



UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
CENTRO DE CIÊNCIAS EXATAS E DA TERRA  
DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA  
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO  
DOUTORADO ACADÊMICO EM CIÊNCIA DA COMPUTAÇÃO



# O Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas

Sílvia Maria Diniz Monteiro Maia

Natal-RN  
Dezembro de 2013

Sílvia Maria Diniz Monteiro Maia

# O Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte como requisito parcial para a obtenção do grau de Doutor em Ciência da Computação.

*Linha de pesquisa:* Algoritmos Experimentais

Orientador

Profa. Dra. Elizabeth Ferreira Gouvêa Goldbarg

PPGSC – PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS E COMPUTAÇÃO  
DIMAP – DEPARTAMENTO DE INFORMÁTICA E MATEMÁTICA APLICADA  
CCET – CENTRO DE CIÊNCIAS EXATAS E DA TERRA  
UFRN – UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE

Natal-RN

Dezembro de 2013

Catálogo da Publicação na Fonte. UFRN / Biblioteca Central  
Zila Mamede.

Maia, Sílvia Maria Diniz Monteiro

O problema biobjetivo da árvore geradora quadrática em adjacência de arestas / Sílvia Maria Diniz Monteiro Maia. – Natal, RN, 2013.  
313 f.; il.

Orientadora: Elizabeth Ferreira Gouvêa Goldberg.

Tese (Doutorado) – Universidade Federal do Rio Grande do Norte.  
Centro de Ciências Exatas e da Terra. Programa de Pós-Graduação em  
Sistemas e Computação.

1. Algoritmos experimentais – Tese. 2. Algoritmos exatos – Tese.  
3. Metaheurísticas – Tese. 4. Otimização multiobjetivo – Tese I.  
Goldberg, Elizabeth Ferreira Gouvêa. II. Universidade Federal do Rio  
Grande do Norte. III. Título.

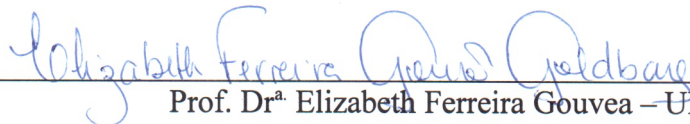
RN/UF/BCZM

CDU 004.021

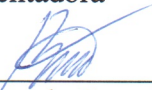
SILVIA MARIA DINIZ MONTEIRO MAIA

## O Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas

Esta Tese foi julgada adequada para a obtenção do título de doutor em Ciência da Computação e aprovado em sua forma final pelo Programa de Pós-Graduação em Sistemas e Computação do Departamento de Informática e Matemática Aplicada da Universidade Federal do Rio Grande do Norte.

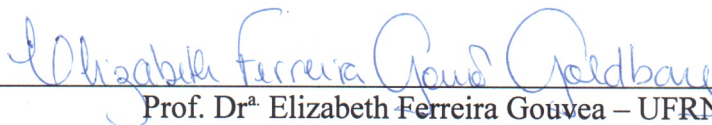


Prof. Dr.ª Elizabeth Ferreira Gouveia – UFRN  
Orientadora

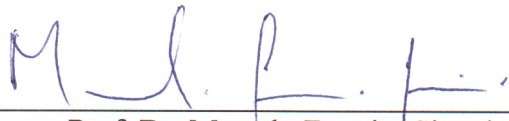


Prof. Dr. Bruno Motta de Carvalho – UFRN  
Vice-coordenador do Programa

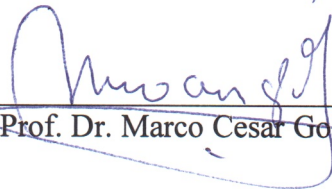
### Banca Examinadora



Prof. Dr.ª Elizabeth Ferreira Gouveia – UFRN  
Presidente



Prof. Dr. Marcelo Ferreira Siqueira – UFRN



Prof. Dr. Marco Cesar Goldberg – UFRN



Prof. Dr. André Luís Vasconcelos Coelho – UNIFOR



Prof. Dr. Lucídio dos Anjos Formiga Cabral – UFPB

Dezembro, 2013

Aos meus pais, Luiz e Ivone:  
pela fé, perseverança, força,  
apoio e amor incondicionais.

Ao meu irmão, Luiz Júnior:  
pelo companheirismo, carinho,  
alegria e pelo melhor presente, Sofia.

Ao meu esposo, Pedro:  
pela dedicação, felicidade, apoio, amor,  
e por estar ao meu lado em todos os momentos.

# Agradecimentos

A Deus, por iluminar meus caminhos e me dar força para seguir sempre em frente, pois só *Ele é o nosso refúgio e a nossa força, socorro que não falta em tempos de aflição*. (Salmo 46.1).

Aos professores Dra. Elizabeth Ferreira Gouvêa Goldberg e Dr. Marco César Goldberg, por serem muito mais que meus orientadores, pelo empenho, competência, disponibilidade, compreensão nos momentos difíceis, críticas construtivas e idéias, que muito colaboraram para a realização deste trabalho. Ao longo de toda a minha formação acadêmica, vocês foram, são e sempre serão, verdadeiros mestres e amigos. O amor de vocês pelo ensino e pesquisa me motiva a cada dia e me faz ter a certeza de que fiz a escolha certa. Tenho vocês como exemplos de pessoas, profissionais, educadores e pesquisadores.

Aos professores Dr. André Luís Vasconcelos Coelho, Dr. Marcelo Ferreira Siqueira e Dr. Marco César Goldberg, pelas valiosas contribuições apresentadas por ocasião da defesa da proposta de tese.

Aos professores Dr. André Luís Vasconcelos Coelho, Dr. Lucídio dos Anjos Formiga Cabral, Dr. Marcelo Ferreira Siqueira e Dr. Marco César Goldberg pela honra de terem aceitado o nosso convite para participar da banca de defesa da tese.

A todos os professores do Departamento de Informática e Matemática Aplicada, os quais contribuíram para minha formação profissional e acadêmica, pelos ensinamentos, apoio, torcida e por estarem sempre dispostos a contribuir, direta ou indiretamente, para que este sonho pudesse ser realizado.

Ao corpo técnico-administrativo do DIMAp, em especial, Hélida Santos e Priscila Farias (secretaria do PPGSC), Vamberto Brito e Anthony Andrade (secretaria do DIMAp) e Denis Medeiros (gerente de redes do DIMAp), pela gentileza, disponibilidade e preocupação em sempre atender, da forma mais eficiente possível, nossas solicitações.

Àqueles a quem Deus ilumina quando quer me ajudar, a quem recorro quando tudo parece faltar, àqueles por quem rezo para Deus abençoar... Vocês são meu exemplo, meu porto seguro, minha alegria. Aos meus pais, Luiz Monteiro dos Santos e Ivone Rodrigues

Diniz Monteiro, e ao meu irmão, Luiz Monteiro dos Santos Júnior, pelos valores ensinados, pelo amor, pela fé e pelo apoio incondicional em todos os momentos.

Aos meus sogros, Pedro Fernandes Maia e Maria das Graças Caetano Maia, pelo carinho, apoio, compreensão e torcida durante esses anos de dedicação aos estudos.

Ao meu esposo, Pedro Petrovitch Caetano Maia, pelo amor, compreensão, carinho, perseverança e por estar sempre ao meu lado em todos os anos desta jornada. Que Deus o abençoe sempre.

Aos colegas do Laboratório de Algoritmos Experimentais, pela amizade, pelos momentos de descontração e de oração, pela força e pelo constante aprendizado.

Ao CNPq/CAPES, pelo apoio financeiro para o desenvolvimento da pesquisa.

*Se eu pudesse deixar algum presente a você, deixaria aceso o sentimento de amar a vida dos seres humanos. A consciência de aprender tudo o que foi ensinado pelo tempo afora. Lembraria os erros que foram cometidos para que não mais se repetissem. A capacidade de escolher novos rumos. Deixaria para você, se pudesse, o respeito àquilo que é indispensável. Além do pão, o trabalho. Além do trabalho, a ação. E, quando tudo mais faltasse, um segredo: o de buscar no interior de si mesmo a resposta e a força para encontrar a saída.*

Mahatma Gandhi



# O Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas

Autor: Sílvia Maria Diniz Monteiro Maia

Orientador(a): Profa. Dra. Elizabeth Ferreira Gouvêa Goldberg

## RESUMO

O problema da Árvore Geradora Mínima Quadrática (AGMQ) é uma versão do problema da Árvore Geradora Mínima na qual se considera, além dos custos lineares tradicionais, uma estrutura de custos quadrática. Tal estrutura quadrática modela efeitos de interação entre pares de arestas. Os custos lineares e quadráticos são somados para compor o custo total da árvore geradora, que deve ser minimizado. Quando as interações são restritas às arestas adjacentes, o problema é denominado Árvore Geradora Mínima Quadrática em Adjacência de Arestas (AGMQA). A AGMQA e a AGMQ são problemas NP-difíceis que modelam diversos problemas de projeto de redes de transporte e distribuição. Em geral, a AGMQA emerge como um modelo mais apropriado para a modelagem de problemas reais. Embora, na literatura, os custos lineares e quadráticos sejam somados, em aplicações reais, os custos linear e quadrático podem ser conflitantes. Neste caso, seria mais interessante considerar os custos separadamente. Neste sentido, a Otimização Multiobjetivo provê uma modelagem mais realista para os problemas da AGMQ e da AGMQA. Uma revisão do estado da arte, até o momento, não foi capaz de encontrar qualquer trabalho que investigue esses problemas sob um ponto de vista biobjetivo. O objetivo desta Tese é, pois, o desenvolvimento de algoritmos exatos e heurísticos para o Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas (AGQA-bi). Para tanto, como fundamentação teórica, discutem-se outros problemas NP-difíceis diretamente relacionados à AGQA-bi, a saber: AGMQ e AGMQA. Algoritmos exatos *backtracking* e *branch-and-bound* são propostos para o problema-alvo desta investigação. Os algoritmos heurísticos desenvolvidos são: busca local *Pareto Local Search*, Busca Tabu com *ejection chain*, Algoritmo Transgenético, NSGA-II e uma hibridização das duas últimas propostas mencionadas denominada NSTA. Os algoritmos propostos são comparados entre si por

meio da análise de seus desempenhos em experimentos computacionais com casos de teste adaptados da literatura da AGMQ. No que se refere aos algoritmos exatos, a análise considera, em especial, o tempo de execução. No caso dos algoritmos heurísticos, além do tempo de execução, a qualidade do conjunto de aproximação gerado é avaliada. Indicadores de qualidade são empregados para aferir tal informação. Ferramentas estatísticas apropriadas são usadas na análise de desempenho dos algoritmos exatos e heurísticos. Para o conjunto de instâncias utilizado e considerando os critérios de qualidade dos conjuntos de aproximação gerados e tempo de execução dos algoritmos, os experimentos mostraram que o algoritmo de Busca Tabu com *ejection chain* obteve melhores resultados e que o algoritmo transgenético ficou em segundo lugar. A busca local PLS obteve soluções de qualidade, mas a um tempo computacional muito alto se comparado às outras (meta)heurísticas. Nesse sentido, ocupa a terceira colocação. Por fim, ficaram os algoritmos NSTA e NSGA-II.

*Palavras-chave:* Algoritmos Experimentais, Algoritmos exatos, Metaheurísticas, Otimização Multiobjetivo, Árvore Geradora Quadrática em Adjacência de Arestas Biobjetivo.

# The Biobjective Adjacent Only Quadratic Spanning Tree Problem

Author: Sílvia Maria Diniz Monteiro Maia

Supervisor: Prof. D.Sc. Elizabeth Ferreira Gouvêa Goldberg

## ABSTRACT

The Quadratic Minimum Spanning Tree Problem (QMST) is a version of the Minimum Spanning Tree Problem in which, besides the traditional linear costs, there is a quadratic structure of costs. This quadratic structure models interaction effects between pairs of edges. Linear and quadratic costs are added up to constitute the total cost of the spanning tree, which must be minimized. When these interactions are restricted to adjacent edges, the problem is named Adjacent Only Quadratic Minimum Spanning Tree (AQ MST). AQ MST and QMST are NP-hard problems that model several problems of transport and distribution networks design. In general, AQ MST arises as a more suitable model for real problems. Although, in literature, linear and quadratic costs are added, in real applications, they may be conflicting. In this case, it may be interesting to consider these costs separately. In this sense, Multiobjective Optimization provides a more realistic model for QMST and AQ MST. A review of the state-of-the-art, so far, was not able to find papers regarding these problems under a biobjective point of view. Thus, the objective of this Thesis is the development of exact and heuristic algorithms for the Biobjective Adjacent Only Quadratic Spanning Tree Problem (bi-AQST). In order to do so, as theoretical foundation, other NP-hard problems directly related to bi-AQST are discussed: the QMST and AQ MST problems. Bracktracking and branch-and-bound exact algorithms are proposed to the target problem of this investigation. The heuristic algorithms developed are: Pareto Local Search, Tabu Search with ejection chain, Transgenetic Algorithm, NSGA-II and a hybridization of the two last-mentioned proposals called NSTA. The proposed algorithms are compared to each other through performance analysis regarding computational experiments with instances adapted from the QMST literature. With regard to exact algorithms, the analysis considers, in particular, the execution time. In case of the heuristic

algorithms, besides execution time, the quality of the generated approximation sets is evaluated. Quality indicators are used to assess such information. Appropriate statistical tools are used to measure the performance of exact and heuristic algorithms. Considering the set of instances adopted as well as the criteria of execution time and quality of the generated approximation set, the experiments showed that the Tabu Search with ejection chain approach obtained the best results and the transgenetic algorithm ranked second. The PLS algorithm obtained good quality solutions, but at a very high computational time compared to the other (meta)heuristics, getting the third place. NSTA and NSGA-II algorithms got the last positions.

*Keywords:* Experimental Algorithms, Exact algorithms, Metaheuristics, Multiobjective Optimization, Biobjective Adjacent Only Quadratic Spanning Tree.

# Lista de figuras

1	Exemplo de Árvore Geradora Mínima. . . . .	p. 26
2	Exemplo de Problema da AGMQ - custos quadráticos, custos lineares e representação dos componentes da rede. . . . .	p. 27
3	Grafo-exemplo para representação <i>Edge-Window-Decoder</i> (SOAK; CORNE; AHN, 2006) . . . . .	p. 53
4	Grafo implícito do algoritmo <i>backtracking</i> . . . . .	p. 72
5	Exemplo de cruzamento (RAIDL, 2000). . . . .	p. 110
6	Seleção de indivíduos para a próxima iteração do algoritmo NSGA-II. . . . .	p. 112
7	Soluções eficientes suportadas e não-suportadas (SOUZA, 2006). . . . .	p. 287
8	Métrica S para os conjuntos de aproximação A e B (SOUZA, 2006). . . . .	p. 291

# Lista de tabelas

1	Exemplificação das características das instâncias CP . . . . .	p. 81
2	Resultados do <i>backtracking</i> para instâncias CP . . . . .	p. 83
3	Resultados do <i>backtracking</i> para instâncias OP1 . . . . .	p. 83
4	Resultados do <i>backtracking</i> para instâncias OP2 . . . . .	p. 86
5	Resultados do <i>backtracking</i> para instâncias OP3 . . . . .	p. 88
6	Resultados do <i>branch-and-bound</i> para instâncias CP . . . . .	p. 91
7	Resultados do <i>branch-and-bound</i> para instâncias OP1 . . . . .	p. 92
8	Resultados do <i>branch-and-bound</i> para instâncias OP2 . . . . .	p. 95
9	Resultados do <i>branch-and-bound</i> para instâncias OP3 . . . . .	p. 98
10	Tempos médios de execução (em segundos) dos algoritmos <i>backtracking</i> e <i>branch-and-bound</i> para as classes de instâncias CP . . . . .	p. 101
11	Tempos médios de execução (em segundos) dos algoritmos <i>backtracking</i> e <i>branch-and-bound</i> para as classes de instâncias OP1 . . . . .	p. 102
12	Tempos médios de execução (em segundos) dos algoritmos <i>backtracking</i> e <i>branch-and-bound</i> para as classes de instâncias OP2 . . . . .	p. 102
13	Tempos médios de execução (em segundos) dos algoritmos <i>backtracking</i> e <i>branch-and-bound</i> para as classes de instâncias OP3 . . . . .	p. 102
14	Tempos de execução (em segundos) para os algoritmos preProc1 e pre- Proc2 no que se refere ao conjunto de instâncias CP . . . . .	p. 123
15	Tempos de execução (em segundos) para os algoritmos preProc1 e pre- Proc2 no que se refere ao conjunto de instâncias OP1 . . . . .	p. 128
16	Tempos de execução (em segundos) para os algoritmos preProc1 e pre- Proc2 no que se refere ao conjunto de instâncias OP2 . . . . .	p. 133

17	Tempos de execução (em segundos) para os algoritmos preProc1 e pre-Proc2 no que se refere ao conjunto de instâncias OP3 . . . . .	p. 138
18	$p$ -valores em favor do NSGA-II para instâncias CP . . . . .	p. 143
19	$p$ -valores em favor do NSGA-II para instâncias OP1 . . . . .	p. 148
20	$p$ -valores em favor do NSGA-II para instâncias OP2 . . . . .	p. 153
21	$p$ -valores em favor do NSGA-II para instâncias OP3 . . . . .	p. 158
22	$p$ -valores em favor do NSTA para instâncias CP . . . . .	p. 164
23	$p$ -valores em favor do NSTA para instâncias OP1 . . . . .	p. 169
24	$p$ -valores em favor do NSTA para instâncias OP2 . . . . .	p. 174
25	$p$ -valores em favor do NSTA para instâncias OP3 . . . . .	p. 179
26	$p$ -valores em favor do AT para instâncias CP . . . . .	p. 185
27	$p$ -valores em favor do AT para instâncias OP1 . . . . .	p. 190
28	$p$ -valores em favor do AT para instâncias OP2 . . . . .	p. 195
29	$p$ -valores em favor do AT para instâncias OP3 . . . . .	p. 200
30	$p$ -valores em favor da Busca Tabu para instâncias CP . . . . .	p. 205
31	$p$ -valores em favor da Busca Tabu para instâncias OP1 . . . . .	p. 211
32	$p$ -valores em favor da Busca Tabu para instâncias OP2 . . . . .	p. 216
33	$p$ -valores em favor da Busca Tabu para instâncias OP3 . . . . .	p. 221
34	$p$ -valores em favor da busca PLS para instâncias CP . . . . .	p. 226
35	$p$ -valores em favor da busca PLS para instâncias OP1 . . . . .	p. 231
36	$p$ -valores em favor da busca PLS para instâncias OP2 . . . . .	p. 236
37	$p$ -valores em favor da busca PLS para instâncias OP3 . . . . .	p. 242
38	Tempos computacionais médios (s) para as instâncias CP . . . . .	p. 247
39	Tempos computacionais médios (s) para as instâncias OP1 . . . . .	p. 252
40	Tempos computacionais médios (s) para as instâncias OP2 . . . . .	p. 257
41	Tempos computacionais médios (s) para as instâncias OP3 . . . . .	p. 262

42	Interpretação dos resultados fornecidos pelos indicadores binários. . . .	p. 292
43	Procedimentos dos vetores transgenéticos. . . . .	p. 307



# Lista de abreviaturas e siglas

AGM – Árvore Geradora Mínima

AGMQ – Árvore Geradora Mínima Quadrática

AGMQA – Árvore Geradora Mínima Quadrática em Adjacência de Arestas

AGQA-bi – Árvore Geradora Quadrática em Adjacência de Arestas Biobjetivo

AGQ-bi – Árvore Geradora Quadrática Biobjetivo

AQ – Alocação Quadrática

AG-bi – Árvore Geradora Biobjetivo

PLS – *Pareto Local Search*

NSGA-II – *Nondominated Sorting Genetic Algorithm II*

NSTA – *Nondominated Sorting Transgenetic Algorithm*

ENDM – Electronic Notes in Discrete Mathematics

LB – Limite Inferior (do inglês, *Lower Bound*)

H1 – Heurística H1, que emprega o método da contribuição média. Proposta por Assad e Xu (1992)

H2 – Heurística H2, que emprega o método da fização sequencial. Proposta por Assad e Xu (1992)

H3 – Heurística H1, que realiza uma execução parcial de um algoritmo *branch-and-bound*. Proposta por Assad e Xu (1992)

AGMQF – Árvore Geradora Mínima Quadrática *Fuzzy*

EP – Espaço de Possibilidades

EF – Evento *Fuzzy*

AGME – Árvore Geradora Mínima Esperada

AGM- $\alpha$ P – Árvore Geradora Mínima  $\alpha$ -Pessimista

MAGM – Menor Árvore Geradora Mínima

ADFAV – Algoritmo de Determinação de Fonte de Alimento Vizinha

SA – *Simulated Annealing*

AG – Árvore Geradora

ITS – *Iterated Tabu Search*

TS – *Tabu Search*

GST – *Generate Spanning Tree*

CP – Extensão do conjunto de instâncias gerado por Cordone e Passeri (2012)

OP1, OP2 e OP3 – Conjuntos de instâncias aleatórias propostas por Öncan e Punnen (2010) e gentilmente cedidas pelos autores

# Lista de símbolos

$G = (V, E)$  – Grafo  $G$ , cujo conjunto de vértices é  $V$  e o conjunto de arestas é  $E$

$V$  – Conjunto de vértices de um grafo  $G$

$v_i$  – Um vértice pertencente ao conjunto  $V$ , com  $0 < i \leq n$

$n$  – Número de vértices em um grafo  $G(n = |V|)$

$E$  – Conjunto de arestas de um grafo  $G$

$e_i$  – Uma aresta pertencente ao conjunto  $E$ , com  $0 < i \leq m$

$m$  – Número de arestas em um grafo  $G(m = |E|)$

$W$  – Vetor  $(w_1, \dots, w_m)$  que contém, para cada aresta de  $E$ , o seu respectivo custo linear

$w_i$  – Custo linear da aresta  $e_i$ ,  $0 < i \leq m$

$x$  – Vetor de variáveis booleanas  $(x_1, \dots, x_m)$ , em que cada variável  $x_i$ ,  $0 < i \leq m$ , vale 1 se a aresta  $e_i$  é faz parte da árvore e 0 caso contrário

$x_i$  – Variável booleana que indica se a aresta  $e_i$  faz parte da árvore ( $x_i = 1$ ) ou não ( $x_i = 0$ ),  $0 < i \leq m$

$T$  – Conjunto de todas as árvores geradoras possíveis de um grafo  $G$

$C$  – Matriz  $C = [c_{ij}]$  contendo, para cada par de arestas  $(e_i, e_j)$ ,  $0 < i, j \leq m$ , o seu respectivo intercusto

$c_{ij}$  – Intercusto entre as arestas  $e_i$  e  $e_j$ ,  $0 < i, j \leq m$

$A_i$  – Conjunto dos índices (números de identificação) de todas as arestas adjacentes à aresta  $e_i$ ,  $0 < i \leq m$

$CPL_i$  – Contribuição parcial da aresta livre  $e_i$ ,  $0 < i \leq m$

$lb_i$  –  $L \cap A_i$ ,  $0 < i \leq m$ , isto é,  $L_i$  é o conjunto dos índices das arestas adjacentes à aresta  $e_i$  que estão livres

$CME_k$  – Contribuição Média Estimada da aresta  $e_k$  para o custo total da árvore,  $0 <$

$k \leq m$

$\Theta^c$  – Conjunto complementar ao conjunto  $\Theta$

$P(\Theta)$  – Conjunto das partes do conjunto  $\Theta$

$Pos\{EF\}$  – Métrica de possibilidade. Indica a possibilidade de ocorrência de um evento qualquer  $EF$

$Nec\{EF\}$  – Indica a necessidade de ocorrência de um evento qualquer  $EF$

$Cr\{EF\}$  – Medida de credibilidade para um evento qualquer  $EF$

$pop\_tam$  – Tamanho da população de um algoritmo genético

$PAI_i$  – Pai de identificador  $i$ , em que  $i$  geralmente vale 1 ou 2. Variável que representa um cromossomo que será empregado como pai na operação de cruzamento de um algoritmo genético

$PAI_i^j$  – Conjunto dos nós adjacentes ao nó  $j$  no cromossomo  $PAI_i$

$CM_i$  – Contribuição Mínima da aresta  $e_i, 0 < i \leq m$

$N_{(t)}^k$  – Vizinhança que seleciona randomicamente  $k$  arestas que estão na árvore geradora  $t$  para serem removidas da árvore. Cada uma das arestas removidas é substituída por uma aresta que está fora de  $t$  de modo que, após a substituição,  $t$  ainda seja uma árvore geradora

$POT_i$  – Custo potencial da aresta  $e_i, 0 < i \leq m$

$FA_i$  – Fonte de Alimento associada à abelha  $i$

$AG^*$  – Melhor Árvore Geradora encontrada por um algoritmo heurístico para o problema da AGMQ ou da AGMQA

$z^*$  – Valor da função-objetivo da árvore geradora em  $AG^*$

$Itr$  – Variável que representa o número de iterações de um algoritmo

$I$  – Nos algoritmos *branch-and-bound*, representa o conjunto dos índices das arestas que estão na árvore, ou seja, para as quais  $x_i = 1$

$O$  – Nos algoritmos *branch-and-bound*, representa o conjunto dos índices das arestas que estão excluídas da árvore, ou seja, para as quais  $x_i = 0$

$L$  – Nos algoritmos *branch-and-bound*, representa o conjunto dos índices das arestas que estão livres para entrar na árvore ou não, ou seja, para as quais  $x_i$  ainda é indefinido

$NV$  – Conjunto com os possíveis tamanhos dos casos de teste

# Lista de algoritmos

1	Algoritmo Backtracking ( $G = (V, E)$ ) . . . . .	p. 73
2	Algoritmo <i>branch-and-bound</i> ( $G = (V, E)$ ) . . . . .	p. 79
3	Algoritmo <i>Pareto Local Search</i> . . . . .	p. 107
4	Algoritmo NSGA-II . . . . .	p. 109
5	Algoritmo NSTA . . . . .	p. 114
6	Algoritmo Transgenético . . . . .	p. 117
7	Algoritmo Busca Tabu . . . . .	p. 119
8	Algoritmo AG . . . . .	p. 305
9	Algoritmo TC . . . . .	p. 308
10	Algoritmo BuscaTabu . . . . .	p. 311

# Sumário

<b>1</b>	<b>Introdução</b>	p. 25
1.1	Objeto de estudo . . . . .	p. 25
1.2	Motivação . . . . .	p. 29
1.3	Aplicações Reais . . . . .	p. 31
1.4	Objetivos . . . . .	p. 32
1.5	Contribuições . . . . .	p. 35
1.6	Organização do trabalho . . . . .	p. 36
<b>2</b>	<b>O Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas</b>	p. 37
2.1	Definição do Problema da Árvore Geradora Mínima Quadrática . . . . .	p. 37
2.2	Definição do Problema da Árvore Geradora Mínima Quadrática em Adjacência de Arestas . . . . .	p. 39
2.3	Definição do Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas . . . . .	p. 39
2.4	Revisão da Literatura para os Problemas da AGMQ e da AGMQA . . . . .	p. 40
2.4.1	Assad e Xu (1992) . . . . .	p. 40
2.4.1.1	<i>Branch-and-bound</i> para o problema da AGMQ . . . . .	p. 41
2.4.1.2	<i>Branch-and-bound</i> para o problema da AGMQA . . . . .	p. 43
2.4.1.3	Heurística H1 (Método da Contribuição Média) . . . . .	p. 45
2.4.1.4	Heurística H2 (Método da Fixação Sequencial) . . . . .	p. 46
2.4.1.5	Heurística H3 . . . . .	p. 47
2.4.2	Zhou e Gen (1998) . . . . .	p. 48

2.4.3	Gao, Lu e Liu (2004) . . . . .	p. 49
2.4.4	Gau e Lu (2005) . . . . .	p. 52
2.4.5	Soak, Corne e Ahn (2006) . . . . .	p. 53
2.4.6	Yu, Shi e Sheu (2008) . . . . .	p. 56
2.4.7	Cordone e Passeri (2008) . . . . .	p. 57
2.4.8	Öncan e Punnen (2010) . . . . .	p. 59
2.4.9	Sundar e Singh (2010) . . . . .	p. 60
2.4.10	Palubeckis, Rubliauskas e Targamadzé (2010) . . . . .	p. 63
2.4.10.1	<i>Simulated annealing multistart</i> . . . . .	p. 63
2.4.10.2	Algoritmo genético . . . . .	p. 64
2.4.10.3	Busca Tabu Iterativa . . . . .	p. 66
2.4.11	Cordone e Passeri (2012) . . . . .	p. 67
2.4.11.1	Busca tabu . . . . .	p. 67
2.4.11.2	Busca em vizinhança variável (VNS) . . . . .	p. 68
2.4.11.3	<i>Branch-and-bound</i> . . . . .	p. 69
<b>3</b>	<b>Algoritmos Exatos</b>	p. 71
3.1	<i>Backtracking</i> . . . . .	p. 71
3.2	<i>Branch-and-bound</i> . . . . .	p. 76
<b>4</b>	<b>Experimentos Computacionais – Algoritmos Exatos</b>	p. 80
4.1	Instâncias . . . . .	p. 80
4.2	Metodologia de Comparação . . . . .	p. 82
4.3	Resultados . . . . .	p. 82
<b>5</b>	<b>Algoritmos Heurísticos</b>	p. 104
5.1	<i>Pareto Local Search</i> . . . . .	p. 104
5.2	NSGA-II . . . . .	p. 107



5.3	NSTA . . . . .	p. 112
5.4	Algoritmo Transgenético . . . . .	p. 115
5.5	Busca Tabu . . . . .	p. 116
<b>6</b>	<b>Experimentos Computacionais – Algoritmos Heurísticos</b>	p. 120
6.1	Instâncias . . . . .	p. 120
6.2	Metodologia de Comparação . . . . .	p. 120
6.3	Configuração dos Parâmetros . . . . .	p. 121
6.4	Resultados . . . . .	p. 122
<b>7</b>	<b>Considerações finais</b>	p. 269
<b>8</b>	<b>Trabalhos Futuros</b>	p. 271
	<b>Referências</b>	p. 273
	<b>Apêndice A – Conceitos Fundamentais em Grafos</b>	p. 280
	<b>Apêndice B – Otimização Multiobjetivo</b>	p. 283
B.1	Definição de um Problema de Otimização Multiobjetivo . . . . .	p. 283
B.2	Dominância de Pareto . . . . .	p. 284
B.3	Soluções Eficientes e Soluções Não-dominadas . . . . .	p. 285
B.4	Comparação entre Algoritmos Multiobjetivos . . . . .	p. 288
B.4.1	Métrica $S$ . . . . .	p. 290
B.4.2	$\epsilon$ -binário . . . . .	p. 291
	<b>Apêndice C – Algoritmos Evolucionários</b>	p. 293
C.1	Teoria de Lamarck . . . . .	p. 293
C.2	Teoria da Seleção Natural . . . . .	p. 294
C.3	Teoria Serial da Endossimbiose . . . . .	p. 298

C.4	Transferência Horizontal de Genes . . . . .	p. 301
C.5	Equilíbrio Pontuado . . . . .	p. 302
C.6	Algoritmos Genéticos . . . . .	p. 303
C.7	Algoritmos Transgenéticos . . . . .	p. 305
<b>Apêndice D – Busca Tabu</b>		p. 309
D.1	Breve histórico . . . . .	p. 309
D.2	Espaço de busca e estrutura de vizinhança . . . . .	p. 309
D.3	Lista Tabu . . . . .	p. 310
D.4	Critério de Aspiração . . . . .	p. 310
D.5	Uma Busca Tabu simples . . . . .	p. 311
D.6	Critério de terminação . . . . .	p. 311

# 1 Introdução

Este capítulo introdutório está organizado em seis seções. Na Seção 1.1, descreve-se o objeto de estudo deste trabalho, ou seja, o problema-alvo da pesquisa. A motivação para esta investigação, bem como aplicações reais que podem ser solucionadas com o auxílio deste estudo são apresentadas nas Seções 1.2 e 1.3 respectivamente. Na Seção 1.4 explicitam-se os objetivos da pesquisa. As contribuições da investigação são detalhadas na Seção 1.5. A Seção 1.6 finaliza esta introdução com a organização geral deste trabalho.

## 1.1 Objeto de estudo

A Árvore Geradora Mínima (AGM) é um conhecido problema de otimização combinatoria solucionável em tempo polinomial. A literatura é rica em trabalhos que propõem algoritmos cada vez mais eficientes para resolvê-la. Uma revisão desses trabalhos pode ser obtida em Bazlamaçci e Hindi (2001). Os algoritmos de Prim (1957) e Kruskal (1956) são exemplos de abordagens tradicionalmente adotadas para este problema.

De maneira informal, a AGM pode ser definida como o problema de, dado um conjunto de cidades, encontrar a estrutura de conexão (rede) de menor custo em que cada cidade consiga se comunicar com as outras. Para tanto, informa-se, para cada par de cidades, o custo de conectá-las. O fato de ser a estrutura de menor custo implica a não existência de caminhos redundantes (ciclos). Na Figura 1, os vértices representam as cidades e as arestas, as conexões entre elas. Cada conexão está associada a um custo. As arestas em cor preta representam a AGM para este caso em particular.

A AGM tem aplicabilidade em áreas como redes de comunicação (JOTHI; RAGHAVACHARI, 2005), expressão de genes (ZHOU et al., 2007) e árvores filogenéticas (LUO; ZHANG, 2007). Além disso, abordagens heurísticas podem utilizar informações obtidas a partir da solução da AGM para auxiliar a resolução de problemas de otimização NP-difíceis, como, por exemplo, o Problema do Caixeiro Viajante (GUTIN; PUNNEN, 2002) e o Problema da

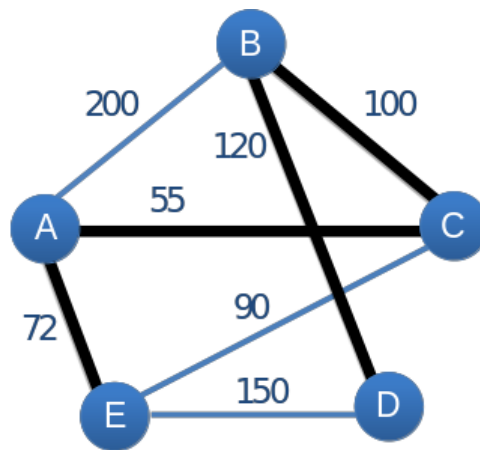


Figura 1: Exemplo de Árvore Geradora Mínima.

Árvore de Steiner (HAKAMI, 1971). Com isso, a AGM passa a ter, além da importância prática, relevância para a teoria da otimização.

Segundo Zhou e Gen (1998), uma das razões para que o problema da AGM seja de fácil resolução é a linearidade da sua função-objetivo. Xu (1984) propôs um novo problema que generaliza a AGM, a Árvore Geradora Mínima Quadrática (AGMQ). Esse problema considera a presença de efeitos de interação entre pares de arestas. Quando tais interações só podem ocorrer entre arestas que sejam adjacentes, o problema é denominado Árvore Geradora Mínima Quadrática em Adjacência de Arestas (AGMQA) (ASSAD; XU, 1992). As modelagens propostas por Xu (1984) para a AGMQ e a AGMQA envolvem, pois, encontrar a árvore geradora de menor custo sob uma estrutura de custos quadrática.

De maneira informal, a AGMQ, em seu caso geral, também pode ser definida como o problema de, dado um conjunto de cidades, encontrar a estrutura de conexão (rede) considerando a minimização dos custos de conexão entre cidades e a minimização dos intercustos entre conexões. Neste caso, o custo apresenta um componente linear e um quadrático, como apresentado na Figura 2, em que os intercustos são representados como os elementos da matriz e os custos de conexão estão representados como os pesos das arestas do grafo.

Assume-se que as cidades serão conectadas por meio de estradas e que os custos de se conectar cada par de cidades (por meio de uma estrada) são dados. Na Figura 2, tais valores são informados próximo à representação pictórica das respectivas conexões entre as cidades. O componente linear corresponde à soma dos custos associados às estradas que foram selecionadas para compor a rede de conexão entre as cidades, assim como ocorre na AGM.

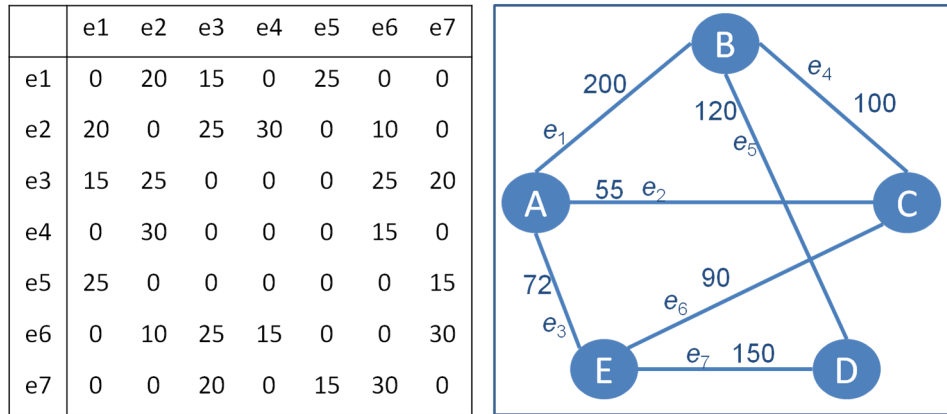


Figura 2: Exemplo de Problema da AGMQ - custos quadráticos, custos lineares e representação dos componentes da rede.

Informa-se, também, para cada par de estradas possível, ou seja, para cada par de conexões, se há interação entre elas e qual o custo desta interação, denominado intercusto e representado pela matriz de valores da Figura 2. Nesse sentido, o componente quadrático é a soma de todos os intercustos resultantes da inclusão, na rede, de conexões que interagem aos pares.

Sendo assim, para a AGMQ, a estrutura de conexão de menor custo entre as cidades é aquela que minimiza a soma dos custos linear e quadrático.

Diz-se que uma estrada é vizinha de (ou adjacente a) outra estrada quando uma mesma cidade é um ponto extremo para ambas as estradas. Quando as interações entre estradas são restritas àquelas que são adjacentes entre si, o problema é denominado Árvore Geradora Mínima Quadrática em Adjacência de Arestas (AGMQA), conforme já mencionado. Neste caso, para o cálculo da função objetivo da AGMQA e levando em conta uma matriz de intercurso, conforme exemplificado na Figura 2, apenas os valores de intercusto entre arestas adjacentes entre si são considerados. Os demais valores são descartados.

A AGMQ e a AGMQA têm especial aplicação no transporte de petróleo e seus derivados de um duto para outro, situação na qual o custo do transporte depende da natureza da interface de conexão entre os dois dutos (ZHOU; GEN, 1998). Outras aplicações contemporâneas dizem respeito ao projeto de redes nas áreas de telecomunicações, transportes e distribuição de energia (PALUBECKIS; RUBLIAUSKAS; TARGAMADZÉ, 2010). A Seção 1.3 abordará em detalhes aplicações para os problemas em questão. Tendo em vista as aplicações para a AGMQ e a AGMQA, a AGMQA emerge como um modelo mais apropriado para problemas reais, pois a interação entre arestas adjacentes é mais comum que

a interação entre duas arestas quaisquer.

