, nov. 2018.
- LI, W. **Multiobjective selection hyper-heuristics using reinforcement learning**. Tese (Doutorado) — University of Nottingham, 2018.
- LI, W.; ÖZCAN, E.; JOHN, R. A learning automata-based multiobjective hyper-heuristic. **IEEE Transactions on Evolutionary Computation**, IEEE, v. 23, n. 1, p. 59–73, 2017.
- LI, Y.; ZHOU, A.; ZHANG, G. Simulated annealing with probabilistic neighborhood for traveling salesman problems. In: IEEE. **2011 Seventh International Conference on Natural Computation**. [S.l.], 2011. v. 3, p. 1565–1569.
- LIBERATORE, S.; SECHI, G. M. Location and calibration of valves in water distribution networks using a scatter-search meta-heuristic approach. **Water resources management**, Springer, v. 23, n. 8, p. 1479–1495, 2009.
- LIN, Y.; BIAN, Z.; LIU, X. Developing a dynamic neighborhood structure for an adaptive hybrid simulated annealing–tabu search algorithm to solve the symmetrical traveling salesman problem. **Applied Soft Computing**, Elsevier, v. 49, p. 937–952, 2016.
- Liu, A. et al. A tabu-based variable neighborhood local search for n-vehicles exploration problem. In: **2016 12th IEEE International Conference on Control and Automation (ICCA)**. [S.l.: s.n.], 2016. p. 951–955.
- LIU, J.; LI, W. Greedy permuting method for genetic algorithm on traveling salesman problem. In: **2018 8th International Conference on Electronics Information and Emergency Communication (ICEIEC)**. [S.l.]: IEEE, 2018.
- LIU, Y.; XIONG, S.; LIU, H. Hybrid simulated annealing algorithm based on adaptive cooling schedule for tsp. In: ACM. **Proceedings of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation**. [S.l.], 2009. p. 895–898.
- LOBO, E. L. M. **Uma Solução do Problema de Horário Escolar Via Algoritmo Genético Paralelo**. Dissertação (Mestrado) — Centro Federal de Educação Tecnológica de Minas Gerais, 2005.
- LORENZONI, L. L.; POLYCARPO, W. M. Scatter search aplicado ao problema de otimização da alocação de sondas de produção em poços de petróleo (scatter search applied to workover rigs optimization problem). In: **Proceedings of XXVII national production engineering meeting**. [S.l.: s.n.], 2007. p. 1–10.

- LOURENÇO, H. R.; MARTIN, O. C.; STTUZLE, T. Iterated local search: Framework and applications. In: _____. [S.l.]: Springer Nature, 2019. v. 272, p. 129–168.
- LUKE, S. **Essentials of Metaheuristics**. second. [S.l.]: Lulu, 2013.
- MAENHOUT, B.; VANHOUCKE, M. A hybrid scatter search heuristic for personalized crew rostering in the airline industry. **European Journal of Operational Research**, Elsevier, v. 206, n. 1, p. 155–167, 2010.
- MAGNANTI, T. L.; ORLIN, J. B. Network flows. **PHI Englewood Cliffs NJ**, 1993.
- MALDONADO, A. E. Explorações sobre a problemática epistemológica no campo das ciências da comunicação. **Epistemologia da Comunicação**., p. 205–25, 2003.
- MARINHO, E. H. **Heurísticas Busca Tabu para o Problema de Programação de Tripulações de Ônibus Urbano**. Dissertação (Mestrado) — Universidade Federal Fluminense (UFF), 2005.
- MARSHALL, R. J. **Adapting a Hyper-heuristic to Respond to Scalability Issues in Combinatorial Optimisation**. Tese (Doutorado) — Victoria University of Wellington, 2015.
- MARTÍ, R.; REY, U.; CARLOS, J. Adaptive Memory Programming: An Empirical Study with the Bandwidth Coloring Problem. 2009.
- MARTIN, O.; OTTO, S. W.; FELTEN, E. W. **Large-step Markov chains for the traveling salesman problem**. [S.l.], 1991.
- MARTIN, O. C.; OTTO, S. W. Combining simulated annealing with local search heuristics. **Annals of Operations Research**, Springer, v. 63, n. 1, p. 57–75, 1996.
- MARTINS, S. L.; RIBEIRO, C. C. Handbook of optimization in telecommunications. In: _____. Boston, MA: Springer US, 2006. p. 103–128.
- MATAI, R.; SINGH, S. P.; MITTAL, M. L. Traveling salesman problem: an overview of applications, formulations, and solution approaches. **Traveling salesman problem, theory and applications**, InTech, Croatia, v. 1, 2010.
- MAVROVOUNIOTIS, M.; MÜLLER, F. M.; YANG, S. Ant colony optimization with local search for dynamic traveling salesman problems. **IEEE Transactions on Cybernetics**, IEEE, v. 47, n. 7, p. 1743–1756, 2016.
- MCCLYMONT, K.; KEEDWELL, E. **A single objective variant of the online selective Markov chain Hyper-heuristic (MCHH-S)**. [S.l.]: CHeSC, 2011b.
- MCCLYMONT, K.; KEEDWELL, E. C. Markov chain hyper-heuristic (mchh): an online selective hyper-heuristic for multi-objective continuous problems. In: ACM. **Proceedings of the 13th annual conference on Genetic and evolutionary computation**. [S.l.], 2011a. p. 2003–2010.
- MCGEOCH, C. C. Toward an experimental method for algorithm simulation. **INFORMS Journal on Computing**, INFORMS, v. 8, n. 1, p. 1–15, 1996.
- METROPOLIS, N. et al. Equation of state calculations by fast computing machines. **The journal of chemical physics**, AIP, v. 21, n. 6, p. 1087–1092, 1953.

MLADENOVIC, N. A variable neighborhood algorithm-a new metaheuristic for combinatorial optimization. In: **papers presented at Optimization Days**. [S.l.: s.n.], 1995. v. 12.

MLADENOVIC, N. et al. General variable neighborhood search for the continuous optimization. **European Journal of Operational Research**, v. 191, n. 3, p. 753–770, 2008.

MLADENOVIC, N.; HANSEN, P. Variable neighborhood search. **Computers & Operations Research**, Elsevier Science Ltd., Oxford, UK, UK, v. 24, n. 11, p. 1097–1100, 11 1997.

MOHAN, B. C.; BASKARAN, R. A survey: Ant colony optimization based recent research and implementation on several engineering domain. **Expert Systems with Applications**, Elsevier, v. 39, n. 4, p. 4618–4627, 2012.

MOHSEN, A. M. Annealing ant colony optimization with mutation operator for solving tsp. **Computational intelligence and neuroscience**, Hindawi, v. 2016, 2016.

MONTGOMERY, D. **Design and Analysis of Experiments**. [S.l.]: John Wiley & Sons, 2013. (Student solutions manual).

MOOD, A. M. F.; GRAYBILL, F. A.; BOES, D. C. **Introduction to the Theory of Statistics**. [S.l.]: McGraw-Hill, 1974. (International Student edition).

MORAES, R. F. D. et al. Implementation of a rvnd, vns, ils heuristic for the traveling car renter problem. In: IEEE. **2018 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2018. p. 1–8.

MORRIS, P. The Breakout Method for Escaping from Local Minima. **Proceedings of the Eleventh National Conference on Artificial Intelligence (AAAI'93)**, p. 40–45, 1993.

MORTEZANEZHAD, A.; DANESHIFAR, E. Big-data clustering with genetic algorithm. In: **2019 5th Conference on Knowledge Based Engineering and Innovation (KBEI)**. [S.l.]: IEEE, 2019.

NARENDRA, K. S.; THATHACHAR, M. Learning automata - a survey. **Systems, Man and Cybernetics, IEEE Transactions on**, SMC-4, n. 4, p. 323–334, 1974.

NARENDRA, K. S.; THATHACHAR, M. A. L. **Learning Automata: An Introduction**. Upper Saddle River, NJ, EUA: Prentice-Hall, Inc., 1989.

OCHOA, G.; QU, R.; BURKE, E. K. Analyzing the landscape of a graph based hyper-heuristic for timetabling problems. In: ACM. **Proceedings of the 11th Annual conference on Genetic and evolutionary computation**. [S.l.], 2009. p. 341–348.

OTTE, G.; VOLPE, M. L. Um olhar constelar sobre o pensamento de Walter Benjamin. **Revista Fragmentos**, v. 18, p. 35–47, 2000.

OZCAN, E. et al. Examination timetabling using late acceptance hyper-heuristics. In: IEEE. **2009 IEEE Congress on Evolutionary Computation**. [S.l.], 2009. p. 997–1004.

ÖZCAN, E. et al. A reinforcement learning: great-deluge hyper-heuristic for examination timetabling. In: **Modeling, Analysis, and Applications in Metaheuristic Computing: Advancements and Trends**. [S.l.]: IGI Global, 2012. p. 34–55.

- PANTRIGO, J. J.; MONTEMAYOR, A. S.; CABIDO, R. Scatter search particle filter for 2d real-time hands and face tracking. In: SPRINGER. **International Conference on Image Analysis and Processing**. [S.l.], 2005. p. 953–960.
- PAPADIMITRIOU, C. H.; STEIGLITZ, K. **Combinatorial optimization: algorithms and complexity**. [S.l.]: Courier Corporation, 1998.
- PARIS, M.; RIBEIRO, C. C. Reactive GRASP: An Application to a Matrix Decomposition Problem in TDMA Traffic Assignment. **INFORMS Journal on Computing**, v. 12, n. 3, p. 164–176, 2000.
- PEISHENG, L. B. M. Simulated annealing-based ant colony algorithm for traveling salesman problems [j]. **Journal of Huazhong University of Science and Technology (Nature Science Edition)**, v. 11, 2009.
- PENG, J.; WILLIAMS, R. J. Efficient learning and planning within the dyna framework. In: **IEEE International Conference on Neural Networks**. [S.l.: s.n.], 1993. v. 1, p. 168–174.
- PILLAY, N. An analysis of representations for hyper-heuristics for the uncapacitated examination timetabling problem in a genetic programming system. In: **ACM. Proceedings of the 2008 annual research conference of the South African Institute of Computer Scientists and Information Technologists on IT research in developing countries: riding the wave of technology**. [S.l.], 2008. p. 188–192.
- PILLAY, N. A study into the use of hyper-heuristics to solve the school timetabling problem. In: **ACM. Proceedings of the 2010 Annual Research Conference of the South African Institute of Computer Scientists and Information Technologists**. [S.l.], 2010. p. 258–264.
- PILLAY, N. Evolving hyper-heuristics for the uncapacitated examination timetabling problem. **Journal of the Operational Research Society**, Taylor & Francis, v. 63, n. 1, p. 47–58, 2012.
- PINTEA, C.-M.; POP, P. C.; CHIRA, C. The generalized traveling salesman problem solved with ant algorithms. **Complex Adaptive Systems Modeling**, SpringerOpen, v. 5, n. 1, p. 8, 2017.
- PLATEAU, B. **De l'évaluation du parallélisme et de la synchronisation**. Tese (Doutorado) — Paris-Sud, Orsay, France, 1984.
- POLYA, G. **How to Solve it**. New Jersey: Princeton University Press, 1945.
- POZNYAK, A. S.; NAJIM, K. **Learning automata and stochastic optimization**. [S.l.]: Springer, 1997. (Lecture notes in control and information sciences).
- PURANEN, T. **Metaheuristics meet metamodels: a modeling language and a product line architecture for route optimization systems**. Tese (Doutorado) — University of Jyväskylä, 2011.
- QU, R.; BURKE, E. K. Hybridizations within a graph-based hyper-heuristic framework for university timetabling problems. **Journal of the Operational Research Society**, Taylor & Francis, v. 60, n. 9, p. 1273–1285, 2009.
- RAGHAVJEE, R.; PILLAY, N. A comparison of genetic algorithms and genetic programming in solving the school timetabling problem. In: **IEEE. 2012 Fourth World Congress on Nature and Biologically Inspired Computing (NaBIC)**. [S.l.], 2012. p. 98–103.