Embora os custos linear e quadrático sejam somados, em aplicações reais, pode ser necessário considerar tais custos separadamente, com o intuito de oferecer informação de melhor qualidade para fundamentação do tomador de decisões. Além disso, em situações reais, é possível que os custos linear e quadrático sejam de naturezas conflitantes, o que inviabiliza a soma. Por exemplo, considere uma rede de infraestrutura em que as conexões são dutos por onde escoam fluidos. O custo dos dutos pode ser dado pela confiabilidade de o mesmo estar operando. O intercusto pode ser dado pelo custo da operação de equipamentos na junção entre os dutos.

Considerando o Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas (AGQA-bi) para o caso de teste da Figura 2, a AGM do grafo apresentado é composta pelo conjunto de arestas  $T_1 = \{e_2, e_3, e_4, e_5\}$ . A árvore  $T_1$  apresenta custo linear igual a 347 e custo quadrático igual a 110. Outra árvore geradora do grafo da Figura 2 é  $T_2 = \{e_3, e_6, e_4, e_5\}$ . Neste caso, o custo linear vale 382 e o quadrático assume o valor 80. Uma vez que os objetivos linear e quadrático representem valores de naturezas diferentes, as árvores  $T_1$  e  $T_2$  ilustram uma situação de conflito.  $T_1$  é melhor com relação ao objetivo linear e  $T_2$  é melhor com relação ao objetivo quadrático. A melhoria de um objetivo implicou a piora do valor do outro objetivo.

Nesse contexto, a otimização multiobjetivo tende a prover um modelo mais realista para os problemas da AGMQ e da AGMQA, uma vez que permite considerar os custos linear e quadrático como objetivos (critérios) separados e independentes. Até o momento, uma revisão da literatura não foi capaz de identificar qualquer trabalho que investigue tais problemas sob um ponto de vista biobjetivo. Neste sentido, esta Tese apresenta uma contribuição para a área expandindo tais investigações.

Assad e Xu (1992) mostram que a AGMQ e a AGMQA são problemas NP-difíceis. Portanto, a versão biobjetivo proposta (AGQA-bi) também é classificada como problema NP-difícil, uma vez que o problema com um único objetivo corresponde ao caso especial do problema biobjetivo onde cada objetivo recebe uma mesma ponderação. Os problemas NP-difíceis correspondem a uma classe de problemas para os quais ainda não existe algoritmo exato de complexidade assintótica polinomial que os resolvam. Assim, dependendo do tamanho do problema, ou seja, do número de variáveis de entrada, os algoritmos exatos existentes para problemas dessa classe podem requerer um tempo de processamento inviável para serem solucionados nas máquinas atuais (CAMPELO; MACULAN, 1994).

Diante do exposto, o objeto de estudo deste trabalho é o Problema Biobjetivo da

Árvore Geradora Quadrática em Adjacência de Arestas (AGQA-bi), um problema ainda não explorado na literatura e relevante tanto pela sua possibilidade de aplicação prática quanto pela sua relação teórica com outros problemas NP-difíceis clássicos da Otimização Combinatória, em especial a Árvore Geradora Biobjetivo e o Problema de Alocação Quadrática.

## 1.2 Motivação

Como mencionam Goldbarg e Luna (2005), o homem tem se mostrado cada vez mais hábil na criação de metáforas para representação e solução de problemas na sua relação com o mundo. Segundo os autores, “os modelos são representações simplificadas da realidade que preservam, para determinadas situações e enfoques, uma equivalência adequada” (GOLDBARG; LUNA, 2005). De posse de um modelo implementável, a Pesquisa Operacional se torna uma ferramenta de inestimável valor para a solução de problemas reais relacionados, por exemplo, à otimização da produtividade de uma indústria, do uso de matéria-prima, da quantidade de petróleo que pode ser extraída, do percurso a ser realizado pelo serviço de entrega de cartas, etc. De acordo com Goldbarg e Luna (2005), “O poder de representatividade é a característica do modelo que o torna desejável”, ou seja, quanto mais próximo do problema real for o modelo, melhor, desde que não seja complexo a ponto de inviabilizar sua resolução.

Observando o equilíbrio entre simplicidade e validade (adequação à realidade) do modelo, a abordagem multiobjetivo pode se mostrar mais apropriada ao lidar com problemas como a AGMQ e AGMQA em situações em que os custos linear e quadrático devam ser tratados separadamente.

Apesar de Weixuan Xu ter proposto, em 1984, as variantes da AGM mencionadas em sua tese de doutorado (XU, 1984), e de Assad e Xu terem publicado um artigo em 1992 sobre o mesmo tema (ASSAD; XU, 1992), aparentemente, os problemas da AGMQ e da AGMQA permaneceram intocados até 1998, quando Zhou e Gen publicaram um algoritmo genético para a AGMQ (ZHOU; GEN, 1998). A partir de 2004, estes problemas começaram a despertar um interesse crescente nos pesquisadores, que desenvolveram abordagens exatas, heurísticas e fuzzy para a AGMQ e a AGMQA, por meio dos trabalhos elencados a seguir: Gao, Lu e Liu (2004); Gao e Lu (2005); Soak, Corne e Ahn (2006); Yu, Shi e Sheu (2008); Cordone e Passeri (2008); Öncan e Punnen (2010); Palubeckis, Rubliauskas e Targamadzé (2010); Sundar e Singh (2010) e Cordone e Passeri (2012). No Capítulo 2, apresentar-se-á

uma revisão dos trabalhos citados.

Como se pode perceber, ainda há poucos trabalhos publicados sobre a AGMQ. Além disso, na maioria dos artigos, os autores assumem que um algoritmo para a AGMQ também será capaz de resolver a AGMQA, e, por isso, não desenvolvem uma abordagem específica para a variante quadrática em adjacência de arestas. Entretanto, como a AGMQA é um subproblema da AGMQ, existe a possibilidade de que possa ser resolvida por meio de abordagens específicas, o que implicaria um menor tempo de processamento em relação às abordagens desenvolvidas para o problema mais geral. Sendo assim, usar as abordagens desenvolvidas para AGMQ para solucionar a AGMQA pode caracterizar desperdício de recursos. Nesse sentido, a literatura ainda é carente de algoritmos desenvolvidos especificamente para a Árvore Geradora Mínima Quadrática em Adjacência de Arestas.

Também realizamos uma busca de artigos na literatura no que se refere aos problemas da AGQ-bi e da AGQA-bi. No entanto, nenhum trabalho foi encontrado. Assumimos, pois, que, até o momento, a abordagem multiobjetivo para os problemas da AGQ-bi e da AGQA-bi é inédita na literatura. Nesse sentido, ainda não há algoritmos desenvolvidos para solução de tais problemas.

Outro fator que contribuiu para a escolha do tema foi o fato de as variantes quadráticas mencionadas do problema da AGM terem relação com dois problemas clássicos de Otimização Combinatória, a saber: o próprio problema da AGM e problema de Alocação Quadrática (AQ), em especial nas suas versões biobjetivo se considerarmos o problema da AGQA-bi. A relação do problema da AGQA-bi com o da Árvore Geradora Biobjetivo (AG-bi) se deve ao compartilhamento de um objetivo em comum. Assim como o problema de AQ, a AGQA-bi tem um de seus objetivos dado por uma função quadrática. Além disso, Assad e Xu (1992) utilizam o problema de AQ para demonstrar que o problema da AGMQ é NP-difícil e Öncan e Punnen (2010) construíram um conjunto de instâncias para a AGMQ pela conversão de instâncias da QAPLib, uma biblioteca de problemas de AQ (BURKARD; KARISCH; RENDL, 1997). Pode-se, portanto, pensar que é possível que técnicas utilizadas para solucionar os problemas AG-bi e AQ possam contribuir para a solução do problema da AGQA-bi e vice-versa.

Diante do exposto, pode-se concluir que a escolha do objeto de estudo desta investigação foi motivada por:

- a AGQA-bi ser um problema de otimização combinatória com possibilidade de modelar problemas reais. Isso implica que o uso das técnicas e ferramentas providas pela



Pesquisa Operacional e pela Otimização Combinatória para a solução da AGQA-bi podem ser aplicadas na solução dos problemas reais modelados pela árvore mencionada.

- na literatura, haver apenas alguns trabalhos que lidam especificamente com a AGMQA e não haver pesquisas relacionadas a AGQA-bi, o que torna este trabalho inédito e relevante, em termos teóricos, para a área de pesquisa.
- a AGQA-bi ter relação teórica com os problemas clássicos de Otimização Combinatória AG-bi e AQ, os quais são bastante estudados pela aplicabilidade na modelagem de problemas reais.

### 1.3 Aplicações Reais

Assad e Xu (1992) citam como aplicação real para os problemas da AGMQ e da AGMQA a transmissão de material, por exemplo, petróleo e/ou seus derivados, de um duto para outro, situação na qual os custos da transmissão podem depender da natureza da interface de conexão entre os dutos e da presença de elementos na rede. Alguns elementos de uma rede de distribuição de gás são, além dos dutos, válvulas de controle e grupos de compressão. O custo de instalação desses elementos pode se caracterizar como um intercusto. Além disso, no caso de o elemento ser ativo, é possível que a sua atuação na rede seja passível de modelagem via intercustos de uma AGQA-bi.

Conseguimos vislumbrar ainda o cenário hipotético a seguir. Suponha que uma cidade possa cobrar impostos em função da transmissão de petróleo e seus derivados por dutos instalados dentro dos limites do município. Neste caso, ao projetar a rede de distribuição, uma estimativa dos impostos que serão pagos pode ser modelada pelo custo quadrático. Dessa forma, na fronteira entre municípios, haveria uma junção entre os dutos de modo que a cobrança de impostos relativos a uma cidade incidiria sobre o fluxo de petróleo e seus derivados que passam pela junção na fronteira do município no sentido de entrar na cidade. Sendo assim, o intercusto poderia modelar os impostos devidos ao fluxo de petróleo de um duto para outro, passando pela junção.

Os efeitos de interação entre pares de arestas também podem ser percebidos na conexão entre cabos de energia aéreos e subterrâneos (ASSAD; XU, 1992). Tal conexão, em geral, ocorre em uma estação de conexão de cabos, empregando equipamento apropriado. Neste caso, um para-raios é colocado no ponto terminal do cabo para proteção do cabo

e de equipamentos contra raios (CHANG et al., 2011). Toda essa infraestrutura representa um custo adicional no projeto da rede de distribuição de energia elétrica.

Redes de transporte ou estradas com penalidades de conversão também são aplicações reais para a AGQA-bi. Neste caso, o projeto da rede leva em conta não apenas os custos de se instalar a rede, mas também a minimização, sobre a rede como um todo, das penalidades de conversão (ASSAD; XU, 1992).

Ao projetar uma rede de transporte apenas com base no custo de instalação da rede, estamos assumindo que os veículos transitarão na rede com velocidade constante e que vão percorrer o caminho determinado no menor tempo possível (JAN et al., 2009). Na realidade, sabemos que, em função do tráfego, ao se aproximar de uma esquina, o veículo deve ter sua velocidade reduzida e, caso seja necessário realizar alguma conversão à esquerda ou à direita, haverá um tempo de espera, a depender do fluxo estimado de veículos, para que tal operação seja concluída. Sendo assim, para cada possível conversão, teremos penalidades associadas. As penalidades são proporcionais ao tempo de espera estimado. A modelagem por meio da AGQA-bi considera como intercusto a penalidade de movimentar-se de uma rua para outra por meio de uma conversão.

## 1.4 Objetivos

Tendo em vista a ausência de trabalhos sobre o tema na literatura, esta pesquisa tem por objetivo geral conduzir um estudo de algoritmos exatos e heurísticos para a solução do problema da AGQA-bi.

Dentre as abordagens exatas, foram desenvolvidos um algoritmo *backtracking* e uma adaptação, para o caso biobjetivo, do *branch-and-bound* proposto em Assad e Xu (1992) para a AGMQA.

Os algoritmos de busca local, seguem a estratégia Pareto Local Search PLS proposta por Paquete e Stützle (2006). Tal estratégia foi selecionada pela simplicidade e por ser própria para problemas com múltiplos objetivos.

A abordagem metaheurística é uma ferramenta da Otimização Combinatória que tem se mostrado muito eficiente e eficaz na solução de problemas (REEVES, 1993). Dentro deste contexto, destaca-se a Computação Evolucionária.

As técnicas metaheurísticas associadas à Computação Evolucionária apresentam as seguintes características: (a) são baseadas em população, (b) possuem arquitetura de

processamento paralelizável e (c) apresentam mecanismo de busca estocástica com viés (GOLDBARG; GOLDBARG, 2007). Os algoritmos evolucionários são, pois, particularmente apropriados para resolver problemas de otimização multicritério, por lidarem com um conjunto de possíveis soluções (população). Com isso, além de permitirem encontrar vários membros do conjunto de soluções ótimas de Pareto em uma única execução do algoritmo, facilitam a implementação de estratégias de exploração e exploração mais eficientes do espaço de busca por meio de mecanismos de interação entre soluções.

Dentre as técnicas da Computação Evolucionária encontram-se os Algoritmos Genéticos (HOLLAND, 1975), Algoritmos Transgenéticos (GOLDBARG; GOLDBARG, 2007) e outras abordagens.

Fundamentados nos algoritmos genéticos, Deb et al. (2002) propuseram o algoritmo NSGA-II (*Nondominated Sorting Genetic Algorithm II*), desenvolvido especificamente para lidar com problemas multiobjetivo. O NSGA-II foi aplicado a diversos problemas e tem sido usado como base de comparação para novas propostas algorítmicas, como pode ser exemplificado pelos trabalhos elencados a seguir: Almeida et al. (2012), Adra e Fleming (2011), Bandyopadhyay (2012), Fallah-Mehdipour et al. (2012), Huang, Buckley e Kechadi (2010), Jeyadevi et al. (2011) e Murugan, Kannan e Baskar (2009). Neste trabalho, a exemplo dos citados anteriormente, foi desenvolvido um NSGA-II para o problema da AGQA-bi.

A inspiração para os Algoritmos Transgenéticos advém da concepção da endossimbiose e da transferência horizontal de genes como duas grandes forças evolucionárias Goldberg e Goldberg (2009). Este tipo de algoritmo foi aplicado a dois problemas relacionados ao objeto de estudo desta tese: AG-bi (MONTEIRO; GOLDBARG; GOLDBARG, 2009, 2010; MONTEIRO, 2011) e AQ com múltiplos objetivos (ALMEIDA et al., 2012). Uma proposta de hibridização do NSGA-II com Algoritmos Transgenéticos foi, também, proposta por Almeida et al. (2012) e aplicada ao AQ multiobjetivo e ao Problema do Caixeiro Comprador Biobjetivo.

Nessa perspectiva, assim como em Almeida et al. (2012), esta Tese, também, inclui a implementação dos algoritmos NSGA-II, do Algoritmo Transgenético e da hibridização entre essas abordagens, que deu origem à abordagem denominada NSTA.

Outra abordagem adotada é a Busca Tabu. O interesse nesta abordagem se dá pelo fato de que, no que se refere ao problema de Alocação Quadrática, a Busca Tabu é uma estratégia que tem obtido excelentes resultados, conforme pode ser verificado nos trabalhos de Drezner (2005), James, Rego e Glover (2009a), James, Rego e Glover (2009b),

Misevicius (2005), Paul (2010, 2011), Fescioglu-Unver e Kokar (2011) e Hussin e Stützle (2014).

A literatura dos problemas da AGMQ e da AGMQA também é rica em abordagens Tabu, implementadas em trabalhos como Cordone e Passeri (2008), Palubeckis, Rubliauskas e Targamadzé (2010) e Cordone e Passeri (2012). Nesse sentido, é interessante averiguar a eficiência e eficácia da Busca Tabu na solução do problema da AGQA-bi.

De maneira geral, o objetivo desta Tese é, pois, a definição do problema da AGQA-bi, assim como a implementação e análise de algoritmos exatos *backtracking* e *branch-and-bound* para a AGQA-bi e das abordagens (meta)heurísticas elencadas a seguir: busca local PLS, NSGA-II, Algoritmo Transgenético, NSTA e Busca Tabu. A escolha dessas metaheurísticas foi motivada pelo sucesso das mesmas no contexto da Otimização Multiobjetivo, da solução do problema da AG-bi ou da solução do problema da AQ, conforme já explicitado.

Experimentos computacionais comparam a eficiência e a eficácia dos algoritmos desenvolvidos. Nesse sentido, esta tese inclui um capítulo específico para experimentos computacionais envolvendo algoritmos exatos e outro para os algoritmos heurísticos.

O conjunto de instâncias (casos de teste) usadas nesta pesquisa está dividido em quatro grupos. O primeiro é uma extensão do conjunto de instâncias proposto em Cordone e Passeri (2012) para o problema da AGMQ. Instâncias com até 50 vértices foram disponibilizadas em: <http://homes.di.unimi.it/~cordone/research/qmst.html> pelos referidos autores. Utilizando o mesmo método, geramos instâncias com até 175 vértices. Os outros três grupos de instâncias foram criados por Öncan e Punnen (2010) e foram gentilmente cedidos pelos autores.

Dois critérios principais de comparação são considerados: o tempo de processamento e a qualidade das soluções geradas. A qualidade das soluções geradas pelos algoritmos heurísticos é comparada com base em indicadores de qualidade apropriados para a avaliação de algoritmos para problemas com múltiplos objetivos (conforme apêndice específico para Otimização Multiobjetivo). Os resultados dos indicadores são submetidos a testes estatísticos não-paramétricos para verificar se há evidência estatística de que um algoritmo tenha desempenho melhor que o outro.

Para o desenvolvimento deste estudo, faz-se necessário conhecer: os fundamentos da Otimização Multiobjetivo, a literatura para a AGMQ e AGMQA bem como as metaheurísticas envolvidas e suas inspirações.

De modo sucinto, podemos estabelecer, pois, os seguintes objetivos específicos:

- Explicar os princípios da Otimização Multiobjetivo e como avaliar algoritmos desenvolvidos para problemas com múltiplos critérios.
- Apresentar os problemas AGMQ e AGMQA, bem como realizar uma revisão da literatura destes problemas.
- Definir o problema da AGQA-bi, o objeto de estudo desta tese.
- Apresentar algoritmos exatos para a AGQA-bi, a saber: *backtracking* e *branch-and-bound*.
- Apresentar algoritmos (meta)heurísticos para a AGQA-bi, seguindo as abordagens busca local PLS, NSGA-II, Algoritmo Transgenético, NSTA e Busca Tabu.
- Realizar estudos comparativos entre os diversos algoritmos desenvolvidos.

## 1.5 Contribuições

Uma das contribuições desta tese advém do fato de que o problema da AGQA-bi ainda é inexplorado. Nesse sentido, a definição formal do problema bem como o estudo de algoritmos para a AGQA-bi proposto nesta pesquisa contribui para ampliar as fronteiras da Otimização Combinatória e da Otimização Multiobjetivo, permitindo caracterizar esta tese como um trabalho original e inovador.

Como demonstração da relevância desta pesquisa, pode-se citar o trabalho intitulado “On the Biobjective Adjacent Only Quadratic Spanning Tree Problem”, apresentado na International Network Optimization Conference (INOC) 2013 e aceito para publicação na Electronic Notes in Discrete Mathematics (ENDM), um periódico classificado no Qualis de Ciência da Computação.

Além disso, esta tese inclui revisão do estado da arte para problemas da AGMQ e da AGMQA. Também são descritos os princípios da Otimização Multiobjetivo e da avaliação de algoritmos para problemas com múltiplos critérios. Assim sendo, esperamos que esta pesquisa possa servir de incentivo e fundamentação teórica para novos estudos na área.

## 1.6 Organização do trabalho

A organização geral deste trabalho compreende 8 capítulos. No primeiro, revelam-se a importância e a aplicabilidade da pesquisa desenvolvida, por meio de aspectos introdutórios acerca da AGQA-bi e dos algoritmos propostos. As definições dos problemas da AGMQ, da AGMQA e da AGQA-bi são apresentadas no Capítulo 2, o qual compreende também o estado da arte para tais problemas.

No Capítulo 3, descrevemos os algoritmos exatos desenvolvidos para a AGQA-bi. Os experimentos computacionais envolvendo tais algoritmos são exibidos no Capítulo 4. Os Capítulos 5 e 6 apresentam, respectivamente, os algoritmos heurísticos implementados para a solução da AGQA-bi e os experimentos computacionais a eles relacionados.

O Capítulo 7 contém considerações finais sobre o trabalho e o 8 contempla os trabalhos futuros. A seguir, temos as referências e alguns apêndices.

No Apêndice A, apresentam-se definições básicas em teoria dos grafos. No Apêndice B, discutem-se aspectos relevantes acerca da otimização multiobjetivo. Tais conhecimentos são necessários para a definição do problema da AGQA-bi. O Apêndice C contempla as descrições das abordagens metaheurísticas evolucionárias adotadas, suas inspirações e implementações computacionais. O Apêndice D é composto pela descrição da Metaheurística Busca Tabu, também abordada neste trabalho. Todos esses conhecimentos são de fundamental importância para a compreensão dos algoritmos desenvolvidos.

## 2 O Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas

O Capítulo 2 diz respeito à definição do objeto de estudo desta tese: o Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas. Para tanto, nas Seções 2.1 e 2.2, apresentam-se o problema da Árvore Geradora Mínima Quadrática e sua versão em Adjacência de Arestas. A Seção 2.3 corresponde à definição da versão biobjetivo.

Tendo em vista que não há trabalhos sobre a AGQA-bi, a literatura para a AGMQ e para a AGMQA representa a melhor fonte de subsídios teóricos para o problema investigado. Nesse sentido, o presente capítulo também contempla uma revisão da literatura para os problemas da AGMQ e AGMQA.

Alguns conceitos básicos em Teoria dos Grafos e a fundamentação teórica em Otimização Multiobjetivo, requisitos para a compreensão deste capítulo, estão disponíveis nos Apêndices A e B desta tese. Vale salientar que, neste trabalho, lidamos apenas com grafos simples, ou seja, sem arestas paralelas nem laços próprios.

### 2.1 Definição do Problema da Árvore Geradora Mínima Quadrática

Os problemas da AGMQ e da AGMQA serão definidos com base no artigo de Assad e Xu (1992). Estes autores partem da definição da AGM para chegar à modelagem da AGMQ e da AGMQA.

Considere um grafo simples, não-dirigido e conexo  $G = (V, E)$ , onde  $V = \{v_1, v_2, \dots, v_n\}$  é um conjunto finito de  $n$  vértices e  $E = \{e_1, e_2, \dots, e_m\}$  é um conjunto finito de  $m$  arestas. A cada aresta de  $E$ , associa-se um valor real positivo denotado por  $W = \{w_1, w_2, \dots, w_m\}$

. Seja  $x = \{x_1, x_2, \dots, x_m\}$  definido como na Equação 2.1:

$$\begin{cases} 1, & \text{se a aresta } e_i \text{ é selecionada} \\ 0, & \text{caso contrário} \end{cases}, \quad i=1, \dots, m \quad (2.1)$$

Neste caso, uma árvore geradora de  $G$  pode ser representada por um vetor  $x$  em que exatamente  $n - 1$  arestas estão selecionadas. Seja  $T$  o conjunto de todas as árvores geradoras de  $G$ . Dado que se deseja encontrar a árvore geradora de menor custo total, pode-se definir a função-objetivo da AGM pela Expressão 2.2 (ASSAD; XU, 1992):

$$\text{Min} \left\{ z(x) = \sum_{i=1}^m w_i x_i \mid x \in T \right\} \quad (2.2)$$

A definição formal da AGMQ parte dos mesmos pressupostos estabelecidos para a definição da AGM. Adicione-se apenas a existência de um custo  $c_{ij}$  associado a cada par de arestas  $(e_i, e_j)$ ,  $i, j = 1, \dots, m$ , denominado intercusto. Suponha que a matriz  $C$  contém os elementos  $c_{ij}$ .

O objetivo da AGMQ é encontrar a árvore geradora que minimiza a soma entre os custos linear e quadrático da árvore. O custo linear advém da formulação da AGM e corresponde, pois, à soma dos pesos  $w_i$  das arestas na árvore. O custo quadrático, por sua vez, se refere à soma dos intercustos  $(c_{ij})$  entre pares de arestas na árvore. Assim, a AGMQ pode ser formulada da seguinte forma (ASSAD; XU, 1992):

$$\text{Min} \left\{ z(x) = \sum_{i=1}^m \sum_{j=1, j \neq i}^m c_{ij} x_i x_j + \sum_{i=1}^m w_i x_i \mid x \in T \right\} \quad (2.3)$$

Como se pode perceber por meio da Expressão 2.3, a inserção do termo referente ao intercusto faz com que a função-objetivo deixe de ser linear e adquira natureza quadrática. Além disso, na formulação proposta, dadas duas arestas  $e_i$  e  $e_j$  ( $0 < i, j \leq m$ ), ambos os intercustos  $c_{ij}$  e  $c_{ji}$  são contabilizados, a exemplo do que ocorre com outros problemas de natureza quadrática, tais como o de Alocação Quadrática (AQ). Sendo assim, considera-se a existência de casos de teste com matriz de intercustos simétrica e não-simétrica.

Assad e Xu (1992) mostram que o problema AQ pode ser transformado, em tempo polinomial, em um problema de AGMQ. Nesse sentido, os referidos autores demonstram que o problema da AGMQ é NP-difícil.



## 2.2 Definição do Problema da Árvore Geradora Mínima Quadrática em Adjacência de Arestas

A diferença entre os problemas da AGMQ e da AGMQA é que, na AGMQ, pode haver interação entre quaisquer duas arestas que estejam na árvore enquanto, na AGMQA, tal interação é restrita às arestas que são adjacentes entre si, ou seja, que possuem vértice-terminal em comum. Seja  $A_i, 0 < i \leq m$ , o conjunto dos índices (números de identificação) de todas as arestas adjacentes à aresta  $e_i$ . Então, o problema da AGMQA pode ser modelado por (ASSAD; XU, 1992):

$$Min \left\{ z(x) = \sum_{i=1}^m \sum_{j \in A_i} c_{ij} x_i x_j + \sum_{i=1}^m w_i x_i \mid x \in T \right\} \quad (2.4)$$

Assad e Xu (1992) mostram que o problema do Caminho Hamiltoniano pode ser transformado, em tempo polinomial, em um problema de AGMQA. Nesse sentido, os autores demonstram que o problema da AGMQA também é NP-difícil.

## 2.3 Definição do Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas

O problema da AGQA-bi leva em conta que os componentes linear e quadrático da Expressão 2.4 podem ser conflitantes. Portanto, tais custos são considerados critérios de otimização separados e independentes, de acordo com a formulação dada pelas Expressões 2.5, 2.6 e 2.7.

$$Min \left\{ z(x) = (z_1(x), z_2(x)) \mid x \in T \right\} \quad (2.5)$$

em que:

$$z_1(x) = \sum_{i=1}^m w_i x_i \quad (2.6)$$

$$z_2(x) = \sum_{i=1}^m \sum_{j \in A_i} c_{ij} x_i x_j \quad (2.7)$$

A literatura apresenta vários problemas que, no contexto mono-objetivo são soluci-

onados em tempo polinomial, mas suas versões com múltiplos objetivos são NP-difíceis. Podemos mencionar, por exemplo, o problema da AG-bi (AGGARWAL; ANEJA; NAIR, 1982) e o problema do Caminho Mais Curto Multiobjetivo (SERAFINI, 1987). No caso da AGQA-bi, o objetivo linear pode ser resolvido em tempo polinomial, mas o objetivo quadrático é NP-difícil, tendo em vista que o problema da AGMQA é NP-difícil conforme Assad e Xu (1992). Nesse sentido, a AGQA-bi é um problema de otimização NP-difícil.

A abordagem multiobjetivo geralmente implica que não há uma única solução para o problema, mas um conjunto de soluções. Assim, a palavra “minimizar” deve ser entendida no contexto da Otimização de Pareto e suas relações de dominância, conforme a revisão dos fundamentos da Otimização Multiobjetivo no Apêndice B desta tese. Para enfatizar esta mudança de perspectiva, apesar de o problema da AGQA-bi ser de minimização, excluimos a palavra “mínima” do nome do problema, a exemplo do que ocorre com outros problemas de otimização multiobjetivo, como a própria AG-bi (MONTEIRO; GOLDBARG; GOLDBARG, 2010). Por isso, o problema é denominado Árvore Geradora Quadrática em Adjacência de Arestas Biobjetivo e não Árvore Geradora Mínima Quadrática em Adjacência de Arestas Biobjetivo.

## 2.4 Revisão da Literatura para os Problemas da AGMQ e da AGMQA

Cada uma das seções abaixo aborda um trabalho da literatura dos problemas da AGMQ e da AGMQA. Vale salientar ainda que alguns trabalhos lidam com abordagens *fuzzy* para a solução de versões *fuzzy* do problema-alvo desta investigação. Algoritmos *fuzzy* não fazem parte do foco deste trabalho. Mesmo assim, por completude, os referidos trabalhos são brevemente descritos, com destaque para os algoritmos utilizados.

### 2.4.1 Assad e Xu (1992)

O trabalho de Assad e Xu (1992) foi o primeiro artigo a apresentar os problemas da AGMQ e da AGMQA. Neste artigo, os autores definem os referidos problemas; provam que são problemas NP-difíceis; desenvolvem limites inferiores para a AGMQ e para a AGMQA; usam tais limites na implementação de um algoritmo *branch-and-bound* para cada problema; implementam três heurísticas e relatam experimentos computacionais envolvendo os algoritmos descritos. Nas subseções a seguir, descreveremos os algoritmos desenvolvidos por Assad e Xu (1992).

### 2.4.1.1 *Branch-and-bound* para o problema da AGMQ

No algoritmo *branch-and-bound* proposto por Assad e Xu (1992), cada nó representa uma partição do conjunto de arestas do grafo - que representa o caso de teste - em três conjuntos: (i) as arestas que estão na árvore; (ii) as arestas que não podem entrar na árvore e (iii) as arestas disponíveis para entrar na árvore. Denominemos por  $I$ ,  $O$  e  $L$ , respectivamente, os conjuntos dos índices das arestas em (i), (ii) e (iii). Assim, temos:

$$x_i = \begin{cases} 1, & \text{se } i \in I \\ 0, & \text{se } i \in O \end{cases}, 0 \leq i \leq m \quad (2.8)$$

Caso  $i \in L$ , o valor de  $x_i$  ainda está indefinido. A função-objetivo definida na Expressão 2.3 pode ser reescrita em termos apenas das arestas em  $L$  ( $x_i$ , tal que  $i \in L$ ):

$$z(x) = z_1(x) + z_2(x) + z_3(x) \quad (2.9)$$

$$z_1(x) = \sum_{i \in I} \left[ w_i + \sum_{j \in I, j \neq i} c_{ij} \right] \quad (2.10)$$

$$z_2(x) = \sum_{i \in L} \left[ \left( w_i + \sum_{j \in I} (c_{ij} + c_{ji}) \right) x_i \right] \quad (2.11)$$

$$z_3(x) = \sum_{i \in L} \sum_{j \in L, j \neq i} c_{ij} x_i x_j \quad (2.12)$$

Dada uma configuração parcial da árvore, ou seja, uma partição do conjunto de arestas em  $I$ ,  $O$  e  $L$ ,  $z_1(x)$  representa o somatório dos custos lineares das arestas que já foram inseridas na árvore e dos intercustos entre pares de arestas que já se encontram na árvore. Isto é,  $z_1$  é a contribuição parcial das arestas em  $I$  para o custo total da árvore. Para cada aresta em  $L$  ( $e_i$ , tal que  $i \in L$ ), caso esta aresta  $e_i$  seja inserida na árvore ( $x_i = 1$ ), o custo total da árvore terá um incremento igual ao somatório entre o custo linear de  $e_i$  e os intercustos entre  $e_i$  e cada uma das arestas em  $I$ . Tal incremento é calculado por meio de  $z_2(x)$ , de modo que o incremento somente é contabilizado se a aresta for inserida na árvore.  $z_3$ , por sua vez, é a função que calcula o somatório dos intercustos entre pares de arestas que estão em  $L$ , de tal forma que, se um dado par de arestas em  $L$  for inserido na árvore, o intercusto entre as mesmas será contabilizado no custo total da árvore. Observe que o custo da árvore não depende dos custos linear ou quadrático de arestas em  $O$  (fora

da árvore) e que a função em  $z_1$  não é multiplicada por  $x_i$ , uma vez que  $z_1$  trata de arestas que já estão na árvore, ou seja, para as quais  $x_i = 1$ . Assad e Xu (1992) desenvolveram uma especialização, para o problema da AGMQ, da técnica proposta em Assad e Xu (1985) para calcular limites inferiores ( $LB$ ) para uma classe de problemas 0 – 1. O limite é definido em função de  $m$  parâmetros  $\pi = (\pi_1, \pi_2, \dots, \pi_m)$ . Seja:

$$w_i(\pi) = w_i - (n - 2)\pi_i, 0 \leq i \leq m \quad (2.13)$$

$$c_{ij}(\pi) = c_{ij} + \pi_j, 0 < i \leq m \quad (2.14)$$

Considerando as informações dadas pelas Expressões 2.13 e 2.14, defina:

$$f_i(\pi) = \min \left\{ w_i(\pi) + \sum_{j=1, j \neq i}^m c_{ij}(\pi)x_j \mid x_i = 1 \wedge x \in T \right\}, \forall i, 0 < i \leq m \quad (2.15)$$

$$LB(\pi) = \min \left\{ \sum_{i=1}^m f_i(\pi)x_i \mid x \in T \right\} \quad (2.16)$$

Os autores provam que  $LB(\pi)$ , dado pela Expressão 2.16, é um limite inferior para o problema da AGMQ para qualquer  $\pi$ .  $LB(\pi)$  é denominado, então, limite inferior parametrizado com os parâmetros  $\pi$ . Observe que, dado  $\pi$ , calcular  $LB(\pi)$  requer a obtenção de  $m + 1$  árvores geradoras mínimas. Além disso, os autores mostram que  $LB(\pi)$  está relacionado a uma relaxação Lagrangeana de uma reformulação linear da AGMQ, de modo que  $LB(\pi)$  corresponde ao próprio valor do problema Lagrangeano. Ao calcular limites inferiores, desejamos encontrar o maior valor possível, aqui denominado  $LB^*$ . Por isso, é importante desenvolver um método para encontrar os parâmetros  $\pi$  que retornam o maior limite inferior possível,  $LB^*$ .

Devido à relação com o problema Lagrangeano, os autores propõem obter  $LB(\pi)$  por um método de ajuste de multiplicadores denominado pelos autores de “Algoritmo por Níveis”. O Algoritmo por Níveis é fundamentado na noção de desvio  $(\varphi)def = (f_1, \dots, f_m)$ , definido como sendo a diferença entre o maior e o menor componente de  $f$ , ou seja, para um dado valor fixo de  $\pi$ :

$$\phi(\pi) = \max_i \{ f_i(\pi) \} - \min_i \{ f_i(\pi) \} \quad (2.17)$$

em que o valor de  $i$  varia de 1 até  $m$ .

Assad e Xu (1992) mostram que  $LB^*$  é atingido quando o desvio de  $f$  é reduzido a zero para algum  $\pi$ . O Algoritmo por Níveis parte de um valor inicial para  $\pi$  - no caso,  $\pi = (0, \dots, 0)$  - e calcula  $LB(\pi)$ . Os valores em  $\pi$  são, então, iterativamente atualizados e um novo  $LB(\pi)$  é calculado até que o desvio  $\varphi$  seja inferior a um valor máximo permitido ou que uma condição de parada seja atingida. Ou seja, o Algoritmo por Níveis busca  $LB^*$  por um procedimento que, iterativamente, reduz a distância entre os componentes máximo e mínimo de  $f$ . Isso significa que, ao executar o Algoritmo por Níveis, o valor de  $LB(\pi)$  obtido converge para  $LB^*$ . O algoritmo *branch-and-bound* proposto por Assad e Xu (1992) para a AGMQ executa uma busca em profundidade. A cada nó na árvore de busca, os limites inferiores e superiores são atualizados. Como já mencionado, o limite inferior é obtido pelo Algoritmo por Níveis. O limite superior, por sua vez, corresponde ao custo real da AGM obtida quando calculamos  $LB(\pi)$  para  $\pi = (0, \dots, 0)$ , ou seja, da AGM correspondente à Expressão 2.16. Ao executar o Algoritmo por Níveis, consideram-se os conjuntos  $I, O, L$ , de modo que as Expressões em 2.8 são satisfeitas. Dessa forma, as arestas em  $I$  são obrigatórias em todas as árvores, as arestas em  $O$  são proibidas em todas as árvores e as arestas em  $L$  podem ou não fazer parte das AGMs obtidas durante o cálculo de  $LB(\pi)$  (Expressões 2.15 e 2.16). Se o limite inferior igualar ou superar o melhor limite superior obtido até o momento, ocorre uma poda, e, pois, o algoritmo volta (realiza *backtracking*) para avaliar outro nó. Caso a poda não ocorra, a busca prossegue escolhendo a próxima aresta que deve ser analisada,  $e_k$ , em que  $k = \operatorname{argmin}_{i \in L} f_i(\pi) | \pi = (0, \dots, 0)$ . Criam-se, então, duas ramificações na árvore de busca: uma para  $x_k = 1$  e outra para  $x_k = 0$ . Os autores justificam esse procedimento de escolha e de ramificação afirmando que, como  $\pi = (0, \dots, 0)$ , a AGM que pode ser construída com  $x_k$  fixo em 1 tem um custo menor que a AGM correspondente à fixação de qualquer outro  $x_i$  em 1. Sendo assim, em uma abordagem de busca em profundidade, espera-se que o nó com  $x_k = 1$  gere um melhor limite superior e que o nó com  $x_k = 0$  gere um limite inferior maior. Assad e Xu (1992) testaram este algoritmo para um conjunto de instâncias aleatórias com tamanho (número de vértices) variando de 6 a 12.

#### 2.4.1.2 *Branch-and-bound* para o problema da AGMQA

O *branch-and-bound* para a AGMQA desenvolvido por Assad e Xu (1992) também realiza uma busca em profundidade e, na ramificação (branching), também seleciona uma aresta que dará origem a dois ramos: um em que a aresta selecionada é obrigatória e

outro em que a mesma é proibida. Isto é, em termos de estrutura geral do algoritmo, este *branch-and-bound* se mostra semelhante ao definido na Subseção 2.4.1.1. Entretanto, a estrutura especial da AGMQA permite que os limites inferior e superior sejam definidos de maneira mais eficiente, o que simplificou também a regra de ramificação. Sejam os conjuntos  $I, O, L$  e  $A_i, 0 < i \leq m$ , definidos como anteriormente neste capítulo. Defina  $L_i = L \cap A_i, 0 < i \leq m$ , ou seja,  $L_i$  corresponde ao conjunto de índices das arestas adjacentes à aresta  $e_i$  que estão livres. A função-objetivo da AGMQA pode, então, ser reescrita como:

$$z(x) = z_1(x) + z_2(x) + z_3(x) \quad (2.18)$$

$$z_1(x) = \sum_{i \in I} \left[ w_i + \sum_{j \in I, j \neq i} c_{ij} \right] \quad (2.19)$$

$$z_2(x) = \sum_{i \in L} \left[ \left( w_i + \sum_{j \in I} (c_{ij} + c_{ji}) \right) x_i \right] \quad (2.20)$$

$$z_3(x) = \sum_{i \in L} \sum_{j \in L_i} c_{ij} x_i x_j \quad (2.21)$$

No caso da AGMQA, as Expressões em 2.18-2.21 podem ser compreendidas da mesma forma que para a AGMQ, na Subseção 2.4.1.1, exceto pelo fato de que, em  $z_3(x)$ , os intercustos são restritos a pares de arestas que possuem relação de adjacência entre si.