- RAMOS, G. d. O.; BAZZAN, A. L. Efficient local search in traffic assignment. In: IEEE. **2016 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2016. p. 1493–1500.
- RARDIN, R. L.; UZSOY, R. Experimental evaluation of heuristic optimization algorithms: A tutorial. **Journal of Heuristics**, Springer, v. 7, n. 3, p. 261–304, 2001.
- REGO, C.; GLOVER, F. The traveling salesman problem and its variations. In: _____. [S.l.]: Springer Science & Business Media, 2006. v. 12, cap. LOCAL SEARCH AND METAHEURISTICS.
- REGO, C.; ROUCAIROL, C. Using tabu search for solving a dynamic multi-terminal truck dispatching problem. **European Journal of Operational Research**, v. 83, n. 2, p. 411–429, 1995.
- RESENDE, M. G. C. Metaheuristic Hybridization with Greedy Randomized Adaptive Search Procedures. **State-of-the-Art Decision-Making Tools in the Information-Intensive Age**, n. April 2016, p. 295–319, 2008.
- REZKI, H.; AGHEZZAF, B. The bi-objective orienteering problem with budget constraint: Grasp_ils. In: IEEE. **2017 International Colloquium on Logistics and Supply Chain Management (LOGISTIQUA)**. [S.l.], 2017. p. 25–30.
- RODRÍGUEZ-GARCÍA, M.; DUARTE, A.; SÁNCHEZ-ORO, J. Grasp with path relinking for 2d-bandwidth minimization problem. In: ACM. **Proceedings of the International Conference on Learning and Optimization Algorithms: Theory and Applications**. [S.l.], 2018. p. 52.
- ROSALES-PÉREZ, A. et al. Evolutionary multilabel hyper-heuristic design. In: IEEE. **2017 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.], 2017. p. 2622–2629.
- ROSS, P. Hyper-heuristics. **Search methodologies**, Springer, p. 529–556, 2005a.
- ROSS, P.; MARFN-BLAZQUEZ, J. G. Constructive hyper-heuristics in class timetabling. In: IEEE. **The 2005 IEEE Congress on Evolutionary Computation**. [S.l.], 2005b. v. 2, p. 1493–1500.
- ROSSI, F.; BEEK, P. v.; WALSH, T. **Handbook of Constraint Programming (Foundations of Artificial Intelligence)**. New York, NY, USA: Elsevier Science Inc., 2006.
- RUSSELL, R. A.; CHIANG, W.-C. Scatter search for the vehicle routing problem with time windows. **European Journal of Operational Research**, Elsevier, v. 169, n. 2, p. 606–622, 2006.
- SAAD, A. et al. A grasp-genetic metaheuristic applied on multi-processor task scheduling systems. In: IEEE. **2018 13th International Conference on Computer Engineering and Systems (ICCES)**. [S.l.], 2018. p. 109–115.
- SABAR, N. R.; AYOB, M. Examination timetabling using scatter search hyper-heuristic. In: IEEE. **2009 2nd Conference on Data Mining and Optimization**. [S.l.], 2009. p. 127–131.
- SABAR, N. R.; TURKY, A.; SONG, A. A genetic programming based iterated local search for software project scheduling. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference**. [S.l.], 2018. p. 1364–1370.
- SAGAN, C. Why we need to understand science. **Parade Magazine**, v. 10, 1989.

- SAGHEER, A. M.; SADIQ, A. T.; IBRAHIM, M. S. Improvement of scatter search using bees algorithm. In: IEEE. **2012 6th International Conference on Signal Processing and Communication Systems**. [S.l.], 2012. p. 1–7.
- SALEHIPOUR, A. et al. Efficient GRASP+VND and GRASP+VNS metaheuristics for the traveling repairman problem. **4or**, v. 9, n. 2, p. 189–209, 2011.
- SALES, A. H. C. **Formalismos Estruturados de Modelagem para Sistemas Markovianos Complexos**. Dissertação (Mestrado) — Pontifícia Universidade Católica do Rio Grande do Sul (PUCRS), 2004.
- SANTOS, V. L. A.; ARROYO, J. E. C.; CARVALHO, T. F. Iterated local search based heuristic for scheduling jobs on unrelated parallel machines with machine deterioration effect. In: ACM. **Proceedings of the 2016 on Genetic and Evolutionary Computation Conference Companion**. [S.l.], 2016. p. 53–54.
- SANTOSO, A. et al. Model development of rescue assignment and scheduling problem using grasp metaheuristic. In: IEEE. **2017 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)**. [S.l.], 2017. p. 1407–1410.
- SARI, T. et al. Evaluation of scatter search and genetic algorithm at resource constrained project scheduling problems. In: IEEE. **2011 15th IEEE International Conference on Intelligent Engineering Systems**. [S.l.], 2011. p. 127–130.
- SEMLALI, S. C. B.; RIFFI, M. E.; CHEBIHI, F. Hybrid chicken swarm optimization with a grasp constructive procedure using multi-threads to solve the quadratic assignment problem. In: IEEE. **2018 6th International Conference on Multimedia Computing and Systems (ICMCS)**. [S.l.], 2018. p. 1–6.
- SIM, K.; HART, E. A combined generative and selective hyper-heuristic for the vehicle routing problem. In: ACM. **Proceedings of the Genetic and Evolutionary Computation Conference 2016**. [S.l.], 2016. p. 1093–1100.
- SIN, E. S.; KHAM, N. S. M. Hyper heuristic based on great deluge and its variants for exam timetabling problem. **arXiv preprint arXiv:1202.1891**, 2012.
- SOGHIER, A. **Novel Hyper-heuristic Approaches in Exam Timetabling**. 1–189 p. Tese (Doutorado) — University of Nottingham, 2012.
- SOGHIER, A.; QU, R. Adaptive selection of heuristics for assigning time slots and rooms in exam timetables. **Applied Intelligence**, Springer, v. 39, n. 2, p. 438–450, 2013.
- SOHRABI, S.; ZIARATI, K.; KESHTKARAN, M. A novel method for solving the orienteering problem with hotel selection. In: IEEE. **2017 International Symposium on Computer Science and Software Engineering Conference (CSSE)**. [S.l.], 2017. p. 7–11.
- SÖRENSEN, K. Metaheuristics-the metaphor exposed. **International Transactions in Operational Research**, v. 22, n. 1, p. 3–18, 2015.
- SORENSEN, K.; SEVAUX, M.; GLOVER, F. A history of metaheuristics. **Handbook of Heuristics**, 01 2017.
- SORIA-ALCARAZ, J. A. et al. Effective learning hyper-heuristics for the course timetabling problem. **European Journal of Operational Research**, Elsevier, v. 238, n. 1, p. 77–86, 2014.

- SPRINGER. Journal of heuristic policies on heuristic search research. **Journal of Heuristics**, 2017.
- STÜTZLE, T. Local search algorithms for combinatorial problems. **Darmstadt University of Technology PhD Thesis**, v. 20, 1998.
- STÜTZLE, T.; RUIZ, R. Iterated local search: A concise review. 2018.
- SUCUPIRA, I. R. **Um Estudo Empírico de Hiper-Heurísticas**. 123 p. Dissertação (Mestrado) — Universidade de São Paulo, 2007.
- SUMAN, B.; KUMAR, P. A survey of simulated annealing as a tool for single and multiobjective optimization. **Journal of the Operational Research Society**, v. 57, n. 10, p. 1143–1160, 2006.
- TAILLARD, E. D. et al. **Adaptive Memory Programming: A Unified View of Meta-Heuristics**. [S.l.], 1998.
- TALBI, E. G. **Metaheuristics: From Design to Implementation**. Hoboken, New Jersey, USA: JohnWiley & Sons, Inc., 2009.
- TAMIMI, A. et al. Intelligent traffic light based on genetic algorithm. In: **2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT)**. [S.l.]: IEEE, 2019.
- TANG, J.; ZHANG, J.; PAN, Z. A scatter search algorithm for solving vehicle routing problem with loading cost. **Expert Systems with Applications**, Elsevier, v. 37, n. 6, p. 4073–4083, 2010.
- TSPLIB. 2018. Disponível em: <<https://www.iwr.uni-heidelberg.de/groups/comopt/software/TSPLIB95/>>. Acesso em: 13 de Junho de 2018.
- TURKY, A. et al. Hyper-heuristic based local search for combinatorial optimisation problems. In: SPRINGER. **Australasian Joint Conference on Artificial Intelligence**. [S.l.], 2018. p. 312–317.
- UNSAI, C. **Intelligent navigation of autonomous vehicles in an automated highway system: learning methods and interacting vehicles approach**. Tese (Doutorado) — Faculty of the Virginia Polytechnic Institute and State University, 1 1997.
- URLI, T. Hybrid meta-heuristics for combinatorial optimization. **Constraints**, v. 20, n. 4, p. 473–473, 2015.
- VAESSENS, R. J. M.; AARTS, E. H. L.; LENSTRA, J. K. A local search template. **Computers & Operations Research**, v. 25, n. 11, p. 969–979, 1998.
- VIANA, G. V. R. **Meta-heurísticas e Programação Paralela em Otimização Combinatória**. [S.l.]: Edições UFC, 1998.
- VILLEGAS, J. G. et al. GRASP/VND and multi-start evolutionary local search for the single truck and trailer routing problem with satellite depots. **Engineering Applications of Artificial Intelligence**, v. 23, n. 5, p. 780–794, 2010.

VOUDOURIS, C.; TSANG, E. P. K.; ALSHEDDY, A. **Guided Local Search**. Colchester, 1995.

Wang, G.; Lu, M.; Zhao, X. An improved bat algorithm with variable neighborhood search for global optimization. In: **2016 IEEE Congress on Evolutionary Computation (CEC)**. [S.l.: s.n.], 2016. p. 1773–1778.

WANG, J. et al. Rough set and scatter search metaheuristic based feature selection for credit scoring. **Expert Systems with Applications**, Elsevier, v. 39, n. 6, p. 6123–6128, 2012.

WANG, Y.-S.; TENG, H.-F.; SHI, Y.-J. Cooperative co-evolutionary scatter search for satellite module layout design. **Engineering Computations**, Emerald Group Publishing Limited, v. 26, n. 7, p. 761–785, 2009.

WHITLEY, D.; COLLINS, F. **Handbook of Natural Computing**. [S.l.]: Springer Berlin Heidelberg, 2012.

WHITLEY, D.; WATSON, J. P. **Complexity theory and the no free lunch theorem**. [S.l.]: Springer, 2006.

WIDIARTHA, I. B. K.; ANJARWANI, S. E.; BIMANTORO, F. Traveling salesman problem using multi-element genetic algorithm. In: **2017 11th International Conference on Telecommunication Systems Services and Applications (TSSA)**. [S.l.]: IEEE, 2017.

WIKIPEDIA. **List of Metaheuristics**. Wikimedia, 2019. Disponível em: <https://en.wikipedia.org/w/index.php?title=Talk:Metaheuristic/List_of_Metaheuristics&oldid=838517553>. Acesso em: 08 de Julho de 2019.

WOLPERT, D. H.; MACREADY, W. G. No free lunch theorems for optimization. **IEEE Transactions on Evolutionary Computation**, v. 1, n. 1, p. 67–82, 1997.

WOO, M.; KIM, S.; KANG, S. Grasp based metaheuristics for layout pattern classification. In: IEEE PRESS. **Proceedings of the 36th International Conference on Computer-Aided Design**. [S.l.], 2017. p. 512–518.

Wu Zixuan; Li Tieke; Wang Bailin. Hybrid variable neighborhood search for batch scheduling of hot rolled steel tube. In: **2016 IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)**. [S.l.: s.n.], 2016. p. 335–338.

XIAO, Y. et al. A multi-strategy improved ant colony algorithm for solving traveling salesman problem. In: IOP PUBLISHING. **IOP Conference Series: Materials Science and Engineering**. [S.l.], 2018. v. 394, n. 4, p. 042101.

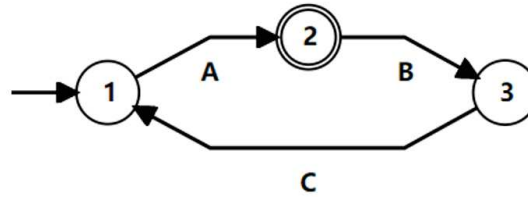
XU, J.; CHIU, S. Y.; GLOVER, F. Fine-tuning a tabu search algorithm with statistical tests. **International Transactions in Operational Research**, Wiley Online Library, v. 5, n. 3, p. 233–244, 1998.

XU, L.; YANPENG, L.; XUAN, J. Based on tabu search and particle swarm optimization algorithms solving job shop scheduling optimization problems. In: IEEE. **2013 Fourth International Conference on Digital Manufacturing & Automation**. [S.l.], 2013. p. 322–324.

- XU, Y.; QU, R. A hybrid scatter search meta-heuristic for delay-constrained multicast routing problems. **Applied Intelligence**, Springer, v. 36, n. 1, p. 229–241, 2012.
- XUE, X.; REN, A.; CHEN, D. An evolutionary tabu search algorithm for matching biomedical ontologies. In: IEEE. **2018 14th International Conference on Computational Intelligence and Security (CIS)**. [S.l.], 2018. p. 191–195.
- XUEYUAN, W. Research on solution of TSP based on improved genetic algorithm. In: **2018 International Conference on Engineering Simulation and Intelligent Control (ESAIC)**. [S.l.]: IEEE, 2018.
- YIQIU, F.; XIA, X.; JUNWEI, G. Cloud computing task scheduling algorithm based on improved genetic algorithm. In: **2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)**. [S.l.]: IEEE, 2019.
- YOUNES, A.; ELKAMEL, A.; AREIBI, S. Meta-heuristics: Evaluation and reporting techniques. In: _____. **Stochastic Global Optimization**. [S.l.: s.n.], 2010. p. 337–352.
- ZHAN, S.-h. et al. List-based simulated annealing algorithm for traveling salesman problem. **Computational intelligence and neuroscience**, Hindawi Publishing Corp., v. 2016, p. 8, 2016.
- ZHANG, T.; CHAOVALITWONGSE, W. A.; ZHANG, Y. Scatter search for the stochastic travel-time vehicle routing problem with simultaneous pick-ups and deliveries. **Computers & Operations Research**, Elsevier, v. 39, n. 10, p. 2277–2290, 2012.
- ZHAO, D.; XIONG, W.; SHU, Z. Simulated annealing with a hybrid local search for solving the traveling salesman problem. **Journal of Computational and Theoretical Nanoscience**, American Scientific Publishers, v. 12, n. 7, p. 1165–1169, 2015.
- ZHONG, Y. et al. The study of neighborhood structure of tabu search algorithm for traveling salesman problem. In: IEEE. **2008 Fourth International Conference on Natural Computation**. [S.l.], 2008. v. 1, p. 491–495.
- ZHOU, Y.; HE, F.; QIU, Y. Dynamic strategy based parallel ant colony optimization on gpus for tsps. **Science China Information Sciences**, Springer, v. 60, n. 6, p. 068102, 2017.

APÊNDICE A – REDE DE AUTÔMATOS ESTOCÁSTICOS

Figura 30 – Autômato $\mathcal{A}^{(1)}$ - Sistema Meta-heurístico Geral

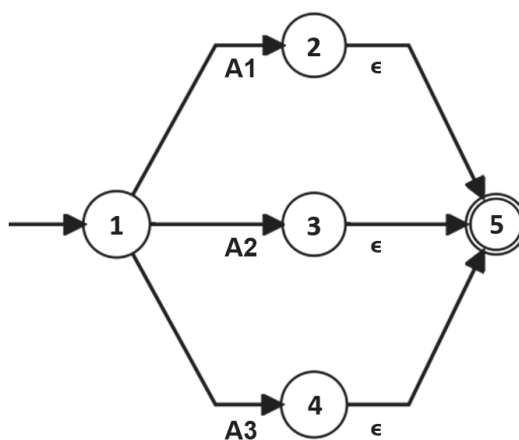


Fonte: Elaborado pelo autor

Tabela 54 – Composição do autômato $\mathcal{A}^{(1)}$ - Sistema Meta-heurístico Geral

Evento (Mecanismo Heurístico)		Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Sistema Meta-heurístico Geral				$A(BCA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	A	$(A1 A2 A3)$
Fase Geracional	$e^{(5)}$	Sincronizante	B	$(D E F J K M O S T)$
Fase Substitutiva	$e^{(6)}$	Sincronizante	C	$(G H I L N Q R U)$

Fonte: Elaborado pelo autor

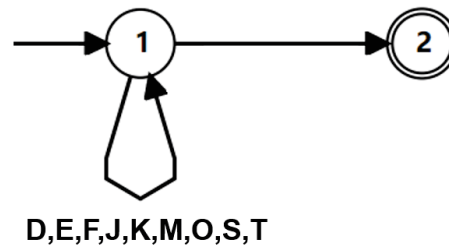
Figura 31 – Autômato $\mathcal{A}^{(2)}$ - Critério de parada

Fonte: Elaborado pelo autor

Tabela 55 – Composição do autômato $\mathcal{A}^{(2)}$ - Critério de parada

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Iterações sem melhora	$e^{(2)}$	Local	
Número de repetições	$e^{(3)}$	Local	
Número de avaliações	$e^{(4)}$	Local	

Fonte: Elaborado pelo autor

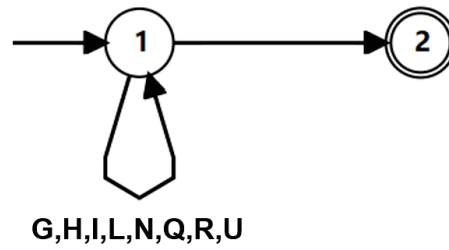
Figura 32 – Autômato $\mathcal{A}^{(3)}$ - Fase Geracional

Fonte: Elaborado pelo autor

Tabela 56 – Composição do autômato $\mathcal{A}^{(3)}$ - Fase Geracional

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)	
Fase Geracional	e ⁽⁵⁾	Sincronizante	B	(D E F J K M O S T)

Fonte: Elaborado pelo autor

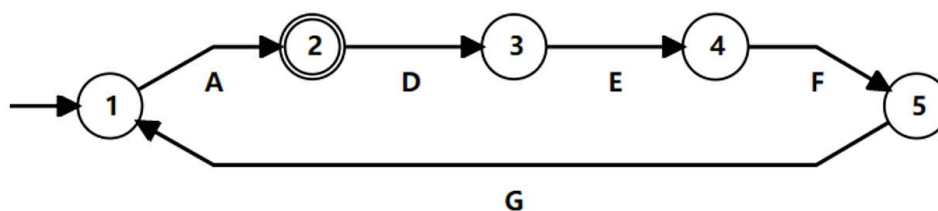
Figura 33 – Autômato $\mathcal{A}^{(4)}$ - Fase Substitutiva

Fonte: Elaborado pelo autor

Tabela 57 – Composição do autômato $\mathcal{A}^{(4)}$ - Fase Substitutiva

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)	
Fase Substitutiva	e ⁽⁶⁾	Sincronizante	C	(G H I L N Q R U)

Fonte: Elaborado pelo autor

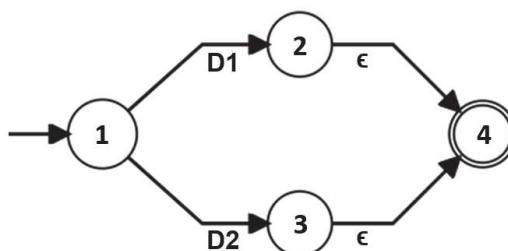
Figura 34 – Autômato $\mathcal{A}^{(5)}$ - Algoritmo Genético

Fonte: Elaborado pelo autor

Tabela 58 – Composição do autômato $\mathcal{A}^{(5)}$ - Algoritmo Genético

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Algoritmo Genético			$A (DEFGA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Seleção	$e^{(7)}$	Sincronizante	$(D1 D2)$
Cruzamento	$e^{(22)}$	Sincronizante	$(E1 E2 E3 E4 E5 E6)$
Mutação	$e^{(29)}$	Sincronizante	$(F1 F2)$
Atualização da População	$e^{(44)}$	Local	

Fonte: Elaborado pelo autor

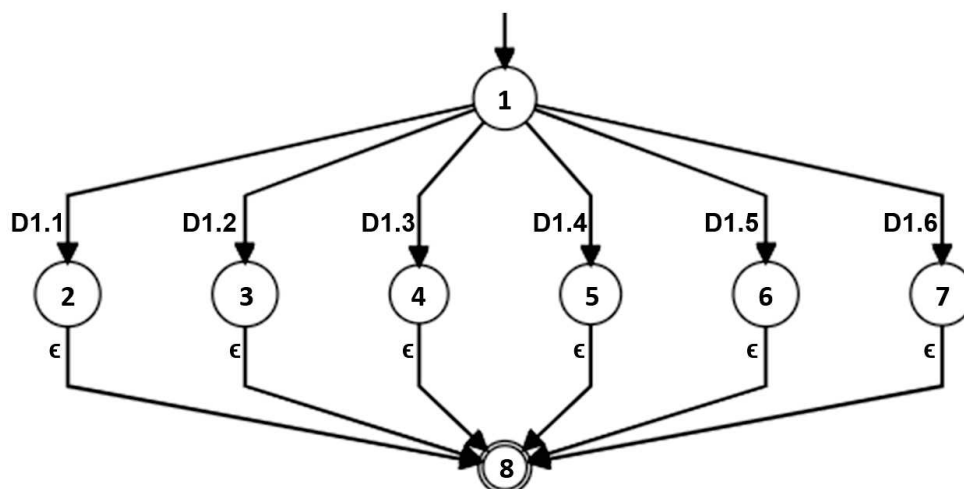
Figura 35 – Autômato $\mathcal{A}^{(6)}$ - Seleção

Fonte: Elaborado pelo autor

Tabela 59 – Composição do autômato $\mathcal{A}^{(6)}$ - Seleção

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Seleção	e ⁽⁷⁾	Sincronizante	(D1 D2)
Roleta	e ⁽⁸⁾	Sincronizante	(D1.1 D1.2 D1.3 D1.4 D1.5 D1.6)
Torneio	e ⁽¹⁵⁾	Sincronizante	(D2.1 D2.2 D2.3 D2.4 D2.5 D2.6)

Fonte: Elaborado pelo autor

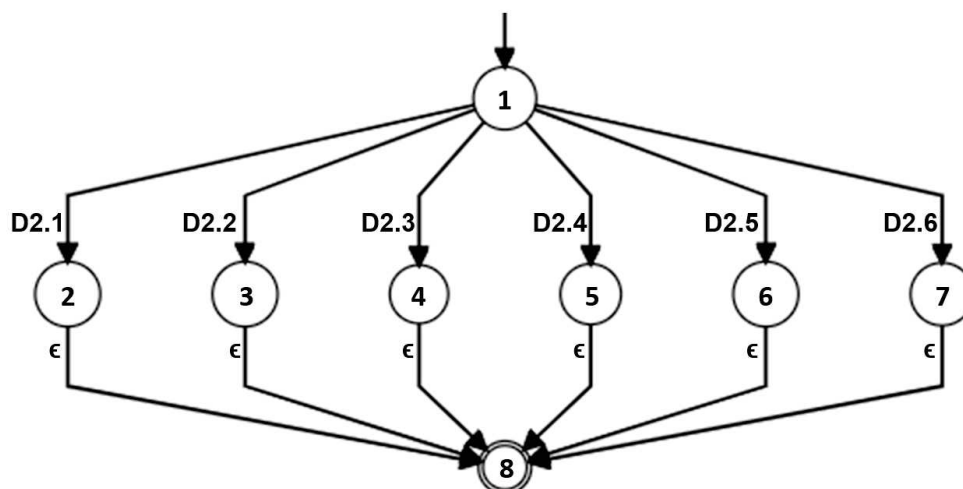
Figura 36 – Autômato $\mathcal{A}^{(7)}$ - Roleta

Fonte: Elaborado pelo autor

Tabela 60 – Composição do autômato $\mathcal{A}^{(7)}$ - Roleta

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Roleta	$e^{(8)}$	Sincronizante	$(D1.1 D1.2 D1.3 D1.4 D1.5 D1.6)$
Taxa de cruzamento 2%	$e^{(9)}$	Local	
Taxa de cruzamento 5%	$e^{(10)}$	Local	
Taxa de cruzamento 10%	$e^{(11)}$	Local	
Taxa de cruzamento 25%	$e^{(12)}$	Local	
Taxa de cruzamento 50%	$e^{(13)}$	Local	
Taxa de cruzamento 75%	$e^{(14)}$	Local	

Fonte: Elaborado pelo autor

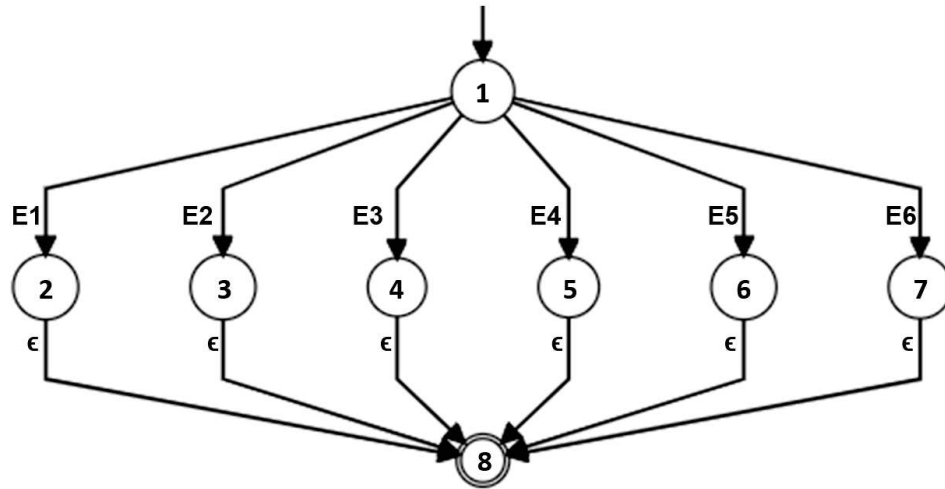
Figura 37 – Autômato $\mathcal{A}^{(8)}$ - Torneio

Fonte: Elaborado pelo autor

Tabela 61 – Composição do autômato $\mathcal{A}^{(8)}$ - Torneio

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Torneio	$e^{(15)}$	Sincronizante	$(D2.1 D2.2 D2.3 D2.4 D2.5 D2.6)$
Taxa de cruzamento 2%	$e^{(16)}$	Local	
Taxa de cruzamento 5%	$e^{(17)}$	Local	
Taxa de cruzamento 10%	$e^{(18)}$	Local	
Taxa de cruzamento 25%	$e^{(19)}$	Local	
Taxa de cruzamento 50%	$e^{(20)}$	Local	
Taxa de cruzamento 75%	$e^{(21)}$	Local	

Fonte: Elaborado pelo autor

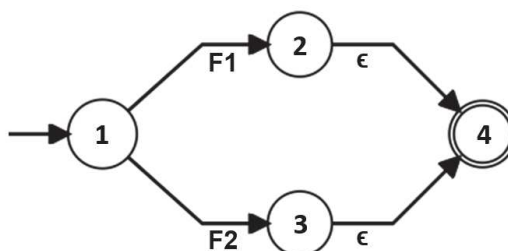
Figura 38 – Autômato $\mathcal{A}^{(9)}$ - Cruzamento

Fonte: Elaborado pelo autor

Tabela 62 – Composição do autômato $\mathcal{A}^{(9)}$ - Cruzamento

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Cruzamento	$e^{(22)}$	Sincronizante	$(E1 E2 E3 E4 E5 E6)$
PMX	$e^{(23)}$	Local	
CX	$e^{(24)}$	Local	
OX	$e^{(25)}$	Local	
POS	$e^{(26)}$	Local	
VR	$e^{(27)}$	Local	
SCX	$e^{(28)}$	Local	