Denominemos por  $CPL_i$  a contribuição parcial da aresta livre  $e_i$ , correspondente à Expressão entre parênteses em  $z_2(x)$ . Então,  $z_2(x) + z_3(x)$  pode ser reescrito como:

$$z_2(x) + z_3(x) = \sum_{i \in L} \left[ CPL_i + \sum_{j \in L_i} c_{ij} x_j \right] x_i \quad (2.22)$$

Assumamos que os termos do tipo  $c_{ij}$  são não-negativos, então os autores mostram que o termo entre colchetes na Expressão 2.22 pode ser limitado inferiormente por:

$$lb_i = \begin{cases} CPL_i + \min \{ c_{ij} | j \in L_i \}, & \text{se } I \cap A_i = \{ \} \\ CPL_i, & \text{caso contrário} \end{cases} \quad (2.23)$$

A partir da Expressão 2.23, observe que, para um dado nó da árvore de busca do *branch-and-bound* para a AGMQA, o limite inferior corresponde à soma entre  $z_1(x)$ , um valor constante, e o custo total da árvore geradora mínima em que os pesos associados às arestas são seus respectivos valores de  $lb_i$ . Diferentemente do caso da AGMQ, apenas uma AGM, em vez de  $m + 1$  AGMs, são necessárias para obter  $LB$ .

Assad e Xu (1992) afirmam que uma Expressão similar, porém mais complicada, pode ser obtida quando os intercustos são negativos, entretanto, nenhuma aplicação real para este caso é mencionada.

No *branch-and-bound* proposto por Assad e Xu (1992) para a AGMQA, então, o limite inferior é calculado de acordo com a Expressão 2.23; o limite superior é o custo real da árvore geradora obtida para solucionar 2.23 e a regra de ramificação consiste em selecionar a aresta que apresenta menor valor de  $lb_i$ .

Os autores testaram este algoritmo em instâncias aleatórias contendo de 6 a 15 vértices. A comparação com o *branch-and-bound* da Subseção 2.4.1.1 mostrou que o *branch-and-bound* específico para a AGMQA foi bem mais rápido, mostrando que vale a pena investir em algoritmos específicos para a AGMQA e aproveitar as vantagens de sua estrutura especial.

#### 2.4.1.3 Heurística H1 (Método da Contribuição Média)

A heurística H1, tal como proposta por Assad e Xu (1992), parte da contribuição individual média de cada aresta para o custo total da AGMQ. Para tanto, reformulemos a Expressão 2.3, colocando o foco em uma única aresta  $e_k, 0 < k \leq m$ :

$$z(x) = z_1(x) + z_2(x) \quad (2.24)$$

em que:

$$z_1(x) = \left[ w_k + \sum_{j \neq k} (c_{jk} + c_{kj}) x_j \right] x_k \quad (2.25)$$

e  $z_2(x)$  é o restante da Expressão 2.3 que não envolve  $x_k$ .

Se a aresta  $e_k$  fizer parte da solução da AGMQ, podemos dizer que, dos  $m - 1$  termos envolvidos no somatório em 2.25,  $n - 2$  termos poderão ter valor diferente de 0 e o restante deve ter valor 0. Isso porque apenas  $n - 1$  arestas fazem parte da solução e uma delas é a aresta  $e_k$ . Sendo assim, a contribuição média estimada de  $e_k$  para o custo total da AGMQ ( $CME_k$ ) é:

$$CME_k = w_k + \frac{n - 2}{m - 1} \sum_{j \neq k} (c_{jk} + c_{kj}), 0 < k \leq m \quad (2.26)$$

A heurística H1 proposta por Assad e Xu (1992) consiste, então, em resolver uma AGM em que os custos lineares associados a cada aresta correspondem aos seus respectivos valores de  $CME_k$ . Nesse sentido, a heurística H1 soluciona o seguinte problema de otimização:

$$\text{Min} \left\{ z(x) = \sum_{k=1}^m CME_k x_k \mid x \in T \right\} \quad (2.27)$$

#### 2.4.1.4 Heurística H2 (Método da Fixação Sequencial)

Na heurística H2, apresentada por Assad e Xu (1992), a estimativa da contribuição média de cada aresta é computada considerando que o fato de uma aresta ser previamente inserida na árvore pode influenciar o cálculo da contribuição de outras arestas. Assim sendo, a cada passo da construção da árvore, uma aresta é inserida (fixada) na solução e as demais arestas têm sua contribuição média novamente estimada.

Sejam os conjuntos  $I$ ,  $O$  e  $L$ , respectivamente, os conjuntos dos índices das arestas que estão na árvore, que estão excluídas da árvore (cuja inclusão gera ciclo) e que estão livres para entrar ou não na árvore, conforme já definido neste capítulo. Relembremos também que a contribuição parcial de uma aresta livre para o custo total da árvore ( $CPL_i$ ) é dado por:

$$CPL_i = w_i + \sum_{j \in I} (c_{ij} + c_{ji}) \quad (2.28)$$

Podemos, então, reformular a Expressão 2.3, com foco em uma única aresta  $e_k$ ,  $0 < k \leq m$ :

$$z(x) = z_1(x) + z_2(x) + z_3(x) \quad (2.29)$$

em que  $z_3(x)$  refere-se aos termos de  $z(x)$  que não dizem respeito a  $e_k$  e:

$$z_1(x) = CPL_k x_k \quad (2.30)$$

$$z_2(x) = \left[ \sum_{j \in L, j \neq k} (c_{jk} + c_{kj}) x_j \right] x_k \quad (2.31)$$

Sendo assim,  $z_1(x) + z_2(x)$  representa a parcela de  $z(x)$  que envolve  $e_k$  e pode ser



reescrito como segue:

$$z_1(x) + z_2(x) = \left[ CPL_k + \sum_{j \in L, j \neq k} (c_{jk} + c_{kj})x_j \right] x_k \quad (2.32)$$

O somatório apresentado na Expressão 2.32 avalia  $|L| - 1$  termos  $x_j$  dos quais apenas  $n - 2 - |I|$  podem assumir valores diferentes de 0. A contribuição média estimada de  $e_k$  considerando as arestas que já foram inseridas na árvore é pois:

$$CME_k = CPL_k + \frac{n - 2 - |I|}{|L| - 1} \sum_{j \in L, j \neq k} (c_{jk} + c_{kj}), 0 < k \leq m \quad (2.33)$$

O procedimento que implementa este algoritmo se assemelha ao algoritmo de Kruskal (1956) para AGM, no sentido de que a aresta mais promissora é inserida na árvore a cada iteração. Isso significa que, durante a construção da árvore, é possível que a solução parcial seja uma floresta e não uma árvore. Em um primeiro momento, as contribuições médias de cada aresta ( $CME_k, 0 < k \leq m$ ) são calculadas conforme a Expressão 2.33. A partir de então, o algoritmo executa um laço que, a cada iteração:

- (i) insere na árvore geradora que representa a solução parcial a aresta  $e_k$  que possui menor valor de  $CME_k$ , o que implica a adição de  $k$  em  $I$  e a remoção de  $k$  de  $L$ ;
- (ii) retira de  $L$  (e insere em  $O$ ) os índices de todas as arestas que geram ciclo na árvore atual;
- (iii) recalcula a contribuição média estimada para todas as arestas que permaneceram em  $L$ , de acordo com a Expressão 2.33.

Os passos de (i) a (iii) são seguidos até que  $|I| = n - 1$ , ou seja, até que a solução seja uma árvore geradora do grafo de entrada.

#### 2.4.1.5 Heurística H3

A heurística H3 proposta por Assad e Xu (1992) consiste simplesmente em uma execução parcial do algoritmo *branch-and-bound* proposto na Subseção 2.4.1.1. A condição de parada para um grafo com  $n$  vértices foi um total de  $2n$  iterações. H3 armazena, então, a melhor solução encontrada pelo *branch-and-bound* até o momento da interrupção.

Embora as heurísticas H1, H2 e H3 tenham sido apresentadas considerando a função-objetivo da AGMQ, essas heurísticas também podem ser executadas para instâncias da

AGMQA. A única ressalva é que tais algoritmos não aproveitam a estrutura simplificada da AGMQA, tratando como uma instância da própria AGMQ.

As três heurísticas foram comparadas para um conjunto de casos de teste constituídos por grafos gerados aleatoriamente com número de vértices variando de 6 a 20. Neste conjunto, há grafos que representam instâncias da AGMQ e também da AGMQA. Os resultados experimentais apresentados pelos autores mostram que, em termos de tempo computacional, temos a seguinte ordenação:  $H1 < H2 < H3$ . Entretanto, quando se trata de qualidade da solução encontrada, os resultados de  $H2$  e  $H3$  são superiores aos de  $H1$  e  $H2$  parece ser ligeiramente superior a  $H3$ .

## 2.4.2 Zhou e Gen (1998)

Zhou e Gen (1998) desenvolveram um algoritmo genético para a AGMQ. Tal algoritmo emprega a codificação de Prüfer (PICCIOTTO, 1999) para representar, em um cromossomo, uma possível solução (ou seja, uma árvore geradora). A codificação de Prüfer permite usar uma permutação de  $n - 2$  dígitos (com valores entre 1 e  $n$ , inclusive) para representar univocamente uma árvore com  $n$  vértices (ZHOU; GEN, 1998). Os autores apresentam, em seu artigo, os procedimentos para codificação e decodificação de números de Prüfer.

As operações de cruzamento e mutação realizadas são estratégias simples e conhecidas na literatura da metaheurística em referência. Zhou e Gen (1998) adotaram o operador de cruzamento uniforme. Dado um cromossomo-pai e um cromossomo-mãe, esse operador cria dois descendentes. Para tanto, gera-se uma máscara binária do mesmo tamanho do cromossomo de forma aleatória. A máscara servirá para promover a combinação de genes dos pais. O primeiro filho receberá, na posição  $i$ , o gene do pai, se o valor da máscara em  $i$  for 0; ou o gene da mãe, se o valor da máscara em  $i$  for 1. O contrário ocorre para o segundo descendente, que recebe genes do pai nas posições em que a máscara assume valor 1 e genes da mãe nas posições em que o valor da máscara é 0. O operador de mutação, por sua vez, corresponde a uma perturbação aleatória. Nesse sentido, o valor de um gene é modificado para um valor inteiro obtido aleatoriamente no intervalo  $[1, n]$ .

O procedimento de avaliação de um cromossomo se constitui de dois passos:

- (i) converter o cromossomo da representação de Prüfer para uma representação por conjunto de arestas, de acordo com o algoritmo de decodificação apresentado no artigo em análise (ZHOU; GEN, 1998).

- (ii) calcular o custo total da árvore, diretamente associado ao *fitness* do cromossomo, de acordo com a formulação apresentada para a AGMQ.

A seleção se caracteriza pela hibridização de duas técnicas, a saber: seleção  $(\mu + \lambda)$  e roleta. A estratégia adotada pelos autores seleciona os  $\mu$  melhores cromossomos distintos dentre os  $\mu$  pais e os  $\lambda$  descendentes. Caso não haja  $\mu$  cromossomos diferentes, a população da próxima geração é preenchida com cromossomos escolhidos por meio de roleta.

O algoritmo genético de Zhou e Gen (1998) segue a estrutura básica de um algoritmo dessa metaheurística. Sendo assim, inicializa e avalia a população da primeira geração e, até que uma condição de parada seja atingida:

- (i) realiza recombinação para gerar os descendentes da população atual;
- (ii) avalia os descendentes gerados; e
- (iii) seleciona os indivíduos que constituirão a população na próxima iteração.

Os autores estabeleceram como critério de parada 500 iterações. Vale salientar que (ZHOU; GEN, 1998) não mencionam no artigo de que forma os pais são escolhidos a cada aplicação do operador de cruzamento.

O experimento computacional realizado pelos autores comparou o algoritmo genético desenvolvido com as heurísticas H1 e H2 de Assad e Xu (1992) com base em um conjunto de 17 instâncias para o problema da AGMQ com número de vértices variando de 6 a 50. Para todas as instâncias, o algoritmo genético obteve melhores resultados que as heurísticas.

### 2.4.3 Gao, Lu e Liu (2004)

Gao, Lu e Liu (2004) apresentam o problema da Árvore Geradora Mínima Quadrática *Fuzzy* (AGMQF), em que os custos linear e quadrático são considerados *fuzzy*. Os autores formulam um modelo de programação *chance-constrained* (restrita a probabilidades) usando como fundamentação o conceito de medida de credibilidade. A caracterização dos custos *fuzzy* como números *fuzzy* trapezoidais permite a obtenção do equivalente *crisp* (ou equivalente determinístico). Um algoritmo genético é delineado, no trabalho em referência, para resolver o modelo de programação *chance-constrained* bem como seu equivalente determinístico. De fato, o algoritmo genético apresentado também pode ser aplicado à solução do próprio problema da AGMQ.

Primeiramente, os autores apresentam algumas definições relacionadas à Teoria da Possibilidade e à Teoria da Credibilidade. Um espaço de possibilidades (EP) é uma tripla  $EP = (\Theta, P(\Theta), Pos)$ , em que  $\Theta$  é um conjunto não-vazio;  $P(\Theta)$  é o conjunto das partes de  $\Theta$  e  $Pos$  é uma métrica de possibilidades. Sendo assim, para cada evento *fuzzy*  $EF \in P(\Theta)$ , sua possibilidade de ocorrência é  $Pos\{EF\}$  e sua necessidade de ocorrência é dada por  $Nec\{EF\} = 1 - Pos\{EF^c\}$ . Com base nessas informações, os autores apresentam a definição de uma medida de credibilidade ( $Cr$ ) para um evento qualquer  $EF \in P(\Theta)$ :

$$Cr\{EF\} = \frac{1}{2}(Pos\{EF\} + Nec\{EF\}) \quad (2.34)$$

Gao, Lu e Liu (2004) mostram que a medida de credibilidade apresentada é mais apropriada como medida de probabilidade do que a medida de possibilidade ou a de necessidade. Nesse sentido, o modelo de programação *chance-constrained* proposto pelos autores para formular a AGMQF é baseada na medida de credibilidade.

Retomemos a formulação para o problema da AGMQ apresentada na Expressão 2.3, para um grafo  $G = (V, E)$ , com custos lineares expressos pelo vetor  $W$  e intercustos pela matriz  $C$ . Considerando a existência de incerteza sobre os valores dos custos linear e quadrático, os autores defendem que tais custos sejam especificados como variáveis *fuzzy*. Isso implica que a própria função-objetivo em 2.3 deve ser tratada como *fuzzy* também.

Nessa perspectiva, os autores estabelecem  $\alpha$  como um nível de confiança para que o valor da função objetivo em 2.3 seja inferior ou igual a um dado valor  $f$ , que diz respeito ao valor  $\alpha$ -pessimista da função. Nesse sentido, a função-objetivo do problema da AGMQF é exibido nas Expressões 2.35-2.37:

$$\min f, \text{ sujeito a:} \quad (2.35)$$

$$Cr \left\{ \sum_{i=1}^m \sum_{j=1}^m c_{ij} x_i x_j + \sum_{i=1}^m w_i x_i \leq f \right\} \geq \alpha \quad (2.36)$$

$$x_i \in \{0, 1\} \wedge x \in T \quad (2.37)$$

Gao, Lu e Liu (2004) também apresentam o equivalente *crisp* da formulação por programação *chance-constrained* em 2.35-2.37 sob a hipótese de que os custos linear e

quadrático são números *fuzzy* trapezoidais. Com isso, é possível calcular facilmente o valor da função-objetivo nas Expressões 2.35-2.37.

Os autores apresentaram um algoritmo genético baseado em simulação para solucionar o problema da AGMQF. Assim como Zhou e Gen (1998), a codificação de cromossomos por meio de números de Prüfer foi adotada.

Os operadores de cruzamento e mutação empregados no algoritmo genético desenvolvido por Gao, Lu e Liu (2004) são bem simples. Dados dois indivíduos para se reproduzirem, o operador de cruzamento simplesmente realiza a troca de seus dígitos em posições escolhidas randomicamente. Os autores não mencionam a metodologia empregada para selecionar os pares a serem submetidos ao operador de cruzamento. A mutação, por sua vez, seleciona uma posição do indivíduo randomicamente e, a seguir, escolhe um valor inteiro entre 1 e  $n$ , inclusive, para substituir o valor original naquela posição.

O procedimento de avaliação é que faz com que esse algoritmo seja adaptável tanto à versão *fuzzy* proposta quanto ao problema da AGMQ comum. No caso da AGMQ comum, é suficiente:

- (i) decodificar o cromossomo (de número de Prüfer para uma representação de conjunto de arestas) e
- (ii) calcular o valor da função objetivo como sendo o *fitness* do cromossomo.

Por outro lado, se estivermos lidando com o modelo *chance-constrained* proposto no artigo em referência, faz-se necessário (GAO; LU; LIU, 2004):

- (i) decodificar o cromossomo;
- (ii) calcular o custo  $\alpha$ -pessimista de acordo com as equações e procedimentos de simulação *fuzzy* apresentados em Gao, Lu e Liu (2004);
- (iii) associar a cada cromossomo um valor de *fitness* via um método baseado em classificação (*rank*) de acordo com o valor do objetivo.

A seleção da população que vigorará na próxima geração é realizada por meio da aplicação de roleta tantas vezes quanto seja o tamanho da população. O algoritmo termina depois de um dado número de iterações executadas. O procedimento geral do algoritmo genético proposto por Gao, Lu e Liu (2004) pode ser descrito pelos seguintes passos:

- (i) Inicie  $pop\_tam$  indivíduos randomicamente.
- (ii) Atualize os cromossomos de acordo com as operações de cruzamento e mutação.
- (iii) Calcule os valores dos objetivos para todos os cromossomos por meio do procedimento de simulação *fuzzy* proposto em Gao, Lu e Liu (2004).
- (iv) Calcule o *fitness* de cada cromossomo de acordo com os valores do objetivo.
- (v) Selecione os cromossomos por meio de roleta.
- (vi) Repita os passos (ii)-(v) por um dado número de ciclos.
- (vii) Retorne o melhor cromossomo como sendo a solução ótima.

Este trabalho não relata experimentos computacionais envolvendo um conjunto de casos de teste, apenas ilustra o funcionamento do algoritmo genético na resolução de um exemplo numérico.

#### 2.4.4 Gau e Lu (2005)

O trabalho de Gao e Lu (2005) é uma extensão do artigo apresentado por Gao, Lu e Liu (2004). Nesse sentido, se dedicam ao estudo do problema da árvore geradora mínima quadrática *fuzzy* (AGMQF).

Os autores se fundamentam na teoria da credibilidade recentemente desenvolvida por Liu (2002, 2004) para propor os conceitos de Árvore Geradora Mínima Esperada (AGME), Árvore Geradora Mínima  $\alpha$ -Pessimista (AGM- $\alpha$ P) e Menor Árvore Geradora Mínima (MAGM). Para encontrar essas árvores, o problema da AGMQF é formulado via modelo de valor esperado, programação *chance-constrained* (restrita a probabilidades) e programação dependent-chance (dependente de probabilidade), respectivamente. Quando os custos *fuzzy* são caracterizados por números *fuzzy* trapezoidais, também se discutem os *crisps* equivalentes.

Perceba que a AGM  $\alpha$ -pessimista corresponde exatamente à AGMQF proposta em Gao, Lu e Liu (2004) e já discutida em subseção anterior. Sendo assim, a título de exemplificação, já apresentamos um dos modelos *fuzzy* propostos. Como a abordagem *fuzzy*, até o momento, não é o foco deste trabalho, não nos ateremos detalhadamente a esse tema.

Após a apresentação dos modelos de AGMQF mencionados, os autores apresentam um algoritmo genético para solucionar as três variantes. Esse algoritmo corresponde exatamente ao algoritmo proposto em Gao, Lu e Liu (2004), exceto no que diz respeito à função de avaliação, a qual exige o cálculo do custo esperado, do custo  $\alpha$ -pessimista e da função de probabilidade de acordo com as equações e procedimentos de simulação *fuzzy* apresentados em Gao e Lu (2005).

Os autores também não reportam experimentos computacionais, apenas apresentam um exemplo numérico. Dessa forma, não há resultados da comparação deste genético com outros algoritmos da literatura no artigo em referência.

#### 2.4.5 Soak, Corne e Ahn (2006)

Neste trabalho, Soak, Corne e Ahn (2006) apresentam uma nova maneira de codificar árvores, denominada representação *Edge-Window-Decoder* (decodificador por janela de aresta). Tal estratégia de codificação consiste em uma lista de nós, que podem se repetir. Para testar a representação sugerida, os autores implementaram algoritmos genéticos para três problemas NP-difíceis relacionados a árvores, dentre os quais está o problema da AGMQ. Explicamos, pois, a codificação e o algoritmo adotados no artigo em referência. Considere o grafo-exemplo da Figura 3.

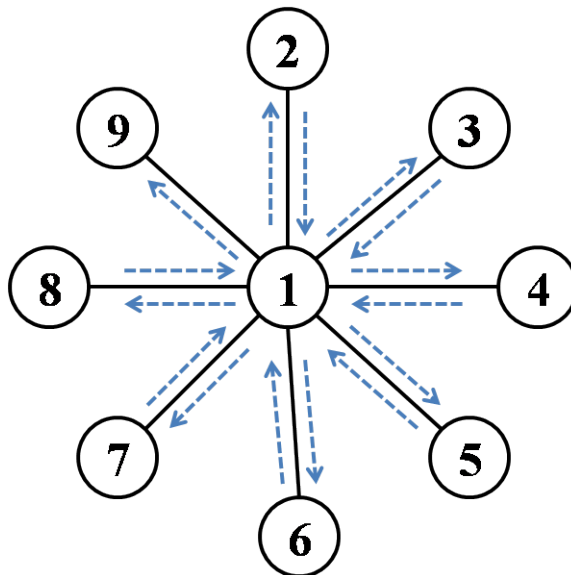


Figura 3: Grafo-exemplo para representação *Edge-Window-Decoder* (SOAK; CORNE; AHN, 2006)

A representação desta árvore por meio de um conjunto de arestas é:  $\{\{1, 2\}, \{1, 3\}, \{1, 4\}, \{1, 5\}, \{1, 6\}, \{1, 7\}, \{1, 8\}, \{1, 9\}\}$ . Agora, imagine-se traçando uma linha do

nó 1 ao nó 9, seguindo as arestas. Neste tracejado, cada nó deve ser visitado sem que se tire o lápis do papel. Esse procedimento resultará em uma sequência de nós visitados em ordem tal como: (1, 2, 1, 3, 1, 4, 1, 5, 1, 6, 1, 7, 1, 8, 1, 9). A codificação foi inspirada neste procedimento. Na prática, a sequência de vértices que representa a árvore pode ser obtida a partir de um simples percurso em profundidade, por exemplo.

A decodificação parte de uma árvore vazia e de uma “janela” de tamanho dois inicialmente posicionada na primeira posição da sequência. No caso do exemplo da Figura 3, a janela conteria, a princípio, os números “1, 2”. Isso significa que a aresta conectando os vértices 1 e 2 deve ser adicionada à árvore. A janela se move, então, uma posição, e passa a ler a informação “2, 1”. Como os autores lidam apenas com grafos não-dirigidos, essa informação corresponde a uma aresta que já está na árvore, então ela é ignorada. Assim, as arestas indicadas por cada janela são, sucessivamente, consideradas e adicionadas ou não à árvore de acordo com uma rotina de construção de árvore.

Soak, Corne e Ahn (2006) apresentam diversas rotinas de construção de árvore. Dentre as rotinas implementadas, os autores constataram que os melhores resultados ocorreram quando empregaram a Rotina de Construção de Árvore Restrita em Grau de Kruskal para o problema da AGMQ.

O procedimento inicial da rotina mencionada consiste em percorrer, usando a janela de tamanho dois, a sequência de vértices que representa a árvore e armazenar todas as arestas visitadas em um grafo  $G = (V, E)$ . Neste caso,  $G$  não necessariamente é acíclico. A partir desse ponto, uma versão do algoritmo de Kruskal (1956) é aplicada a  $G$  para dar origem à árvore. Sendo assim, o segundo passo consiste em ordenar todas as arestas de  $G$  em ordem crescente de pesos. Nesse ponto, os autores não deixam evidente que valor é tomado como peso da aresta para o problema da AGMQ. Isto é, não fica claro como os custos lineares e quadráticos associados a cada aresta são combinados (ou não) para formar o peso da aresta.

Dada a ordenação das arestas, o algoritmo prossegue inserindo na árvore, iterativamente, a aresta de menor peso que não gera ciclo nem viola a restrição de grau. Esta rotina não se aplica apenas a AGMs restritas em grau, uma vez que se pode ignorar tal restrição. O procedimento termina quando a árvore contém  $n - 1$  arestas. Soak, Corne e Ahn (2006) mencionam que cada indivíduo é uma sequência de  $2(n - 1)$  nós.

Os autores propuseram alguns operadores genéticos e um procedimento de seleção. O algoritmo genético delineado por Soak, Corne e Ahn (2006) para a AGMQ emprega o operador de cruzamento de nós adjacentes. Dados dois pais,  $PAI_1$  e  $PAI_2$ , o operador



funciona como descrito a seguir. Primeiramente, um mapa de adjacência de nós é construído para cada pai. Por exemplo, o conjunto  $PAI\ 1_i$  contém todos os nós adjacentes ao nó  $i$ ,  $0 < i \leq n$ , em  $PAI_1$ . O segundo passo é a escolha aleatória de um nó para ser o nó inicial do descendente. Denominemos o nó selecionado de vértice atual e o representemos por  $va$ . O operador de cruzamento executa, iterativamente, os passos a seguir, até que o cromossomo-filho possua  $2(n - 1)$  vértices:

- (i) Encontrar  $PAI_1^{va} \cap PAI_2^{va}$ . Se a interseção em (i) não for vazia, então escolha o  $va$  cuja aresta que o próximo vértice,  $pv$ , como sendo o vértice em  $PAI_1^{va} \cap PAI_2^{va}$  conecta a  $va$  possui menor peso. Desempate com escolha de  $pv$  aleatória quando necessário.
- (ii) Por outro lado, se a interseção em (i) for vazia, faça  $PAI_1^{va} \cup PAI_2^{va}$ . Se a união não for vazia escolha o próximo vértice,  $pv$ , como sendo o vértice em  $PAI_1^{va} \cup PAI_2^{va}$  cuja aresta que o conecta a  $va$  possui menor peso. Desempate com escolha de  $pv$  aleatória quando necessário.
- (iii) Caso contrário, escolha aleatoriamente o próximo nó  $pv$  a ser inserido no cromossomo-filho.
- (iv) Remova  $pv$  de  $PAI_1^{va}$  e de  $PAI_2^{va}$  remova  $va$  de  $PAI_1^{pv}$  e de  $PAI_2^{pv}$ .
- (v) Adicione o vértice  $pv$  imediatamente após  $va$  no descendente e atualize  $va$  com  $pv$ .

Este operador não garante que todos os nós sejam incluídos na árvore. Por isso, se algum nó tem número de ocorrências 0 na sequência de nós gerada para o cromossomo-filho, um procedimento de reparação deve ser utilizado. Os autores sugerem que, para cada nó não inserido no filho, se escolha, aleatoriamente, um nó com pelo menos 2 ocorrências. Então, o nó é inserido em lugar de uma das ocorrências do nó selecionado. O descendente gerado substitui o pior dentre os dois pais.

Soak, Corne e Ahn (2006) adotaram um operador de mutação por troca recíproca. Tal operador simplesmente seleciona, aleatoriamente, duas posições no cromossomo e troca os genes nestas posições.

Os experimentos computacionais relatados envolvem instâncias geradas aleatoriamente e compostas por grafos completos de 50 a 100 nós. O tamanho da população foi definido em 100 e a condição de parada é a execução de 10.000 iterações.

Tendo em vista que o objetivo do principal do artigo de Soak, Corne e Ahn (2006) é a apresentação de uma nova representação para estruturas de árvores, os experimentos não envolvem comparação com outros algoritmos da literatura para a AGMQ. Os experimentos consistiram, de fato, na comparação entre diferentes representações para árvores. Sendo assim, os autores mostram que o algoritmo que adotou a representação por *Edge-Window-Decoder* obteve melhores resultados em termos de qualidade de solução, embora seu desempenho em tempo computacional tenha sido inferior a abordagem via conjunto de arestas (RAIDL; JULSTROM, 2003).

#### 2.4.6 Yu, Shi e Sheu (2008)

O trabalho de Yu, Shi e Sheu (2008) apresenta uma metodologia para linearizar as formulações da AGMQ. Tal linearização é aplicada aos modelos apresentados por Gao e Lu (2005) de forma que os problemas de AGMQ *fuzzy* e seus *crisps* equivalentes possam ser solucionados via programação linear.

A linearização proposta é aplicada à formulação da AGMQ propriamente dita. Seja a definição da AGMQ apresentada no Capítulo 2 desta Tese, na Expressão 2.3. A linearização proposta pelos autores modificaria o modelo para:

$$\text{Min } \left\{ z(x) = \sum_{i=1}^m \sum_{j=1, j \neq i}^m a_{ij} + \sum_{i=1}^m w_i x_i \mid x \in T \right\}, \text{ sujeito a:} \quad (2.38)$$

$$a_{ij} \geq c_{ij} - c_{ij}(2 - x_i - x_j) \quad (2.39)$$

$$a_{ij} \geq 0 \quad (2.40)$$

Os autores mostram dois exemplos ilustrativos do algoritmo e afirmam que o problema da AGMQF e seu respectivo *crisp* podem ser resolvidos via programação linear, obtendo uma resposta ótima e não uma resposta aproximada tal como fornecido pelas abordagens heurísticas. Nenhuma consideração é realizada com relação ao tempo computacional despendido para encontrar o ótimo via programação linear e experimentos computacionais não são relatados.

### 2.4.7 Cordone e Passeri (2008)

Cordone e Passeri (2008) implementaram três abordagens heurísticas e uma busca tabu para o problema da AGMQ. Os autores também desenvolveram um algoritmo *branch-and-bound* fundamentado na relaxação da função-objetivo quadrática para uma função linear aproximada.

Duas das heurísticas implementadas por Cordone e Passeri foram as heurísticas H1 - Método da Contribuição Média e H2 - Método da Fixação Sequencial, ambas inicialmente propostas por Assad e Xu (1992) e já descritas nesta tese.

O terceiro algoritmo denomina-se Método da Contribuição Mínima e, assim como H1 e H2, também se caracteriza por ser um construtivo guloso. Esta heurística estima a contribuição mínima  $CM_i$  como sendo a soma entre o custo linear associado a cada aresta e os  $(n - 2)$  menores custos quadráticos entre  $i$  e as outras arestas. A lógica que sustenta essa estratégia é que se  $i$  é selecionada, então essas arestas têm maior probabilidade de comporem a solução. Neste caso, o problema da AGMQ é convertido em um problema de AGM com a seguinte função objetivo:

$$\text{Min} \left\{ z(x) = \sum_{i=1}^m CM_i x_i \mid x \in T \right\}, \text{ sujeito a:} \quad (2.41)$$

$$CM_i \approx w_i + \sum_{j=1}^m c_{ij} x_j, 0 < i \leq m \quad (2.42)$$

A busca tabu implementada por Cordone e Passeri (2008) emprega uma vizinhança bem simples: a cada movimento, ocorre a adição de uma aresta que está fora da árvore e a remoção de uma aresta na árvore dentre as que estão formando ciclo. Os autores afirmam que utilizam estruturas de dados adequadas para ter rápido acesso à lista de arestas candidatas à remoção, contudo, não mencionam que estruturas são essas.

Cada movimento é avaliado em  $O(1)$  haja vista que se mantém em memória a contribuição parcial de cada aresta, definida exatamente como a contribuição parcial de aresta livre  $CPL_i$  na Expressão 2.28. Nesse momento, aplicaremos a definição de  $CPL_i$  mesmo para arestas que não estejam livres, apenas para não adicionar desnecessariamente variáveis com mesmo significado matemático. Com isso, a variação da função objetivo em virtude da troca das arestas  $i \in I$  e  $i' \in L$ , é:

$$\delta_{ii'} = CPL_{i'} - CPL_i - q_{ii'} - q_{i'i} \quad (2.43)$$

Os autores implementaram mecanismo lista tabu para impedir que arestas recentemente removidas entrem novamente na árvore e arestas recentemente inseridas saiam rapidamente da árvore. O critério de aspiração é considerado, ou seja, movimentos que melhoram o melhor resultado conhecido até então são realizados, independentemente de respeitarem ou não a lista tabu.

Cordone e Passeri (2008) também implementaram um *branch-and-bound* com base em uma linearização aproximada da função-quadrática. O subproblema relaxado é, pois, uma AGM. Para cada aresta  $i$ , a estimativa linear  $EL_i$  inclui o custo linear  $w_i$ , os custos quadráticos com as arestas que já estão na árvore e um número adequado dos menores custos quadráticos com as arestas da árvore que ainda não foram definidas. Nesse sentido, o primeiro e o segundo termos dessa aproximação correspondem à contribuição real de cada aresta que já está na árvore para o custo parcial atual da árvore e o terceiro termo corresponde a uma estimativa, por baixo, da contribuição das arestas que ainda serão inseridas para completar a árvore.

A abordagem implementada por Cordone e Passeri (2008) considera como limite superior a solução do subproblema relaxado avaliada de acordo com a função-objetivo original. Um limite superior inicial é obtido pela aplicação da busca tabu inicializada pela heurística H2 - Método da Fixação Sequencial.

A etapa de *branching* (ramificação) consiste em fixar uma determinada aresta dentro ou fora de solução. A aresta de *branching* é a mais barata não fixada que pertence à solução relaxada. A estratégia de visita implementada é uma combinação das estratégias melhor-limite-superior-primeiro e melhor-limite-inferior-primeiro. Os autores iniciam a fase de *branching* seguindo a primeira estratégia e, quando o limite superior não melhora para um determinado número de nós de *branching*, o algoritmo passa a adotar a segunda estratégia.

Os experimentos computacionais relatados no trabalho de Cordone e Passeri (2008) contemplam um conjunto de 60 instâncias aleatórias do problema da AGMQ com tamanhos variando entre 10 e 30 e densidades de 33 grafos simples, a densidade pode ser calculada como a razão entre o número de arestas do grafo e o número máximo de arestas que o grafo poderia ter, correspondente ao número de arestas de um grafo completo com a mesma quantidade de nós. Os autores afirmam que todas as heurísticas executaram em

menos de 1 segundo e, dentre as três heurísticas descritas, a H2 - Método da Fixação Sequencial foi a que obteve melhores resultados. Além disso, a busca tabu executa em um tempo que varia de menos de 1 segundo a menos de 1 minuto. Cordone e Passeri não relatam o desempenho da busca tabu em termos de qualidade de solução encontrada, apenas comentam que melhoram o resultado de H2. O *branch-and-bound*, por sua vez, consegue resolver em menos de 1 minuto um subconjunto das instâncias de até 15 vértices. Os outros casos de teste não puderam ser solucionados em menos de 1 hora e, pois, apresentaram *gaps* significativos.

## 2.4.8 Öncan e Punnen (2010)

O trabalho de Öncan e Punnen (2010) apresenta um procedimento de relaxação lagrangeana para obtenção de limites inferiores e um algoritmo de busca local com limitação tabu para o problema da AGMQ.

Assad e Xu (1992), além de terem proposto os algoritmos descritos na Subseção 2.4.1, também apresentaram uma formulação linear para a AGMQ. A relaxação Lagrangeana proposta por Öncan e Punnen (2010) consiste em uma extensão dessa formulação anterior de Assad e Xu (1992), acrescentando mais duas inequações como restrições.

A vizinhança  $N_{(t)}^k$ , proposta por Öncan e Punnen (2010), seleciona aleatoriamente  $k$  arestas básicas (arestas cujos índices estão em  $I$ ) da árvore geradora  $t$ . Cada aresta  $e_i$  escolhida,  $i \in I$ , é substituída por uma aresta não-básica (aresta cujo índice está em  $L$ )  $e_j, j \in L$ , de tal forma que o resultado da operação ainda seja uma árvore geradora.  $e_j$  é escolhida de maneira que, dentre todas as candidatas, seja a que permite que a árvore geradora resultante tenha o melhor valor de função-objetivo possível. O algoritmo de busca local delineado no trabalho em referência adota  $k = 1$ , ou seja, uma vizinhança  $1 - \text{aresta}$ .

O algoritmo de busca local com limitação tabu proposto pelos autores consiste basicamente de duas fases. A primeira delas corresponde à fase de busca na vizinhança. O algoritmo é iniciado com uma solução randômica e, enquanto um mínimo local não é atingido, avalia-se o melhor indivíduo da vizinhança da árvore atual. Se ele melhorar o mínimo local, a árvore atual é atualizada com a nova solução encontrada. Após a primeira fase, a melhor solução já encontrada pelo algoritmo é atualizada com a árvore atual, em caso de melhoria.

A segunda fase consiste em um movimento randômico a partir de um ótimo local. Essa

fase alterna movimento randômico e busca local até que um número limite de iterações é atingido. O movimento randômico consiste em atualizar a árvore atual  $t$  com a  $k$  melhor solução na vizinhança de  $t$ , em que  $k$  tem seu valor escolhido aleatoriamente no intervalo  $[1, l]$ , sendo  $l$  um parâmetro do algoritmo. Se em algum momento da segunda fase do algoritmo a melhor solução encontrada for melhorada, então a segunda fase termina prematuramente. O critério de parada do algoritmo é por número de iterações.

O trabalho de Öncan e Punnen (2010) traz uma vasta seção de experimentos computacionais, que se inicia pela geração de instâncias para a AGMQ a partir das instâncias na QAPLIB.

Também há relatos de experimentos envolvendo obtenção de limites inferiores com base no método Lagrangeano desenvolvido por Öncan e Punnen (2010), no método de Gilmore (1962) e Lawler (1963) e nos métodos propostos por Assad e Xu (1992). Para o conjunto de instâncias testado, o limite Lagrangeano proposto pelos autores obteve limites inferiores melhores, porém, utilizando um tempo computacional maior.

Os autores acreditam que o desempenho da busca local desenvolvida é bastante encorajadora, haja vista que, em comparação com a heurística H2, o algoritmo obteve resultados, em média, 16,7% melhores para um primeiro grupo de instâncias e 62,3% melhores para outro grupo de instâncias. Vale salientar que o tempo computacional utilizado pela busca local é significativamente superior ao tempo de H2. Enquanto H2 executa em até 3,8 segundos, a busca local chega a executar em 6227 segundos.