Fonte: Elaborado pelo autor

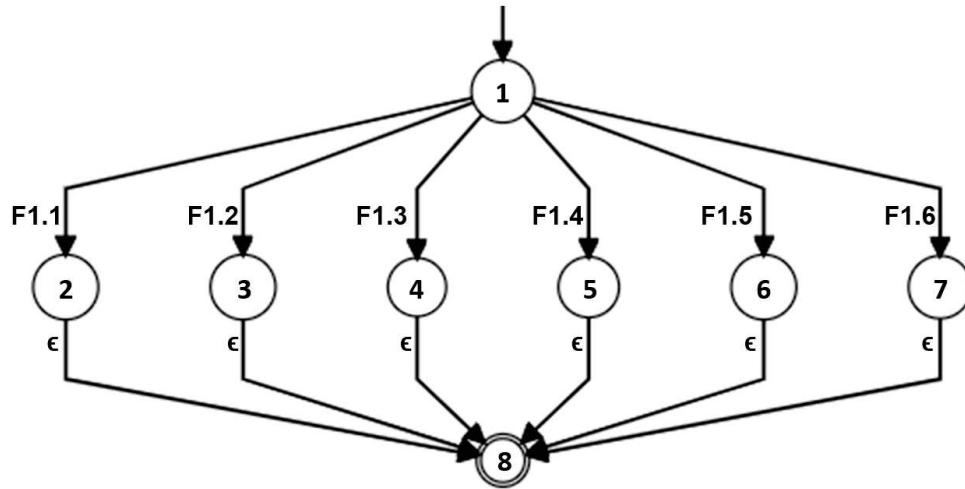
Figura 39 – Autômato $\mathcal{A}^{(10)}$ - Mutaç o

Fonte: Elaborado pelo autor

Tabela 63 – Composi o do aut mato $\mathcal{A}^{(10)}$ - Muta o

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Fun�o de Atingibilidade (Express�o Regular)
Muta�o	$e^{(29)}$	Sincronizante	$(F1 F2)$
Swap	$e^{(30)}$	Sincronizante	$(F1.1 F1.2 F1.3 F1.4 F1.5 F1.6)$
Invers�o de Bit	$e^{(37)}$	Sincronizante	$(F2.1 F2.2 F2.3 F2.4 F2.5 F2.6)$

Fonte: Elaborado pelo autor

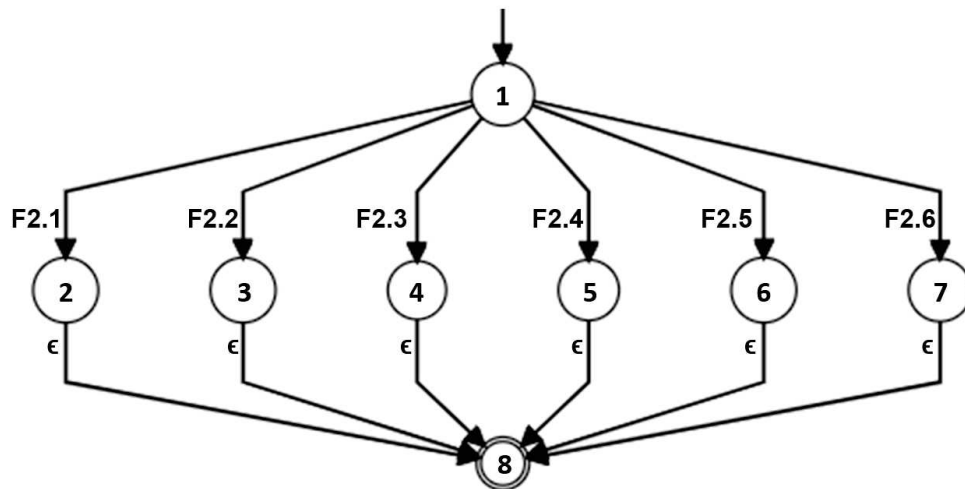
Figura 40 – Autômato $\mathcal{A}^{(11)}$ - Swap

Fonte: Elaborado pelo autor

Tabela 64 – Composição do autômato $\mathcal{A}^{(11)}$ - Swap

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Swap	$e^{(30)}$	Sincronizante	$(F1.1 F1.2 F1.3 F1.4 F1.5 F1.6)$
Taxa de mutação 1%	$e^{(31)}$	Local	
Taxa de mutação 2%	$e^{(32)}$	Local	
Taxa de mutação 5%	$e^{(33)}$	Local	
Taxa de mutação 10%	$e^{(34)}$	Local	
Taxa de mutação 15%	$e^{(35)}$	Local	
Taxa de mutação 20%	$e^{(36)}$	Local	

Fonte: Elaborado pelo autor

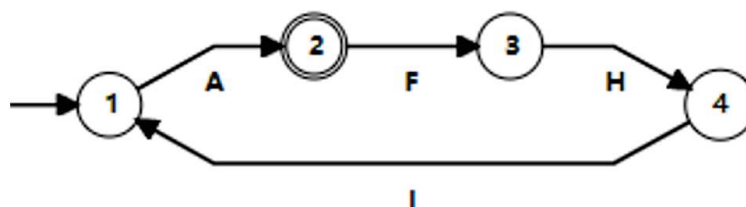
Figura 41 – Autômato $\mathcal{A}^{(12)}$ - Inversão de Bit

Fonte: Elaborado pelo autor

Tabela 65 – Composição do autômato $\mathcal{A}^{(12)}$ - Inversão de Bit

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Inversão de Bit	$e^{(37)}$	Sincronizante	$(F2.1 F2.2 F2.3 F2.4 F2.5 F2.6)$
Taxa de mutação 1%	$e^{(38)}$	Local	
Taxa de mutação 2%	$e^{(39)}$	Local	
Taxa de mutação 5%	$e^{(40)}$	Local	
Taxa de mutação 10%	$e^{(41)}$	Local	
Taxa de mutação 15%	$e^{(42)}$	Local	
Taxa de mutação 20%	$e^{(43)}$	Local	

Fonte: Elaborado pelo autor

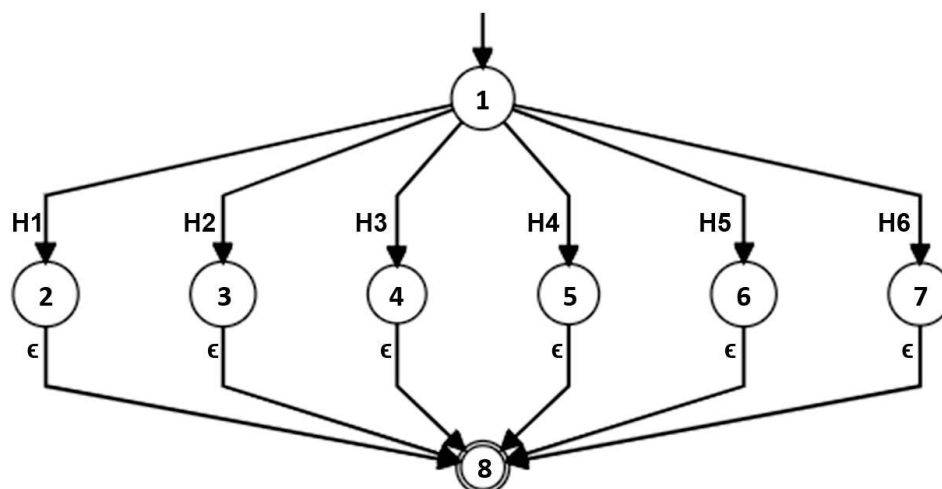
Figura 42 – Autômato $\mathcal{A}^{(13)}$ - Arrefecimento Simulado

Fonte: Elaborado pelo autor

Tabela 66 – Composição do autômato $\mathcal{A}^{(13)}$ - Arrefecimento Simulado

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Arrefecimento Simulado			$A(FHIA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Criação solução vizinha	$e^{(29)}$	Sincronizante	$(F1 F2)$
Temperatura Inicial	$e^{(45)}$	Sincronizante	$(H1 H2 H3 H4 H5 H6)$
Redução da Temperatura	$e^{(52)}$	Sincronizante	$(I1 I2 I3 I4 I5 I6)$

Fonte: Elaborado pelo autor

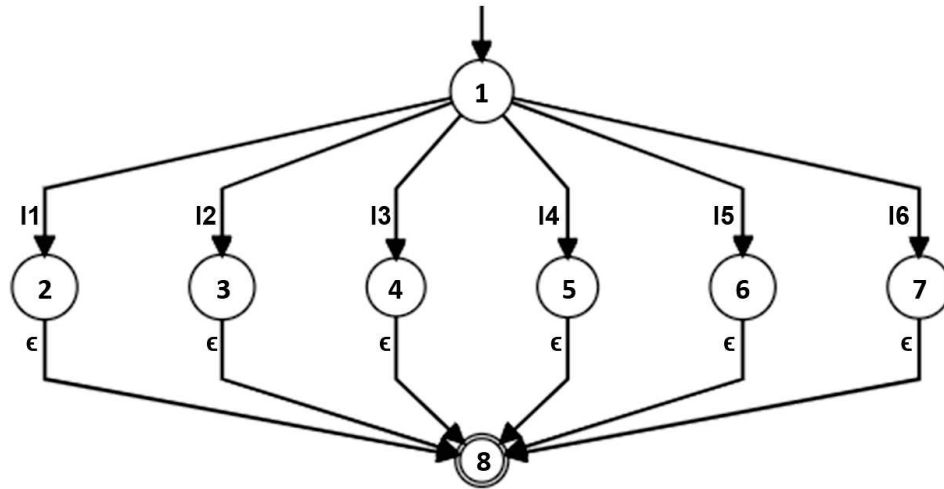
Figura 43 – Autômato $\mathcal{A}^{(14)}$ - Temperatura Inicial

Fonte: Elaborado pelo autor

Tabela 67 – Composição do autômato $\mathcal{A}^{(14)}$ - Temperatura Inicial

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Temperatura Inicial	$e^{(45)}$	Sincronizante	$(H1 H2 H3 H4 H5 H6)$
Temperatura Inicial 100	$e^{(46)}$	Local	
Temperatura Inicial 500	$e^{(47)}$	Local	
Temperatura Inicial 1000	$e^{(48)}$	Local	
Temperatura Inicial 2500	$e^{(49)}$	Local	
Temperatura Inicial 5000	$e^{(50)}$	Local	
Temperatura Inicial 10000	$e^{(51)}$	Local	

Fonte: Elaborado pelo autor

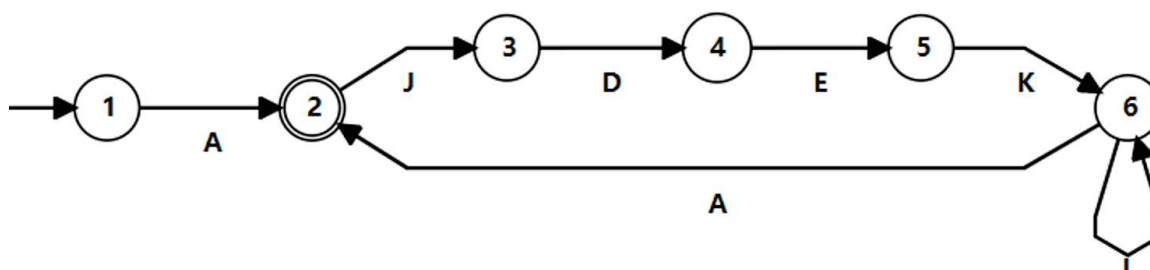
Figura 44 – Autômato $\mathcal{A}^{(15)}$ - Redução da Temperatura

Fonte: Elaborado pelo autor

Tabela 68 – Composição do autômato $\mathcal{A}^{(15)}$ - Redução da Temperatura

Evento (Mecanismo Heurístico)		Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Redução da Temperatura	$e^{(52)}$	Sincronizante	I	$(I1 I2 I3 I4 I5 I6)$
Redução da Temperatura 0,001	$e^{(53)}$	Local	I1	
Redução da Temperatura 0,01	$e^{(54)}$	Local	I2	
Redução da Temperatura 0,05	$e^{(55)}$	Local	I3	
Redução da Temperatura 0,1	$e^{(56)}$	Local	I4	
Redução da Temperatura 0,5	$e^{(57)}$	Local	I5	
Redução da Temperatura 1,0	$e^{(58)}$	Local	I6	

Fonte: Elaborado pelo autor

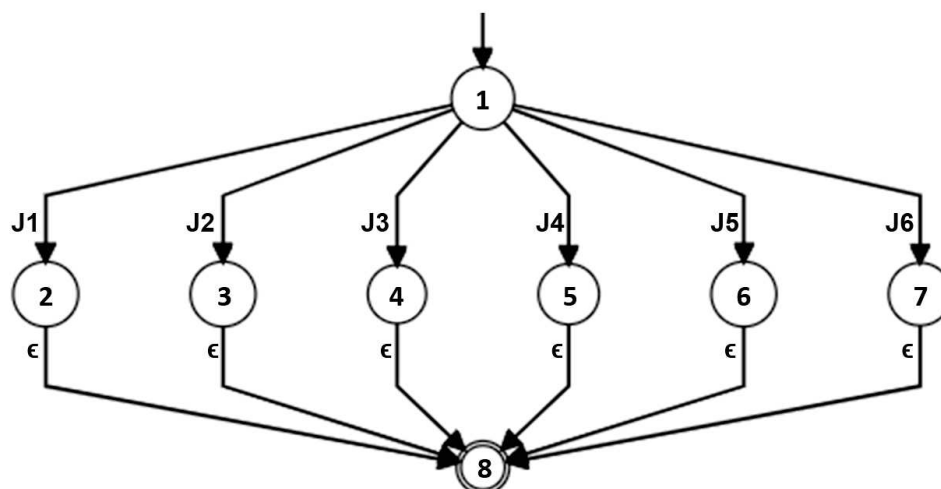
Figura 45 – Autômato $\mathcal{A}^{(16)}$ - Busca Dispersa

Fonte: Elaborado pelo autor

Tabela 69 – Composição do autômato $\mathcal{A}^{(16)}$ - Busca Dispersa

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Busca Dispersa			$A(JDEKL^*A)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Tamanho Conjunto de Diversificação	$e^{(59)}$	Sincronizante	$(J1 J2 J3 J4 J5 J6)$
Seleção de um subconjunto	$e^{(7)}$	Sincronizante	$(D1 D2)$
Recombinação do subconjunto	$e^{(22)}$	Sincronizante	$(E1 E2 E3 E4 E5 E6)$

Fonte: Elaborado pelo autor

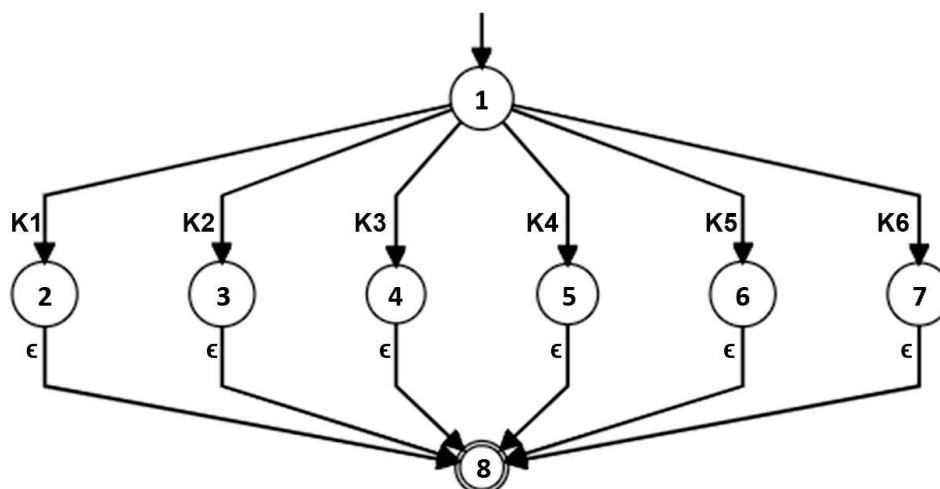
Figura 46 – Autômato $\mathcal{A}^{(17)}$ - Tamanho Conjunto de Diversificação

Fonte: Elaborado pelo autor

Tabela 70 – Composição do autômato $\mathcal{A}^{(17)}$ - Tamanho Conjunto de Diversificação

Evento (Mecanismo Heurístico)		Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Tamanho Conjunto de Diversificação	$e^{(59)}$	Sincronizante	J	$(J1 J2 J3 J4 J5 J6)$
Conjunto de Diversificação 25	$e^{(60)}$	Local	J1	
Conjunto de Diversificação 50	$e^{(61)}$	Local	J2	
Conjunto de Diversificação 75	$e^{(62)}$	Local	J3	
Conjunto de Diversificação 100	$e^{(63)}$	Local	J4	
Conjunto de Diversificação 150	$e^{(64)}$	Local	J5	
Conjunto de Diversificação 200	$e^{(65)}$	Local	J6	

Fonte: Elaborado pelo autor

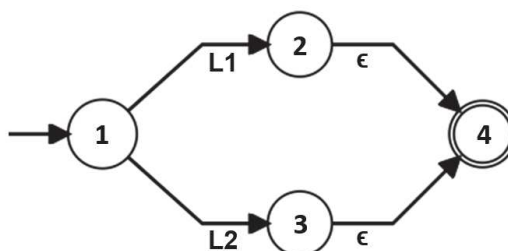
Figura 47 – Autômato $\mathcal{A}^{(18)}$ - Tamanho Conjunto de Referência

Fonte: Elaborado pelo autor

Tabela 71 – Composição do autômato $\mathcal{A}^{(18)}$ - Tamanho Conjunto de Referência

Evento (Mecanismo Heurístico)		Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Tamanho Conjunto de Referência	$e^{(66)}$	Sincronizante	K	$(K1 K2 K3 K4 K5 K6)$
Conjunto de Referência 2	$e^{(67)}$	Local	K1	
Conjunto de Referência 5	$e^{(68)}$	Local	K2	
Conjunto de Referência 10	$e^{(69)}$	Local	K3	
Conjunto de Referência 15	$e^{(70)}$	Local	K4	
Conjunto de Referência 20	$e^{(71)}$	Local	K5	
Conjunto de Referência 25	$e^{(72)}$	Local	K6	

Fonte: Elaborado pelo autor

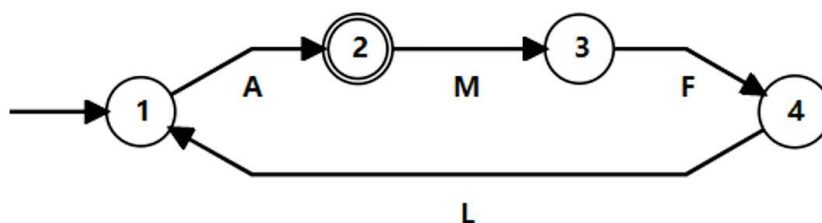
Figura 48 – Autômato $\mathcal{A}^{(19)}$ - Busca Local

Fonte: Elaborado pelo autor

Tabela 72 – Composição do autômato $\mathcal{A}^{(19)}$ - Busca Local

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Busca Local	$e^{(73)}$	Sincronizante	$(L1 L2)$
2-opt	$e^{(74)}$	Local	
3-opt	$e^{(75)}$	Local	

Fonte: Elaborado pelo autor

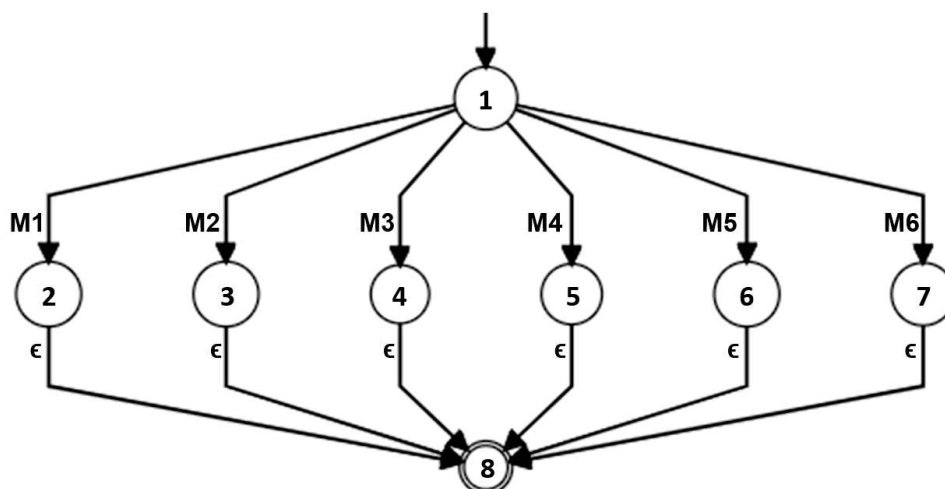
Figura 49 – Autômato $\mathcal{A}^{(20)}$ - Busca em Vizinhaça Variável

Fonte: Elaborado pelo autor

Tabela 73 – Composição do autômato $\mathcal{A}^{(20)}$ - Busca em Vizinhaça Variável

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Busca em Vizinhaça Variável			$A(MFLA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Vizinhaças	$e^{(76)}$	Sincronizante	$(M1 M2 M3 M4 M5 M6)$
Geração da vizinhaça	$e^{(29)}$	Sincronizante	$(F1 F2)$
Busca Local	$e^{(73)}$	Sincronizante	$(L1 L2)$

Fonte: Elaborado pelo autor

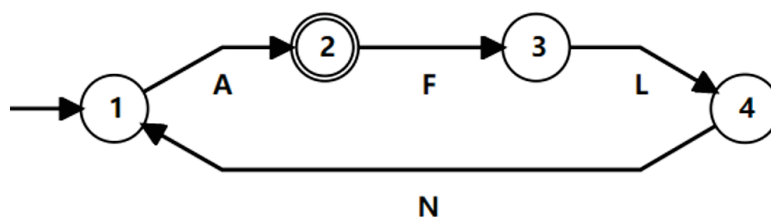
Figura 50 – Autômato $\mathcal{A}^{(21)}$ - Vizinhanças

Fonte: Elaborado pelo autor

Tabela 74 – Composição do autômato $\mathcal{A}^{(21)}$ - Vizinhanças

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Vizinhanças	$e^{(76)}$	Sincronizante	M
Vizinhanças 5	$e^{(77)}$	Local	$M1$
Vizinhanças 10	$e^{(78)}$	Local	$M2$
Vizinhanças 25	$e^{(79)}$	Local	$M3$
Vizinhanças 50	$e^{(80)}$	Local	$M4$
Vizinhanças 75	$e^{(81)}$	Local	$M5$
Vizinhanças 100	$e^{(82)}$	Local	$M6$

Fonte: Elaborado pelo autor

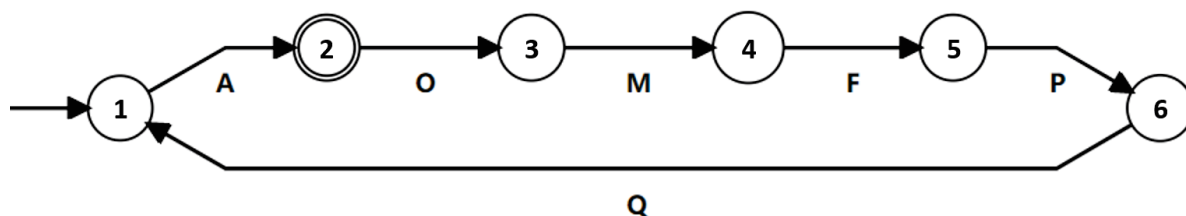
Figura 51 – Autômato $\mathcal{A}^{(22)}$ - Busca Local Iterada

Fonte: Elaborado pelo autor

Tabela 75 – Composição do autômato $\mathcal{A}^{(22)}$ - Busca Local Iterada

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Busca Local Iterada			$A (FLNA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	A
Perturbação	$e^{(29)}$	Sincronizante	F
Busca Local	$e^{(73)}$	Sincronizante	L
Critério de aceitação	$e^{(83)}$	Local	N

Fonte: Elaborado pelo autor

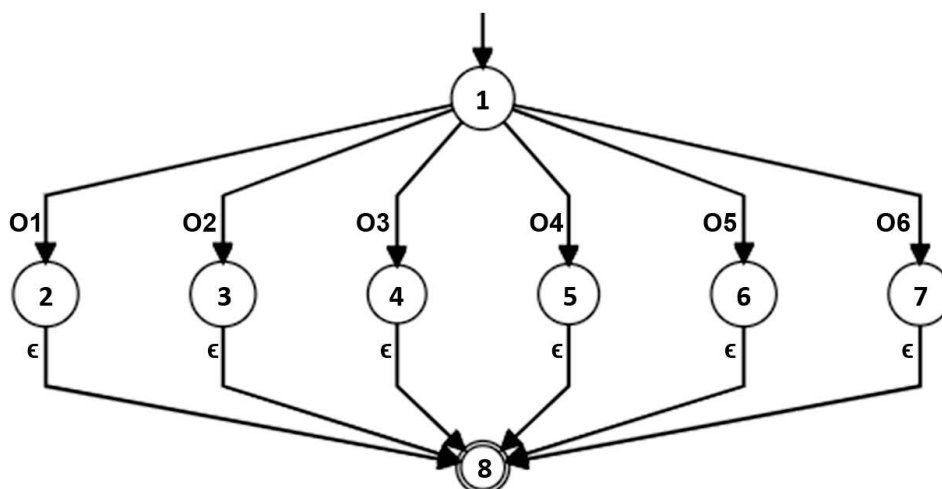
Figura 52 – Autômato $\mathcal{A}^{(23)}$ - Busca Tabu

Fonte: Elaborado pelo autor

Tabela 76 – Composição do autômato $\mathcal{A}^{(23)}$ - Busca Tabu

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Busca Tabu			$A(O M F P Q A)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Tamanho Lista Tabu	$e^{(84)}$	Sincronizante	$(O1 O2 O3 O4 O5 O6)$
NBMax (Vizinhanças)	$e^{(76)}$	Sincronizante	$(M1 M2 M3 M4 M5 M6)$
Geração da vizinhança	$e^{(29)}$	Sincronizante	$(F1 F2)$
Atualização da Lista Tabu	$e^{(91)}$	Local	
Critério de aspiração	$e^{(92)}$	Local	

Fonte: Elaborado pelo autor

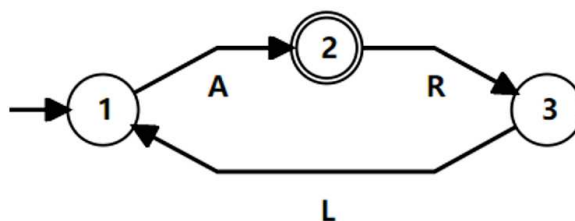
Figura 53 – Autômato $\mathcal{A}^{(24)}$ - Tamanho Lista Tabu