### 2.4.9 Sundar e Singh (2010)

Sundar e Singh (2010) propõem uma abordagem via colônia artificial de abelhas para lidar com o problema da AGMQ. Essa abordagem é inspirada no comportamento das abelhas para aquisição de alimentos. As abelhas reais são classificadas em três grupos: empregadas, escoteiras e observadoras. As empregadas estão associadas a uma fonte de alimento. São elas que trazem os carregamentos de néctar para a colméia. As observadoras ficam na colméia à espera das abelhas empregadas, as quais, por meio de uma dança, informaram as abelhas observadoras sobre a qualidade das fontes de alimento encontradas. Já as escoteiras, são as abelhas que, em um dado momento, estão à procura de fonte de alimento na vizinhança da colméia. Nesse sentido, os autores afirmam que, em termos de algoritmos, as abelhas empregadas e observadoras realizam exploração enquanto as escoteiras são responsáveis pela exploração (SUNDAR; SINGH, 2010).

No algoritmo descrito pelos autores, uma solução (árvore geradora) é representada de acordo com a codificação por conjuntos de arestas, conforme Raidl e Julstrom (2003). O procedimento de inicialização do algoritmo associa uma árvore gerada aleatoriamente a cada uma das abelhas empregadas. Os custos quadráticos e linear de cada aresta são reunidos em um único custo, denominado custo potencial da aresta,  $POT_i$ , o qual pode ser calculado como na Expressão em 2.44.

$$POT_i = w_i + \sum_{j=1, j \neq i}^m (c_{ij} + c_{ji}), \forall i, 0 < i \leq m \quad (2.44)$$

Sendo assim, as soluções randômicas iniciais são obtidas por meio de um procedimento construtivo similar ao algoritmo de Prim (1957) que considera o custo da aresta como sendo o custo potencial. Além disso, ao invés de selecionar a aresta de menor custo que conecta um vértice na árvore em construção a um vértice fora dela, a adaptação do algoritmo de Prim (1957) usa roleta para escolher uma aresta aleatoriamente dentre as todas as candidatas. Nessa perspectiva, a probabilidade de escolher uma aresta é inversamente proporcional ao seu custo. A cada árvore gerada (uma para cada empregada) a melhor solução já encontrada pelo algoritmo é atualizada, se for o caso.

Depois do processo de inicialização, o algoritmo de Sundar e Singh (2010) se constitui basicamente de um laço de repetição, até que um critério de parada seja satisfeito.

O primeiro passo dentro do laço tenta gerar, para cada uma das abelhas empregadas  $i$ , uma nova fonte de alimento  $FA_{i'}$  que esteja na vizinhança da respectiva fonte atual  $FA_i$ . Esse procedimento segue o Algoritmo de Determinação de Fonte de Alimento Vizinha (ADFAV), o qual será explicado adiante. Se, para alguma empregada  $i$ , a árvore  $FA_{i'}$  gerada for vazia, isso significa que o ADFAV não conseguiu gerar uma nova árvore e, pois,  $FA_i$  é substituída por uma solução (fonte de alimento) gerada randomicamente. Senão, caso  $FA_{i'}$  seja melhor que  $FA_i$ ,  $FA_i$  é substituída por  $FA_{i'}$  e checa-se a possibilidade de a solução ótima até o momento ter sido melhorada. Por outro lado, se o ADFAV não conseguir melhorar  $FA_i$  por um determinado número limite de iterações, então  $FA_i$  é substituída por uma solução aleatória.

De acordo com Sundar e Singh (2010), o método ADFAV recebe como entrada uma solução, denominada  $s$ , a qual é copiada para a solução  $s'$ . O ADFAV escolhe randomicamente uma aresta  $e_r$  para ser removida da árvore  $s'$ . Para que  $s'$  volte a ser uma árvore geradora, faz-se necessário, pois, re-conectar os dois componentes criados. Para tanto, o ADFAV escolhe aleatoriamente uma solução  $u$  diferente de  $s'$ . Procura-se, em  $u$ , uma

aresta  $e_g$  diferente de  $e_r$  que possa re-conectar  $s'$  ao menor custo possível dentre todas as arestas candidatas presentes em  $u$ . Tal custo é analisado com base no custo linear de  $e_g$  e nos intercustos de  $e_g$  com as arestas remanescentes em  $s'$ . No caso de não haver aresta candidata  $e_g$  tal que seja possível realizar a substituição de  $e_r$  por  $e_g$ , a aresta  $e_r$  é re-estabelecida na árvore. A aresta  $e_r$  é inserida em uma lista tabu para que não possa ser selecionada novamente nas próximas iterações. O algoritmo continua escolhendo novas arestas a serem removidas e adicionadas segundo o procedimento detalhado acima até que alguma substituição seja efetuada. Se o algoritmo falhar mesmo após um determinado número de iterações, retorna-se um conjunto vazio indicando que a população precisa ser diversificada. Senão, retorna-se  $s'$ .

Ainda dentro do laço principal do algoritmo colônia de abelha, após a manipulação das soluções das abelhas empregadas, o algoritmo de Sundar e Singh (2010) tenta gerar, para cada abelha observadora  $j$  uma fonte de alimentação  $FA_j$ . O procedimento de seleção de uma fonte de alimentação para uma abelha observadora emprega a estratégia de seleção de torneio binário. Duas soluções são escolhidas aleatoriamente do conjunto de soluções manipuladas pelo algoritmo. A melhor dentre as duas soluções é escolhida para ser associada à abelha observadora com uma probabilidade  $bt$ . A pior dentre as duas soluções é associada à observadora com probabilidade  $1 - bt$ . Logo após, tenta-se gerar, utilizando a heurística ADFAV, uma nova fonte de alimentação  $FA_{j'}$  na vizinhança de  $FA_j$ . Se  $FA_{j'}$  for vazio ou pior que  $FA_j$ ,  $FA_j$  não é atualizado com  $FA_{j'}$ . Caso contrário,  $FA_j$  é atualizado com  $FA_{j'}$  e checka-se a possibilidade de a solução ótima até o momento ter sido melhorada. A qualquer momento, se uma árvore gerada for melhor que a árvore ótima armazenada, a solução ótima é atualizada.

Após o laço principal do algoritmo, Sundar e Singh (2010) utilizam um procedimento de busca local com o objetivo de melhorar ainda mais a qualidade da solução ótima encontrada até o momento. Tal busca local consiste em um processo iterativo. A cada iteração, a busca local considera cada aresta da solução isoladamente. A aresta em questão é excluída da árvore. Todas as arestas do grafo que podem conectar os dois componentes resultantes são avaliadas. A aresta que resultar no menor valor possível para a nova solução será inserida na árvore. A busca local é aplicada até que uma iteração completa falhe em melhorar a solução.

O algoritmo implementado por Sundar e Singh (2010) foi comparado ao algoritmo genético proposto por Zhou e Gen (1998) e a dois algoritmos genéticos propostos em Soak, Corne e Ahn (2006). Segundo os autores, o algoritmo de colônia de abelhas foi



melhor que todos os outros em termos de qualidade da solução encontrada. No quesito tempo computacional, o algoritmo proposto no artigo em análise também demonstrou excelente desempenho, sendo superado apenas pelo algoritmo de (ZHOU; GEN, 1998) que apresentou os piores resultados em termos de qualidade de solução dentre os algoritmos testados.

## 2.4.10 Palubeckis, Rubliauskas e Targamadzé (2010)

Palubeckis, Rubliauskas e Targamadzé (2010) apresentam três abordagens heurísticas para a AGMQ, a saber: um *simulated annealing multistart*, um algoritmo genético híbrido e uma busca tabu iterativa. Essas implementações são discutidas, respectivamente, nas Subseções 2.4.10.1, 2.4.10.2 e 2.4.10.3.

### 2.4.10.1 *Simulated annealing multistart*

O algoritmo de *simulated annealing multistart* implementado pelos autores se inicia gerando aleatoriamente uma árvore geradora  $AG$ .  $AG^*$ , variável que representa a melhor solução encontrada até um determinado momento, é inicializada com  $AG$ .  $z^*$ , o valor da função-objetivo de  $AG^*$ , é inicializado com  $z(AG)$ .

A seguir, calcula-se a (alta) temperatura inicial, denominada  $\bar{t}$ . O procedimento para obtenção de  $\bar{t}$  passa pela avaliação de um determinado número de vizinhos de  $AG$ . A estrutura de vizinhança adotada considera vizinho de  $AG$  a árvore geradora  $AG'$  que difere de  $AG$  em apenas uma aresta. Ou seja,  $AG'$  pode ser obtida a partir de  $AG$  pela troca de uma aresta  $e_r$  que está em  $AG$  por uma aresta  $e_g$  que não pertence a  $AG$ . De maneira geral, esse procedimento de assemelha à vizinhança  $N_{(AG)}^k$ , quando  $k$  igual a 1, apresentada por Öncan e Punnen (2010) e discutida na Subseção 2.4.8.

Seja  $\delta(AG, r, g) = z(AG') - z(AG)$ , a mudança no valor da função objetivo quando movemos de  $AG$  para  $AG'$  em função da remoção da aresta  $e_r$  e inserção da  $e_g$ . O valor atribuído a  $t$  é o maior valor absoluto de  $\delta(AG, r, g)$  obtido de uma amostra de 10000 árvores geradoras escolhidas aleatoriamente dentre os vizinhos de (PALUBECKIS; RUBLIAUSKAS; TARGAMADZÉ, 2010).

Outros parâmetros do algoritmo são: a taxa de resfriamento  $\alpha$ , a temperatura mínima  $t_0$  (próxima de 0) e o fator de repetição  $R_0$ . Esses parâmetros são empregados para definir o número de reduções de temperatura  $K = \lfloor (\log(t_0) - \log(\bar{t}))/\log\alpha \rfloor$ , bem como o número de árvores avaliadas a cada nível de temperatura  $R = R_0 n$ .

Após a inicialização desses parâmetros, o procedimento de *Simulated Annealing* (SA) propriamente dito é invocado. Esse procedimento, denominado pelos autores de SA será explicado em detalhes a seguir.

O primeiro passo de SA verifica se é a primeira execução do *simulated annealing*. Em caso negativo, a variável  $foa$  é inicializada com  $z^*$ . Senão, gera-se uma árvore aleatória  $AG$  e  $foa$  é inicializada com  $z(AG)$ . A variável  $foa$  representa o valor da função-objetivo da árvore atual. O fato de o SA sempre executar a partir de uma solução gerada aleatoriamente caracteriza o *multistart*.

Depois de gerada a solução inicial, arestas  $e_r$  e  $e_g$  são selecionadas aleatoriamente de maneira que a árvore resultante da substituição de  $e_r$  por  $e_g$ ,  $AG' = AG(e_r, e_g)$ , seja vizinha de  $AG$ . Seja  $\delta' = \delta(AG, e_r, e_g)$ . Se  $\delta' \leq 0$ , a troca resultou em uma melhor solução, então a árvore  $AG = AG'$  e  $foa = foa + \delta'$ . Se a nova solução é melhor que o ótimo atual, o ótimo também é atualizado. Caso a árvore obtida pela troca entre as arestas  $e_r$  e  $e_g$  não resulte em uma solução melhor, a solução  $AG$  ainda pode ser atualizada com  $AG'$  com uma probabilidade de  $\exp(-\delta'/t)$ , em que  $t$  é a temperatura corrente, inicializada, a princípio, com o valor de  $\bar{t}$ .

A escolha aleatória de arestas  $e_r$  e  $e_g$  ocorre iterativamente, até que o número de árvores avaliadas por nível de temperatura ( $R$ ) seja atingido. Quando isso ocorre, o número de reduções de temperatura  $i$  é incrementado e  $t = \alpha t$ . No caso de o número máximo de reduções de temperatura  $K$  ter sido atingido, o SA pára e retorna  $AG^*$  e  $z^*$ . Senão, o algoritmo volta a gerar arestas aleatórias  $e_r$  e  $e_g$  a serem trocadas de forma a gerar vizinhos de  $AG$  que serão avaliados.

Após a execução do SA, o algoritmo de Palubeckis, Rubliauskas e Targamadzé (2010) checa a condição de parada. Para fins de comparação entre as três abordagens propostas neste artigo, os autores definiram o mesmo critério de parada para todos os algoritmos, a saber: um determinado período de tempo transcorrido. Se a referida condição for satisfeita, o algoritmo retorna a árvore  $AG^*$  com valor de função objetivo igual a  $z^*$ . Caso contrário, SA é novamente aplicado.

#### 2.4.10.2 Algoritmo genético

O algoritmo genético proposto por Palubeckis, Rubliauskas e Targamadzé (2010) codifica a solução (ou seja, uma árvore) em um cromossomo seguindo a representação de lista de arestas. Na realidade, os autores apresentam dois algoritmos genéticos: um gené-

tico puro e a hibridização do mesmo com um procedimento de busca local. Vale salientar que a única diferença entre os dois métodos é a existência da busca local no algoritmo híbrido. Nesse sentido, a descrição a seguir corresponde ao algoritmo híbrido, pois assim teremos explicado as duas versões propostas.

O primeiro passo do algoritmo genético é a criação da população de indivíduos. Esse procedimento é realizado por meio de uma versão randomizada do algoritmo de Kruskal (1956). A cada iteração, o algoritmo leva em conta o conjunto de todas as arestas que podem ser utilizadas para ampliar a floresta  $(GF = (V, E_{sel}))$  atualmente composta pelos vértices e arestas já adicionados ao indivíduo. Nesse sentido, forma-se um subconjunto contendo as melhores candidatas a compor a árvore, ou seja, as arestas que não formam ciclo e que apresentam os menores valores para a soma  $Q_i = w_i + \sum_{e_j \in E_{sel}} c_{ij}$ . Uma das arestas é randomicamente selecionada a partir do subconjunto obtido. Se o novo indivíduo guarda semelhança acentuada com algum outro cromossomo da população, essa nova solução é descartada. Caso contrário, o indivíduo passa a compor a população (PALUBECKIS; RUBLIAUSKAS; TARGAMADZÉ, 2010).

Uma vez que a população foi criada, o algoritmo inicia a fase de evolução da população, constituída pelos seguintes passos: reprodução da população, aplicação de busca local e atualização da população (PALUBECKIS; RUBLIAUSKAS; TARGAMADZÉ, 2010).

Seja a população denotada por  $\Pi$ . A operação de cruzamento é realizada com base em duas soluções escolhidas randomicamente de  $\Pi$ . A princípio, identificam-se as arestas comuns aos dois indivíduos selecionados para reprodução. Essas arestas são inseridas no descendente. Em seguida, o descendente é completado com arestas que pertencem a apenas um dos pais, seguindo a mesma versão randomizada do algoritmo de Kruskal (1956) que já foi descrita.

Após a reprodução, o descendente é submetido a um procedimento de busca local que pode ser entendido como uma mutação. Tal busca local é a implementação do método de melhoria local *1-opt*. A cada iteração da busca local, a vizinhança  $N(AG)$  da árvore corrente  $AG$  é explorada. O algoritmo tenta trocar cada aresta da árvore com cada aresta fora da árvore que não implique a criação de um ciclo.

A estratégia de atualização da população é bastante conhecida da literatura. O pior indivíduo da população é substituído pelo descendente, caso este seja suficientemente diferente dos indivíduos em  $\Pi$ . Senão, o descendente é simplesmente descartado.

### 2.4.10.3 Busca Tabu Iterativa

Palubeckis, Rubliauskas e Targamadzé (2010) também apresentam uma busca tabu iterativa, denominada *ITS* (do inglês, *Iterated Tabu Search*). O primeiro passo de *ITS* é obter uma árvore geradora inicial, tal como os autores fazem no algoritmo *simulated annealing* já descrito. Em seguida, o procedimento *TS*, a busca tabu propriamente dita, é invocado com os seguintes parâmetros:  $AG$ , a árvore inicial;  $AG^*$ , a melhor árvore até o momento;  $z^*$ , o valor da função objetivo de  $AG^*$  e  $Itr$ , o número de iterações de uma execução da busca tabu, cujo valor é escolhido aleatoriamente no intervalo inteiro  $[Itr_{min}, Itr_{max}]$ .

O primeiro passo do *TS* é a inicialização de variáveis locais. Depois, inicia-se um laço que, em geral, executará  $Itr$  vezes. *TS* pode ter seu término antecipado no caso de o tempo máximo para execução do *ITS* ter sido atingido. A cada iteração do laço, o algoritmo tenta avaliar toda a vizinhança da árvore atualmente levada em consideração. A estrutura de vizinhança adotada é a mesma já mencionada durante a análise deste artigo. Checa-se, então, se o vizinho gerado melhora a solução corrente. Em caso afirmativo, o mesmo procedimento de busca local do algoritmo genético é empregado (PALUBECKIS; RUBLIAUSKAS; TARGAMADZÉ, 2010).

Após a execução de *TS*, checka-se a condição de parada do algoritmo. Caso a condição ainda não tenha sido satisfeita, o procedimento *GST* (do inglês, *Generate Spanning Tree*) é executado para gerar uma nova árvore inicial. Senão, o algoritmo pára e retorna a melhor solução até o momento.

O procedimento *GTS* proposto por Palubeckis, Rubliauskas e Targamadzé (2010) realiza uma perturbação na árvore atual  $AG$ . Além de  $AG$ , os parâmetros de *GST* incluem  $\bar{p}$ , que representa o número de arestas a serem removidas de  $AG$ ; e  $\bar{q}$ . O procedimento é randomizado. A cada iteração, escolhem-se aleatoriamente uma aresta da árvore para ser removida e uma aresta que não está na árvore para substituí-la. Esta escolha é realizada a partir de uma lista de candidatos de tamanho, no máximo  $q$ . Esta lista é construída pela inclusão de pares de arestas  $(e_r, e_g)$  do tipo (aresta na árvore, aresta fora da árvore) para os quais os valores  $\delta(AG, e_r, e_g)$  são os menores possíveis. Uma restrição adicional deste procedimento é que uma aresta só pode ser escolhida uma vez, quer seja para entrar ou para sair da árvore.

Para os experimentos computacionais, os autores levaram em consideração as instâncias disponibilizadas em Cordone e Passeri (2011), que possuem tamanhos variando entre

10 e 50 vértices, inclusive. Os resultados mostraram que, em relação à qualidade da solução encontrada, o melhor algoritmo foi a busca tabu (*ITS*), seguida pelo genético híbrido (*HGA*) e pelo *simulated annealing multistart* (*MSA*). O pior algoritmo foi o genético puro. Já em termos de tempo computacionais, o *HGA* é comparável ao *ITS* e ambos são significativamente mais rápidos que o *MSA*.

### 2.4.11 Cordone e Passeri (2012)

Cordone e Passeri (2012) apresentam três algoritmos para o problema da AGMQ, a saber: uma busca tabu, uma busca em vizinhança variável (VNS, do inglês, *Variable Neighbourhood Search*) e um *branch-and-bound*. Tais abordagens são objeto de estudo das Subseções 2.4.11.1, 2.4.11.2 e 2.4.11.3, respectivamente.

O artigo em referência é uma extensão do trabalho de Cordone e Passeri (2008), com a apresentação de um algoritmo novo, a abordagem *VNS*, assim como a descrição detalhada de versões melhoradas dos algoritmos busca tabu e *branch-and-bound* previamente discutidos.

#### 2.4.11.1 Busca tabu

A busca tabu proposta pelos autores inicia gerando uma árvore geradora aleatória *AG*. O mecanismo de obtenção de *AG* consiste em, simplesmente, escolher aleatoriamente  $n - 1$  arestas de *E* que não gerem ciclo.

Em seguida, busca-se melhorar *AG* com base em uma vizinhança simples: a inserção de uma aresta  $e_g$  que não está em *AG* e a remoção de uma aresta  $e_r$  de *AG* no ciclo gerado. A busca tabu de Cordone e Passeri (2012) tenta realizar todos os movimentos possíveis a partir da vizinhança especificada.

A visita de soluções já exploradas é evitada utilizando memória. Há duas listas tabu independentes: uma que proíbe a inserção de arestas removidas recentemente e outra que proíbe a remoção de arestas inseridas recentemente. Dessa forma, os autores armazenam, para cada aresta, o valor  $Itrio_e$ , que representa o número da última iteração em que a aresta *e* foi inserida ou removida da árvore. Para avaliar se uma aresta  $e_g$  pode ser inserida e se uma aresta  $e_r$  pode ser removida, o número da iteração atual do algoritmo tabu deve ser maior que  $Itrio_{e_g} + tabuten_{in}$  e que  $Itrio_{e_r} + tabuten_{out}$ . Caso contrário, o movimento é tabu.  $tabuten_{in}$  e  $tabuten_{out}$  representam o número de iterações em que a inserção e a remoção, respectivamente, são proibidas, ou seja, o tabu *tenure*.

Cordone e Passeri (2012) empregam uma estratégia de memória de curto prazo para favorecer intensificação ou diversificação de acordo com os resultados recentes. A estratégia consiste em, dentro de limites máximo e mínimo pré-estabelecidos, reduzir os valores de tabu *tenure* em uma unidade se houve melhoria da função-objetivo na iteração mais recente e incrementar tais valores em uma unidade no caso de piora da função-objetivo.

Vale salientar que o critério de aspiração é empregado. Nesse sentido, caso um movimento melhore o melhor resultado já encontrado, tal movimento é realizado mesmo que seja tabu.

O algoritmo executa um total de  $Itr$  iterações. A cada  $Itr/numAG$  iterações, a árvore  $AG$  é reinicializada com uma solução aleatória.  $numAG$  é um parâmetro que informa o número de reinicializações de  $AG$ .

#### 2.4.11.2 Busca em vizinhança variável (VNS)

O algoritmo *VNS* requer a definição de uma hierarquia de vizinhanças. A hierarquia definida por Cordone e Passeri (2012) é uma extensão da vizinhança empregada pelos autores nos movimentos do algoritmo tabu em 2.4.11.1. Denominemos tal hierarquia por  $NC_k$ , em que  $k$  varia de  $k_{min}$  até  $k_{max}$ . Sendo assim, na vizinhança  $NC_k$ ,  $k$  arestas que estão fora da árvore são inseridas nela e, pois,  $k$  arestas da árvore são removidas.

O *VNS* se inicia com uma solução gerada aleatoriamente denominada  $AG$ .  $k$  é inicializado com  $k_{min}$  e  $AG^*$ , a melhor solução encontrada pelo algoritmo, com  $AG$ . A vizinhança  $NC_k$  é aplicada. Ao atingir o ótimo local, verifica-se se  $AG^*$  foi melhorado. Em caso afirmativo,  $AG^*$  é atualizado e  $k = k_{min}$ . Senão,  $k$  é incrementado em uma unidade.

Sempre que  $k$  ultrapassa  $k_{max}$ ,  $k$  é atualizado com  $k_{min}$  e um procedimento de *shaking* (perturbação) extrai uma nova solução para atualizar  $AG$ . O procedimento de *shaking* consiste em gerar uma solução aleatória na vizinhança  $NC_k$  de  $AG$  evitando, se possível, que as arestas inseridas sejam retiradas e que as arestas removidas sejam recolocadas na árvore. A solução gerada substitui  $AG$ .

O algoritmo encerra sua execução quando um número predefinido de iterações de busca local é realizado. Seu retorno é  $AG^*$ .

### 2.4.11.3 *Branch-and-bound*

O algoritmo *branch-and-bound* proposto por Cordone e Passeri (2012) é uma reimplementação do algoritmo proposto por Assad e Xu (1992) para o problema da AGMQ, com algumas modificações.

A principal alteração diz respeito ao procedimento para calcular o limite inferior. De maneira geral, os autores seguem o procedimento descrito em Assad e Xu (1992). Entretanto, há uma relaxação, de modo que a necessidade de gerar um grafo acíclico é ignorada. Sendo assim, o limite inferior gerado é mais fraco (porém mais rápido) que o proposto por Assad e Xu (1992). Os autores também modificaram a condição para término do cálculo do limite inferior. O “Algoritmo por Níveis” é executado até que uma estimativa superior para o melhor limite inferior seja estritamente menor que o limite superior. Neste caso, não há possibilidade de melhorar o limite inferior a ponto de realizar uma poda, então o cálculo do limite inferior é interrompido (CORDONE; PASSERI, 2012).

O limite superior empregado consiste no valor da função-objetivo original da AGMQ aplicado à AGM que soluciona o subproblema relaxado a cada iteração do “Algoritmo por Níveis” (ASSAD; XU, 1992).

A etapa de *branching*, assim como em Cordone e Passeri (2008), corresponde à fixação de uma aresta dentro ou fora da solução. A regra para escolher a aresta que será examinada no *branching*, entretanto, é diferente. Neste trabalho, os autores seguem a estratégia melhor-limite-primeiro. Nesse sentido, a aresta de *branching* é a aresta não-fixada mais barata que pertence à solução do problema relaxado, assim como em Assad e Xu (1992).

Os autores acrescentam dois testes lógicos que permitem definir se uma determinada aresta deve ser inserida ou removida da solução ótima. Considere o grafo  $G = (V, E)$  e uma aresta  $e_k$  tal que  $k \in L$ , o conjunto das arestas que ainda podem ser inseridas na árvore. A primeira regra testa se o subgrafo  $G' = (V, e_i | i \in L \vee i \in I \setminus \{e_k\})$  é conexo. Se  $G'$  não for conexo, a aresta  $e_k$  é adicionada a  $I$ , o conjunto das arestas que compõem a árvore geradora. Por outro lado, a segunda regra testa se uma dada aresta  $e_k$  deve ser incluída em  $O$ , o conjunto das arestas que estão excluídas da árvore. Para tanto, o segundo teste verifica se o subgrafo  $G' = (V, e_i | i \in I \cup \{e_k\})$  é cíclico. Se  $G'$  for cíclico, então  $e_k$  deve ser adicionada ao conjunto  $O$ . Cordone e Passeri (2012) também empregam penalidades Lagrangeanas para fixar dentro (ou fora) da solução ótima uma aresta que pertence (ou não) à solução relaxada atual.

Cordone e Passeri (2012) apresentam experimentos computacionais envolvendo casos

de testes propostos pelos autores como também conjuntos de instâncias relatadas na literatura. O algoritmo de busca tabu se mostrou melhor e mais robusto que o VNS. Os resultados do algoritmo exato para as diversas instâncias empregadas também foram descritos.



## 3 Algoritmos Exatos

Este capítulo apresenta dois algoritmos exatos desenvolvidos para o problema da AGQA-bi, a saber: um algoritmo *backtracking*, na Seção 3.1, e um algoritmo *branch-and-bound* na Seção 3.2. O algoritmo da Seção 3.2 foi desenvolvido com base no *branch-and-bound* apresentado por Assad e Xu (1992) para o problema da AGMQA.

### 3.1 *Backtracking*

De acordo com Brassard e Bratley (1996), a técnica *backtracking* utiliza uma representação por grafo implícito para realizar uma busca no espaço de soluções. Em sua forma mais comum, o *backtracking* simula uma busca em profundidade em um grafo implícito dirigido (uma árvore). Isso significa que o grafo em si não é construído, mas há uma descrição disponível para seus vértices e arestas de modo que se possa simular a construção de partes relevantes do grafo no decorrer da busca.

O algoritmo *backtracking* implementado considera a divisão das arestas nos conjuntos  $I$ ,  $O$  e  $L$ , os quais representam, respectivamente, os índices das arestas que estão na árvore (solução) parcial; os índices das arestas que estão proibidas de entrar na árvore; e os índices das arestas que ainda não foram analisadas e, pois, estão livres para entrar nos conjuntos  $I$  ou  $O$ . Tal organização foi proposta por Assad e Xu (1992), conforme já mencionado no Capítulo 2.

Nesse sentido, cada nó do grafo implícito do *backtracking* representa uma solução parcial, ou seja, uma árvore geradora incompleta. As folhas do grafo implícito são os únicos nós que contêm soluções completas.

No algoritmo implementado, cada nó do grafo implícito tem, no máximo, dois filhos: um em que uma dada aresta do grafo de entrada (caso de teste) é inserida em  $I$  e outro em que tal aresta é inserida em  $O$ . Sendo assim, simular uma busca em profundidade no grafo implícito significa manipular as arestas em  $I$ ,  $O$  e  $L$ .

Para a implementação realizada adota-se a representação da árvore por meio de seu conjunto de arestas, tal como proposto em Raidl e Julstrom (2003). Neste caso, uma árvore geradora quadrática de um grafo  $G = (V, E)$  é um vetor contendo  $n - 1$  arestas. Esta estrutura será utilizada para a representação do grafo implícito em construção, representando o nó atual do grafo implícito que está sendo visitado.

A Figura 4 ilustra o grafo implícito construído ao longo da busca e o que cada um de seus nós representa. Tal grafo é percorrido em profundidade durante o processo de busca. Inicialmente, a solução em construção está vazia. A cada passo, uma aresta é escolhida para ser mandatória ou proibida.

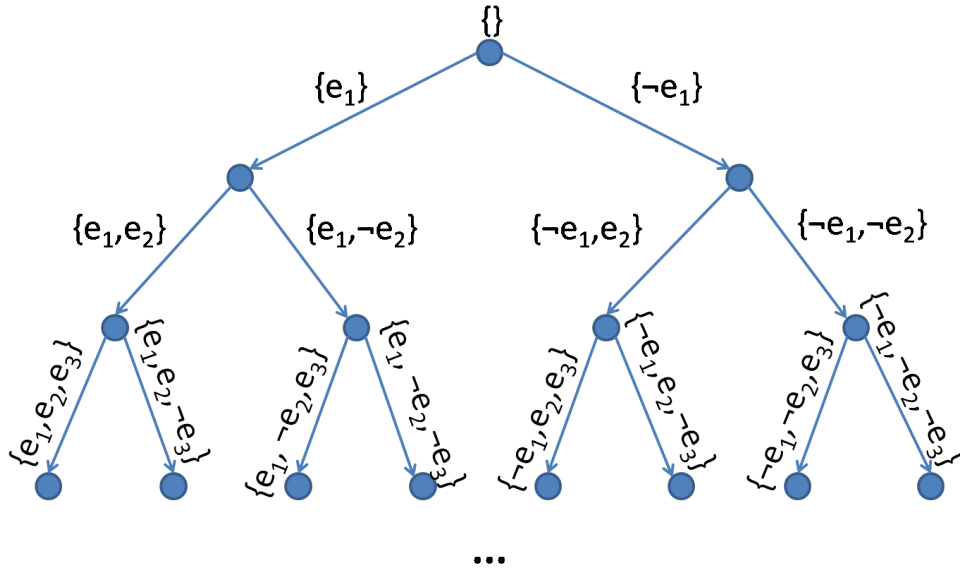


Figura 4: Grafo implícito do algoritmo *backtracking*.

O conjunto  $I$  é, pois, a solução parcial armazenada em uma solução. Emprega-se uma lista para representar o conjunto  $L$  e uma pilha para armazenar as arestas em  $O$ . Essas estruturas são denominadas, respectivamente, por *listaL* e *pilhaO*. No caso de *pilhaO* cada elemento armazenado na estrutura é uma dupla, composta pelo índice da aresta em  $O$  e por um número inteiro que representa o nível do grafo implícito em que a aresta foi inserida/excluída da solução. Como *pilhaO* é a estrutura de dados usada para o controle de *backtrack*, as arestas em  $I$  (que, temporariamente, também não podem ser inseridas na árvore) também são incluídas em *pilhaO*.

O pseudo-código do algoritmo *backtracking* implementado está disponível em 1. Como entrada do algoritmo tem-se um grafo  $G = (V, E)$  e, como saída, o conjunto de soluções não-dominadas, o qual corresponde à fronteira de Pareto ótima.

O algoritmo inicia com uma árvore vazia, com a pilha *pilhaO* vazia e com todas as

---

**Algoritmo 1** Algoritmo Backtracking ( $G = (V, E)$ )

---

Adicione todas as arestas de  $G$  a  $listaL$   
 Inicie  $sol$  com uma árvore vazia  
 Inicie  $pilhaO$  também vazia  
 Inicie  $ParetoFront$  com soluções da heurística H2 escalarizada  
 $end \leftarrow$  falso e  $tamSolAtual \leftarrow 0$   
**enquanto** ! $end$  **faça**  
      $e \leftarrow listaL.primeiroElemento()$   
     **se**  $e$  não existe porque  $ListaL$  está vazia **então**  
         **se**  $tamSolAtual = 0$  **então**  
              $end \leftarrow$  verdadeiro  
         **senão**  
             Retire de  $pilhaO$  os elementos associados com  $tamSolAtual$   
             Reinsira tais elementos em  $listaL$   
              $tamSolAtual \leftarrow tamSolAtual - 1$   
             Atualize o valor função-objetivo (parcial) de  $sol$   
         **fim se**  
     **senão se** a inserção de  $e$  em  $sol$  não gerar ciclo **então**  
          $sol[tamSolAtual] \leftarrow e$   
         Adicione  $(e, tamSolAtual)$  a  $pilhaO$   
         Remova  $e$  de  $listaL$   
         Atualize o valor da função-objetivo (parcial) para  $sol$   
         **se**  $sol$  corresponde a uma árvore completa **então**  
             Atualize  $ParetoFront$  de acordo com  $sol$   
         **senão se** a função-objetivo parcial de  $sol$  já é dominada por alguma solução em  $ParetoSet$  **então**  
             Não atualize  $tamSolAtual$   
             Atualize o valor da função-objetivo (parcial) de  $sol$  de forma a excluir  $e$   
         **senão**  
              $tamSolAtual \leftarrow tamSolAtual + 1$   
         **fim se**  
     **senão**  
         Adicione  $(e, tamSolAtual)$  a  $pilhaO$   
         Remova  $e$  de  $listaL$   
     **fim se**  
**fim enquanto**  
 retorne  $ParetoFront$ 


---

arestas em *listaL*. Um contador denominado *tamSolAtual* armazena o tamanho atual da solução parcial em construção.

Antes de iniciar o laço principal do algoritmo, o conjunto de soluções não-dominadas *ParetoSet* é iniciado com soluções obtidas a partir da escalarização da heurística H2. Esse pré-processamento melhora a eficiência do algoritmo, tendo em vista que as soluções iniciais obtidas funcionam como um limite superior. São gerados 21 vetores de escalarização, com um passo de 0,05, indo de  $(1, 0; 0, 0)$  até  $(0, 0; 1, 0)$ . A heurística H2 proposta por Assad e Xu (1992) (vide Capítulo 2) é, então, executada uma vez para cada vetor de escalarização definido. Neste caso, o valor da contribuição estimada de cada aresta não é dado pela simples soma da contribuição linear e da quadrática, mas pelo produto interno entre o vetor de escalarização e um vetor de contribuição, em que a primeira posição diz respeito à estimativa do custo linear e, a segunda, do custo quadrático.

Vale ressaltar que, em função dos valores escolhidos para a geração das instâncias (vide Capítulo 4) bem como da forma como são calculados, é possível que os valores dos custos totais linear e quadrático sejam demasiado discrepantes. Isso faz com que, ao escalarizar, um dos objetivos (em especial o linear) não consiga influenciar o valor escalarizado na proporção definida pelo vetor de escalarização. Nesse sentido, ao realizar escalarizações, os valores dos custos são normalizados para o intervalo de 0 a 1. Essa informação é válida para todos os algoritmos implementados nesta Tese.

A cada iteração  $i$ , uma aresta em *listaL*,  $e$ , é avaliada. Se  $e$  puder ser inserida na árvore sem gerar ciclos, essa aresta é removida de *listaL* e inserida na solução na posição *tamSolAtual*. A aresta  $e$  também é inserida em *pilhaO* com o valor do nível no grafo igual a *tamSolAtual*. O contador é incrementado em uma unidade.

A estrutura de dados utilizada para verificar a existência de ciclo é a *Union Find* Sedgewick (1983). Tal estrutura é adotada por todos os algoritmos implementados. A *Union Find* consiste em uma floresta em que cada árvore representa um conjunto conexo e é identificada pelo vértice que ocupa a posição da raiz. Se dois vértices fazem parte de árvores diferentes da floresta (que possuem vértices raiz diferentes), a inserção de uma aresta que os conecta não gera ciclo pois fazem parte de componentes conexos distintos. Neste caso, uma das árvores é anexada à outra de forma que a árvore final fique com a menor altura possível. Isso configura a união entre os dois conjuntos conexos em questão. A verificação de ciclo ocorre em tempo amortizado  $O(1)$ , tendo em vista que a cada vez que se consulta o identificador do componente conexo de um vértice, este vértice é conectado diretamente à raiz. Isso significa que, com as consultas, as árvores da *Union*

*Find* degeneram para uma configuração estrela (em que os filhos da raiz são folhas). Sendo assim, a remoção de arestas implica a reconstrução da estrutura.

Se, ao adicionar a aresta  $e$  na árvore uma árvore geradora tenha sido completada, calcula-se o valor da função-objetivo para a árvore criada (de acordo com as Expressões 2.5-2.7) e a mesma é adicionada ao conjunto de soluções não-dominadas encontradas até o momento, *ParetoFront*. Como o próprio nome do conjunto já indica, apenas as soluções não- dominadas são mantidas. Nesse sentido, se a solução gerada for dominada, ela será descartada. De modo análogo, se alguma solução do conjunto for dominada pela árvore recém gerada, então a solução dominada também será descartada.

Caso a adição da aresta  $e$  à solução não resulte ainda em uma árvore completa, verifica-se a possibilidade de o custo parcial da solução ainda em construção já ser dominado por alguma solução em *ParetoSet*. Em caso afirmativo, *tamSolAtual* não é atualizado, de forma que ocorre uma poda na árvore de busca. Do contrário, *tamSolAtual* é atualizado, o que significa que a aresta  $e$  está inserida na árvore. A busca continua, examinando a próxima aresta em *listaL*. Observe que esse algoritmo *backtracking* permite a realização de podas, entretanto, tais podas ocorrem em função apenas do valor da função-objetivo referente à árvore parcial construída. Nenhuma estimativa do custo total da árvore é realizada e o valor da função-objetivo parcial não influencia a escolha do próximo nó da árvore de busca a ser visitado.

Já foi comentada a situação em que a aresta  $e$  avaliada na iteração  $i$  pode ser inserida na árvore. Entretanto, pode ser que a aresta  $e$  não possa ser inserida na árvore, em virtude de sua inserção gerar ciclo. Neste caso, a aresta é incluída em *pilhaO*, juntamente com o valor de *tamSolAtual*.

Chegará um momento em que *listaL* estará vazia. Duas condições podem ocorrer neste caso, a saber: *tamSolAtual* igual a 0 ou *tamSolAtual* diferente de zero.

Se *tamSolAtual* for diferente de zero, isso implica que não existe nenhuma aresta disponível para ocupar a posição *tamSolAtual* da solução, seja porque não é possível obter uma árvore completa com a combinação de arestas já inserida em  $I$  ou porque todas as possibilidades já foram testadas. Neste caso, ocorre o *backtrack*, ou retorno.

Em termos do grafo implícito, o *backtrack* corresponde ao fato de, durante a busca em profundidade, voltar a um vértice já visitado e continuar a busca considerando um filho desse vértice que ainda não foi visitado.

Na prática, no algoritmo implementado, significa que:

- (i) As arestas em *pilhaO* com valor de nível associado igual a *tamSolAtual* serão removidas de *pilhaO* e reinseridas em *listaL*. Assim, estão novamente disponíveis para entrar na árvore.
- (ii) *tamSolAtual* será reduzido em uma unidade, representando o retorno da busca.
- (iii) A aresta  $e'$  que ocupa a posição *tamSolAtual* será substituída, ficando proibida de entrar na árvore.  $e'$  é inserida em *pilhaO* associada ao valor de *tamSolAtual*.

Ou seja, analisa-se o caso em que a aresta na posição *tamSolAtual* está em *O* e, pois, está fora da árvore.