Fonte: Elaborado pelo autor

Tabela 77 – Composição do autômato $\mathcal{A}^{(24)}$ - Tamanho Lista Tabu

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Tamanho Lista Tabu	$e^{(84)}$	Sincronizante	$(O1 O2 O3 O4 O5 O6)$
Lista Tabu 10	$e^{(85)}$	Local	
Lista Tabu 25	$e^{(86)}$	Local	
Lista Tabu 50	$e^{(87)}$	Local	
Lista Tabu 75	$e^{(88)}$	Local	
Lista Tabu 100	$e^{(89)}$	Local	
Lista Tabu 200	$e^{(90)}$	Local	

Fonte: Elaborado pelo autor

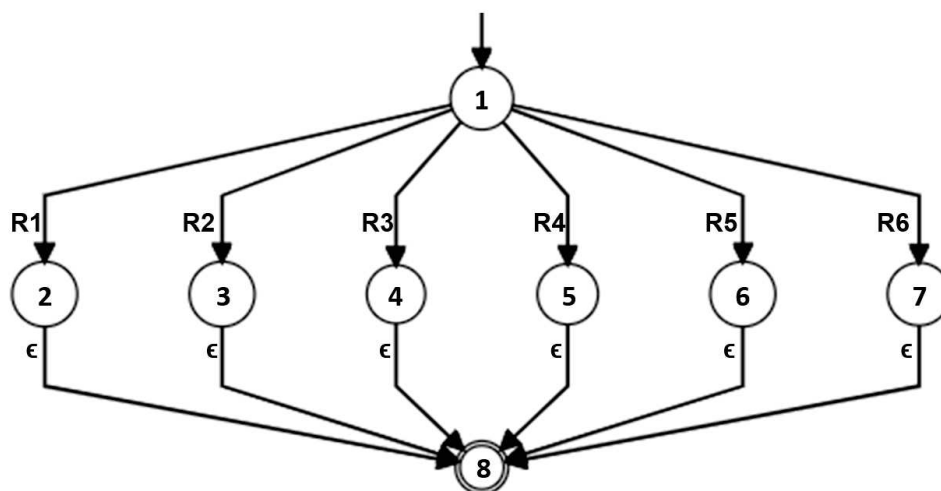
Figura 54 – Autômato $\mathcal{A}^{(25)}$ - GRASP

Fonte: Elaborado pelo autor

Tabela 78 – Composição do autômato $\mathcal{A}^{(25)}$ - GRASP

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
GRASP			$A(RLA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Construção Aleatória	$e^{(93)}$	Sincronizante	$(R1 R2 R3 R4 R5 R6)$
Busca Local	$e^{(73)}$	Sincronizante	$(L1 L2)$

Fonte: Elaborado pelo autor

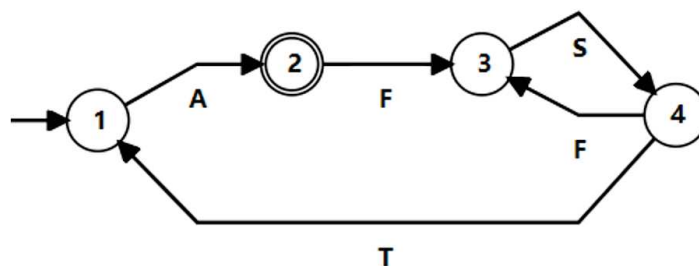
Figura 55 – Autômato $\mathcal{A}^{(26)}$ - Construção Aleatória

Fonte: Elaborado pelo autor

Tabela 79 – Composição do autômato $\mathcal{A}^{(26)}$ - Construção Aleatória

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Construção Aleatória	$e^{(93)}$	Sincronizante	$(R1 R2 R3 R4 R5 R6)$
Lista de Candidatos $\alpha 0,1$	$e^{(94)}$	Local	
Lista de Candidatos $\alpha 0,2$	$e^{(95)}$	Local	
Lista de Candidatos $\alpha 0,3$	$e^{(96)}$	Local	
Lista de Candidatos $\alpha 0,5$	$e^{(97)}$	Local	
Lista de Candidatos $\alpha 0,7$	$e^{(98)}$	Local	
Lista de Candidatos $\alpha 0,9$	$e^{(99)}$	Local	

Fonte: Elaborado pelo autor

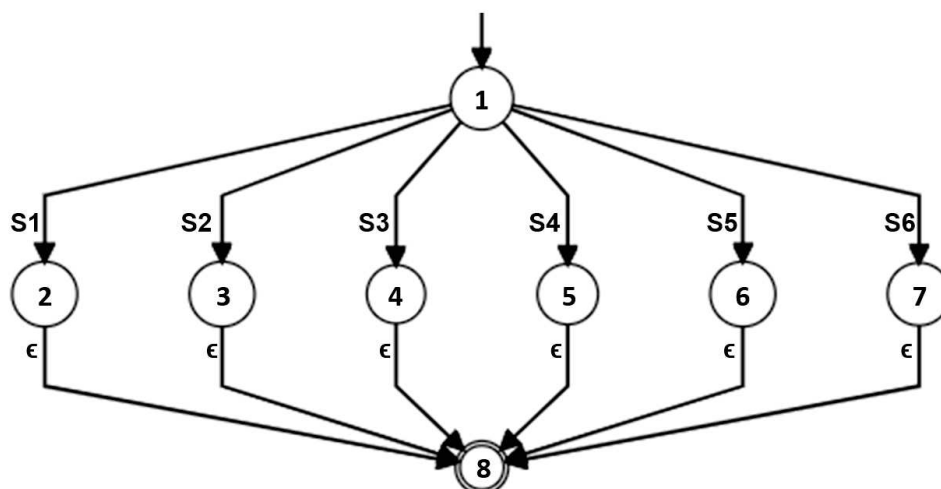
Figura 56 – Autômato $\mathcal{A}^{(27)}$ - Otimização por Colônia de Formigas

Fonte: Elaborado pelo autor

Tabela 80 – Composição do autômato $\mathcal{A}^{(27)}$ - Otimização por Colônia de Formigas

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Otimização por Colônia de Formigas			$A((FS) + TA)^*$
Critério de parada	$e^{(1)}$	Sincronizante	$(A1 A2 A3)$
Construção da Solução	$e^{(29)}$	Sincronizante	$(F1 F2)$
Atração do Feromônio	$e^{(100)}$	Sincronizante	$(S1 S2 S3 S4 S5 S6)$
Evaporação do Feromônio	$e^{(107)}$	Sincronizante	$(T1 T2 T3 T4 T5 T6)$

Fonte: Elaborado pelo autor

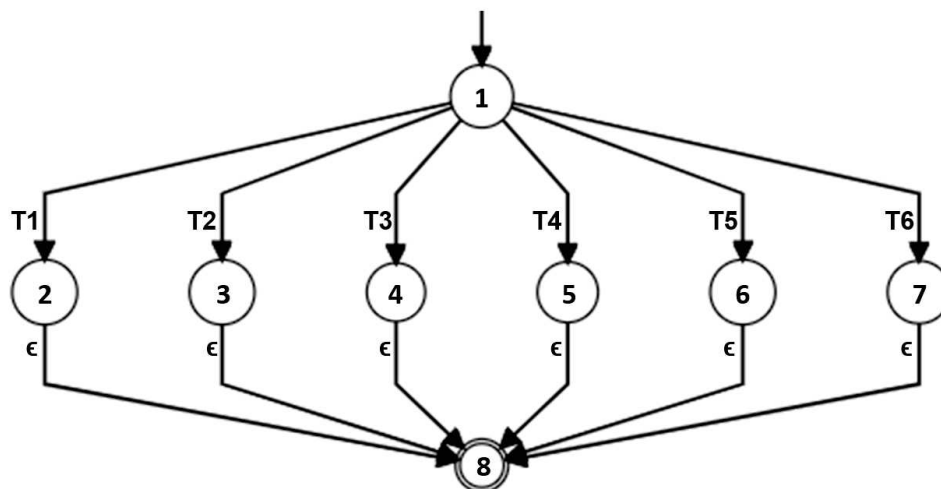
Figura 57 – Autômato $\mathcal{A}^{(28)}$ - Atração do Feromônio

Fonte: Elaborado pelo autor

Tabela 81 – Composição do autômato $\mathcal{A}^{(28)}$ - Atração do Feromônio

Evento (Mecanismo Heurístico)		Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Atração do Feromônio	$e^{(100)}$	Sincronizante	S	$(S1 S2 S3 S4 S5 S6)$
Atração do Feromônio 0,5	$e^{(101)}$	Local	S1	
Atração do Feromônio 1,0	$e^{(102)}$	Local	S2	
Atração do Feromônio 2,0	$e^{(103)}$	Local	S3	
Atração do Feromônio 3,0	$e^{(104)}$	Local	S4	
Atração do Feromônio 4,0	$e^{(105)}$	Local	S5	
Atração do Feromônio 5,0	$e^{(106)}$	Local	S6	

Fonte: Elaborado pelo autor

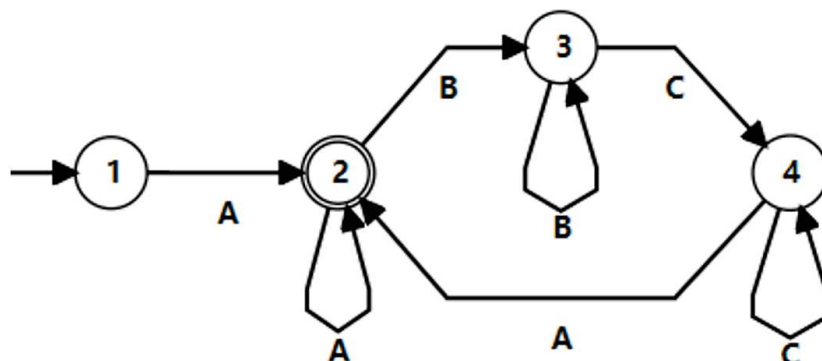
Figura 58 – Autômato $\mathcal{A}^{(29)}$ - Evaporação do Feromônio

Fonte: Elaborado pelo autor

Tabela 82 – Composição do autômato $\mathcal{A}^{(29)}$ - Evaporação do Feromônio

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Evaporação do Feromônio	$e^{(107)}$	Sincronizante	$(T1 T2 T3 T4 T5 T6)$
Evaporação do Feromônio 0,01	$e^{(108)}$	Local	
Evaporação do Feromônio 0,02	$e^{(109)}$	Local	
Evaporação do Feromônio 0,03	$e^{(110)}$	Local	
Evaporação do Feromônio 0,05	$e^{(111)}$	Local	
Evaporação do Feromônio 0,10	$e^{(112)}$	Local	
Evaporação do Feromônio 0,15	$e^{(113)}$	Local	

Fonte: Elaborado pelo autor

Figura 59 – Autômato $\mathcal{A}^{(30)}$ - Hiper-heurística

Fonte: Elaborado pelo autor

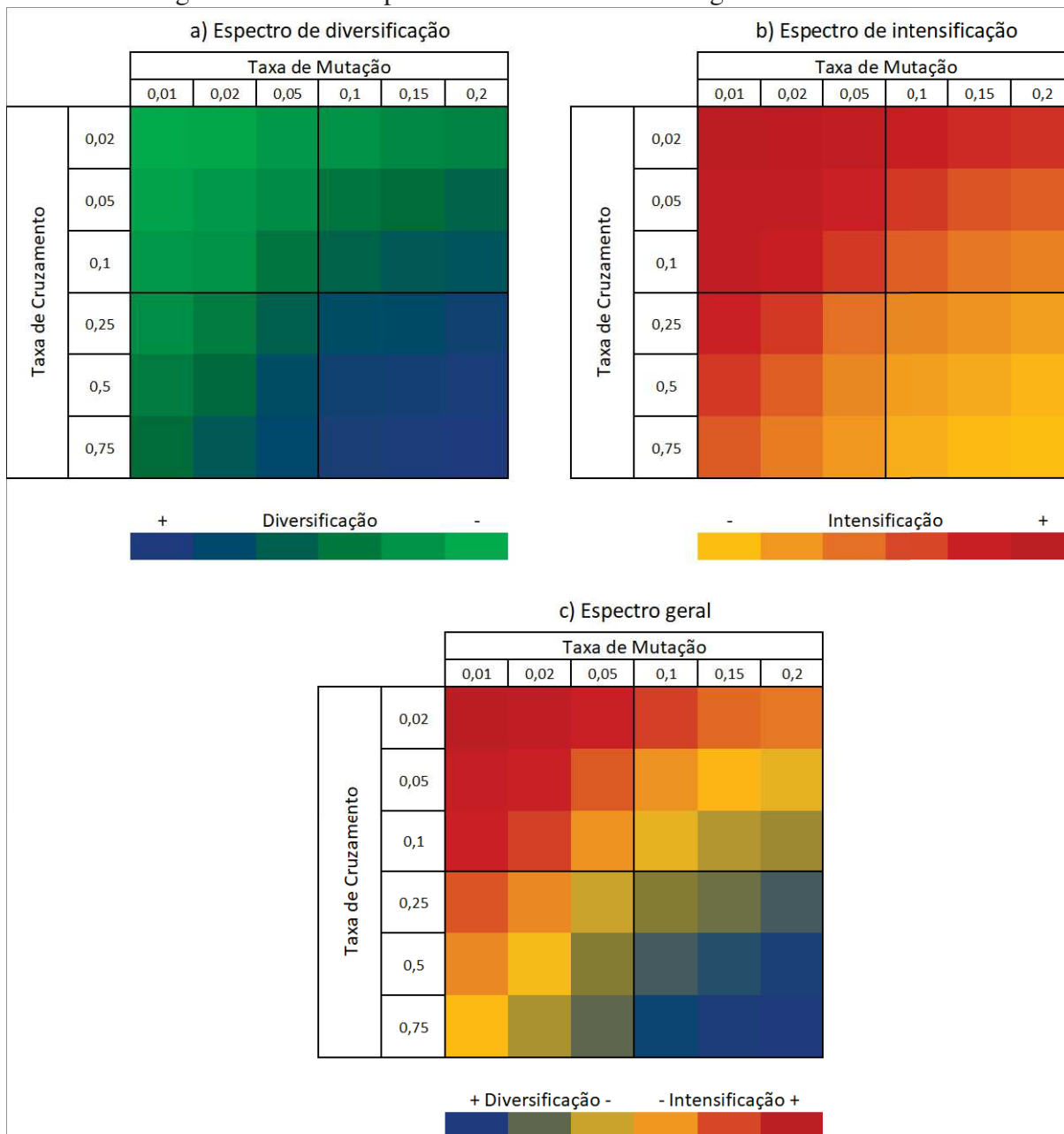
Tabela 83 – Composição do autômato $\mathcal{A}^{(30)}$ - Hiper-heurística