Por outro lado, se *tamSolAtual* for igual a zero, então todas as arestas estão proibidas, indicando que todo o espaço de soluções já foi explorado e que a busca foi concluída.

## 3.2 *Branch-and-bound*

A técnica *branch-and-bound* também baseia seu mecanismo de busca no espaço de soluções no percorrimto de um grafo acíclico implícito. Segundo Brassard e Bratley (1996), a diferença básica entre essas técnicas é que, no caso do *branch-and-bound*, a cada nó visitado no grafo implícito, limites inferiores e superiores são calculados para definir se a busca naquele subgrafo deve prosseguir ou não, enquanto que o *backtracking* clássico não adota esse tipo de critério.

No *branch-and-bound* desenvolvido, o cálculo dos limites é combinado com uma busca em profundidade no grafo implícito. Os limites são usados para podar ramificações da árvore de busca e também para definir qual subárvore do grafo implícito parece ser mais promissora, de forma que este ramo possa ser explorado primeiro.

Considerando que o problema da AGQA-bi é um problema de minimização, um limite superior representa a melhor solução de fato encontrada até o momento e o limite inferior é uma estimativa “otimista” da melhor solução que ainda pode ser encontrada a partir da solução parcial construída. Por “otimista” quer-se dizer que o valor real da melhor solução que pode ser encontrada a partir da solução parcial construída nunca será melhor que o valor do limite inferior. Nesse sentido, se, para um dado nó do grafo implícito (uma configuração parcial da árvore) o limite inferior é pior que o limite superior, não vale a pena continuar a busca na subárvore enraizada pelo nó visitado, pois nenhuma árvore contendo tal configuração parcial pode ser melhor que a solução já encontrada. Neste

caso, ocorre uma poda, ou seja, o procedimento de *backtrack* é antecipado com o objetivo de evitar a exploração desnecessária de espaço de busca.

Tendo em vista que o problema da AGQA-bi possui múltiplos objetivos, o limite superior adotado neste algoritmo é o próprio conjunto de soluções não-dominadas encontrado até o momento, ou seja, *ParetoFront*.

O limite inferior, por sua vez, é uma adaptação do limite inferior empregado por Assad e Xu (1992) para a solução do problema da AGMQA, conforme subseção 2.4.1.2. Para o caso biobjetivo, o limite inferior é dado por um par ordenado, em que o primeiro elemento representa o limite inferior para o custo linear e, o segundo, o limite inferior para o custo quadrático.

O limite inferior no que se refere ao objetivo linear é o custo da AGM de  $G$  quando define-se que as arestas que compõem a configuração parcial da árvore são obrigatórias na AGM e considera-se como pesos das arestas unicamente seus respectivos custos lineares. O limite inferior para o objetivo quadrático também corresponde ao custo total de uma AGM de  $G$  em que as arestas da configuração parcial da árvore são obrigatórias. Entretanto, neste caso, o custo  $lb_i$  associado a cada aresta  $e_i$  de  $G$  que está fora da árvore é dado pelas Expressões 3.1.

$$lb_i = \begin{cases} \min \{c_{ij} | j \in L_i\}, & \text{se } I \cap A_i = \{\} \\ \sum_{j \in I} (c_{ij} + c_{ji}), & \text{caso contrário} \end{cases} \quad (3.1)$$

O algoritmo *branch-and-bound* desenvolvido emprega a mesma estrutura do algoritmo *backtracking* descrito. Neste sentido, as seguintes variáveis são utilizadas: *listaL*, *pilhaO*, *tamSolAtual*, *ParetoFront*. A solução também é representada por um conjunto de arestas, tal como proposto por Raidl e Julstrom (2003). O algoritmo inicia com todas as arestas do grafo de entrada disponíveis para entrar na árvore. Ou seja, *listaL* é iniciada com as arestas de  $G$ . O heurística H2 escalarizada é adotada para iniciar *ParetoFront*, assim como no *backtracking*. Dessa forma, obtém-se um limite superior inicial, o qual possibilita que as podas ocorram em profundidades mais baixas da árvore de busca. As outras variáveis são inicialmente vazias.

A cada passo do algoritmo, a aresta disponível com menor limite inferior no que se refere ao objetivo quadrático, e, é selecionada para entrar na árvore. Se a inclusão da aresta e não induz a existência de um ciclo na árvore parcial, essa aresta é removida de *listaL* e inserida na solução na posição *tamSolAtual*. A aresta e também é inserida em

*pilhaO* com o valor do nível no grafo igual a *tamSolAtual*.

Se, ao adicionar a aresta e na árvore uma árvore geradora tenha sido completada, calcula-se o valor da função-objetivo para a árvore criada (de acordo com as Expressões 2.5-2.7) e a mesma é adicionada ao conjunto de soluções não-dominadas encontradas até o momento, *ParetoFront*.

Por outro lado, caso a adição de *e* não seja suficiente para completar a árvore, o limite inferior é recalculado considerando que *e* seja aresta mandatória. O valor do limite inferior (um par ordenado) é comparado com os valores da função-objetivo das soluções, não-dominadas, em *ParetoFront*. Se o valor do novo limite for dominado em relação ao conjunto *ParetoFront*, a árvore é podada. Isso significa que a busca nesta subárvore do grafo implícito é prematuramente concluída e um *backtrack* é realizado. Senão, a busca prossegue e o contador é incrementado em uma unidade, confirmando que a aresta *e* permanece na configuração parcial da árvore.

Assim como no *backtracking*, quando a aresta *e* avaliada na iteração *i* não pode ser inserida na árvore em virtude de sua inserção gerar ciclo, e é incluída em *pilhaO*, juntamente com o valor de *tamSolAtual*.

No *branch-and-bound*, o mesmo procedimento do *backtracking* ocorre quando *listaL* se torna vazia. Nessa perspectiva, se *tamSolAtual* for diferente de zero, ocorre o *backtrack*, ou retorno. Por outro lado, se *tamSolAtual* for igual a zero, então todas as arestas estão proibidas, indicando que todo o espaço de soluções já foi explorado e que a busca foi concluída.

O pseudo-código do algoritmo *branch-and-bound* implementado está disponível a seguir. Como entrada do algoritmo temos um grafo  $G = (V, E)$  e, como saída, o conjunto de soluções não-dominadas, o qual corresponde à fronteira de Pareto ótima.

A literatura considera que, em geral, há uma grande diferença de desempenho entre as abordagens *backtracking* e *branch-and-bound* em favor do algoritmo *branch-and-bound*. Isso ocorre porque, quanto melhor for o limite inferior e superior calculado, maior o número de podas realizadas, menor a profundidade nas quais elas ocorrem e, pois, menor o espaço de busca pesquisado. É interessante notar que, para que tal fato seja verdade, deve ser possível calcular os limites de forma relativamente eficiente, para que o tempo de cálculo desses valores não supere o tempo que será economizado por meio das podas. Os algoritmos exatos implementados foram submetidos a experimentos computacionais, os quais estão relatados no Capítulo 4 e confirmam os resultados esperados.



---

**Algoritmo 2** Algoritmo *branch-and-bound* ( $G = (V, E)$ )

---

Adicione todas as arestas de  $G$  a *listaL*  
 Inicialize *sol* com uma árvore vazia  
 Inicie *pilhaO* também vazia  
 Inicie *ParetoFront* com soluções da heurística H2 escalarizada  
 $end \leftarrow falso$  e  $tamSolAtual \leftarrow 0$   
 $lb \leftarrow \text{CalculeLimiteInferior}(sol)$   
**enquanto**  $!end$  **faça**  
      $e \leftarrow$  Aresta de *listaL* com menor valor de  $lb_i$   
     **se**  $e$  não existe porque *listaL* está vazia **então**  
         **se**  $tamSolAtual = 0$  **então**  
              $end \leftarrow verdadeiro$   
         **senão**  
             Retire de *pilhaO* os elementos associado com  $tamSolAtual$   
             Reinsira tais elementos em *listaL*  
              $tamSolAtual \leftarrow tamSolAtual - 1$   
         **fim se**  
     **senão se** a inserção de  $e$  em *sol* não gerar ciclo **então**  
          $sol[tamSolAtual] \leftarrow e$   
         Adicione  $(e, tamSolAtual)$  a *pilhaO*  
         Remova  $e$  de *listaL*  
         **se** *sol* corresponde a uma árvore completa **então**  
             Calcule o valor da função-objetivo para *sol*  
             Atualize *ParetoFront* de acordo com *sol*  
         **senão**  
              $lb \leftarrow \text{CalculeLimiteInferior}(sol)$   
             **se** *ParetoFront*.Domina ( $lb$ ) for falso **então**  
                  $tamSolAtual \leftarrow tamSolAtual + 1$   
             **fim se**  
         **fim se**  
     **senão**  
         Adicione  $(e, tamSolAtual)$  a *pilhaO*  
         Remova  $e$  de *listaL*  
     **fim se**  
**fim enquanto**  
 retorne *ParetoFront*

---

## 4 Experimentos Computacionais – Algoritmos Exatos

Este capítulo é dedicado aos experimentos computacionais envolvendo os algoritmos exatos abordados no Capítulo 3, a saber: um *backtracking* e um *branch-and-bound*. Nesse sentido, os experimentos relatados dizem respeito à comparação entre os dois algoritmos mencionados. Antes de descrever os experimentos propriamente ditos, apresentam-se o conjunto de instâncias adotado bem como a metodologia empregada para comparação.

### 4.1 Instâncias

São adotados 4 conjuntos de instâncias para os experimentos relatados nessa Tese. O primeiro deles, denominado CP, é uma extensão do conjunto proposto por Cordone e Passeri (2012) para o problema da AGMQ. Os referidos autores disponibilizaram as instâncias, grafos de até 50 vértices, em <http://homes.di.unimi.it/~cordone/research/qmst.html>. O conjunto foi estendido para contemplar grafos com até 175 nós. As instâncias com 200 ou mais vértices podem ocupar mais de 4GB de memória, em função do armazenamento dos valores de intercustos. Assim, o limite de 175 vértices foi escolhido de forma a viabilizar experimentos computacionais em máquinas com até 8GB de memória e dois processos executando simultaneamente. Vale ressaltar que todos os algoritmos apresentados nesta Tese executarão sem problemas para instâncias maiores, caso haja máquinas com disponibilidade de memória adequada. Os outros três conjuntos de instâncias são denominados, respectivamente, OP1, OP2 e OP3. Esses conjuntos foram propostos por Öncan e Punnen (2010) e consistem em grafos completos, gerados de maneira aleatória, com até 50 nós. A seguir, descrever-se-ão os quatro conjuntos em detalhes.

O conjunto CP é originalmente composto por 108 grafos, divididos em 9 classes, de acordo com o número de vértices, o qual é um valor pertencente ao seguinte conjunto  $NV = 10, 15, 20, 25, 30, 35, 40, 45, 50$ . Cada classe é composta por 12 grafos, gera-

dos de acordo com uma combinação de três características, a saber: densidade do grafo (33%, 67% ou 100%); intervalo de escolha dos custos lineares ( $[0, 10]$  ou  $[0, 100]$ ) e intervalo de escolha dos custos quadráticos ou intercustos ( $[0, 10]$  ou  $[0, 100]$ ).

A extensão proposta para o conjunto CP contempla, além das instâncias já mencionadas, grafos com  $NV = 75, 100, 125, 150, 175$ , totalizando 168 casos de teste. Cada nova classe também se constitui em 12 grafos, gerados aleatoriamente, de acordo com a combinação das três características citadas. Como os grafos são maiores, os intervalos de escolha dos custos lineares e quadráticos foram ambos modificados para ( $[0, 100]$  ou  $[0, 1000]$ ).

As instâncias são nomeadas por meio do número de vértices e de um número identificador variando de 1 a 12, de acordo com a combinação das características mencionadas. Por exemplo, o caso de teste 10.1 é um grafo com 10 vértices, cuja densidade é 33%, o custo linear está no intervalo  $[0, 10]$  e o quadrático também. A tabela 1 ilustra as características das instâncias CP com 10 vértices e com 100 vértices. O mesmo padrão se repete para todas as outras instâncias do conjunto. As colunas Inst, Dens, CLin e CQuad apresentam, respectivamente, o rótulo da instância, a densidade do grafo, o valor máximo para o custo linear e para o custo quadrático.

Tabela 1: Exemplificação das características das instâncias CP

Inst	Dens	CLin	CQuad	Inst	Dens	CLin	CQuad
10.1	33%	10	10	100.1	33%	100	100
10.2	33%	10	100	100.2	33%	100	1000
10.3	33%	100	10	100.3	33%	1000	100
10.4	33%	100	100	100.4	33%	1000	1000
10.5	67%	10	10	100.5	67%	100	100
10.6	67%	10	100	100.6	67%	100	1000
10.7	67%	100	10	100.7	67%	1000	100
10.8	67%	100	100	100.8	67%	1000	1000
10.9	100%	10	10	100.9	100%	100	100
10.10	100%	10	100	100.10	100%	100	1000
10.11	100%	100	10	100.11	100%	1000	100
10.12	100%	100	100	100.12	100%	1000	1000

Os conjuntos de instâncias OP1, OP2 e OP3 são compostos por grafos completos com  $NV = 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 20, 30, 50$ . Para cada classe de instâncias definida em função do número de vértices do grafo, foram geradas 10 instâncias.

Nesse sentido, esses três conjuntos totalizam 480 instâncias. O conjunto OP1 contém grafos com matriz de intercustos simétrica. Tanto os custos lineares quanto os intercustos são escolhidos de acordo com distribuição de probabilidade uniforme nos intervalos  $[1, 100]$  e  $[1, 20]$ , respectivamente. Para OP2, os custos lineares são escolhidos no intervalo  $[1, 10.000]$ . Os custos quadráticos, por sua vez, são definidos por um produtório. Para cada vértice, associa-se, no intervalo uniforme  $[1, 10]$ , um peso aleatório. Os autores definiram, pois, o intercusto entre duas arestas como sendo o produtório entre os pesos associados aos quatro vértices terminais das arestas. Para gerar as instâncias no conjunto OP3, escolhem-se coordenadas aleatórias para cada vértice no interior do retângulo de coordenadas  $(0, 0)(0, 100)(100, 0)(100, 100)$ . O custo linear de uma aresta é calculada, pois, como a distância euclidiana entre os seus vértices terminais. O intercusto entre duas arestas é, por sua vez, definido como a distância entre as coordenadas dos pontos médios das arestas.

## 4.2 Metodologia de Comparação

A comparação entre os algoritmos exatos tem como objetivo ilustrar a diferença de desempenho entre o algoritmo backtracking e o algoritmo branch-and-bound. Tendo em vista que ambos produzem os mesmos resultados em termos de fronteira de Pareto, a comparação será realizada com base em três informações, a saber: tempo de processamento (em segundos), número de nós visitados no grafo implícito e o número de árvores geradoras mínimas completas geradas. De qualquer modo, também será exibido o número de soluções não-dominadas pertencentes às fronteiras de Pareto geradas.

Os algoritmos foram implementados em C++, usando o compilador g++ e o sistema operacional Scientific Linux 5.5 de 64 bits. Todos os experimentos foram realizados em máquinas HP Z400, com processador Intel Xeon QuadCore W3520 de 2.8 GHz e 8Gb de RAM.

## 4.3 Resultados

As Tabelas 2, 3, 4 e 5 apresentam as informações de execução do algoritmo *backtracking* para as instâncias CP, OP1, OP2 e OP3, respectivamente. O mesmo é válido para as tabelas 6, 7, 8 e 9 no que se refere ao algoritmo *branch-and-bound*. Tais informações são: tempo de processamento (em segundos), número de nós visitados no grafo

implícito, número de árvores geradoras mínimas completas geradas e número de soluções não-dominadas obtidas. Os algoritmos foram executados até a classe de instâncias na qual o tempo de execução atingiu, em média, a cada do milhar.

Tabela 2: Resultados do *backtracking* para instâncias CP

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
10.1	0,00	7.097	80	5
10.2	0,00	7.098	80	5
10.3	0,00	7.105	85	7
10.4	0,00	7.106	87	7
10.5	0,28	1.220.213	6.816	7
10.6	0,25	1.136.869	5.554	11
10.7	0,27	1.207.395	7.585	9
10.8	0,26	1.124.700	6.478	14
10.9	3,03	14.188.485	96.837	13
10.10	2,58	11.708.418	77.708	21
10.11	3,19	13.803.348	109.104	17
10.12	2,66	11.374.045	87.842	30
15.1	71,29	232.409.214	84.022	16
15.2	70,10	221.495.709	79.799	26
15.3	68,69	217.248.471	73.617	21
15.4	70,34	205.119.467	68.563	40
15.5	3.589,50	13.792.420.130	807.407	13
15.6	3.392,37	13.494.355.363	714.132	16

Tabela 3: Resultados do *backtracking* para instâncias OP1

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
6.1	0,00	1.978	470	9
6.2	0,00	2.468	856	14
6.3	0,00	1.789	386	3
6.4	0,00	2.233	678	8
6.5	0,00	1.273	181	4
6.6	0,00	1.627	345	8
6.7	0,00	2.155	616	5

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
6.8	0,00	2.165	616	8
6.9	0,00	1.556	327	4
6.10	0,00	1.968	464	8
7.1	0,00	17.344	2.029	16
7.2	0,00	13.745	1.510	6
7.3	0,00	11.342	1.074	7
7.4	0,00	7.779	563	5
7.5	0,00	15.236	1.868	13
7.6	0,00	18.133	2.526	12
7.7	0,00	13.974	1.373	6
7.8	0,00	12.337	1.189	7
7.9	0,00	18.531	2.646	12
7.10	0,00	16.347	2.080	14
8.1	0,02	113.698	5.889	15
8.2	0,04	175.381	11.909	20
8.3	0,04	148.231	10.102	16
8.4	0,03	138.003	7.090	18
8.5	0,03	149.199	10.266	14
8.6	0,04	188.164	17.368	18
8.7	0,02	89.408	4.031	12
8.8	0,05	203.572	22.730	25
8.9	0,03	134.197	7.015	16
8.10	0,03	152.861	11.452	15
9.1	0,17	783.023	14.099	12
9.2	0,14	693.174	10.103	9
9.3	0,32	1.429.627	37.656	17
9.4	0,21	997.487	13.437	17
9.5	0,18	863.931	11.999	18
9.6	0,25	1.106.355	20.108	17
9.7	0,25	1.112.333	20.838	21
9.8	0,17	814.556	16.005	15
9.9	0,40	1.701.038	50.121	27

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
9.10	0,10	523.006	5.300	7
10.1	1,95	8.771.838	45.386	16
10.2	1,64	7.198.225	40.643	33
10.3	2,33	9.861.308	56.598	23
10.4	3,33	14.395.395	114.353	24
10.5	3,07	13.163.108	93.115	25
10.6	4,69	19.158.246	343.032	28
10.7	1,02	4.614.273	34.706	28
10.8	2,62	10.459.257	94.894	33
10.9	1,48	6.838.906	44.833	16
10.10	1,57	7.261.846	42.157	19
11.1	16,76	70.471.290	209.886	24
11.2	17,58	79.983.780	96.127	22
11.3	13,43	62.032.843	173.324	23
11.4	6,80	31.039.001	54.257	24
11.5	23,18	95.764.689	285.611	32
11.6	12,02	56.306.640	51.911	18
11.7	12,39	52.871.577	112.370	22
11.8	9,92	43.969.444	121.306	26
11.9	6,94	31.362.807	48.048	18
11.10	16,91	71.985.103	141.119	27
12.1	67,42	270.933.405	233.319	29
12.2	129,58	528.209.934	228.565	32
12.3	202,40	802.577.344	561.487	25
12.4	125,84	502.326.496	466.400	32
12.5	105,96	434.680.597	267.095	30
12.6	134,66	614.579.242	164.754	26
12.7	126,88	516.279.026	322.907	34
12.8	156,12	609.065.612	724.657	47
12.9	312,16	1.204.921.769	1.060.170	31
12.10	182,98	735.060.745	693.458	31
13.1	480,36	2.152.932.011	172.775	28

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
13.2	603,72	2.649.269.565	185.828	29
13.3	1.194,13	4.994.490.559	592.027	34
13.4	3.183,62	11.833.899.267	3.095.023	46
13.5	1.770,87	7.020.140.107	1.388.690	42
13.6	447,71	1.633.459.211	349.928	33
13.7	1.328,07	5.396.246.841	1.769.520	33
13.8	500,76	2.259.663.380	192.357	28
13.9	1.943,76	8.084.290.038	1.328.788	37
13.10	629,06	2.606.390.051	329.655	25

Tabela 4: Resultados do *backtracking* para instâncias OP2

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
6.1	0,00	2.345	739	10
6.2	0,00	1.956	465	9
6.3	0,00	2.015	483	9
6.4	0,00	1.983	516	11
6.5	0,00	1.925	479	10
6.6	0,00	1.224	204	4
6.7	0,00	1.449	295	6
6.8	0,00	1.630	354	10
6.9	0,00	1.489	329	9
6.10	0,00	1.692	337	3
7.1	0,00	14.195	1.784	15
7.2	0,00	15.807	2.228	21
7.3	0,00	10.047	826	7
7.4	0,00	10.897	1.250	12
7.5	0,00	11.534	1.244	10
7.6	0,00	14.275	2.224	16
7.7	0,00	11.922	1.319	15
7.8	0,00	9.971	886	7
7.9	0,00	16.282	2.418	19
7.10	0,00	9.939	831	7



Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
8.1	0,02	104.083	6.458	11
8.2	0,03	130.447	7.370	18
8.3	0,04	174.749	15.727	24
8.4	0,02	94.966	7.481	15
8.5	0,01	63.068	2.965	10
8.6	0,02	89.542	4.815	13
8.7	0,03	132.250	9.640	21
8.8	0,02	109.619	7.545	19
8.9	0,03	109.644	6.995	24
8.10	0,03	107.897	7.496	25
9.1	0,15	683.025	16.621	10
9.2	0,13	624.596	12.905	13
9.3	0,16	751.301	21.526	16
9.4	0,18	793.157	27.197	22
9.5	0,17	732.819	18.794	20
9.6	0,14	674.129	16.332	19
9.7	0,12	558.234	12.581	10
9.8	0,18	789.593	32.957	23
9.9	0,20	824.791	39.203	24
9.10	0,29	1.180.965	56.974	39
10.1	4,88	18.262.826	409.912	48
10.2	1,74	7.067.950	130.087	33
10.3	1,13	4.657.530	70.819	24
10.4	1,81	7.614.841	114.450	37
10.5	2,14	8.371.125	87.044	29
10.6	1,28	5.556.405	68.617	22
10.7	2,20	9.201.437	114.368	30
10.8	1,77	6.994.561	114.014	33
10.9	1,60	6.515.457	105.280	27
10.10	0,91	3.867.311	36.867	23
11.1	34,06	127.462.747	595.837	56
11.2	13,62	57.281.210	655.374	30

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
11.3	15,94	53.037.544	482.160	54
11.4	22,94	85.823.622	740.551	51
11.5	11,40	43.579.310	222.032	32
11.6	11,68	46.240.422	340.803	38
11.7	26,70	90.044.909	1.009.629	55
11.8	14,48	58.785.573	657.701	46
11.9	15,53	55.076.552	784.356	65
11.10	21,26	80.459.748	508.877	39
12.1	216,57	787.483.044	4.741.542	62
12.2	108,62	417.878.723	2.257.982	26
12.3	250,33	1.006.223.899	5.567.484	42
12.4	171,66	604.196.290	3.159.099	67
12.5	130,53	483.429.029	1.885.034	49
12.6	172,89	668.146.997	1.716.881	45
12.7	99,07	376.191.873	2.301.451	28
12.8	137,73	513.635.734	2.795.667	47
12.9	121,65	458.797.254	1.337.600	43
12.10	151,24	561.490.074	1.987.036	46
13.1	1.661,20	5.757.057.568	9.343.495	40
13.2	3.468,83	9.700.272.782	13.360.875	94
13.3	1.990,73	6.618.831.496	18.075.040	62
13.4	1.826,48	6.137.799.710	12.349.021	111
13.5	2.794,62	8.608.954.070	9.999.801	62
13.6	1.343,78	4.951.618.639	8.061.761	52
13.7	581,94	2.161.572.702	4.952.084	62
13.8	2.178,91	7.394.514.431	44.305.650	44
13.9	1.337,52	5.452.906.151	13.527.635	27
13.10	1.491,14	5.324.638.372	10.354.876	53

Tabela 5: Resultados do *backtracking* para instâncias OP3

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
6.1	0,00	2.788	1.153	11
6.2	0,00	2.696	1.064	7

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
6.3	0,00	2.546	915	10
6.4	0,00	2.667	1.032	15
6.5	0,00	2.676	1.041	12
6.6	0,00	2.717	1.082	7
6.7	0,00	2.831	1.196	9
6.8	0,00	2.585	950	13
6.9	0,00	2.691	1.056	9
6.10	0,00	2.534	914	9
7.1	0,01	29.060	9.363	11
7.2	0,01	28.870	9.038	20
7.3	0,01	26.459	6.912	25
7.4	0,01	31.714	11.708	15
7.5	0,00	24.790	5.823	21
7.6	0,01	27.945	8.261	17
7.7	0,01	28.689	9.084	9
7.8	0,01	28.014	9.001	19
7.9	0,00	24.656	6.243	17
7.10	0,01	28.868	9.351	8
8.1	0,08	325.502	64.361	13
8.2	0,08	347.525	96.071	11
8.3	0,07	304.112	55.745	17
8.4	0,06	261.539	41.341	21
8.5	0,10	349.804	84.320	28
8.6	0,06	266.961	38.351	18
8.7	0,07	280.790	50.460	24
8.8	0,06	229.523	36.134	21
8.9	0,07	296.331	55.868	17
8.10	0,09	348.689	74.359	19
9.1	1,09	4.190.368	450.824	23
9.2	0,90	3.680.099	277.108	19
9.3	0,86	3.536.502	294.714	18
9.4	0,62	2.685.883	264.164	17

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
9.5	0,76	3.261.746	265.800	17
9.6	0,90	3.715.440	321.824	17
9.7	0,93	3.723.938	346.244	20
9.8	1,13	4.533.178	557.113	13
9.9	0,89	3.508.465	296.523	24
9.10	0,89	3.705.259	315.731	15
10.1	11,39	45.236.527	1.551.600	21
10.2	9,21	37.060.959	2.545.476	27
10.3	11,20	45.126.070	1.784.752	19
10.4	9,58	40.937.643	2.580.983	20
10.5	10,66	44.400.093	1.427.894	22
10.6	8,18	35.053.171	1.906.675	19
10.7	9,77	41.985.082	1.603.869	21
10.8	10,20	42.052.608	1.396.592	19
10.9	11,46	45.045.611	1.586.195	27
10.10	8,56	34.714.112	1.517.382	28
11.1	149,91	581.787.153	6.753.426	27
11.2	118,21	488.577.782	5.636.711	29
11.3	113,97	453.213.289	9.518.385	27
11.4	138,82	531.812.045	11.455.624	36
11.5	126,56	501.394.736	7.939.680	36
11.6	163,26	666.475.097	8.042.689	29
11.7	189,65	788.429.860	23.139.431	17
11.8	107,99	463.704.275	6.247.221	18
11.9	130,82	525.741.060	7.864.569	29
11.10	144,48	563.818.697	9.924.705	29
12.1	2.321,85	8.608.234.158	37.965.996	32
12.2	1.701,42	6.648.660.569	38.866.385	29
12.3	1.135,36	4.627.478.847	76.117.450	28
12.4	1.710,75	6.722.547.318	23.728.097	31
12.5	1.559,98	6.073.126.182	29.536.378	25
12.6	2.487,12	9.734.670.225	60.182.282	21

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
12.7	1.766,65	7.021.067.819	20.029.771	18
12.8	931,28	3.890.754.971	37.331.607	26
12.9	320,29	1.272.252.045	17.821.634	24
12.10	1.878,76	7.439.413.558	34.653.757	28

Tabela 6: Resultados do *branch-and-bound* para instâncias CP

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
10.1	0,00	369	14	5
10.2	0,00	316	14	5
10.3	0,00	381	17	7
10.4	0,00	333	17	7
10.5	0,04	18.502	140	7
10.6	0,03	19.302	207	11
10.7	0,03	16.757	137	9
10.8	0,03	20.098	234	14
10.9	0,37	190.445	1.421	13
10.10	0,42	208.648	2.218	21
10.11	0,39	199.599	1.615	17
10.12	0,44	221.473	2.195	30
15.1	0,39	248.587	277	16
15.2	0,50	325.329	846	26
15.3	0,41	264.167	323	21
15.4	0,54	349.268	1.077	40
15.5	49,41	18.811.547	2.123	14
15.6	42,08	15.379.606	5.134	19
15.7	51,63	19.803.561	4.145	25
15.8	42,67	15.567.627	7.485	35
15.9	2.248,20	635.894.199	16.984	25
15.10	2.541,25	693.058.692	39.226	33
15.11	1.910,92	535.708.586	28.292	43
15.12	2.333,70	632.304.396	54.178	58
20.1	804,86	375.600.900	16.865	31

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
20.2	999,51	467.708.070	15.339	39
20.3	1.085,81	504.485.162	27.701	60
20.4	1.424,98	656.363.595	31.717	86
20.5	3.419,92	778.515.188	42.455	26

Tabela 7: Resultados do *branch-and-bound* para instâncias OP1

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
6.1	0,00	593	95	9
6.2	0,00	858	215	14
6.3	0,00	378	44	3
6.4	0,00	576	74	8
6.5	0,00	294	52	4
6.6	0,00	361	58	8
6.7	0,00	294	25	5
6.8	0,00	504	88	8
6.9	0,00	253	38	4
6.10	0,00	459	74	8
7.1	0,00	2.273	317	16
7.2	0,00	1.777	106	6
7.3	0,00	1.628	141	7
7.4	0,00	973	72	5
7.5	0,00	2.193	211	13
7.6	0,00	2.421	199	12
7.7	0,00	751	53	6
7.8	0,00	1.793	140	7
7.9	0,01	3.558	275	12
7.10	0,00	2.828	235	14
8.1	0,01	7.909	366	15
8.2	0,02	10.472	701	20
8.3	0,02	8.540	348	16
8.4	0,01	8.151	440	18
8.5	0,02	12.278	575	14

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
8.6	0,02	14.585	874	18
8.7	0,01	6.926	234	12
8.8	0,02	15.874	835	25
8.9	0,02	9.556	479	16
8.10	0,02	10.330	549	15
9.1	0,05	22.554	407	12
9.2	0,05	25.418	405	9
9.3	0,13	69.179	1.994	17
9.4	0,05	25.862	356	17
9.5	0,06	29.500	394	18
9.6	0,08	44.307	897	17
9.7	0,10	50.829	841	21
9.8	0,05	25.848	348	15
9.9	0,13	71.162	2.000	27
9.10	0,02	10.714	59	7
10.1	0,28	133.053	1.146	16
10.2	0,37	168.857	1.748	33
10.3	0,30	141.832	1.624	23
10.4	0,58	284.228	2.953	24
10.5	0,71	349.678	3.860	25
10.6	0,59	282.023	5.251	28
10.7	0,22	98.067	1.240	28
10.8	0,66	310.728	5.088	33
10.9	0,23	103.522	1.438	16
10.10	0,31	149.382	1.179	19
11.1	2,02	854.982	4.109	24
11.2	2,16	913.595	3.132	22
11.3	1,65	692.912	3.185	23
11.4	0,91	367.858	1.284	24
11.5	3,07	1.366.049	8.102	32
11.6	1,41	581.118	1.126	18
11.7	1,47	614.407	3.322	22

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
11.8	1,17	478.948	3.304	26
11.9	0,92	377.924	2.024	18
11.10	2,07	888.399	6.178	27
12.1	5,95	2.132.257	6.859	29
12.2	9,84	3.758.552	7.361	32
12.3	11,25	4.183.153	11.079	25
12.4	6,21	2.267.218	10.355	32
12.5	6,48	2.396.663	6.365	30
12.6	4,56	1.659.772	2.056	26
12.7	11,79	4.291.441	8.654	34
12.8	9,15	3.417.237	7.618	47
12.9	18,20	6.948.148	12.337	31
12.10	16,28	6.246.751	15.838	31
13.1	24,58	7.959.694	3.892	28
13.2	30,13	9.658.147	4.976	29
13.3	44,68	14.455.727	12.730	34
13.4	87,55	29.544.315	30.244	46
13.5	49,43	16.396.424	17.276	42
13.6	25,32	7.903.780	12.495	33
13.7	77,00	26.398.114	19.658	33
13.8	19,86	6.362.024	3.871	28
13.9	65,76	22.221.070	16.373	37
13.10	25,82	8.273.932	4.547	25
14.1	309,57	89.521.386	25.264	42
14.2	370,81	109.752.002	19.909	56
14.3	112,74	32.162.942	12.414	35
14.4	239,28	69.190.840	26.242	43
14.5	240,76	70.527.966	24.515	44
14.6	187,66	54.755.661	33.053	47
14.7	154,67	43.432.414	11.645	30
14.8	432,20	126.747.259	46.134	54
14.9	144,45	41.055.214	18.729	43



<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
14.10	308,11	89.282.129	40.547	38
15.1	717,05	186.409.537	20.147	44
15.2	902,02	235.040.917	20.115	39
15.3	1.235,54	323.670.230	31.130	44
15.4	761,25	197.889.072	15.348	39
15.5	1.032,80	269.392.447	27.034	53
15.6	895,30	230.141.706	22.762	49
15.7	1.371,44	367.236.012	50.082	45
15.8	1.196,06	313.171.754	42.368	52
15.9	851,65	217.599.753	19.601	37
15.10	1.996,88	529.431.756	42.998	47

Tabela 8: Resultados do *branch-and-bound* para instâncias OP2

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
6.1	0,00	485	76	10
6.2	0,00	375	49	9
6.3	0,00	445	70	9
6.4	0,00	382	55	11
6.5	0,00	599	123	10
6.6	0,00	126	10	4
6.7	0,00	405	81	6
6.8	0,00	377	55	10
6.9	0,00	358	55	9
6.10	0,00	103	4	3
7.1	0,00	923	105	15
7.2	0,00	1.782	177	21
7.3	0,00	595	35	7
7.4	0,00	1.415	183	12
7.5	0,00	1.325	103	10
7.6	0,00	2.221	205	16
7.7	0,00	1.691	116	15
7.8	0,00	710	34	7

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
7.9	0,00	2.419	233	19
7.10	0,00	782	27	7
8.1	0,01	2.827	194	11
8.2	0,01	6.581	366	18
8.3	0,02	10.308	570	24
8.4	0,01	5.243	234	15
8.5	0,01	2.719	122	10
8.6	0,01	2.209	94	13
8.7	0,01	6.809	335	21
8.8	0,01	7.802	255	19
8.9	0,01	6.602	282	24
8.10	0,02	10.086	506	25
9.1	0,02	10.163	494	10
9.2	0,02	9.751	395	13
9.3	0,03	16.497	612	16
9.4	0,04	21.596	784	22
9.5	0,05	22.547	891	20
9.6	0,02	11.220	318	19
9.7	0,02	10.150	418	10
9.8	0,06	30.133	1.006	23
9.9	0,06	31.808	967	24
9.10	0,11	56.594	1.360	39
10.1	1,01	462.969	7.070	48
10.2	0,25	109.121	1.492	33
10.3	0,14	64.644	1.557	24
10.4	0,16	71.319	1.154	37
10.5	0,42	186.211	2.189	29
10.6	0,13	55.916	894	22
10.7	0,20	87.270	1.398	30
10.8	0,32	139.655	2.051	33
10.9	0,27	122.873	2.067	27
10.10	0,11	43.680	840	23

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
11.1	2,07	816.550	4.917	56
11.2	0,82	286.900	1.785	30
11.3	2,06	780.213	6.474	54
11.4	2,13	829.071	7.146	51
11.5	0,93	352.962	2.258	32
11.6	0,45	157.338	2.343	38
11.7	4,53	1.800.765	14.787	55
11.8	1,22	438.914	3.810	46
11.9	1,57	590.363	5.040	65
11.10	1,98	775.508	3.376	39
12.1	14,17	4.600.973	9.286	62
12.2	4,23	1.256.961	2.941	26
12.3	9,30	2.852.177	15.530	42
12.4	8,63	2.724.358	8.238	67
12.5	3,68	1.132.309	4.035	49
12.6	11,00	3.849.971	6.467	45
12.7	4,30	1.303.599	3.648	28
12.8	4,82	1.540.343	5.476	47
12.9	4,11	1.275.477	3.895	43
12.10	6,91	2.169.307	5.832	46
13.1	40,92	11.300.221	9.325	40
13.2	117,94	34.247.686	30.467	94
13.3	49,55	13.026.213	21.094	62
13.4	36,20	10.667.527	20.869	111
13.5	91,41	25.713.824	19.482	62
13.6	37,12	10.248.100	9.232	52
13.7	53,17	15.059.241	12.242	65
13.8	29,41	7.686.951	5.518	44
13.9	14,06	3.600.876	4.086	27
13.10	32,89	8.779.941	9.172	53
14.1	198,93	44.075.140	9.116	57
14.2	148,56	32.681.981	17.741	53

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
14.3	421,73	93.398.166	34.424	110
14.4	218,22	47.857.384	35.927	49
14.5	130,06	27.905.903	8.997	41
14.6	319,17	73.255.697	26.885	68
14.7	409,01	96.365.583	34.631	89
14.8	266,24	59.186.475	32.941	75
14.9	270,48	61.118.576	44.308	80
14.10	236,33	53.824.624	21.862	87
15.1	2.588,77	509.786.222	56.617	101
15.2	1.078,02	205.599.169	22.839	67
15.3	2.643,32	515.542.152	93.780	94
15.4	1.985,62	387.371.889	78.292	100
15.5	798,90	146.506.300	26.956	61
15.6	974,68	186.820.480	36.825	69
15.7	707,76	131.454.330	14.932	51
15.8	836,93	155.335.439	16.346	57
15.9	649,19	120.574.162	27.577	56
15.10	1.366,85	264.875.683	59.755	97

Tabela 9: Resultados do *branch-and-bound* para instâncias OP3

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
6.1	0,00	841	103	11
6.2	0,00	709	72	7
6.3	0,00	762	88	10
6.4	0,00	973	201	15
6.5	0,00	890	156	12
6.6	0,00	996	161	7
6.7	0,00	772	91	9
6.8	0,00	1.003	226	13
6.9	0,00	957	196	9
6.10	0,00	995	205	9
7.1	0,01	3.820	178	11
7.2	0,01	4.781	433	20

Instância	Tempo(s)	Nós visitados	AGM geradas	Soluções
7.3	0,01	5.074	690	25
7.4	0,01	4.490	314	15
7.5	0,01	4.073	525	21
7.6	0,01	4.363	337	17
7.7	0,01	3.582	282	9
7.8	0,01	4.180	431	19
7.9	0,01	4.887	790	17
7.10	0,01	4.197	321	8
8.1	0,03	24.916	1.109	13
8.2	0,03	23.704	588	11
8.3	0,03	22.915	1.153	17
8.4	0,03	20.119	789	21
8.5	0,04	24.465	1.595	28
8.6	0,03	23.124	1.710	18
8.7	0,04	26.421	2.036	24
8.8	0,04	24.748	1.961	21
8.9	0,03	20.541	686	17
8.10	0,03	22.462	706	19
9.1	0,25	151.932	3.623	23
9.2	0,21	128.788	2.065	19
9.3	0,19	112.864	2.388	18
9.4	0,14	84.453	1.797	17
9.5	0,19	117.742	2.146	17
9.6	0,19	113.778	1.822	17
9.7	0,19	113.623	2.584	20
9.8	0,21	126.155	1.631	13
9.9	0,20	119.791	2.270	24
9.10	0,17	102.679	974	15
10.1	1,12	599.457	4.493	21
10.2	1,98	1.037.819	21.255	27
10.3	1,02	547.802	3.305	19
10.4	1,35	730.361	5.160	20

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
10.5	0,98	526.438	2.319	22
10.6	0,87	460.382	4.007	19
10.7	1,11	583.988	3.890	21
10.8	0,78	420.645	1.972	19
10.9	1,18	625.988	3.672	27
10.10	1,14	609.647	13.279	28
11.1	5,73	2.773.772	7.823	27
11.2	6,52	3.104.989	7.949	29
11.3	5,91	2.742.770	7.438	27
11.4	7,18	3.462.453	15.233	36
11.5	5,77	2.804.988	7.139	36
11.6	7,06	3.434.145	4.407	29
11.7	6,82	3.410.945	2.006	17
11.8	4,95	2.425.254	3.836	18
11.9	5,51	2.636.690	3.986	29
11.10	6,34	3.057.595	7.968	29
12.1	40,39	17.379.642	21.133	32
12.2	30,96	13.197.246	8.868	29
12.3	42,24	17.615.254	74.571	28
12.4	36,45	15.761.727	11.496	31
12.5	35,07	14.629.082	15.588	25
12.6	36,94	16.398.785	5.402	21
12.7	35,28	14.934.095	10.013	18
12.8	33,15	14.013.588	90.585	26
12.9	26,03	11.283.398	14.708	23
12.10	28,75	12.388.930	11.809	28
13.1	242,25	93.432.434	22.370	32
13.2	213,36	81.420.599	27.111	33
13.3	184,41	69.193.910	19.437	42
13.4	228,79	87.527.607	36.322	40
13.5	195,50	73.356.700	19.188	35
13.6	273,00	104.561.270	29.212	29

<b>Instância</b>	<b>Tempo(s)</b>	<b>Nós visitados</b>	<b>AGM geradas</b>	<b>Soluções</b>
13.7	203,87	79.484.746	43.900	49
13.8	230,13	86.339.499	73.533	52
13.9	211,70	83.747.614	23.565	27
13.10	180,38	69.231.015	18.081	36
14.1	1.162,88	398.861.689	25.504	34
14.2	1.330,39	459.400.975	26.388	53
14.3	1.249,65	425.312.517	23.719	34
14.4	1.641,13	559.411.064	27.268	29
14.5	987,44	334.582.485	34.452	28
14.6	1.282,56	416.840.455	400.731	37
14.7	1.341,46	449.557.227	41.083	45
14.8	1.328,51	443.710.548	106.504	46
14.9	930,36	324.858.733	11.653	40
14.10	1.474,29	494.487.708	40.029	37

O algoritmo *backtracking* foi executado até a instância 15.6 de CP, 13.10 de OP1, 13.10 de OP2 e 12.10 de OP3. O algoritmo *branch-and-bound*, por sua vez, executou até a instância 20.5 de CP, 15.10 de OP1, 15.10 de OP2 e 14.10 de OP3, em tempos computacionais significativamente inferiores. Por exemplo, para as instâncias 15.6 de CP, 13.10 de OP1, 13.10 de OP2 e 12.10 de OP3, o algoritmo *backtracking* necessita de um tempo computacional, respectivamente, de 3392s, 629s, 1491s e 1878s. O algoritmo *branch-and-bound*, por sua vez, utiliza apenas 42s, 26s, 33s e 29s, respectivamente, para executar as mesmas instâncias.

As tabelas 10, 11, 12 e 13 apresentam os tempos médios de execução (em segundos), por classe, dos algoritmos analisados para cada um dos grupos de instâncias.

Tabela 10: Tempos médios de execução (em segundos) dos algoritmos *backtracking* e *branch-and-bound* para as classes de instâncias CP

<b>Classe</b>	<b>Backtracking</b>	<b>Branch-and-bound</b>
10	1,05	0,15
15	1.210,38	768,48
20	—	1.547,02

Tabela 11: Tempos médios de execução (em segundos) dos algoritmos *backtracking* e *branch-and-bound* para as classes de instâncias OP1

Classe	Backtracking	Branch-and-bound
6	0,00	0,00
7	0,00	0,00
8	0,03	0,02
9	0,22	0,07
10	2,37	0,42
11	13,59	1,69
12	154,40	9,97
13	1.208,21	45,01
14	–	250,02
15	–	1.096,00

Tabela 12: Tempos médios de execução (em segundos) dos algoritmos *backtracking* e *branch-and-bound* para as classes de instâncias OP2

Classe	Backtracking	Branch-and-bound
6	0,00	0,00
7	0,00	0,00
8	0,03	0,01
9	0,17	0,05
10	1,95	0,30
11	18,76	1,78
12	156,03	7,12
13	1.867,51	50,27
14	–	261,87
15	–	1.363,00

Tabela 13: Tempos médios de execução (em segundos) dos algoritmos *backtracking* e *branch-and-bound* para as classes de instâncias OP3

Classe	Backtracking	Branch-and-bound
6	0,00	0,00
7	0,01	0,01



Classe	Backtracking	Branch-and-bound
8	0,07	0,03
9	0,90	0,19
10	10,02	1,15
11	138,37	6,18
12	1.581,35	34,53
13	–	216,34
14	–	1.272,86

Os resultados exibidos resumem as tabelas anteriores e explicitam o melhor desempenho da abordagem *branch-and-bound* frente ao algoritmo *backtracking*. É importante notar que os tempos computacionais médios das instâncias OP3 são superiores aos de OP2 e OP1, o que sugere que OP3 é um conjunto de instâncias ligeiramente mais difícil de ser solucionado pelos algoritmos exatos propostos.

Observa-se, também, que as quantidades de nós visitados no grafo implícito e de AGMs completamente construídas durante a busca são drasticamente reduzidas na abordagem *branch-and-bound* quando comparada à abordagem *backtracking*. Nesse sentido, o tempo gasto no cálculo de limites e verificação de condição de poda vale a pena, pois implica uma redução significativa no espaço de busca que será explorado, permitindo encontrar a fronteira de Pareto de modo mais eficiente. Com isso, confirma-se que a abordagem *branch-and-bound* é mais eficiente e, pois, apropriada, para solucionar de forma exata o problema da AGQA-bi. Ainda assim, tendo em vista que o problema da AGQA-bi é NP-difícil, as abordagens exatas demandam um tempo computacional inviável para encontrar a fronteira de Pareto ótima mesmo para instâncias pequenas, com número de vértices menor que 100. Eis o motivo pelo qual se faz necessário investigar as abordagens metaheurísticas.

## 5 Algoritmos Heurísticos

Este capítulo apresenta os cinco algoritmos heurísticos desenvolvidos para esta tese. Na Seção 5.1, descreve-se a busca local *Pareto Local Search*. A Seção 5.2 contém detalhes sobre o NSGA-II. Uma abordagem utilizando Algoritmos Transgenéticos é apresentada na Seção 5.3. A hibridização do NSGA-II com o Algoritmo Transgenético proposto, denominada NSTA, é tema da Seção 5.4. A Seção 5.5 é dedicada ao estudo do algoritmo de Busca Tabu implementado.

### 5.1 *Pareto Local Search*

A busca local *Pareto Local Search* (PLS) foi apresentada por Paquete e Stützle (2006) e consiste em um *framework* geral para projeto de algoritmos de busca local em cenários com múltiplos objetivos. Esta abordagem caracteriza-se por ser uma extensão natural, para o caso multiobjetivo, dos algoritmos de busca local tradicionais para o caso mono-objetivo.

O primeiro passo do algoritmo PLS discutido nesta seção é gerar uma solução inicial  $sol\pi$ . A princípio, foi adotada uma heurística baseada em H1 para gerar tal solução. Nos outros algoritmos desenvolvidos para esta tese, tal procedimento para iniciar soluções é auxiliado por um pré-processamento, conforme será discutido adiante. Entretanto, para manter a maior diversidade possível entre as soluções iniciais de cada execução da busca local e tendo em vista que tal iniciação ocorre apenas uma vez no algoritmo, optamos por não utilizar o arquivo de pré-processamento.

O procedimento de construção de uma solução inicial  $sol\pi$  se inicia com a determinação de uma estimativa da contribuição de cada aresta para o custo total da árvore. A estimativa para o custo linear é dada pelo próprio custo linear. Para o custo quadrático, tal estimativa é calculada como em H1 (apresentada no Capítulo 2), exceto pelo fato de que o custo linear não é considerado. Um vetor de escalarização contendo dois elementos é então sorteado de forma que a soma de seus valores seja 1 e ambos os valores sejam

maiores que 0. O produto interno do vetor de escalarização com o vetor de estimativas é considerado o custo estimado total de cada aresta. O vetor de arestas é então ordenado com base no custo total estimado. A partir desse momento, o algoritmo prossegue escolhendo, a cada iteração, uma aresta para entrar na árvore. A aresta a ser inserida é escolhida aleatoriamente dentre as que compõem uma lista de arestas candidatas. A lista de arestas candidatas é formada pelas  $\#tamCand$  melhores arestas que podem ser inseridas na árvore sem gerar ciclo. O valor de  $\#tamCand$  é definido aleatoriamente no intervalo  $[0, 1 * n; 0, 3 * n]$ . Para instâncias muito pequenas, o algoritmo assegura que  $\#tamCand$  vale pelo menos 2. Após a inserção de uma aresta na árvore, a lista de candidatas é atualizada, excluindo as arestas que geram ciclo e adicionando novas arestas, de acordo com a ordenação pelo custo total estimado, até completar as  $\#tamCand$  arestas. Quando a árvore está completa, o algoritmo para.

Testou-se o algoritmo PLS em que a iniciação da solução ocorre por meio de heurística baseada em H1 contra um procedimento totalmente aleatório. Como resultado, o procedimento aleatório conseguia gerar soluções iniciais mais diversificadas, conduzindo o algoritmo PLS a melhores resultados. Portanto, a abordagem inspirada em H1 foi abandonada e, neste trabalho, descreve-se a busca PLS com a solução inicial gerada de forma aleatória. De acordo com o procedimento implementado, a princípio, uma aresta aleatória  $e$  é selecionada para compor a solução inicial. Em seguida, uma lista de arestas candidatas para entrar na árvore é criada. Esta lista é composta por todas as arestas adjacentes àquelas que já estão na árvore e cuja inserção não gera ciclo. Sendo assim, neste primeiro momento, a lista de candidatos conterá as arestas adjacentes a  $e$ . Então, inicia-se um laço que só termina quando a árvore estiver completa. A cada passo do laço, escolhe-se aleatoriamente uma aresta  $e'$  da lista de candidatos para entrar na árvore e atualiza-se a lista de candidatos. Arestas são removidas da lista de sua inserção gerar ciclo com as arestas que estão na árvore. Arestas são inseridas na lista se sua inserção não gera ciclo e se são adjacentes a  $e'$ .

Após a construção de  $sol\pi$ , esta solução é marcada como não-visitada e adicionada à lista atual de soluções não-dominadas,  $listaSND$ . Então, um procedimento iterativo é iniciado. A cada passo, uma solução não-visitada,  $sol$ , pertencente à  $listaSND$  é escolhida para ser visitada. Todos os vizinhos de  $sol$  são gerados de acordo com a vizinhança adotada. Implementamos duas versões da busca local PLS. Na primeira, adota-se a vizinhança  $N(t)^1$ , conforme proposto por Öncan e Punnen (2010) para a AGMQ (para detalhes, veja seção específica sobre este trabalho no Capítulo 2). Na segunda, uma estratégia de busca *ejectionchain* é empregada para gerar as soluções vizinhas. Denominemos esses algoritmos,

respectivamente, por PLS1 e PLS2.

Conforme já mencionado, no algoritmo PLS, toda a vizinhança de  $sol$  é examinada. Nesse sentido, no algoritmo PLS1, em vez de escolher aleatoriamente arestas para deixarem a solução, realizamos uma busca sistemática, substituindo cada aresta na solução  $sol$  por cada aresta fora dela que não induz um ciclo. Se um vizinho  $\alpha$  de  $sol$  é não-dominado com relação a todas as soluções em  $listaSND$ ,  $\alpha$  é marcado como não-visitado e adicionado à  $listaSND$ . Soluções de  $listaSND$  que sejam dominadas por  $\alpha$  são removidas do conjunto.

No caso de PLS2, a busca *ejection chain* implementada é composta por dois laços principais. O laço mais externo é executado  $\#maxTent$  vezes. A cada iteração desse laço, a solução é reiniciada com  $sol$ , a solução que está sendo visitada. Para cada iteração desse laço mais externo, associa-se um vetor de escalarização escolhido aleatoriamente e proíbe-se que a primeira aresta removida em cada uma das iterações anteriores seja novamente escolhida para deixar a solução. Como o algoritmo é determinístico, tal proibição impede que se gere a mesma sequência de trocas.

O laço mais interno escolhe, a cada passo, uma aresta para ser removida e outra para ser inserida na solução. A aresta a ser removida é a que mais contribui para o custo total da árvore, de acordo com o vetor de escalarização definido. A aresta que vai entrar, por sua vez, é a aresta que permite a melhor troca e cuja inserção não gera ciclo, com base no mesmo vetor de escalarização. A solução vizinha  $\alpha$  gerada é, então, submetida ao conjunto de soluções não-dominadas atual,  $listaSND$ . Caso o vizinho seja não-dominado, a troca é efetivamente realizada, a solução vizinha é inserida em  $listaSND$  e é marcada como não-visitada. Além disso, um contador de falhas é zerado e as soluções de  $listaSND$  que sejam dominadas por  $\alpha$  são removidas do conjunto. Caso contrário, o contador de falhas é incrementado em uma unidade. Esse processo de remoção e inserção de arestas continua até que o número total de falhas atinja um valor máximo dado pelo parâmetro  $\#maxFalhas$ .

Enquanto  $\#maxFalhas$  não é atingido, as trocas bem sucedidas vão se acumulando na solução, de forma que uma aresta que foi removida não pode entrar novamente e uma aresta inserida não pode deixar a solução. O conjunto de trocas realizadas corresponde à cadeia de trocas do *ejection chain*.

Uma vez que uma troca é realizada, as contribuições de cada aresta da solução para o custo total da árvore bem como a variação no custo da árvore em função da inserção de outra aresta são modificados. Isso se deve à própria formulação do problema, em que

os intercustos dependem das arestas que constituem a solução. Neste sentido, esse procedimento pode ser classificado como *ejection chain* pois uma cadeia de trocas (envolvendo ejeção de arestas) é realizada e as trocas futuras são influenciadas pelas trocas anteriores.

---

**Algoritmo 3** Algoritmo *Pareto Local Search*

---

```

 $\pi \leftarrow$  Solução gerada aleatoriamente
visitado( $sol\pi$ )  $\leftarrow$  falso
listaSND  $\leftarrow sol\pi$ 
enquanto  $\exists \mu \in listaSND \mid$  visitado( $\mu$ ) = falso faça
  Escolher uma solução qualquer  $\mu \in listaSND \mid$  visitado( $\mu$ ) = falso
  para todo  $\alpha \in$  Vizinhos( $\mu$ ) faça
    se não existe solução em listaSND que domine  $\alpha$  então
      visitado( $\alpha$ )  $\leftarrow$  falso
      listaSND  $\leftarrow \{listaSND - \{\beta \in listaSND \mid \alpha \text{ domina } \beta\}\} \cup \alpha$ 
    fim se
  fim para
  visitado( $\mu$ )  $\leftarrow$  verdadeiro
fim enquanto
Retornar listaSND

```

---

PLS1 e PLS2 continuam de forma idêntica após o exame da vizinhança. Nesse momento,  $sol$  é marcado como visitado. O procedimento iterativo continua até que todas as soluções em listaSND sejam visitadas. Ao final da execução, listaSND corresponde ao conjunto de aproximação gerado pelos algoritmos PLS1 e PLS2.

O pseudo-código do algoritmo proposto segue a proposta de Paquete e Stützle (2006) e é apresentado em Algoritmo *Pareto Local Search*. Esse algoritmo representa, de maneira geral, as duas buscas implementadas.

Os experimentos computacionais prévios realizados mostraram que o algoritmo PLS2 apresenta melhores resultados que o PLS1 tanto no que se refere ao tempo de execução quanto em relação à qualidade dos conjuntos de aproximação gerados. Nesse sentido, ao longo no texto, ao nos referirmos ao algoritmo PLS implementado, estaremos abordando o algoritmo PLS2, em que a solução inicial é construída de maneira aleatória e uma vizinhança *ejection chain* é empregada.

## 5.2 NSGA-II

O algoritmo *Nondominated Sorting Genetic Algorithm II*, NSGA-II, foi proposto por Deb et al. (2002) e é bastante aplicado na solução de diversos problemas de otimização multiobjetivo. Neste trabalho, o NSGA-II é adaptado para lidar com o problema da

AGQA-bi.

O pseudo-código do Algoritmo NSGA-II apresenta o algoritmo em questão. O primeiro passo do algoritmo é a criação de uma população inicial de indivíduos. Essa geração utiliza o mesmo procedimento descrito na Seção 5.1 para gerar aleatoriamente uma solução. A diferença é que, neste caso, o procedimento é auxiliado por uma etapa de pré-processamento, denominada *preProc1*.

O pré-processamento calcula a ordenação das arestas do grafo correspondente a cada um dos vetores de escalarização no conjunto  $\{[0; 1], [0,1; 0,9], [0,2; 0,8], [0,3; 0,7], [0,4; 0,6], [0,5; 0,5], [0,6; 0,4], [0,7; 0,3], [0,8; 0,2], [0,9; 0,1], [1; 0]\}$ . Observe que o resultado de *preProc1* é tão somente as ordenações das arestas do grafo em função de cada um dos vetores de escalarização adotados, de forma que *preProc1* não gera nenhuma solução. O algoritmo NSGA-II gera cada solução de acordo com um vetor de escalarização do conjunto pré-determinado e, pois, de acordo com a respectiva ordenação gerada no pré-processamento. O procedimento de escolha da aresta que será inserida na solução a cada iteração é determinado da mesma maneira que no procedimento de iniciação descrito na Seção 5.1.

Caso a solução gerada já exista na população, um método aleatório para geração de soluções é executado até que uma solução diferente seja encontrada, conforme descrito a seguir. A princípio, uma aresta aleatória  $e$  é selecionada para compor a solução inicial. Em seguida, uma lista de arestas candidatas para entrar na árvore é criada. Esta lista é composta por todas as arestas adjacentes àquelas que já estão na árvore e cuja inserção não gera ciclo. Sendo assim, neste primeiro momento, a lista de candidatos conterá as arestas adjacentes a  $e$ . Então, inicia-se um laço que só termina quando a árvore estiver completa. A cada passo do laço, escolhe-se aleatoriamente uma aresta  $e'$  da lista de candidatos para entrar na árvore e atualiza-se a lista de candidatos. Arestas são removidas da lista se sua inserção gerar ciclo com as arestas que estão na árvore. Arestas são inseridas na lista se sua inserção não gera ciclo e se são adjacentes a  $e'$ .

Em seguida, a população criada no passo 1 é avaliada, por meio do cálculo das funções-objetivo das árvores que a constituem. Os passos 3 e 4 apresentam a estratégia diferenciada do NSGA-II. A população é ordenada com base em duas características: o valor de *rank* definido em função da dominância de Pareto e o valor de *crowding distance*, que é uma medida de proximidade entre soluções.

No passo 3, a população é dividida em fronteiras. O conjunto de soluções não-dominadas da população como um todo tem *rank* 1, pois está na primeira fronteira. Ao desconsiderar

os indivíduos com *rank* 1, os indivíduos não-dominados da sub-população constituem a segunda fronteira e possuem valor de *rank* igual a 2. O algoritmo segue classificando os indivíduos até que todos tenham seus valores de *rank* definidos.

---

**Algoritmo 4** Algoritmo NSGA-II
 

---

```

Gere a população inicial
Calcule objetivos
Associe a cada indivíduo um rank baseado na dominância de Pareto
Associe um valor de crowding distance a cada indivíduo
para  $i \leftarrow 1$  até um máximo de iterações faça
  para  $j \leftarrow 1$  até o número de indivíduos na população faça
    Selecione dois pais por torneio binário
    Recombine pais para gerar filho[ $j$ ]
    Realize mutação no filho[ $j$ ]
  fim para
  Considere a população de pais e filhos como um todo
  Associe a cada indivíduo um rank baseado na dominância de Pareto
  Associe um valor de crowding distance a cada indivíduo
  Selecione a próxima população com base no rank e no crowding distance
fim para
Retorne os indivíduos de rank 1

```

---

Apesar de a explicação dada para o passo 3 ser a mais didática, Deb et al. (2002) propõem uma maneira mais eficiente de definir os *ranks* com base no contador de dominância de uma solução *sol* (número de soluções que dominam *sol*) e no conjunto de soluções da população dominadas por *sol*. A princípio, os valores dessas variáveis são definidos para todos os indivíduos da população. As soluções que pertencerem à primeira fronteira têm contador de dominância 0. Essas soluções são colocadas em uma fila e o seu *rank* é definido em 1. A partir de então, inicia-se um processo iterativo que retira, uma por uma, as soluções da fila. Denominemos a solução retirada da fila por *sol'*. Todas as soluções *sol'* que estiverem no conjunto de soluções dominadas por *sol* têm seu contador de dominância reduzido em uma unidade. Se o contador de dominância de *sol'* se tornar 0, isso significa que o *rank* de *sol'* é o *rank* de *sol* mais 1 e que *sol'* deve ser adicionado à fila. Quando a fila se torna vazia, todos os indivíduos da população estão com seus *ranks* definidos.

Dada uma ordenação da população em função do *rank*, indivíduos que possuem um mesmo *rank* são ordenados pela *crowding distance*. O valor da *crowding distance* oferece uma estimativa da densidade de soluções ao redor de uma outra solução e corresponde à soma, para todos os objetivos, da distância média entre a solução antecessora e sucessora no que diz respeito a um dado objetivo. O passo 4 diz respeito ao cálculo do valor de *crowding distance* para todos os indivíduos da população. Tal valor é utilizado em um





em uma adaptação, para o caso multiobjetivo, do operador proposto por Raidl (2000). A Figura 5 mostra o procedimento utilizado por Raidl (2000) no operador de cruzamento. Tal operador é implementado com base no princípio de se manter o maior número possível de arestas por herança. De forma breve, o operador atua em três etapas. Na primeira etapa, o descendente recebe todas as arestas em comum entre os pais. Na segunda, tenta-se inserir todas as arestas dos pais que não foram inseridas na primeira etapa. A escolha da próxima aresta a tentar entrar na árvore ocorre por seleção binária. Para realizar a seleção binária faz-se necessário obter uma forma de comparar as arestas. Sendo assim, sorteamos um vetor de escalarização. O valor usado para a comparação é, pois, uma combinação linear entre o custo linear e o custo quadrático parcial. Denominamos custo quadrático parcial a soma dos intercustos da aresta que não está no descendente com as arestas que estão na árvore do descendente. Ou seja, essa definição é similar à contribuição parcial de aresta livre. Ao final da segunda etapa, é possível que o descendente ainda não constitua uma árvore. Neste caso, arestas aleatórias do grafo de entrada são escolhidas para completar o filho.

Após a recombinação, o descendente é submetido à mutação. O operador de mutação utilizado é bem simples. Consiste em substituir uma aresta aleatória do descendente por uma aresta escolhida aleatoriamente dentre as arestas do grafo que não formam ciclo. Dessa forma, o indivíduo resultante da mutação é um vizinho do descendente obtido na recombinação, de acordo com a estrutura de vizinhança  $N(t)^1$ , usada na busca local PLS1.

Testamos também uma implementação, NSGAII2, em que a mutação ocorre de forma que arestas com maior contribuição para o custo total da árvore tivessem maior probabilidade de serem removidas. A aresta a ser inserida também seria determinada de forma aleatória com viés. Neste caso, um dos vetores do pré-processamento seria escolhido e as arestas que ocupam melhor posição no vetor de ordenação correspondente teriam maior prioridade. Entretanto, os resultados experimentais não mostraram diferença significativa quanto ao tempo e, quanto à qualidade de solução, a proposta totalmente aleatória obtém resultados ligeiramente superiores. Nesse sentido, consideraremos, a partir desse instante, que o operador de mutação implementado não é o descrito em NSGAII2, com viés, mas sim o que foi apresentado a princípio.

Em seguida, a população “original” é unida à população de filhos e selecionam-se os indivíduos que permanecerão na população para a próxima iteração, o que ocorre nos passos de 11 a 14. Tal seleção se fundamenta na ordenação dos indivíduos com base no *rank* e na *crowding distance*, conforme Figura 6. Considerando que a população tem um

tamanho fixo e que, no momento em que os filhos são inseridos nessa população a mesma fica com o dobro da sua capacidade, metade dos indivíduos permanecerão na população para a próxima iteração e a outra metade será descartada.

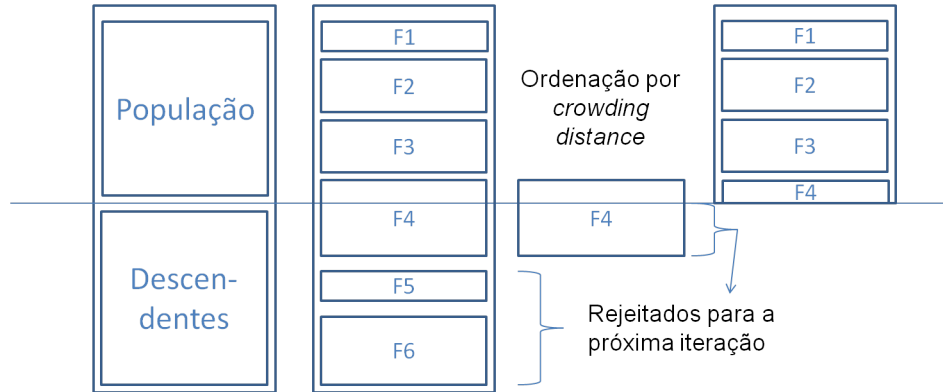


Figura 6: Seleção de indivíduos para a próxima iteração do algoritmo NSGA-II.

O retorno do algoritmo NSGA-II é o conjunto de soluções que pertencem à primeira fronteira, pois constituem o conjunto de soluções não-dominadas encontradas pelo algoritmo.

### 5.3 NSTA

O algoritmo NSTA consiste em uma hibridização do NSGA-II de Deb et al. (2002) com os operadores dos Algoritmos Transgenéticos. Almeida et al. (2012) introduziram essa hibridização para o Problema do Caixeiro Comprador Biobjetivo. O pseudo-código do Algoritmo NSTA apresenta o algoritmo desenvolvido nesta tese para o problema da AGQA-bi.

No NSTA implementado, a população inicial é gerada tal como no NSGA-II, descrito na Seção 5.2, o que significa que o algoritmo NSTA também utiliza o pré-processamento preProc1. A ordenação em função do *rank* e da *crowding distance* é mantida. No passo 5, inicializa-se um repositório do hospedeiro. O repositório do hospedeiro contém informações que serão utilizadas durante a busca e pode ser constituído por informações *a priori* e *a posteriori*.

A informação *a priori* inserida no banco de informações é um conjunto de, no máximo, 22 soluções, 11 geradas a partir da heurística H1 (ASSAD; XU, 1992) e 11 geradas a partir da heurística H2 (ASSAD; XU, 1992). Os 11 vetores de escalarização  $\{[0,0; 1,0]; [0,1; 0,9]; [0,2; 0,8]; \dots; [1,0; 0,0]\}$  são utilizados para gerar as soluções por meio da combinação linear entre os objetivos. Para tanto, um segundo pré-processamento, preProc2 é realizado.

O segundo pré-processamento inicia-se com a leitura do resultado de *preProc1*, gerando as 11 soluções de acordo com cada uma das ordenações obtidas. Em seguida, a heurística H2 é executada para gerar mais 11 soluções, considerando como custo das arestas a combinação linear entre a estimativa de custos linear e quadrática. No caso, a estimativa quadrática é calculada e atualizada a cada iteração, tal como na heurística H2. Ao final desse processo, apenas as soluções diferentes (não necessariamente não-dominadas) são inseridas no repositório do hospedeiro. Outras informações *a priori* armazenadas no repositório do hospedeiro são os vetores empregados nos pré-processamentos e as ordenações de arestas provenientes de *preProc1*.

Ainda na fase de inicialização, a probabilidade de se manipular um indivíduo usando plasmídeo ou transposon é fixada. No caso, a probabilidade inicial para plasmídeos foi fixada em  $\#probPlasm = 0,8$  e, pois, para transposons, em  $\#probTransp = 0,2$ . A cada iteração, se for o caso, essas probabilidades serão atualizadas. No caso, a atualização ocorre a cada  $\#intervAlterProb$  iterações e consiste em diminuir a probabilidade dos plasmídeos em  $\#passoAgentes$ , o que implica o aumento da probabilidade do transposon no mesmo valor.

A preferência pela atuação dos plasmídeos no início do algoritmo se deve ao fato de o banco de informações do hospedeiro ser iniciado com soluções de qualidade superior à qualidade dos indivíduos da população. Nesse sentido, a injeção de bons trechos de informação do repositório nas soluções da população levará a uma melhoria da qualidade dos indivíduos da população e pois do repositório, que será atualizado. A atuação do transposon, por sua vez, tem pouco a contribuir no início do algoritmo, tendo em vista que a sua manipulação emprega as ordenações de arestas que foram utilizadas na geração da população inicial. Entretanto, a atuação dos transposons se torna imprescindível nas etapas finais do algoritmo. Isso porque as informações das soluções do repositório já foram deveras utilizadas. Neste ponto, a recuperação de boas informações que podem ter se perdido durante o processo de busca é essencial para dar continuidade ao processo de melhoria com um pouco de diversificação.

A cada iteração, todos os indivíduos da população serão manipulados ou por um plasmídeo ou por um transposon. Neste caso, não há recombinação nem mutação.

O plasmídeo implementado foi inicialmente proposto em Rocha, Goldberg e Goldberg (2007). A cada iteração do algoritmo, um novo conjunto de plasmídeos é criado. Para obter a cadeia de informações desse agente, escolhe-se aleatoriamente um cromossomo no repositório do hospedeiro. Então, um determinado número de arestas é escolhido aleato-

riamente O tamanho da cadeia de informações é escolhido com probabilidade uniforme no intervalo entre 0,15% e 0,35% do número de arestas na árvore. Uma vez que a cadeia de informação do plasmídeo é gerada apenas com base em informações já disponíveis, os plasmídeos adotados não são recombinados.

---

**Algoritmo 5** Algoritmo NSTA
 

---

```

Gere a população inicial pop
Calcule objetivos
Associe a cada indivíduo um rank baseado na dominância de Pareto
Associe um valor de crowding distance a cada indivíduo
Inicie o repositório do hospedeiro
 $probPlasmideo \leftarrow$  probabilidade inicial para plasmídeos
 $probTransposon \leftarrow 1.0 - probPlasmideo$ 
para  $i \leftarrow 1$  até um máximo de iterações faça
  Atualize probabilidades dos agentes se preciso e gere conjunto de plasmídeos
  para  $j \leftarrow 1$  até o número de indivíduos na população faça
    se  $random() < probPlasmideo$  então
       $filho[j] \leftarrow$  ManipulaçãoPlasmídeo( $pop[j]$ )
    senão
       $filho[j] \leftarrow$  ManipulaçãoTransposon( $pop[j]$ )
    fim se
  Atualize o repositório do hospedeiro com  $filho[j]$ 
fim para
Considere a população de pais e filhos como um todo
Associe a cada indivíduo um rank baseado na dominância de Pareto
Associe um valor de crowding distance a cada indivíduo
Selecione a próxima população com base no rank e no crowding distance
fim para
Retorne os indivíduos com rank 1
  
```

---

A cada iteração em que os plasmídeos são os agentes transgenéticos atuantes, seleciona-se aleatoriamente um dos plasmídeos do conjunto para manipular o indivíduo. A manipulação realizada pelos plasmídeos injeta no descendente todas as arestas armazenadas na cadeia de informação do agente transgenético. Em seguida, tenta-se inserir no descendente as arestas do indivíduo selecionado para manipulação, até que o descendente seja uma AGM. Se, ao final desse processo, o descendente ainda não constituir uma árvore, arestas aleatórias do grafo são selecionadas para completar o indivíduo.

O transposon, por sua vez, seleciona um trecho aleatório do indivíduo da população para atacar. Este trecho tem um tamanho escolhido aleatoriamente entre 0,15% e 0,35% do número de arestas na árvore. Nesse sentido, o descendente conterà todas as arestas do indivíduo manipulado, exceto pelas arestas que estão no trecho selecionado. Tais arestas são substituídas uma a uma por arestas que estão fora da árvore. A aresta a ser removida

é escolhida de forma aleatória, segundo uma distribuição não-uniforme, de maneira que as arestas que apresentam maior custo escalarizado tenham maior probabilidade de sair. O custo escalarizado é obtido por meio do produto interno do vetor de custos de cada aresta com um vetor de escalarização escolhido aleatoriamente dentre os vetores armazenados durante o primeiro pré-processamento.

Para cada aresta que será substituída, cria-se uma lista de arestas candidatas com as 5 melhores arestas que podem ser inseridas na árvore de acordo com a ordenação correspondente ao vetor de escalarização escolhido na remoção. A aresta a ser inserida é escolhida aleatoriamente dentre o conjunto de arestas candidatas. A aresta do transposon original somente é inserida de volta caso não haja outra aresta que possa ser inserida em seu lugar.

A cada manipulação, o repositório do hospedeiro é atualizado. Se o descendente gerado dominar alguma solução do repositório, ele substitui a solução dominada no repositório. A estratégia utilizada é substituir a primeira solução dominada encontrada, de forma a tentar evitar o gasto de tempo computacional para averiguar a quantidade total de soluções dominadas ou mesmo para decidir que solução remover.

Os mesmos procedimentos para atualização da população para a próxima geração realizados no NSGA-II são adotados pelo NSTA. Após um determinado número de iterações, o algoritmo termina. O retorno, assim como no NSGA-II, é o conjunto de soluções que situam-se na primeira fronteira.

## 5.4 Algoritmo Transgenético

Com o intuito de poder comparar adequadamente o potencial de cada estratégia para a solução do problema da AGQA-bi, o algoritmo Transgenético (AT) implementado utiliza os mesmos procedimentos definidos para o algoritmo NSTA.

Sendo assim, a população inicial é gerada com base no procedimento H1 aleatorizado e auxiliado por preProc1, tal como em NSTA. O repositório do hospedeiro também é inicializado e atualizado do mesmo modo que em NSTA.

O comportamento dos agentes também foi mantido, de forma que se possa comparar efetivamente o potencial das estratégias NSGA-II, NSTA e Algoritmos Transgenéticos. Uma grande diferença é que, no caso do arcabouço do NSGA-II e do NSTA, um descendente sempre tem que ser gerado, independentemente de implicar melhoria ou

não das soluções ou do conjunto de soluções não-dominadas. A abordagem transgenética desenvolvida, por outro lado, só aceita a manipulação dos agentes transgenéticos caso acarretem melhoria para o endossimbionte manipulado ou para o conjunto de soluções não-dominadas como um todo.

No caso dos algoritmos NSGA-II e NSTA, a própria população armazena o conjunto de soluções não-dominadas que será retornado. Este conjunto corresponde às soluções na primeira fronteira. No algoritmo Transgenético, emprega-se um arquivo externo para armazenar tais soluções. Como o número de soluções retornadas pelo NSGA-II é limitado ao dobro do tamanho da população, também limitou-se o tamanho do arquivo externo do AT neste valor. A estratégia usada para implementar o arquivo externo foi o *Adaptive Grid Archiving* – AGA, proposto por Knowles (2005). Segundo esta abordagem, as soluções não-dominadas são organizadas em um *grid*. Dessa forma, enquanto o arquivo não estiver lotado, todas as soluções não-dominadas são aceitas e o arquivo é atualizado para remover todas as soluções que sejam dominadas pela que está sendo inserida no arquivo. Caso o arquivo esteja lotado, a nova solução somente poderá entrar se a sua inserção provocar uma variação no valor máximo ou mínimo para um dos objetivos no arquivo ou se ocupar uma célula do *grid* que apresente menos elementos que alguma outra célula do *grid*. Neste último caso, uma solução qualquer da célula mais populosa do *grid* será removida aleatoriamente. A seguir, apresentamos o pseudo-código do Algoritmo Transgenético.

## 5.5 Busca Tabu

A literatura ainda é escassa em termos de algoritmos de Busca Tabu para problemas de Otimização Multiobjetivo. Em um dos poucos trabalhos na área Hansen (1997) propõem o MOTS (Busca Tabu para Otimização Multiobjetivo, do inglês, *Tabu Search for Multiobjective Optimization*). Assim como o NSGA-II, o MOTS não foi pensado para um problema específico, mas para servir como um *framework* para implementação de algoritmos de Busca Tabu multiobjetivos. Propomos, pois, o emprego do arcabouço do MOTS, com adaptações, para a solução do problema da AGQA-bi.

O MOTS otimiza de modo simultâneo um conjunto de soluções iniciais em direção ao conjunto de soluções eficientes. As soluções iniciais são espalhadas de maneira uniforme e, a cada iteração, escolhe-se uma direção para modificar a solução que está sendo manipulada na tentativa de que a solução se aproxime da fronteira de Pareto e mantenha distância das outras soluções no conjunto (HANSEN, 1997).

---

**Algoritmo 6** Algoritmo Transgenético
 

---

```

Gere a população inicial pop
Inicie o repositório do hospedeiro
 $probPlasmideo \leftarrow$  probabilidade inicial para plasmídeos
 $probTransposon \leftarrow 1.0 - probPlasmideo$ 
para  $i \leftarrow 1$  até um máximo de iterações faça
  Atualize as probabilidades dos agentes se preciso e gere conjunto de plasmídeos
  para  $j \leftarrow 1$  até o número de indivíduos na população faça
    se  $random() < probPlasmideo$  então
       $resp \leftarrow$  ManipulaçãoPlasmídeo( $pop[j]$ )
      se  $resp$  domina  $pop[j]$  ou é inserida no conjunto de não-dominadas então
         $pop[j] \leftarrow resp$ 
      fim se
    senão
       $resp \leftarrow$  ManipulaçãoTransposon( $pop[j]$ )
      se  $resp$  domina  $pop[j]$  ou é inserida no conjunto de não-dominadas então
         $pop[j] \leftarrow resp$ 
      fim se
    Atualize o repositório do hospedeiro
  fim se
fim para
fim para
Retorne o conjunto de soluções não-dominadas

```

---

No caso do trabalho de Hansen (1997), a direção a ser explorada por cada solução é definida com base em uma estratégia determinística que leva em conta as relações de dominância entre cada par de soluções do conjunto. A princípio, também empregamos esta mesma estratégia. Entretanto, os resultados preliminares mostraram que a escolha aleatória de um vetor de escalarização poderia produzir resultados melhores e em um tempo computacional relativamente inferior. Nesse sentido, optamos pelo vetor escolhido de forma aleatória.