Evento (Mecanismo Heurístico)	Tipo do evento	Elemento Funcional (Alfabeto)	Função de Atingibilidade (Expressão Regular)
Hiper-heurística			$A^+(B+C+A)^*$

Fonte: Elaborado pelo autor

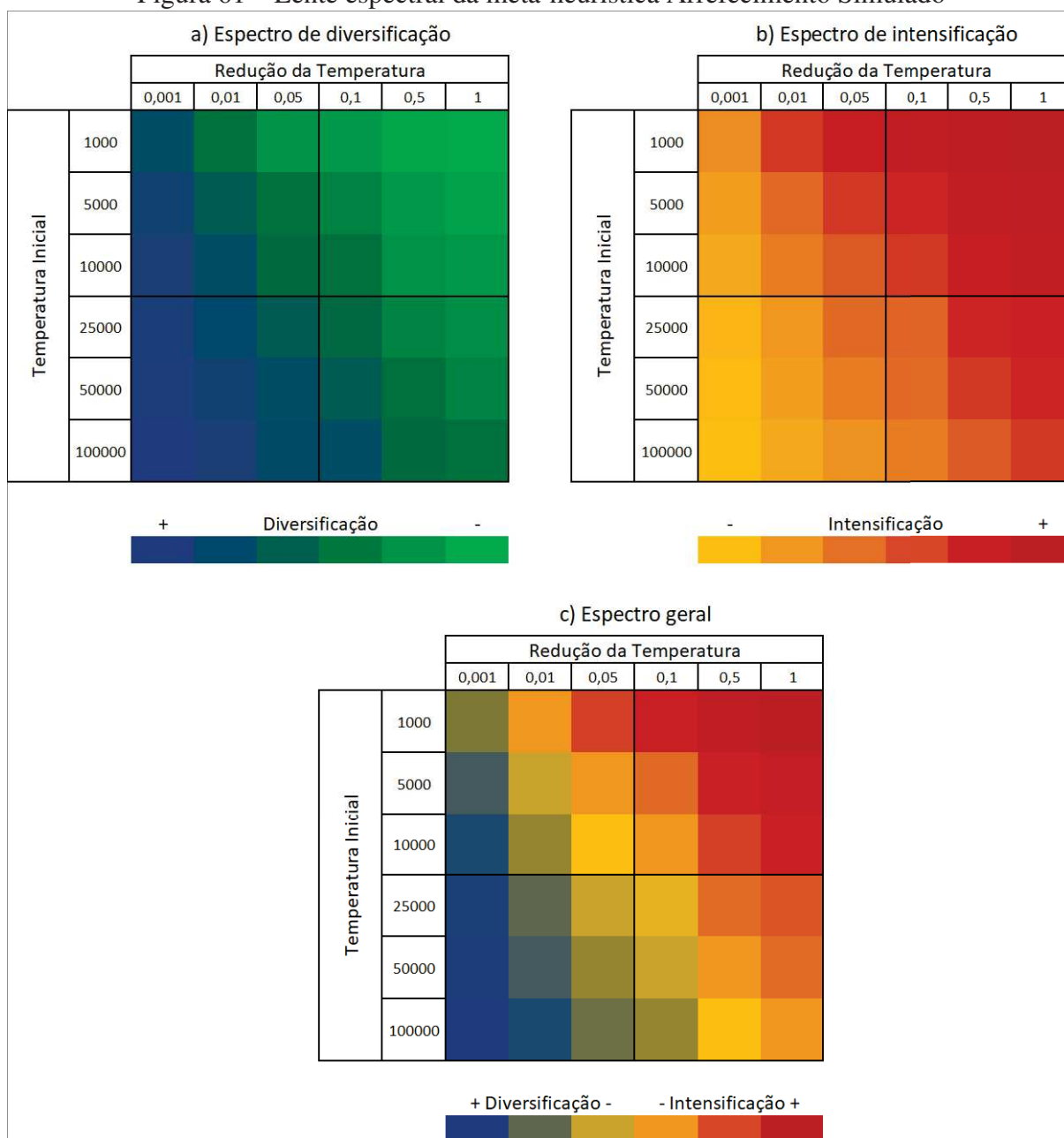
APÊNDICE B – LENTES ESPECTRAIS DE ANÁLISE DAS META-HEURÍSTICAS

Figura 60 – Lente espectral da meta-heurística Algoritmos Genéticos



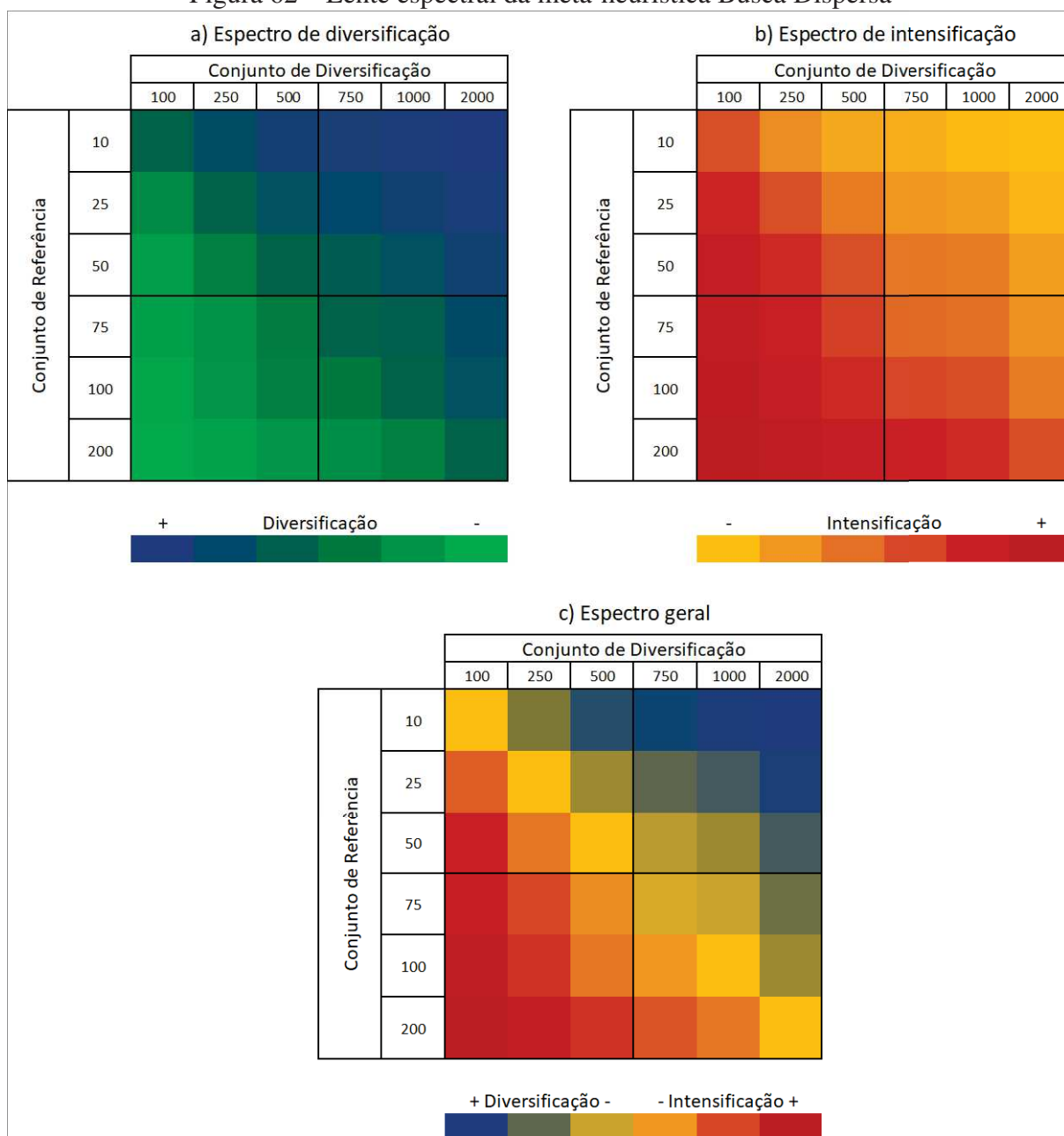
Fonte: Elaborado pelo autor

Figura 61 – Lente espectral da meta-heurística Arrefecimento Simulado



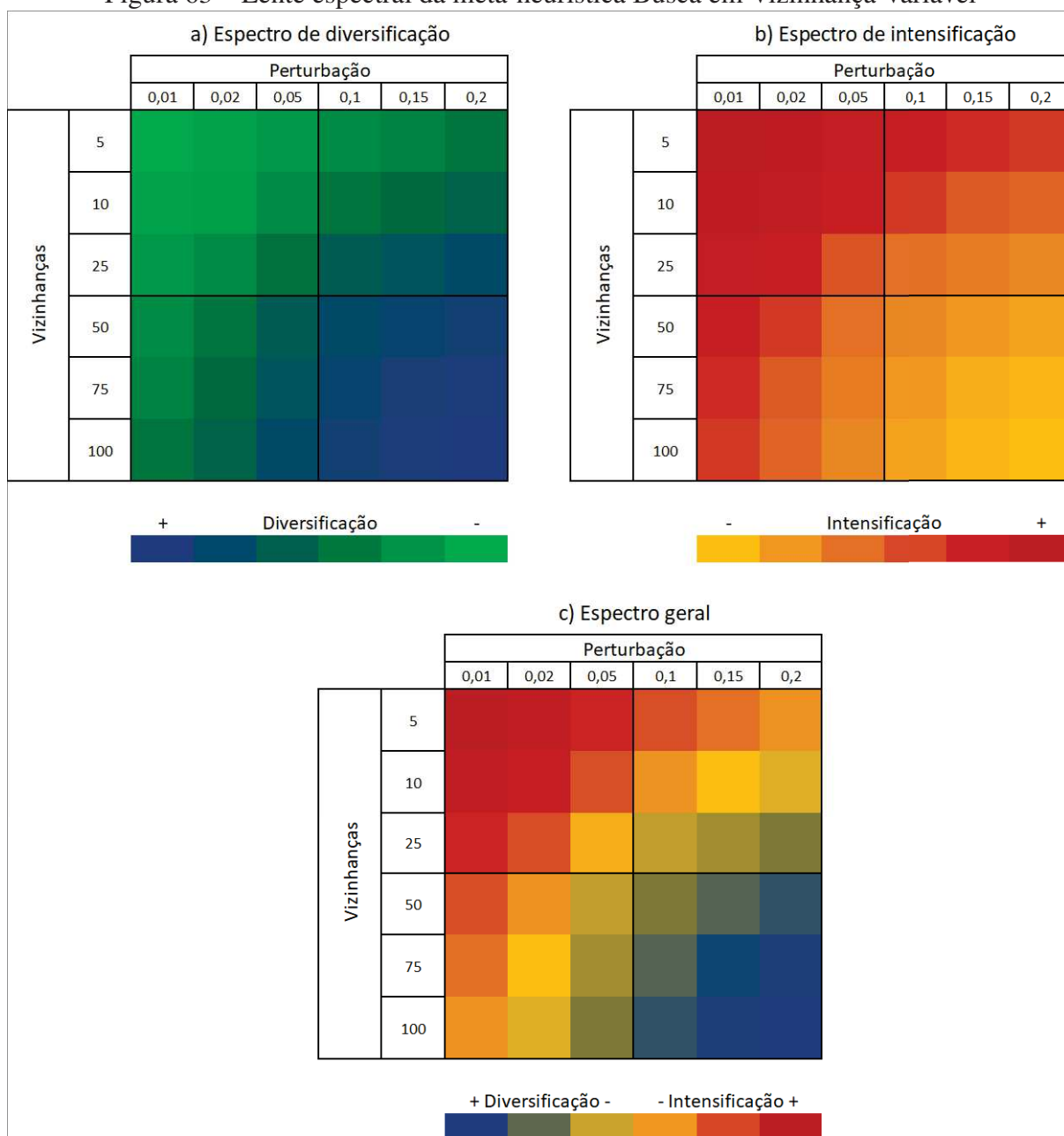
Fonte: Elaborado pelo autor

Figura 62 – Lente espectral da meta-heurística Busca Dispersa



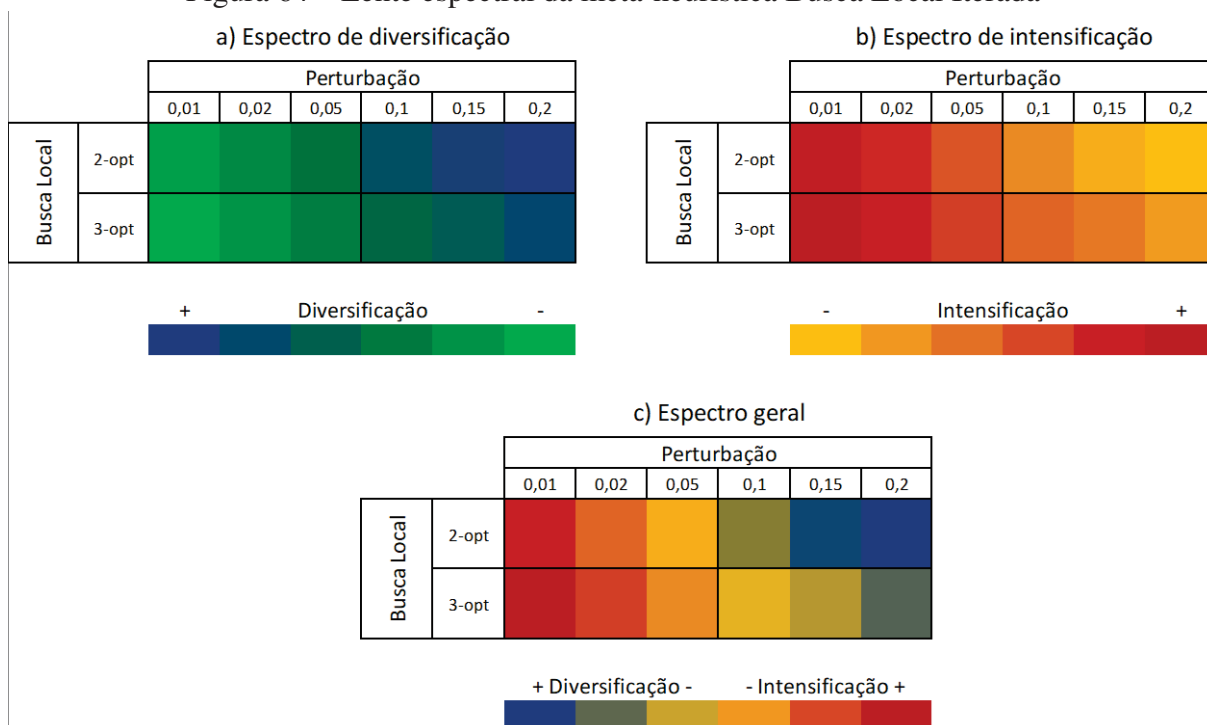