O procedimento adotado neste trabalho para iniciar as soluções do conjunto é o mesmo empregado para iniciar as soluções dos algoritmos NSGA-II, NSTA e Transgenético. Após iniciar as soluções e escolher aleatoriamente um vetor de escalarização, o algoritmo segue, para cada iteração, visitando uma solução *sol*. Nesse sentido, o algoritmo busca uma solução *sol'*, vizinha de *sol*, que minimiza o valor escalarizado da função-objetivo do problema e que não é movimento Tabu. A única situação em que uma restrição Tabu não é levada em consideração é no caso em que a solução *sol'* gerada atende ao critério de aspiração. No caso, o critério de aspiração usado é que a solução seja não-dominada em relação ao conjunto atual de soluções não-dominadas. Caso isso ocorra, o movimento será realizado mesmo que seja tabu.

O método usado para gerar a vizinhança é uma adaptação da busca *ejection chain* proposta na Seção 5.1. Tendo em vista que a busca Tabu apresenta seu próprio laço principal, optamos por realizar apenas o laço mais interno da busca *ejection chain* proposta em 5.1. A cada passo, uma aresta é escolhida para ser removida e outra para ser inserida na solução. A aresta a ser removida é a que mais contribui para o custo total da árvore, de acordo com o vetor de escalarização definido. A aresta que vai entrar, por sua vez, é a aresta que permite a melhor troca e cuja inserção não gera ciclo, com base no mesmo vetor de escalarização. A solução vizinha  $\alpha$  gerada é, então, submetida ao conjunto de soluções não-dominadas atual, *listaSND*. Caso o vizinho seja não-dominado, a troca é efetivamente realizada e a solução vizinha é inserida em *listaSND*. Além disso, um contador de falhas é zerado e as soluções de *listaSND* que sejam dominadas por  $\alpha$  são removidas do conjunto. Caso contrário, verifica-se se a troca constitui movimento tabu. Se o movimento não for tabu, mesmo que a solução  $\alpha$  gerada não entre no conjunto de soluções não-dominadas, a troca é efetuada. De qualquer forma, o contador de falhas é incrementado em uma unidade. Esse processo de remoção e inserção de arestas continua até que o número total de falhas atinja um valor máximo dado pelo parâmetro *#maxFalhas*.

Enquanto *#maxFalhas* não é atingido, as trocas bem sucedidas vão se acumulando na solução, de forma que uma aresta que foi removida não pode entrar novamente e uma aresta inserida não pode deixar a solução. O conjunto de trocas realizadas corresponde à cadeia de trocas do *ejection chain*.

Após a geração do vizinho, as informações Tabu são atualizadas. Hansen sugere, no MOTS, que essa informação seja um atributo do movimento de *sol* para *sol'*, de forma que a revisita de soluções seja evitada. No algoritmo Tabu implementado para esta tese de doutorado, seguimos a ideia de Cordone e Passeri (2012), em que há um limite mínimo de iterações para que uma aresta seja inserida ou removida da solução, dado que essa aresta já esteve envolvida em um procedimento de troca. A informação tabu consiste, pois, no movimento de inserção ou remoção de uma aresta e o valor da iteração em que cada uma das arestas está habilitada a participar de uma troca representa o *tabu tenure*.

Hansen (1997) também propõe que, de tempos em tempos, as soluções sejam atualizadas de forma que se gere diversificação. No caso, implementamos esse critério pela substituição de uma solução do conjunto atual por uma solução do conjunto de não-dominadas. O método AGA de arquivamento também foi utilizado neste algoritmo. A seguir, apresentamos o pseudo-código da busca tabu implementada.



---

**Algoritmo 7** Algoritmo Busca Tabu
 

---

```

CriarConjuntoInicial(conj[TAMCONJ])
para  $itr \leftarrow 1$  até NUMITER faça
  para  $i \leftarrow 1$  até TAMCONJ faça
    gerarVizinhosEjectionChain( $pop[i], i, itr$ )
    se critério de diversificação for satisfeito então
       $conj[i] \leftarrow$  Solução aleatória  $\in listaSND$ 
    fim se
  fim para
fim para

```

---

## 6 Experimentos Computacionais – Algoritmos Heurísticos

Este capítulo é dedicado aos experimentos computacionais envolvendo os algoritmos (meta)heurísticos abordados no Capítulo 5, a saber: Busca Local *PLS*, Busca Tabu com *ejection chain*, NSGAI, NSTA e Transgenético. Nesse sentido, os experimentos relatados dizem respeito à comparação entre os algoritmos mencionados. Antes de descrever os experimentos propriamente ditos, apresentam-se o conjunto de instâncias adotado bem como a metodologia empregada para comparação.

### 6.1 Instâncias

Os conjuntos de instâncias adotados para a realização dos experimentos com algoritmos heurísticos são os mesmos 4 conjuntos anteriormente empregados para a realização de comparações entre algoritmos exatos. Neste caso, todos os algoritmos foram executados com sucesso para todas as instâncias propostas. A exceção ocorre para as instâncias geradas de acordo com o método de Cordone e Passeri (2012) contendo 150 ou 175 vértices e 100% de densidade de arestas, ou seja, grafos completos. Para essas instâncias, a quantidade de memória total requisitada é maior que a memória disponível, o que provoca uma exceção na alocação de memória e encerra o processo. Conforme já mencionado, os algoritmos desenvolvidos nessa Tese são passíveis de execução para grandes instâncias, desde que os recursos de *hardware* necessários (em especial, memória RAM) estejam disponíveis.

### 6.2 Metodologia de Comparação

O ambiente de desenvolvimento e execução dos algoritmos heurísticos foi o mesmo dos algoritmos exatos. Assim, os algoritmos foram implementados em C++, usando o compilador g++ e o sistema operacional Scientific Linux 5.5 de 64 bits. Todos os experimentos foram realizados em máquinas HP Z400, com processador Intel Xeon QuadCore W3520 de 2.8 GHz e 8Gb de RAM. Para cada instância, 30 execuções independentes de cada um

desses algoritmos foram realizadas.

A comparação entre os algoritmos heurísticos utilizou o indicador de qualidade hipervolume. O indicador epsilon binário aditivo também foi adotado para realizar as comparações e os resultados foram compatíveis com os obtidos para o hipervolume. Nesse sentido, para evitar redundância de informações, optou-se por apresentar apenas os resultados para o hipervolume. Ambos indicadores estão descritos no anexo B desta tese.

O teste estatístico de Mann-Whitney (U-teste) é utilizado para verificar a significância estatística dos resultados (CONOVER, 2001). O U-teste é um teste não-paramétrico que indica se há evidência estatística de que duas amostras de observações independente advenham da mesma distribuição. Este é um dos testes de significância mais conhecidos. A hipótese nula do Mann-Whitney é que as duas amostras foram obtidas da mesma população. O teste envolve o cálculo de uma estatística cuja distribuição na hipótese nula seja conhecida. A partir de 20 amostras, os resultados do U-teste têm elevada precisão, haja vista o emprego da distribuição normal. Para conjuntos de amostras menores, o resultado do Mann-Whitney é uma aproximação.

Como resultado do teste de Mann-Whitney, obtêm-se  $p$ -valores. O nível de significância adotado é de 5% (0,05). Assim, se o  $p$ -valor da comparação de um algoritmo A com um algoritmo B for menor que 0,05, o resultado é favorável a A. Por outro lado, se o  $p$ -valor for superior a 0,95, o resultado é favorável a B.  $p$ -valores entre 0,05 e 0,95 não permitem decidir se um algoritmo pode ser melhor que outro, situação na qual consideramos os algoritmos equivalentes. Os  $p$ -valores exibidos nos experimentos computacionais são considerados 0 quando são inferiores a  $10^{-7}$ .

Além da qualidade dos conjuntos de aproximação encontrados, a comparação entre algoritmos envolve, também, o tempo computacional despendido.

## 6.3 Configuração dos Parâmetros

Os algoritmos NSGA-II e NSTA exigem que alguns parâmetros sejam configurados. No caso do NSGA-II, faz-se necessário indicar o número de iterações do algoritmo e o tamanho da população. Tais parâmetros foram fixados, respectivamente, em 60 iterações e um máximo de 300 indivíduos, se o tamanho da instância for maior que 40 nós. Se o número de vértices for inferior a 15, 20 e 40 vértices, a população assume tamanhos iguais a 80, 100 e 180 respectivamente.

O NSTA e o algoritmo transgenético também utilizam os dois parâmetros mencionados para o NSGA-II, com os mesmos valores. Além disso, outros parâmetros devem ser estabelecidos e já tiveram seus valores mencionados ao longo do texto. Ainda assim, recordemos os parâmetros adotados, a saber: número máximo de soluções no repositório do hospedeiro (22 soluções); tamanho da cadeia de informações do plasmídeo (entre 0,15% e 0,35% do número de arestas na árvore); trecho do indivíduo que será manipulado pelo transposon (entre 0,15% e 0,35% do número de arestas na árvore); probabilidade inicial de manipular indivíduo com plasmídeo (80%); probabilidade inicial de manipular indivíduo com transposon (20%); intervalo de iterações para atualização de probabilidades dos agentes (5 iterações); variação na probabilidade dos agentes ao realizar atualização (5%).

A busca tabu realiza 30 iterações. A cada 5 iterações as soluções do conjunto são atualizadas com soluções do conjunto de não-dominadas. O número mínimo de iterações para que uma aresta possa participar de uma nova troca é 3 e o número máximo de falhas é 5. O tamanho do conjunto de soluções inicial foi definido em função do número de vértices do grafo. Caso tal número seja inferior a 15, 20 ou 40 nós, o tamanho do conjunto será 15, 25 ou 50, respectivamente. Para um grafo com mais de 40 vértices, o algoritmo considera um conjunto inicial com 100 soluções.

Vale salientar que os parâmetros foram definidos por meio de testes de parâmetros, equilibrando qualidade do conjunto de aproximação gerado e tempo de computação. Foram realizados testes estatísticos nos mesmos moldes dos experimentos computacionais apresentados nesta Tese para determinar a melhor combinação de parâmetros. Tendo em vista que se trata de um grande volume de dados e que a sua apresentação não influencia diretamente as conclusões dessa pesquisa, optou-se por apresentar tais resultados em outro momento.

## 6.4 Resultados

Esta seção trata especificamente da comparação entre as heurísticas apresentadas nesta tese.

A princípio, apresentam-se as tabelas relativas ao tempo de execução dos algoritmos de pré-processamento para os 4 grupos de instâncias considerados. Tais dados podem ser visualizados nas Tabelas 14, 15, 16 e 17. O algoritmo NSGA-II e o de Busca Tabu com *ejection chain* utilizam apenas o pré-processamento 1, ao passo que NSTA e Transgenético empregam os dois pré-processamentos. Nenhum processamento é utilizado pelo algoritmo

de busca PLS.

Tabela 14: Tempos de execução (em segundos) para os algoritmos preProc1 e preProc2 no que se refere ao conjunto de instâncias CP

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
10.1	0,00	0,00
10.2	0,00	0,00
10.3	0,00	0,00
10.4	0,00	0,00
10.5	0,00	0,00
10.6	0,00	0,00
10.7	0,00	0,00
10.8	0,00	0,00
10.9	0,00	0,00
10.10	0,00	0,00
10.11	0,00	0,00
10.12	0,00	0,00
15.1	0,00	0,00
15.2	0,00	0,00
15.3	0,00	0,00
15.4	0,00	0,00
15.5	0,01	0,00
15.6	0,01	0,00
15.7	0,01	0,00
15.8	0,01	0,00
15.9	0,01	0,00
15.10	0,01	0,00
15.11	0,01	0,00
15.12	0,01	0,00
20.1	0,00	0,00
20.2	0,00	0,00
20.3	0,00	0,00
20.4	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
20.5	0,01	0,00
20.6	0,01	0,01
20.7	0,01	0,01
20.8	0,01	0,00
20.9	0,02	0,01
20.10	0,02	0,02
20.11	0,02	0,02
20.12	0,02	0,02
25.1	0,01	0,00
25.2	0,01	0,00
25.3	0,01	0,00
25.4	0,01	0,00
25.5	0,02	0,02
25.6	0,02	0,02
25.7	0,02	0,02
25.8	0,02	0,02
25.9	0,05	0,04
25.10	0,05	0,04
25.11	0,05	0,04
25.12	0,05	0,04
30.1	0,01	0,01
30.2	0,01	0,01
30.3	0,01	0,01
30.4	0,01	0,01
30.5	0,04	0,04
30.6	0,05	0,04
30.7	0,05	0,04
30.8	0,05	0,04
30.9	0,10	0,10
30.10	0,10	0,10
30.11	0,10	0,10
30.12	0,10	0,10

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
35.1	0,02	0,02
35.2	0,02	0,02
35.3	0,02	0,02
35.4	0,02	0,02
35.5	0,08	0,09
35.6	0,08	0,09
35.7	0,08	0,09
35.8	0,09	0,09
35.9	0,19	0,21
35.10	0,19	0,19
35.11	0,19	0,19
35.12	0,19	0,19
40.1	0,04	0,04
40.2	0,04	0,04
40.3	0,04	0,04
40.4	0,04	0,04
40.5	0,15	0,16
40.6	0,15	0,15
40.7	0,15	0,16
40.8	0,15	0,16
40.9	0,33	0,38
40.10	0,33	0,38
40.11	0,33	0,39
40.12	0,33	0,40
45.1	0,06	0,06
45.2	0,06	0,06
45.3	0,06	0,06
45.4	0,06	0,06
45.5	0,23	0,26
45.6	0,24	0,26
45.7	0,23	0,28
45.8	0,24	0,27

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
45.9	0,53	0,82
45.10	0,55	0,79
45.11	0,53	0,91
45.12	0,54	0,88
50.1	0,09	0,09
50.2	0,09	0,09
50.3	0,09	0,09
50.4	0,09	0,09
50.5	0,36	0,42
50.6	0,38	0,42
50.7	0,36	0,43
50.8	0,37	0,43
50.9	0,83	1,85
50.10	0,82	1,85
50.11	0,82	2,05
50.12	0,82	2,09
75.1	0,45	0,56
75.2	0,45	0,56
75.3	0,45	0,57
75.4	0,46	0,56
75.5	1,92	5,58
75.6	1,93	5,69
75.7	1,90	5,71
75.8	1,91	5,67
75.9	4,24	15,37
75.10	4,29	14,87
75.11	4,27	15,83
75.12	4,32	15,48
100.1	1,46	3,79
100.2	1,47	3,78
100.3	1,46	3,79
100.4	1,47	3,78



<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
100.5	6,11	25,97
100.6	6,19	26,53
100.7	6,15	26,87
100.8	6,22	26,04
100.9	13,74	59,47
100.10	13,85	59,41
100.11	13,78	59,65
100.12	14,14	59,92
125.1	3,67	13,06
125.2	3,70	13,01
125.3	3,70	13,27
125.4	3,72	13,32
125.5	15,27	63,31
125.6	15,36	61,79
125.7	15,48	62,23
125.8	15,52	61,73
125.9	34,13	154,29
125.10	34,80	155,45
125.11	34,80	156,84
125.12	34,84	156,31
150.1	7,73	28,26
150.2	7,82	28,01
150.3	7,79	28,58
150.4	7,86	28,52
150.5	32,12	145,04
150.6	32,53	142,73
150.7	32,71	144,85
150.8	32,87	143,67
175.1	14,49	55,81
175.2	14,62	57,80
175.3	14,60	57,91
175.4	14,76	57,93

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
175.5	60,41	285,80
175.6	61,16	289,66
175.7	61,22	296,34
175.8	61,72	292,92

Tabela 15: Tempos de execução (em segundos) para os algoritmos preProc1 e preProc2 no que se refere ao conjunto de instâncias OP1

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
6.1	0,00	0,00
6.2	0,00	0,00
6.3	0,00	0,00
6.4	0,00	0,00
6.5	0,00	0,00
6.6	0,00	0,00
6.7	0,00	0,00
6.8	0,00	0,00
6.9	0,00	0,00
6.10	0,00	0,00
7.1	0,00	0,00
7.2	0,00	0,00
7.3	0,00	0,00
7.4	0,00	0,00
7.5	0,00	0,00
7.6	0,00	0,00
7.7	0,00	0,00
7.8	0,00	0,00
7.9	0,00	0,00
7.10	0,00	0,00
8.1	0,00	0,00
8.2	0,00	0,00
8.3	0,00	0,00
8.4	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
8.5	0,00	0,00
8.6	0,00	0,00
8.7	0,00	0,00
8.8	0,00	0,00
8.9	0,00	0,00
8.10	0,00	0,00
9.1	0,00	0,00
9.2	0,00	0,00
9.3	0,00	0,00
9.4	0,00	0,00
9.5	0,00	0,00
9.6	0,00	0,00
9.7	0,00	0,00
9.8	0,00	0,00
9.9	0,00	0,00
9.10	0,00	0,00
10.1	0,00	0,00
10.2	0,00	0,00
10.3	0,00	0,00
10.4	0,00	0,00
10.5	0,00	0,00
10.6	0,00	0,00
10.7	0,00	0,00
10.8	0,00	0,00
10.9	0,00	0,00
10.10	0,00	0,00
11.1	0,00	0,00
11.2	0,00	0,00
11.3	0,00	0,00
11.4	0,00	0,00
11.5	0,00	0,00
11.6	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
11.7	0,00	0,00
11.8	0,00	0,00
11.9	0,00	0,00
11.10	0,00	0,00
12.1	0,00	0,00
12.2	0,00	0,00
12.3	0,00	0,00
12.4	0,00	0,00
12.5	0,00	0,00
12.6	0,00	0,00
12.7	0,00	0,00
12.8	0,00	0,00
12.9	0,00	0,00
12.10	0,00	0,00
13.1	0,00	0,00
13.2	0,00	0,00
13.3	0,00	0,00
13.4	0,00	0,00
13.5	0,00	0,00
13.6	0,00	0,00
13.7	0,00	0,00
13.8	0,00	0,00
13.9	0,00	0,00
13.10	0,00	0,00
14.1	0,00	0,00
14.2	0,00	0,00
14.3	0,00	0,00
14.4	0,00	0,00
14.5	0,00	0,00
14.6	0,00	0,00
14.7	0,00	0,00
14.8	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
14.9	0,00	0,00
14.10	0,00	0,00
15.1	0,00	0,00
15.2	0,00	0,00
15.3	0,00	0,00
15.4	0,01	0,00
15.5	0,00	0,00
15.6	0,00	0,00
15.7	0,00	0,00
15.8	0,00	0,00
15.9	0,00	0,00
15.10	0,00	0,00
16.1	0,00	0,00
16.2	0,01	0,00
16.3	0,00	0,00
16.4	0,01	0,00
16.5	0,00	0,00
16.6	0,00	0,00
16.7	0,00	0,00
16.8	0,00	0,00
16.9	0,00	0,00
16.10	0,00	0,00
17.1	0,00	0,01
17.2	0,00	0,01
17.3	0,00	0,01
17.4	0,00	0,00
17.5	0,00	0,00
17.6	0,00	0,00
17.7	0,00	0,00
17.8	0,01	0,00
17.9	0,01	0,00
17.10	0,01	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
18.1	0,01	0,01
18.2	0,01	0,01
18.3	0,01	0,01
18.4	0,01	0,01
18.5	0,01	0,01
18.6	0,01	0,01
18.7	0,01	0,01
18.8	0,01	0,01
18.9	0,01	0,01
18.10	0,01	0,01
20.1	0,01	0,01
20.2	0,01	0,01
20.3	0,01	0,01
20.4	0,01	0,01
20.5	0,01	0,01
20.6	0,01	0,01
20.7	0,01	0,01
20.8	0,01	0,01
20.9	0,01	0,01
20.10	0,01	0,01
30.1	0,03	0,06
30.2	0,03	0,06
30.3	0,03	0,06
30.4	0,04	0,05
30.5	0,03	0,06
30.6	0,03	0,06
30.7	0,03	0,06
30.8	0,04	0,06
30.9	0,03	0,06
30.10	0,03	0,06
50.1	0,28	1,02
50.2	0,28	1,03

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
50.3	0,28	1,03
50.4	0,28	1,02
50.5	0,29	1,05
50.6	0,28	1,03
50.7	0,28	1,04
50.8	0,28	1,03
50.9	0,28	1,02
50.10	0,28	1,05

Tabela 16: Tempos de execução (em segundos) para os algoritmos preProc1 e preProc2 no que se refere ao conjunto de instâncias OP2

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
6.1	0,00	0,00
6.2	0,00	0,00
6.3	0,00	0,00
6.4	0,00	0,00
6.5	0,00	0,00
6.6	0,00	0,00
6.7	0,00	0,00
6.8	0,00	0,00
6.9	0,00	0,00
6.10	0,00	0,00
7.1	0,00	0,00
7.2	0,00	0,00
7.3	0,00	0,00
7.4	0,00	0,00
7.5	0,00	0,00
7.6	0,00	0,00
7.7	0,00	0,00
7.8	0,00	0,00
7.9	0,00	0,00
7.10	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
8.1	0,00	0,00
8.2	0,00	0,00
8.3	0,00	0,00
8.4	0,00	0,00
8.5	0,00	0,00
8.6	0,00	0,00
8.7	0,00	0,00
8.8	0,00	0,00
8.9	0,00	0,00
8.10	0,00	0,00
9.1	0,00	0,00
9.2	0,00	0,00
9.3	0,00	0,00
9.4	0,00	0,00
9.5	0,00	0,00
9.6	0,00	0,00
9.7	0,00	0,00
9.8	0,00	0,00
9.9	0,00	0,00
9.10	0,00	0,00
10.1	0,00	0,00
10.2	0,00	0,00
10.3	0,00	0,00
10.4	0,00	0,00
10.5	0,00	0,00
10.6	0,00	0,00
10.7	0,00	0,00
10.8	0,00	0,00
10.9	0,00	0,00
10.10	0,00	0,00
11.1	0,00	0,00
11.2	0,00	0,00



<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
11.3	0,00	0,00
11.4	0,00	0,00
11.5	0,00	0,00
11.6	0,00	0,00
11.7	0,00	0,00
11.8	0,00	0,00
11.9	0,00	0,00
11.10	0,00	0,00
12.1	0,00	0,00
12.2	0,00	0,00
12.3	0,00	0,00
12.4	0,00	0,00
12.5	0,00	0,00
12.6	0,00	0,00
12.7	0,00	0,00
12.8	0,00	0,00
12.9	0,00	0,00
12.10	0,00	0,00
13.1	0,00	0,00
13.2	0,00	0,00
13.3	0,00	0,00
13.4	0,00	0,00
13.5	0,00	0,00
13.6	0,00	0,00
13.7	0,00	0,00
13.8	0,00	0,00
13.9	0,00	0,00
13.10	0,00	0,00
14.1	0,00	0,00
14.2	0,00	0,00
14.3	0,00	0,00
14.4	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
14.5	0,00	0,00
14.6	0,00	0,00
14.7	0,00	0,00
14.8	0,00	0,00
14.9	0,00	0,00
14.10	0,00	0,00
15.1	0,00	0,00
15.2	0,00	0,00
15.3	0,00	0,00
15.4	0,00	0,00
15.5	0,00	0,00
15.6	0,00	0,00
15.7	0,00	0,00
15.8	0,00	0,00
15.9	0,00	0,00
15.10	0,00	0,00
16.1	0,00	0,00
16.2	0,00	0,00
16.3	0,00	0,00
16.4	0,01	0,00
16.5	0,00	0,00
16.6	0,00	0,00
16.7	0,00	0,00
16.8	0,00	0,00
16.9	0,01	0,00
16.10	0,00	0,00
17.1	0,00	0,00
17.2	0,00	0,00
17.3	0,00	0,00
17.4	0,00	0,00
17.5	0,00	0,00
17.6	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
17.7	0,00	0,00
17.8	0,00	0,00
17.9	0,00	0,00
17.10	0,00	0,00
18.1	0,01	0,00
18.2	0,00	0,01
18.3	0,01	0,01
18.4	0,01	0,00
18.5	0,00	0,00
18.6	0,01	0,01
18.7	0,01	0,01
18.8	0,00	0,00
18.9	0,01	0,00
18.10	0,01	0,00
20.1	0,01	0,01
20.2	0,01	0,01
20.3	0,01	0,01
20.4	0,01	0,01
20.5	0,01	0,01
20.6	0,01	0,01
20.7	0,01	0,01
20.8	0,01	0,01
20.9	0,01	0,01
20.10	0,01	0,01
30.1	0,04	0,05
30.2	0,04	0,05
30.3	0,04	0,05
30.4	0,04	0,04
30.5	0,04	0,05
30.6	0,04	0,05
30.7	0,04	0,04
30.8	0,04	0,04

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
30.9	0,04	0,04
30.10	0,04	0,05
50.1	0,29	0,81
50.2	0,29	0,79
50.3	0,30	0,79
50.4	0,29	0,83
50.5	0,29	0,83
50.6	0,29	0,78
50.7	0,30	0,81
50.8	0,30	0,80
50.9	0,29	0,81
50.10	0,29	0,80

Tabela 17: Tempos de execução (em segundos) para os algoritmos preProc1 e preProc2 no que se refere ao conjunto de instâncias OP3

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
6.1	0,00	0,00
6.2	0,00	0,00
6.3	0,00	0,00
6.4	0,00	0,00
6.5	0,00	0,00
6.6	0,00	0,00
6.7	0,00	0,00
6.8	0,00	0,00
6.9	0,00	0,00
6.10	0,00	0,00
7.1	0,00	0,00
7.2	0,00	0,00
7.3	0,00	0,00
7.4	0,00	0,00
7.5	0,00	0,00
7.6	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
7.7	0,00	0,00
7.8	0,00	0,00
7.9	0,00	0,00
7.10	0,00	0,00
8.1	0,00	0,00
8.2	0,00	0,00
8.3	0,00	0,00
8.4	0,00	0,00
8.5	0,00	0,00
8.6	0,00	0,00
8.7	0,00	0,00
8.8	0,00	0,00
8.9	0,00	0,00
8.10	0,00	0,00
9.1	0,00	0,00
9.2	0,00	0,00
9.3	0,00	0,00
9.4	0,00	0,00
9.5	0,00	0,00
9.6	0,00	0,00
9.7	0,00	0,00
9.8	0,00	0,00
9.9	0,00	0,00
9.10	0,00	0,00
10.1	0,00	0,00
10.2	0,00	0,00
10.3	0,00	0,00
10.4	0,00	0,00
10.5	0,00	0,00
10.6	0,00	0,00
10.7	0,00	0,00
10.8	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
10.9	0,00	0,00
10.10	0,00	0,00
11.1	0,00	0,00
11.2	0,00	0,00
11.3	0,00	0,00
11.4	0,00	0,00
11.5	0,00	0,00
11.6	0,00	0,00
11.7	0,00	0,00
11.8	0,00	0,00
11.9	0,00	0,00
11.10	0,00	0,00
12.1	0,00	0,00
12.2	0,00	0,00
12.3	0,00	0,00
12.4	0,00	0,00
12.5	0,00	0,00
12.6	0,00	0,00
12.7	0,00	0,00
12.8	0,00	0,00
12.9	0,00	0,00
12.10	0,00	0,00
13.1	0,00	0,00
13.2	0,00	0,00
13.3	0,00	0,00
13.4	0,00	0,00
13.5	0,00	0,00
13.6	0,00	0,00
13.7	0,00	0,00
13.8	0,00	0,00
13.9	0,00	0,00
13.10	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
14.1	0,00	0,00
14.2	0,00	0,00
14.3	0,00	0,00
14.4	0,00	0,00
14.5	0,00	0,00
14.6	0,00	0,00
14.7	0,00	0,00
14.8	0,00	0,00
14.9	0,00	0,00
14.10	0,00	0,00
15.1	0,00	0,00
15.2	0,01	0,00
15.3	0,00	0,00
15.4	0,00	0,00
15.5	0,00	0,00
15.6	0,01	0,00
15.7	0,01	0,00
15.8	0,01	0,00
15.9	0,00	0,00
15.10	0,00	0,00
16.1	0,01	0,00
16.2	0,01	0,00
16.3	0,01	0,00
16.4	0,01	0,00
16.5	0,00	0,00
16.6	0,00	0,00
16.7	0,00	0,00
16.8	0,00	0,00
16.9	0,00	0,00
16.10	0,00	0,00
17.1	0,00	0,00
17.2	0,00	0,00

<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
17.3	0,00	0,00
17.4	0,00	0,00
17.5	0,00	0,00
17.6	0,00	0,00
17.7	0,00	0,00
17.8	0,00	0,00
17.9	0,00	0,00
17.10	0,00	0,00
18.1	0,01	0,01
18.2	0,01	0,01
18.3	0,01	0,01
18.4	0,01	0,01
18.5	0,01	0,01
18.6	0,01	0,01
18.7	0,01	0,01
18.8	0,01	0,01
18.9	0,01	0,01
18.10	0,01	0,01
20.1	0,00	0,00
20.2	0,00	0,00
20.3	0,00	0,00
20.4	0,00	0,00
20.5	0,00	0,00
20.6	0,00	0,00
20.7	0,00	0,00
20.8	0,00	0,00
20.9	0,00	0,00
20.10	0,00	0,00
30.1	0,04	0,05
30.2	0,03	0,06
30.3	0,03	0,06
30.4	0,03	0,06



<b>Inst.</b>	<b>preProc1</b>	<b>preProc2</b>
30.5	0,04	0,06
30.6	0,03	0,05
30.7	0,03	0,06
30.8	0,04	0,06
30.9	0,03	0,06
30.10	0,03	0,06
50.1	0,27	1,02
50.2	0,27	1,05
50.3	0,27	1,00
50.4	0,27	1,02
50.5	0,27	1,04
50.6	0,27	1,02
50.7	0,27	1,02
50.8	0,27	1,00
50.9	0,27	1,00
50.10	0,27	1,05

As Tabelas 18 a 21 exibem os  $p$ -valores da comparação entre o NSGA-II e os outros algoritmos no que se refere ao indicador hipervolume, para os quatro grupos de instâncias adotados. Os  $p$ -valores exibidos são em favor do NSGA-II. Isso significa que valores inferiores a 0,05 apontam que o NSGA-II é melhor e valores superiores a 0,95 indicam que o algoritmo com o qual estamos comparando o NSGA-II é que é melhor. O rótulo nas colunas indica o algoritmo de comparação e o rótulo nas linhas identifica a instância.

Tabela 18:  $p$ -valores em favor do NSGA-II para instâncias CP

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
10.1	0,00	0,50	0,50	0,50
10.2	0,00	0,50	0,50	0,50
10.3	0,02	0,50	0,50	0,50
10.4	0,04	0,50	0,50	0,50
10.5	0,00	0,50	0,50	0,50
10.6	0,56	1,00	0,00	1,00
10.7	0,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
10.8	0,35	1,00	0,00	0,94
10.9	1,00	1,00	1,00	1,00
10.10	1,00	1,00	1,00	1,00
10.11	1,00	1,00	1,00	1,00
10.12	1,00	1,00	1,00	1,00
15.1	1,00	1,00	1,00	1,00
15.2	1,00	1,00	1,00	1,00
15.3	1,00	1,00	1,00	1,00
15.4	1,00	1,00	1,00	1,00
15.5	1,00	1,00	1,00	1,00
15.6	1,00	1,00	1,00	1,00
15.7	1,00	1,00	1,00	1,00
15.8	1,00	1,00	1,00	1,00
15.9	1,00	1,00	1,00	1,00
15.10	1,00	1,00	1,00	1,00
15.11	1,00	1,00	1,00	1,00
15.12	1,00	1,00	1,00	1,00
20.1	1,00	1,00	1,00	1,00
20.2	1,00	1,00	1,00	1,00
20.3	1,00	1,00	1,00	1,00
20.4	1,00	1,00	1,00	1,00
20.5	1,00	1,00	1,00	1,00
20.6	1,00	1,00	1,00	1,00
20.7	1,00	1,00	1,00	1,00
20.8	1,00	1,00	1,00	1,00
20.9	1,00	1,00	1,00	1,00
20.10	1,00	1,00	1,00	1,00
20.11	1,00	1,00	1,00	1,00
20.12	1,00	1,00	1,00	1,00
25.1	1,00	1,00	1,00	1,00
25.2	1,00	1,00	1,00	1,00
25.3	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
25.4	1,00	1,00	1,00	1,00
25.5	1,00	1,00	1,00	1,00
25.6	1,00	1,00	1,00	1,00
25.7	1,00	1,00	1,00	1,00
25.8	1,00	1,00	1,00	1,00
25.9	1,00	1,00	1,00	1,00
25.10	1,00	1,00	1,00	1,00
25.11	1,00	1,00	1,00	1,00
25.12	1,00	1,00	1,00	1,00
30.1	1,00	1,00	1,00	1,00
30.2	1,00	1,00	1,00	1,00
30.3	1,00	1,00	1,00	1,00
30.4	1,00	1,00	1,00	1,00
30.5	1,00	1,00	1,00	1,00
30.6	1,00	1,00	1,00	1,00
30.7	1,00	1,00	1,00	1,00
30.8	1,00	1,00	1,00	1,00
30.9	1,00	1,00	1,00	1,00
30.10	1,00	1,00	1,00	1,00
30.11	1,00	1,00	1,00	1,00
30.12	1,00	1,00	1,00	1,00
35.1	1,00	1,00	1,00	1,00
35.2	1,00	1,00	1,00	1,00
35.3	1,00	1,00	1,00	1,00
35.4	1,00	1,00	1,00	1,00
35.5	1,00	1,00	1,00	1,00
35.6	1,00	1,00	1,00	1,00
35.7	1,00	1,00	1,00	1,00
35.8	1,00	1,00	1,00	1,00
35.9	1,00	1,00	1,00	1,00
35.10	1,00	1,00	1,00	1,00
35.11	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
35.12	1,00	1,00	1,00	1,00
40.1	1,00	1,00	1,00	1,00
40.2	1,00	1,00	1,00	1,00
40.3	1,00	1,00	1,00	1,00
40.4	1,00	1,00	1,00	1,00
40.5	1,00	1,00	1,00	1,00
40.6	1,00	1,00	1,00	1,00
40.7	1,00	1,00	1,00	1,00
40.8	1,00	1,00	1,00	1,00
40.9	1,00	1,00	1,00	1,00
40.10	1,00	1,00	1,00	1,00
40.11	1,00	1,00	1,00	1,00
40.12	1,00	1,00	1,00	1,00
45.1	1,00	1,00	1,00	1,00
45.2	1,00	1,00	1,00	1,00
45.3	1,00	1,00	1,00	1,00
45.4	1,00	1,00	1,00	1,00
45.5	1,00	1,00	1,00	1,00
45.6	1,00	1,00	1,00	1,00
45.7	1,00	1,00	1,00	1,00
45.8	1,00	1,00	1,00	1,00
45.9	1,00	1,00	1,00	1,00
45.10	1,00	1,00	1,00	1,00
45.11	1,00	1,00	1,00	1,00
45.12	1,00	1,00	1,00	1,00
50.1	1,00	1,00	1,00	1,00
50.2	1,00	1,00	1,00	1,00
50.3	1,00	1,00	1,00	1,00
50.4	1,00	1,00	1,00	1,00
50.5	1,00	1,00	1,00	1,00
50.6	1,00	1,00	1,00	1,00
50.7	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
50.8	1,00	1,00	1,00	1,00
50.9	1,00	1,00	1,00	1,00
50.10	1,00	1,00	1,00	1,00
50.11	1,00	1,00	1,00	1,00
50.12	1,00	1,00	1,00	1,00
75.1	1,00	1,00	1,00	1,00
75.2	1,00	1,00	1,00	1,00
75.3	1,00	1,00	1,00	1,00
75.4	1,00	1,00	1,00	1,00
75.5	1,00	1,00	1,00	1,00
75.6	1,00	1,00	1,00	1,00
75.7	1,00	1,00	1,00	1,00
75.8	1,00	1,00	1,00	1,00
75.9	1,00	1,00	1,00	1,00
75.10	1,00	1,00	1,00	1,00
75.11	1,00	1,00	1,00	1,00
75.12	1,00	1,00	1,00	1,00
100.1	1,00	1,00	1,00	1,00
100.2	1,00	1,00	1,00	1,00
100.3	1,00	1,00	1,00	1,00
100.4	1,00	1,00	1,00	1,00
100.5	1,00	1,00	1,00	1,00
100.6	1,00	1,00	1,00	1,00
100.7	1,00	1,00	1,00	1,00
100.8	1,00	1,00	1,00	1,00
100.9	1,00	1,00	1,00	1,00
100.10	1,00	1,00	1,00	1,00
100.11	1,00	1,00	1,00	1,00
100.12	1,00	1,00	1,00	1,00
125.1	1,00	1,00	1,00	1,00
125.2	1,00	1,00	1,00	1,00
125.3	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
125.4	1,00	1,00	1,00	1,00
125.5	1,00	1,00	1,00	1,00
125.6	1,00	1,00	1,00	1,00
125.7	1,00	1,00	1,00	1,00
125.8	1,00	1,00	1,00	1,00
125.9	1,00	1,00	1,00	1,00
125.10	1,00	1,00	1,00	1,00
125.11	1,00	1,00	1,00	1,00
125.12	1,00	1,00	1,00	1,00
150.1	1,00	1,00	1,00	1,00
150.2	1,00	1,00	1,00	1,00
150.3	1,00	1,00	1,00	1,00
150.4	1,00	1,00	1,00	1,00
150.5	1,00	1,00	1,00	1,00
150.6	1,00	1,00	1,00	1,00
150.7	1,00	1,00	1,00	1,00
150.8	1,00	1,00	1,00	1,00
175.1	1,00	1,00	1,00	1,00
175.2	1,00	1,00	1,00	1,00
175.3	1,00	1,00	1,00	1,00
175.4	1,00	1,00	1,00	1,00
175.5	1,00	1,00	1,00	1,00
175.6	1,00	1,00	1,00	1,00
175.7	1,00	1,00	1,00	1,00
175.8	1,00	1,00	1,00	1,00

Tabela 19:  $p$ -valores em favor do NSGA-II para instâncias OP1

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,61	0,00	0,97
6.2	0,00	0,81	0,01	1,00
6.3	0,00	0,02	0,02	1,00
6.4	0,00	1,00	0,74	1,00
6.5	0,01	0,50	0,00	0,50