Fonte: Elaborado pelo autor

Figura 63 – Lente espectral da meta-heurística Busca em Vizinhança Variável



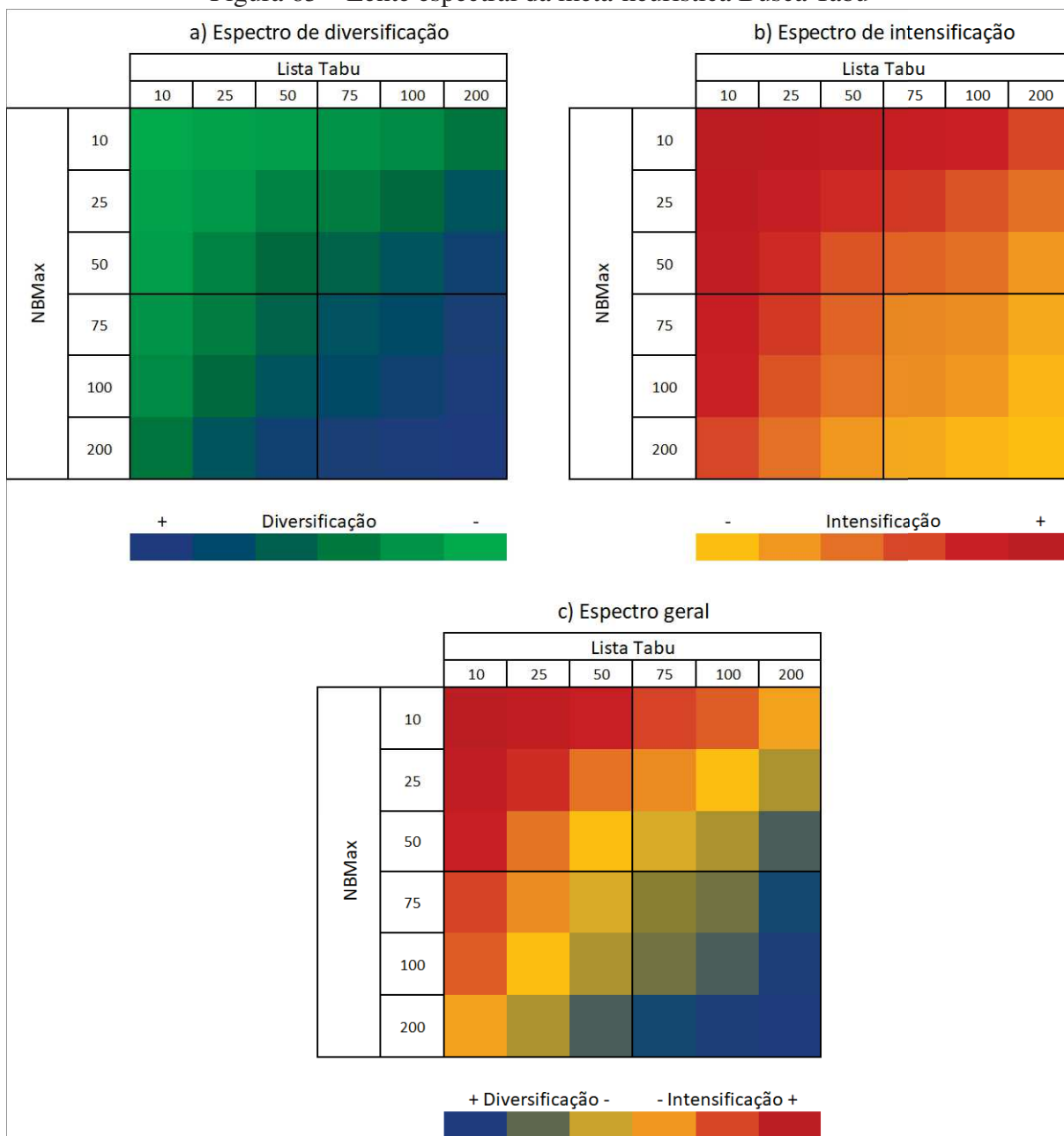
Fonte: Elaborado pelo autor

Figura 64 – Lente espectral da meta-heurística Busca Local Iterada



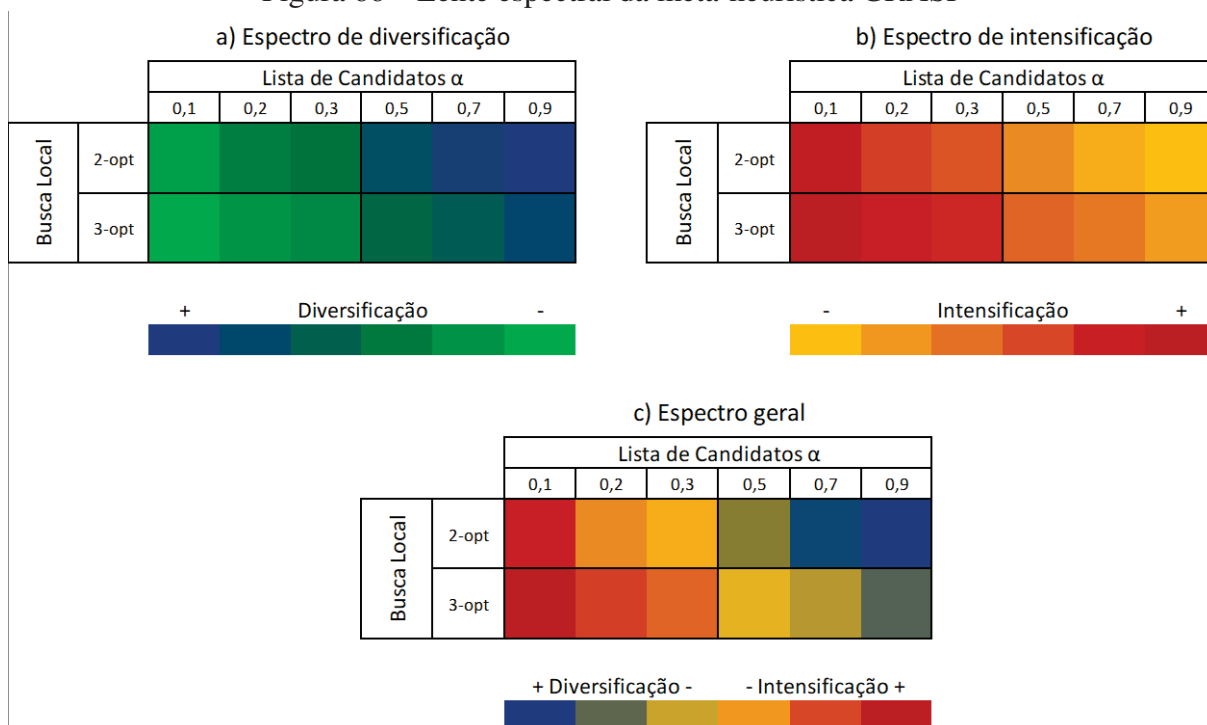
Fonte: Elaborado pelo autor

Figura 65 – Lente espectral da meta-heurística Busca Tabu



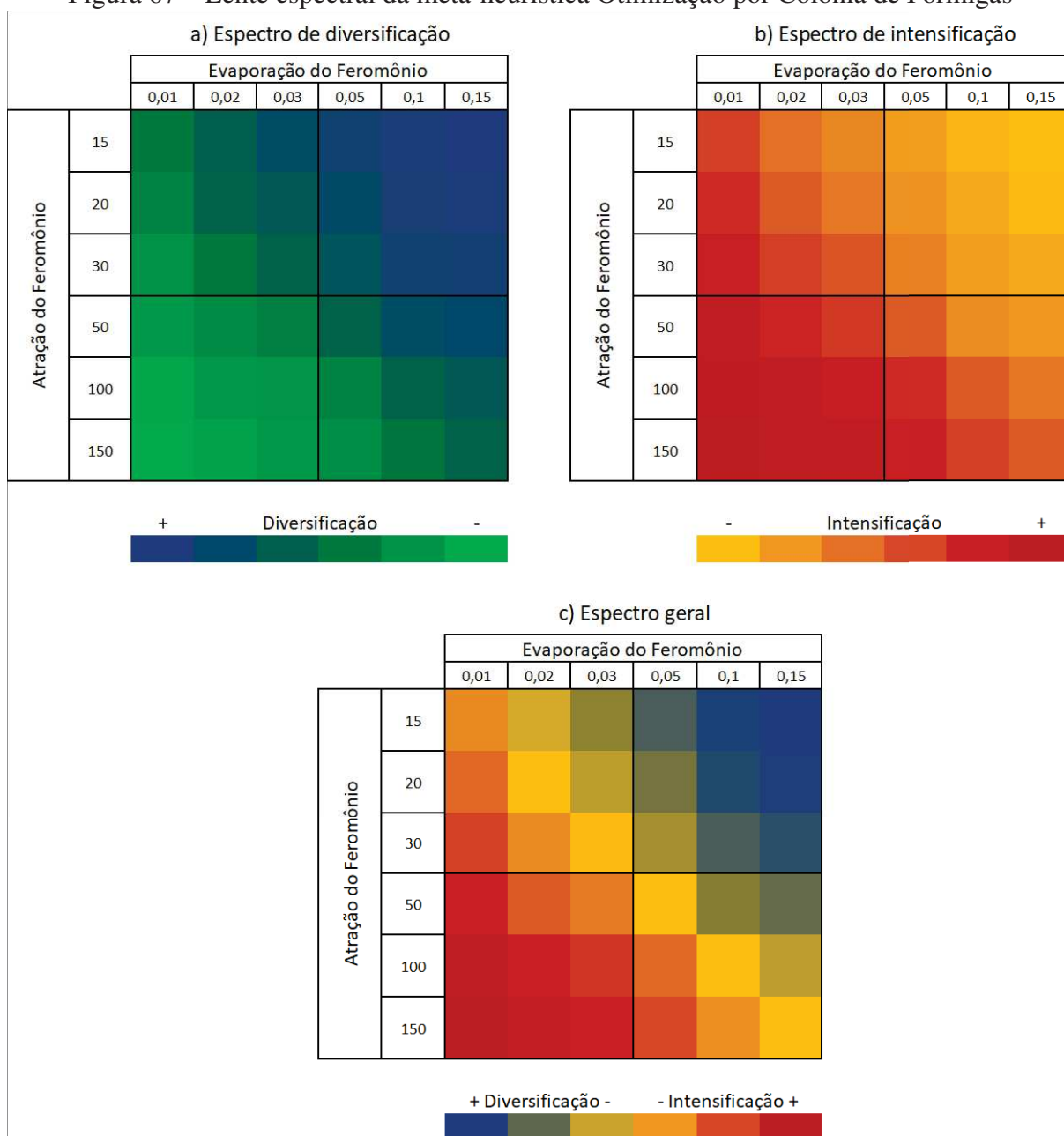
Fonte: Elaborado pelo autor

Figura 66 – Lente espectral da meta-heurística GRASP



Fonte: Elaborado pelo autor

Figura 67 – Lente espectral da meta-heurística Otimização por Colônia de Formigas



Fonte: Elaborado pelo autor