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
6.6	0,00	0,99	0,19	1,00
6.7	0,00	0,50	0,00	0,50
6.8	0,00	1,00	0,00	1,00
6.9	0,00	0,50	0,50	0,50
6.10	0,00	0,08	0,00	0,16
7.1	0,01	1,00	1,00	1,00
7.2	0,00	1,00	0,00	1,00
7.3	0,00	1,00	0,84	1,00
7.4	0,01	1,00	0,01	0,92
7.5	0,49	1,00	0,23	1,00
7.6	0,00	0,98	0,00	0,58
7.7	0,00	0,50	0,00	0,08
7.8	0,95	1,00	1,00	1,00
7.9	0,00	1,00	0,00	0,10
7.10	0,00	0,80	0,00	0,00
8.1	0,00	0,87	0,00	0,01
8.2	1,00	1,00	1,00	1,00
8.3	0,00	0,30	0,00	0,00
8.4	0,00	0,75	0,00	0,29
8.5	0,00	1,00	0,02	1,00
8.6	1,00	1,00	1,00	1,00
8.7	0,00	1,00	0,00	0,20
8.8	1,00	1,00	0,00	0,99
8.9	0,00	0,37	0,68	1,00
8.10	0,93	1,00	1,00	1,00
9.1	0,03	1,00	0,91	1,00
9.2	0,52	1,00	0,00	0,00
9.3	1,00	1,00	0,57	1,00
9.4	0,00	0,80	0,04	1,00
9.5	0,87	1,00	0,00	1,00
9.6	0,88	1,00	0,00	1,00
9.7	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
9.8	0,00	1,00	0,81	1,00
9.9	0,98	1,00	0,00	0,00
9.10	0,00	0,92	1,00	1,00
10.1	1,00	1,00	0,00	0,70
10.2	1,00	1,00	1,00	1,00
10.3	1,00	1,00	0,00	0,32
10.4	1,00	1,00	1,00	1,00
10.5	1,00	1,00	1,00	1,00
10.6	1,00	1,00	1,00	1,00
10.7	0,99	1,00	1,00	1,00
10.8	1,00	1,00	1,00	1,00
10.9	1,00	1,00	0,00	0,70
10.10	1,00	1,00	0,34	1,00
11.1	1,00	1,00	0,98	0,99
11.2	1,00	1,00	1,00	1,00
11.3	1,00	1,00	1,00	1,00
11.4	1,00	1,00	1,00	1,00
11.5	1,00	1,00	1,00	1,00
11.6	1,00	1,00	1,00	1,00
11.7	1,00	1,00	0,01	1,00
11.8	1,00	1,00	0,00	1,00
11.9	1,00	1,00	1,00	1,00
11.10	1,00	1,00	0,99	1,00
12.1	1,00	1,00	1,00	1,00
12.2	1,00	1,00	0,13	1,00
12.3	1,00	1,00	1,00	1,00
12.4	1,00	1,00	1,00	1,00
12.5	1,00	1,00	0,99	1,00
12.6	1,00	1,00	1,00	1,00
12.7	1,00	1,00	1,00	1,00
12.8	1,00	1,00	1,00	1,00
12.9	1,00	1,00	1,00	1,00



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
12.10	1,00	1,00	1,00	1,00
13.1	1,00	1,00	1,00	1,00
13.2	1,00	1,00	0,00	1,00
13.3	1,00	1,00	1,00	1,00
13.4	1,00	1,00	1,00	1,00
13.5	1,00	1,00	1,00	1,00
13.6	1,00	1,00	1,00	1,00
13.7	1,00	1,00	1,00	1,00
13.8	1,00	1,00	1,00	1,00
13.9	1,00	1,00	1,00	1,00
13.10	1,00	1,00	1,00	1,00
14.1	1,00	1,00	1,00	1,00
14.2	1,00	1,00	1,00	1,00
14.3	1,00	1,00	1,00	1,00
14.4	1,00	1,00	1,00	1,00
14.5	1,00	1,00	1,00	1,00
14.6	1,00	1,00	1,00	1,00
14.7	1,00	1,00	1,00	1,00
14.8	1,00	1,00	1,00	1,00
14.9	1,00	1,00	1,00	1,00
14.10	1,00	1,00	1,00	1,00
15.1	1,00	1,00	1,00	1,00
15.2	1,00	1,00	1,00	1,00
15.3	1,00	1,00	1,00	1,00
15.4	1,00	1,00	1,00	1,00
15.5	1,00	1,00	1,00	1,00
15.6	1,00	1,00	1,00	1,00
15.7	1,00	1,00	1,00	1,00
15.8	1,00	1,00	1,00	1,00
15.9	1,00	1,00	1,00	1,00
15.10	1,00	1,00	1,00	1,00
16.1	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
16.2	1,00	1,00	1,00	1,00
16.3	1,00	1,00	1,00	1,00
16.4	1,00	1,00	1,00	1,00
16.5	1,00	1,00	1,00	1,00
16.6	1,00	1,00	1,00	1,00
16.7	1,00	1,00	1,00	1,00
16.8	1,00	1,00	1,00	1,00
16.9	1,00	1,00	1,00	1,00
16.10	1,00	1,00	1,00	1,00
17.1	1,00	1,00	1,00	1,00
17.2	1,00	1,00	1,00	1,00
17.3	1,00	1,00	1,00	1,00
17.4	1,00	1,00	1,00	1,00
17.5	1,00	1,00	1,00	1,00
17.6	1,00	1,00	1,00	1,00
17.7	1,00	1,00	1,00	1,00
17.8	1,00	1,00	1,00	1,00
17.9	1,00	1,00	1,00	1,00
17.10	1,00	1,00	1,00	1,00
18.1	1,00	1,00	1,00	1,00
18.2	1,00	1,00	1,00	1,00
18.3	1,00	1,00	1,00	1,00
18.4	1,00	1,00	1,00	1,00
18.5	1,00	1,00	1,00	1,00
18.6	1,00	1,00	1,00	1,00
18.7	1,00	1,00	1,00	1,00
18.8	1,00	1,00	1,00	1,00
18.9	1,00	1,00	1,00	1,00
18.10	1,00	1,00	1,00	1,00
20.1	1,00	1,00	1,00	1,00
20.2	1,00	1,00	1,00	1,00
20.3	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
20.4	1,00	1,00	1,00	1,00
20.5	1,00	1,00	1,00	1,00
20.6	1,00	1,00	1,00	1,00
20.7	1,00	1,00	1,00	1,00
20.8	1,00	1,00	1,00	1,00
20.9	1,00	1,00	1,00	1,00
20.10	1,00	1,00	1,00	1,00
30.1	1,00	1,00	1,00	1,00
30.2	1,00	1,00	1,00	1,00
30.3	1,00	1,00	1,00	1,00
30.4	1,00	1,00	1,00	1,00
30.5	1,00	1,00	1,00	1,00
30.6	1,00	1,00	1,00	1,00
30.7	1,00	1,00	1,00	1,00
30.8	1,00	1,00	1,00	1,00
30.9	1,00	1,00	1,00	1,00
30.10	1,00	1,00	1,00	1,00
50.1	1,00	1,00	1,00	1,00
50.2	1,00	1,00	1,00	1,00
50.3	1,00	1,00	1,00	1,00
50.4	1,00	1,00	1,00	1,00
50.5	1,00	1,00	1,00	1,00
50.6	1,00	1,00	1,00	1,00
50.7	1,00	1,00	1,00	1,00
50.8	1,00	1,00	1,00	1,00
50.9	1,00	1,00	1,00	1,00
50.10	1,00	1,00	1,00	1,00

Tabela 20:  $p$ -valores em favor do NSGA-II para instâncias OP2

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,16	0,00	0,29
6.2	0,00	0,50	0,02	0,50
6.3	0,00	0,50	0,50	0,50

Inst.	PLS	Tabu	NSTA	AT
6.4	0,00	0,16	0,50	0,50
6.5	0,00	0,00	0,00	0,08
6.6	0,00	0,96	0,53	1,00
6.7	0,00	0,02	0,00	0,94
6.8	0,00	0,00	0,00	0,16
6.9	0,00	0,50	0,50	0,50
6.10	0,08	0,50	0,50	0,50
7.1	0,00	0,00	0,00	0,00
7.2	0,00	0,96	1,00	1,00
7.3	0,00	0,50	0,00	0,50
7.4	0,00	0,00	0,00	0,50
7.5	0,00	0,00	0,00	0,00
7.6	0,00	1,00	0,76	1,00
7.7	0,00	0,95	0,99	0,99
7.8	0,00	1,00	1,00	1,00
7.9	0,00	0,00	0,94	0,98
7.10	0,00	1,00	1,00	1,00
8.1	0,00	0,00	0,00	0,95
8.2	0,00	1,00	1,00	1,00
8.3	0,00	0,73	0,00	0,07
8.4	0,00	0,00	0,00	0,99
8.5	0,00	0,00	0,00	0,01
8.6	0,04	1,00	1,00	1,00
8.7	0,00	1,00	1,00	0,98
8.8	0,00	0,00	0,00	0,03
8.9	0,00	0,77	0,00	0,00
8.10	0,66	1,00	1,00	1,00
9.1	0,99	1,00	0,00	1,00
9.2	0,01	0,00	0,00	1,00
9.3	0,00	1,00	0,00	1,00
9.4	1,00	1,00	1,00	1,00
9.5	0,99	1,00	0,02	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
9.6	0,00	0,84	0,00	0,04
9.7	0,00	1,00	0,04	1,00
9.8	0,99	1,00	0,72	0,99
9.9	0,96	1,00	0,96	0,98
9.10	1,00	1,00	0,99	1,00
10.1	0,00	1,00	1,00	1,00
10.2	0,90	1,00	0,02	1,00
10.3	1,00	1,00	0,21	1,00
10.4	0,83	1,00	0,30	0,55
10.5	1,00	1,00	0,99	1,00
10.6	1,00	1,00	1,00	1,00
10.7	1,00	1,00	0,91	1,00
10.8	1,00	1,00	1,00	1,00
10.9	1,00	1,00	1,00	1,00
10.10	0,28	1,00	0,86	1,00
11.1	1,00	1,00	1,00	1,00
11.2	0,99	1,00	1,00	1,00
11.3	1,00	1,00	1,00	1,00
11.4	1,00	1,00	1,00	1,00
11.5	1,00	1,00	1,00	1,00
11.6	1,00	1,00	1,00	1,00
11.7	1,00	1,00	1,00	1,00
11.8	1,00	1,00	1,00	1,00
11.9	1,00	1,00	1,00	1,00
11.10	1,00	1,00	1,00	1,00
12.1	1,00	1,00	1,00	1,00
12.2	1,00	1,00	1,00	1,00
12.3	1,00	1,00	1,00	1,00
12.4	1,00	1,00	1,00	1,00
12.5	1,00	1,00	1,00	1,00
12.6	0,73	1,00	1,00	0,00
12.7	1,00	1,00	0,47	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
12.8	1,00	1,00	0,10	0,99
12.9	1,00	1,00	1,00	1,00
12.10	1,00	1,00	1,00	1,00
13.1	1,00	1,00	1,00	1,00
13.2	1,00	1,00	1,00	1,00
13.3	0,79	1,00	0,00	0,86
13.4	1,00	1,00	1,00	0,83
13.5	0,95	1,00	1,00	1,00
13.6	1,00	1,00	1,00	1,00
13.7	1,00	1,00	1,00	1,00
13.8	1,00	1,00	0,00	0,30
13.9	1,00	1,00	0,93	1,00
13.10	1,00	1,00	1,00	1,00
14.1	1,00	1,00	1,00	1,00
14.2	1,00	1,00	1,00	1,00
14.3	1,00	1,00	1,00	1,00
14.4	1,00	1,00	1,00	1,00
14.5	1,00	1,00	1,00	1,00
14.6	1,00	1,00	1,00	1,00
14.7	1,00	1,00	1,00	0,00
14.8	1,00	1,00	1,00	0,02
14.9	1,00	1,00	1,00	1,00
14.10	1,00	1,00	1,00	1,00
15.1	1,00	1,00	1,00	1,00
15.2	1,00	1,00	1,00	1,00
15.3	1,00	1,00	1,00	1,00
15.4	1,00	1,00	1,00	1,00
15.5	1,00	1,00	1,00	1,00
15.6	1,00	1,00	1,00	1,00
15.7	1,00	1,00	1,00	1,00
15.8	1,00	1,00	1,00	1,00
15.9	1,00	1,00	0,00	0,82

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
15.10	1,00	1,00	1,00	1,00
16.1	1,00	1,00	1,00	1,00
16.2	1,00	1,00	1,00	1,00
16.3	1,00	1,00	1,00	0,09
16.4	1,00	1,00	1,00	1,00
16.5	0,95	1,00	1,00	1,00
16.6	1,00	1,00	1,00	1,00
16.7	1,00	1,00	1,00	1,00
16.8	1,00	1,00	1,00	1,00
16.9	0,63	1,00	1,00	1,00
16.10	0,04	1,00	1,00	1,00
17.1	1,00	1,00	1,00	1,00
17.2	1,00	1,00	1,00	1,00
17.3	1,00	1,00	1,00	1,00
17.4	1,00	1,00	1,00	1,00
17.5	1,00	1,00	1,00	1,00
17.6	1,00	1,00	1,00	0,97
17.7	0,00	1,00	1,00	0,92
17.8	1,00	1,00	1,00	1,00
17.9	1,00	1,00	1,00	1,00
17.10	0,00	1,00	0,43	0,15
18.1	1,00	1,00	0,99	1,00
18.2	1,00	1,00	1,00	1,00
18.3	1,00	1,00	1,00	1,00
18.4	1,00	1,00	1,00	1,00
18.5	1,00	1,00	0,96	0,00
18.6	1,00	1,00	1,00	1,00
18.7	1,00	1,00	1,00	1,00
18.8	1,00	1,00	1,00	1,00
18.9	1,00	1,00	1,00	0,13
18.10	1,00	1,00	1,00	0,09
20.1	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
20.2	1,00	1,00	1,00	1,00
20.3	1,00	1,00	1,00	1,00
20.4	1,00	1,00	1,00	1,00
20.5	1,00	1,00	1,00	1,00
20.6	1,00	1,00	1,00	1,00
20.7	1,00	1,00	1,00	1,00
20.8	1,00	1,00	1,00	1,00
20.9	1,00	1,00	1,00	1,00
20.10	1,00	1,00	1,00	1,00
30.1	1,00	1,00	1,00	1,00
30.2	1,00	1,00	1,00	1,00
30.3	1,00	1,00	1,00	1,00
30.4	1,00	1,00	1,00	1,00
30.5	1,00	1,00	1,00	1,00
30.6	1,00	1,00	1,00	1,00
30.7	1,00	1,00	1,00	1,00
30.8	1,00	1,00	1,00	1,00
30.9	1,00	1,00	1,00	1,00
30.10	1,00	1,00	1,00	1,00
50.1	1,00	1,00	1,00	1,00
50.2	1,00	1,00	1,00	1,00
50.3	1,00	1,00	1,00	1,00
50.4	1,00	1,00	1,00	1,00
50.5	1,00	1,00	1,00	1,00
50.6	1,00	1,00	1,00	1,00
50.7	1,00	1,00	1,00	1,00
50.8	1,00	1,00	1,00	1,00
50.9	1,00	1,00	1,00	1,00
50.10	1,00	1,00	1,00	1,00

Tabela 21:  $p$ -valores em favor do NSGA-II para instâncias OP3

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,16	0,00	0,16



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
6.2	0,00	1,00	0,99	1,00
6.3	0,00	0,67	0,46	1,00
6.4	0,00	0,46	0,01	0,97
6.5	0,00	1,00	0,01	1,00
6.6	0,00	0,02	0,00	0,96
6.7	0,00	1,00	0,00	1,00
6.8	0,00	0,35	0,00	1,00
6.9	0,00	0,97	0,24	0,98
6.10	0,00	0,03	0,00	0,84
7.1	0,00	1,00	0,77	1,00
7.2	0,00	1,00	0,00	1,00
7.3	0,00	1,00	0,03	1,00
7.4	0,00	0,93	0,01	1,00
7.5	0,00	1,00	0,99	1,00
7.6	0,00	1,00	1,00	1,00
7.7	0,09	1,00	0,69	1,00
7.8	0,00	1,00	1,00	1,00
7.9	0,00	0,96	0,00	0,38
7.10	0,32	1,00	1,00	1,00
8.1	0,35	1,00	1,00	1,00
8.2	0,57	1,00	1,00	1,00
8.3	0,91	1,00	1,00	1,00
8.4	0,07	1,00	1,00	1,00
8.5	0,94	1,00	1,00	1,00
8.6	0,00	1,00	0,93	1,00
8.7	0,11	1,00	1,00	1,00
8.8	0,00	1,00	1,00	1,00
8.9	0,26	1,00	0,99	1,00
8.10	0,98	1,00	1,00	1,00
9.1	1,00	1,00	1,00	1,00
9.2	0,99	1,00	1,00	1,00
9.3	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
9.4	0,94	1,00	1,00	1,00
9.5	0,57	1,00	1,00	1,00
9.6	0,99	1,00	1,00	1,00
9.7	1,00	1,00	1,00	1,00
9.8	1,00	1,00	1,00	1,00
9.9	0,66	1,00	1,00	1,00
9.10	1,00	1,00	1,00	1,00
10.1	1,00	1,00	1,00	1,00
10.2	0,00	1,00	1,00	1,00
10.3	1,00	1,00	0,99	1,00
10.4	0,99	1,00	1,00	1,00
10.5	1,00	1,00	1,00	1,00
10.6	1,00	1,00	0,98	1,00
10.7	1,00	1,00	1,00	1,00
10.8	1,00	1,00	1,00	1,00
10.9	1,00	1,00	1,00	1,00
10.10	1,00	1,00	1,00	1,00
11.1	1,00	1,00	1,00	1,00
11.2	0,99	1,00	1,00	1,00
11.3	1,00	1,00	1,00	1,00
11.4	1,00	1,00	1,00	1,00
11.5	1,00	1,00	1,00	1,00
11.6	1,00	1,00	1,00	1,00
11.7	1,00	1,00	1,00	1,00
11.8	1,00	1,00	1,00	1,00
11.9	1,00	1,00	1,00	1,00
11.10	1,00	1,00	1,00	1,00
12.1	1,00	1,00	1,00	1,00
12.2	1,00	1,00	0,13	1,00
12.3	1,00	1,00	1,00	1,00
12.4	1,00	1,00	0,60	1,00
12.5	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
12.6	1,00	1,00	0,96	1,00
12.7	1,00	1,00	0,00	1,00
12.8	1,00	1,00	1,00	1,00
12.9	1,00	1,00	0,99	1,00
12.10	1,00	1,00	0,98	1,00
13.1	1,00	1,00	1,00	1,00
13.2	1,00	1,00	0,00	1,00
13.3	1,00	1,00	0,97	1,00
13.4	1,00	1,00	1,00	1,00
13.5	1,00	1,00	1,00	1,00
13.6	1,00	1,00	1,00	1,00
13.7	1,00	1,00	1,00	1,00
13.8	1,00	1,00	0,06	1,00
13.9	1,00	1,00	1,00	1,00
13.10	1,00	1,00	1,00	1,00
14.1	1,00	1,00	1,00	1,00
14.2	1,00	1,00	1,00	1,00
14.3	1,00	1,00	1,00	1,00
14.4	1,00	1,00	1,00	1,00
14.5	1,00	1,00	0,66	1,00
14.6	0,95	1,00	1,00	1,00
14.7	1,00	1,00	1,00	1,00
14.8	1,00	0,88	0,00	1,00
14.9	1,00	1,00	1,00	1,00
14.10	1,00	1,00	1,00	1,00
15.1	1,00	1,00	1,00	1,00
15.2	1,00	1,00	1,00	1,00
15.3	1,00	1,00	1,00	1,00
15.4	1,00	1,00	1,00	1,00
15.5	1,00	1,00	1,00	1,00
15.6	1,00	1,00	1,00	1,00
15.7	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
15.8	1,00	1,00	0,00	1,00
15.9	1,00	1,00	1,00	1,00
15.10	1,00	1,00	0,89	1,00
16.1	1,00	1,00	1,00	1,00
16.2	1,00	1,00	1,00	1,00
16.3	1,00	1,00	1,00	1,00
16.4	1,00	1,00	1,00	1,00
16.5	1,00	1,00	0,02	1,00
16.6	1,00	1,00	1,00	1,00
16.7	1,00	1,00	0,00	1,00
16.8	1,00	1,00	1,00	1,00
16.9	1,00	1,00	1,00	1,00
16.10	1,00	1,00	1,00	1,00
17.1	1,00	0,98	1,00	1,00
17.2	1,00	0,99	1,00	1,00
17.3	1,00	1,00	1,00	1,00
17.4	1,00	1,00	0,04	1,00
17.5	1,00	1,00	1,00	1,00
17.6	1,00	1,00	1,00	1,00
17.7	1,00	1,00	0,02	1,00
17.8	1,00	1,00	1,00	1,00
17.9	1,00	1,00	0,00	1,00
17.10	1,00	1,00	1,00	1,00
18.1	1,00	1,00	0,00	1,00
18.2	1,00	1,00	0,26	1,00
18.3	1,00	1,00	1,00	1,00
18.4	1,00	0,95	1,00	1,00
18.5	1,00	1,00	1,00	1,00
18.6	1,00	1,00	1,00	1,00
18.7	1,00	0,54	1,00	1,00
18.8	1,00	1,00	0,00	1,00
18.9	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSTA</b>	<b>AT</b>
18.10	1,00	1,00	0,00	1,00
20.1	0,00	0,02	0,00	0,04
20.2	0,00	0,50	0,04	0,16
20.3	0,00	0,50	0,02	0,50
20.4	0,00	0,50	0,02	0,50
20.5	0,00	0,00	0,00	0,50
20.6	0,00	0,91	0,95	1,00
20.7	0,00	0,21	0,01	0,99
20.8	0,00	0,01	0,00	0,16
20.9	0,00	0,50	0,50	0,50
20.10	0,01	0,50	0,50	0,50
30.1	1,00	1,00	1,00	1,00
30.2	1,00	1,00	1,00	1,00
30.3	1,00	1,00	1,00	1,00
30.4	1,00	1,00	1,00	1,00
30.5	1,00	1,00	1,00	1,00
30.6	1,00	1,00	0,98	1,00
30.7	1,00	1,00	1,00	1,00
30.8	1,00	1,00	1,00	1,00
30.9	1,00	1,00	1,00	1,00
30.10	1,00	1,00	1,00	1,00
50.1	1,00	1,00	1,00	1,00
50.2	1,00	1,00	1,00	1,00
50.3	1,00	1,00	1,00	1,00
50.4	1,00	1,00	1,00	1,00
50.5	1,00	1,00	1,00	1,00
50.6	1,00	1,00	1,00	1,00
50.7	1,00	1,00	1,00	1,00
50.8	1,00	1,00	1,00	1,00
50.9	1,00	1,00	1,00	1,00
50.10	1,00	1,00	1,00	1,00

Pode-se observar, a partir das Tabelas 18, 19, 20 e 21, que, em termos de qualidade de solução, o algoritmo NSGA-II foi considerado pior que todas as outras abordagens

desenvolvidas. Para instâncias pequenas, com até 10 vértices, o NSGA-II ainda consegue superar a busca local PLS. Entretanto, o quadro se inverte conforme o tamanho das instâncias aumenta.

As Tabelas 22 a 25 exibem os  $p$ -valores da comparação entre o NSTA e os outros algoritmos no que se refere ao indicador hipervolume, para os quatro grupos de instâncias adotados. Os  $p$ -valores exibidos são em favor do NSTA. O rótulo nas colunas indica o algoritmo de comparação e o rótulo nas linhas identifica a instância.

Tabela 22:  $p$ -valores em favor do NSTA para instâncias CP

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>AT</b>
10.1	0,00	0,50	0,50	0,50
10.2	0,00	0,50	0,50	0,50
10.3	0,02	0,50	0,50	0,50
10.4	0,04	0,50	0,50	0,50
10.5	0,00	0,50	0,50	0,50
10.6	0,98	1,00	1,00	1,00
10.7	0,02	0,99	1,00	1,00
10.8	1,00	1,00	1,00	1,00
10.9	1,00	1,00	0,00	1,00
10.10	1,00	1,00	0,00	1,00
10.11	0,45	1,00	0,00	0,99
10.12	1,00	1,00	0,00	1,00
15.1	0,00	1,00	0,00	0,05
15.2	0,00	1,00	0,00	0,00
15.3	0,00	1,00	0,00	0,29
15.4	0,31	1,00	0,00	0,00
15.5	1,00	1,00	0,00	1,00
15.6	0,52	1,00	0,00	1,00
15.7	0,21	1,00	0,00	0,03
15.8	0,71	1,00	0,00	1,00
15.9	0,05	1,00	0,00	0,00
15.10	1,00	1,00	0,00	0,00
15.11	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
15.12	1,00	1,00	0,00	0,00
20.1	0,02	0,71	0,00	0,00
20.2	0,02	0,24	0,00	0,00
20.3	0,00	0,84	0,00	0,00
20.4	0,95	1,00	0,00	0,00
20.5	1,00	1,00	0,00	0,29
20.6	1,00	1,00	0,00	0,08
20.7	1,00	1,00	0,00	0,26
20.8	1,00	1,00	0,00	0,61
20.9	1,00	1,00	0,00	0,00
20.10	1,00	1,00	0,00	0,98
20.11	1,00	1,00	0,00	0,00
20.12	1,00	1,00	0,00	0,00
25.1	1,00	1,00	0,00	0,59
25.2	1,00	1,00	0,00	0,00
25.3	1,00	1,00	0,00	0,11
25.4	1,00	1,00	0,00	0,00
25.5	1,00	1,00	0,00	0,16
25.6	1,00	1,00	0,00	0,00
25.7	1,00	1,00	0,00	0,69
25.8	1,00	1,00	0,00	1,00
25.9	1,00	1,00	0,00	0,00
25.10	1,00	1,00	0,00	1,00
25.11	1,00	1,00	0,00	1,00
25.12	1,00	1,00	0,00	1,00
30.1	1,00	1,00	0,00	1,00
30.2	1,00	1,00	0,00	0,71
30.3	1,00	1,00	0,00	1,00
30.4	1,00	1,00	0,00	1,00
30.5	1,00	1,00	0,00	1,00
30.6	1,00	1,00	0,00	1,00
30.7	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
30.8	1,00	1,00	0,00	1,00
30.9	1,00	1,00	0,00	0,02
30.10	1,00	1,00	0,00	0,00
30.11	1,00	1,00	0,00	1,00
30.12	1,00	1,00	0,00	0,69
35.1	1,00	1,00	0,00	1,00
35.2	1,00	1,00	0,00	0,01
35.3	1,00	1,00	0,00	1,00
35.4	1,00	1,00	0,00	1,00
35.5	1,00	1,00	0,00	1,00
35.6	1,00	1,00	0,00	1,00
35.7	1,00	1,00	0,00	1,00
35.8	1,00	1,00	0,00	1,00
35.9	1,00	1,00	0,00	1,00
35.10	1,00	1,00	0,00	0,11
35.11	1,00	1,00	0,00	1,00
35.12	1,00	1,00	0,00	1,00
40.1	1,00	1,00	0,00	1,00
40.2	1,00	1,00	0,00	1,00
40.3	1,00	1,00	0,00	1,00
40.4	1,00	1,00	0,00	1,00
40.5	1,00	1,00	0,00	1,00
40.6	1,00	1,00	0,00	0,03
40.7	1,00	1,00	0,00	1,00
40.8	1,00	1,00	0,00	1,00
40.9	1,00	1,00	0,00	1,00
40.10	1,00	1,00	0,00	1,00
40.11	1,00	1,00	0,00	1,00
40.12	1,00	1,00	0,00	1,00
45.1	1,00	1,00	0,00	1,00
45.2	1,00	1,00	0,00	0,97
45.3	1,00	1,00	0,00	1,00



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
45.4	1,00	1,00	0,00	1,00
45.5	1,00	1,00	0,00	1,00
45.6	1,00	1,00	0,00	1,00
45.7	1,00	1,00	0,00	1,00
45.8	1,00	1,00	0,00	1,00
45.9	1,00	1,00	0,00	1,00
45.10	1,00	1,00	0,00	0,03
45.11	1,00	1,00	0,00	1,00
45.12	1,00	1,00	0,00	1,00
50.1	1,00	1,00	0,00	1,00
50.2	1,00	1,00	0,00	1,00
50.3	1,00	1,00	0,00	1,00
50.4	1,00	1,00	0,00	1,00
50.5	1,00	1,00	0,00	1,00
50.6	1,00	1,00	0,00	1,00
50.7	1,00	1,00	0,00	1,00
50.8	1,00	1,00	0,00	1,00
50.9	1,00	1,00	0,00	1,00
50.10	1,00	1,00	0,00	1,00
50.11	1,00	1,00	0,00	1,00
50.12	1,00	1,00	0,00	1,00
75.1	1,00	1,00	0,00	1,00
75.2	1,00	1,00	0,00	1,00
75.3	1,00	1,00	0,00	1,00
75.4	1,00	1,00	0,00	1,00
75.5	1,00	1,00	0,00	1,00
75.6	1,00	1,00	0,00	1,00
75.7	1,00	1,00	0,00	1,00
75.8	1,00	1,00	0,00	1,00
75.9	1,00	1,00	0,00	1,00
75.10	1,00	1,00	0,00	1,00
75.11	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
75.12	1,00	1,00	0,00	1,00
100.1	1,00	1,00	0,00	1,00
100.2	1,00	1,00	0,00	1,00
100.3	1,00	1,00	0,00	1,00
100.4	1,00	1,00	0,00	1,00
100.5	1,00	1,00	0,00	1,00
100.6	1,00	1,00	0,00	1,00
100.7	1,00	1,00	0,00	1,00
100.8	1,00	1,00	0,00	1,00
100.9	1,00	1,00	0,00	1,00
100.10	1,00	1,00	0,00	1,00
100.11	1,00	1,00	0,00	1,00
100.12	1,00	1,00	0,00	1,00
125.1	1,00	1,00	0,00	1,00
125.2	1,00	1,00	0,00	1,00
125.3	1,00	1,00	0,00	1,00
125.4	1,00	1,00	0,00	1,00
125.5	1,00	1,00	0,00	1,00
125.6	1,00	1,00	0,00	1,00
125.7	1,00	1,00	0,00	1,00
125.8	1,00	1,00	0,00	1,00
125.9	1,00	1,00	0,00	1,00
125.10	1,00	1,00	0,00	1,00
125.11	1,00	1,00	0,00	1,00
125.12	1,00	1,00	0,00	1,00
150.1	1,00	1,00	0,00	1,00
150.2	1,00	1,00	0,00	1,00
150.3	1,00	1,00	0,00	1,00
150.4	1,00	1,00	0,00	1,00
150.5	1,00	1,00	0,00	1,00
150.6	1,00	1,00	0,00	1,00
150.7	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>AT</b>
150.8	1,00	1,00	0,00	1,00
175.1	1,00	1,00	0,00	1,00
175.2	1,00	1,00	0,00	1,00
175.3	1,00	1,00	0,00	1,00
175.4	1,00	1,00	0,00	1,00
175.5	1,00	1,00	0,00	1,00
175.6	1,00	1,00	0,00	1,00
175.7	1,00	1,00	0,00	1,00
175.8	1,00	1,00	0,00	1,00

Tabela 23:  $p$ -valores em favor do NSTA para instâncias OP1

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>AT</b>
6.1	0,00	1,00	1,00	1,00
6.2	0,00	1,00	0,99	1,00
6.3	0,03	0,78	0,98	1,00
6.4	0,00	0,99	0,26	1,00
6.5	0,84	1,00	1,00	1,00
6.6	0,00	0,98	0,81	1,00
6.7	0,69	1,00	1,00	1,00
6.8	0,59	1,00	1,00	1,00
6.9	0,00	0,50	0,50	0,50
6.10	0,00	0,95	1,00	0,99
7.1	0,00	0,01	0,00	0,92
7.2	0,05	1,00	1,00	1,00
7.3	0,00	0,94	0,16	1,00
7.4	0,43	1,00	0,99	1,00
7.5	0,80	1,00	0,77	1,00
7.6	0,00	1,00	1,00	1,00
7.7	0,00	1,00	1,00	1,00
7.8	0,00	0,59	0,00	1,00
7.9	0,15	1,00	1,00	1,00
7.10	0,00	0,98	1,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
8.1	0,99	1,00	1,00	1,00
8.2	0,93	1,00	0,00	1,00
8.3	1,00	1,00	1,00	1,00
8.4	1,00	1,00	1,00	1,00
8.5	0,01	1,00	0,98	1,00
8.6	0,98	1,00	0,00	1,00
8.7	0,37	1,00	1,00	1,00
8.8	1,00	1,00	1,00	1,00
8.9	0,00	0,15	0,32	1,00
8.10	0,00	1,00	0,00	1,00
9.1	0,00	1,00	0,09	1,00
9.2	1,00	1,00	1,00	1,00
9.3	1,00	1,00	0,43	1,00
9.4	0,00	1,00	0,96	1,00
9.5	1,00	1,00	1,00	1,00
9.6	1,00	1,00	1,00	1,00
9.7	0,95	1,00	0,00	1,00
9.8	0,00	1,00	0,19	1,00
9.9	1,00	1,00	1,00	1,00
9.10	0,00	0,00	0,00	1,00
10.1	1,00	1,00	1,00	1,00
10.2	0,00	1,00	0,00	1,00
10.3	1,00	1,00	1,00	1,00
10.4	0,97	1,00	0,00	0,24
10.5	0,01	1,00	0,00	0,94
10.6	0,00	1,00	0,00	0,88
10.7	0,55	1,00	0,00	1,00
10.8	1,00	1,00	0,00	1,00
10.9	1,00	1,00	1,00	1,00
10.10	1,00	1,00	0,66	1,00
11.1	1,00	1,00	0,02	0,63
11.2	0,97	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
11.3	1,00	1,00	0,00	1,00
11.4	0,00	1,00	0,00	0,73
11.5	1,00	1,00	0,00	0,99
11.6	0,00	1,00	0,00	0,98
11.7	1,00	1,00	0,99	1,00
11.8	1,00	1,00	1,00	1,00
11.9	1,00	1,00	0,00	1,00
11.10	1,00	1,00	0,01	1,00
12.1	1,00	1,00	0,00	0,93
12.2	1,00	1,00	0,87	1,00
12.3	1,00	1,00	0,00	0,00
12.4	1,00	1,00	0,00	1,00
12.5	1,00	1,00	0,01	1,00
12.6	1,00	1,00	0,00	1,00
12.7	0,98	1,00	0,00	0,78
12.8	1,00	1,00	0,00	0,36
12.9	1,00	1,00	0,00	0,12
12.10	1,00	1,00	0,00	0,75
13.1	0,66	1,00	0,00	1,00
13.2	1,00	1,00	1,00	1,00
13.3	0,94	1,00	0,00	1,00
13.4	1,00	1,00	0,00	0,01
13.5	1,00	1,00	0,00	0,00
13.6	1,00	1,00	0,00	1,00
13.7	1,00	1,00	0,00	0,00
13.8	1,00	1,00	0,00	0,78
13.9	1,00	1,00	0,00	0,85
13.10	1,00	1,00	0,00	1,00
14.1	1,00	1,00	0,00	0,00
14.2	1,00	1,00	0,00	1,00
14.3	1,00	1,00	0,00	0,05
14.4	0,00	0,52	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
14.5	1,00	1,00	0,00	0,02
14.6	1,00	1,00	0,00	0,00
14.7	1,00	1,00	0,00	0,83
14.8	1,00	1,00	0,00	0,72
14.9	1,00	1,00	0,00	0,08
14.10	1,00	1,00	0,00	0,00
15.1	1,00	1,00	0,00	0,05
15.2	1,00	1,00	0,00	1,00
15.3	1,00	1,00	0,00	0,00
15.4	1,00	1,00	0,00	0,00
15.5	1,00	1,00	0,00	1,00
15.6	1,00	1,00	0,00	0,79
15.7	1,00	1,00	0,00	0,18
15.8	1,00	1,00	0,00	0,85
15.9	1,00	1,00	0,00	0,31
15.10	1,00	1,00	0,00	0,00
16.1	1,00	1,00	0,00	0,00
16.2	1,00	1,00	0,00	0,00
16.3	1,00	1,00	0,00	0,10
16.4	1,00	1,00	0,00	0,55
16.5	1,00	1,00	0,00	0,14
16.6	1,00	1,00	0,00	0,23
16.7	0,84	1,00	0,00	0,00
16.8	1,00	1,00	0,00	0,00
16.9	1,00	1,00	0,00	0,00
16.10	0,99	1,00	0,00	0,00
17.1	1,00	1,00	0,00	0,84
17.2	1,00	1,00	0,00	0,00
17.3	1,00	1,00	0,00	1,00
17.4	1,00	1,00	0,00	0,07
17.5	1,00	1,00	0,00	1,00
17.6	1,00	1,00	0,00	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
17.7	1,00	1,00	0,00	0,01
17.8	1,00	1,00	0,00	0,68
17.9	1,00	1,00	0,00	0,00
17.10	1,00	1,00	0,00	0,02
18.1	1,00	1,00	0,00	0,00
18.2	1,00	1,00	0,00	1,00
18.3	1,00	1,00	0,00	1,00
18.4	1,00	1,00	0,00	1,00
18.5	1,00	1,00	0,00	1,00
18.6	1,00	1,00	0,00	0,00
18.7	1,00	1,00	0,00	0,01
18.8	1,00	1,00	0,00	0,00
18.9	1,00	1,00	0,00	0,15
18.10	1,00	1,00	0,00	0,41
20.1	1,00	1,00	0,00	1,00
20.2	1,00	1,00	0,00	0,94
20.3	1,00	1,00	0,00	0,00
20.4	1,00	1,00	0,00	0,00
20.5	1,00	1,00	0,00	0,01
20.6	1,00	1,00	0,00	0,00
20.7	1,00	1,00	0,00	0,00
20.8	1,00	1,00	0,00	0,60
20.9	1,00	1,00	0,00	1,00
20.10	1,00	1,00	0,00	1,00
30.1	1,00	1,00	0,00	1,00
30.2	1,00	1,00	0,00	0,99
30.3	1,00	1,00	0,00	1,00
30.4	1,00	1,00	0,00	1,00
30.5	1,00	1,00	0,00	1,00
30.6	1,00	1,00	0,00	1,00
30.7	1,00	1,00	0,00	0,00
30.8	1,00	1,00	0,00	0,92

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
30.9	1,00	1,00	0,00	1,00
30.10	1,00	1,00	0,00	1,00
50.1	1,00	1,00	0,00	1,00
50.2	1,00	1,00	0,00	1,00
50.3	1,00	1,00	0,00	1,00
50.4	1,00	1,00	0,00	1,00
50.5	1,00	1,00	0,00	1,00
50.6	1,00	1,00	0,00	1,00
50.7	1,00	1,00	0,00	1,00
50.8	1,00	1,00	0,00	1,00
50.9	1,00	1,00	0,00	1,00
50.10	1,00	1,00	0,00	1,00

Tabela 24:  $p$ -valores em favor do NSTA para instâncias OP2

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
6.1	0,00	1,00	1,00	1,00
6.2	0,00	0,98	0,98	0,98
6.3	0,00	0,50	0,50	0,50
6.4	0,00	0,16	0,50	0,50
6.5	0,00	0,49	1,00	0,99
6.6	0,00	0,95	0,47	1,00
6.7	0,00	0,93	1,00	1,00
6.8	0,00	0,80	1,00	1,00
6.9	0,00	0,50	0,50	0,50
6.10	0,08	0,50	0,50	0,50
7.1	0,00	1,00	1,00	1,00
7.2	0,00	0,18	0,00	1,00
7.3	0,00	1,00	1,00	1,00
7.4	0,00	1,00	1,00	1,00
7.5	0,00	1,00	1,00	1,00
7.6	0,00	1,00	0,24	1,00
7.7	0,00	0,08	0,01	0,50



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
7.8	0,00	0,62	0,00	0,37
7.9	0,00	0,00	0,06	0,58
7.10	0,00	1,00	0,00	0,78
8.1	0,00	1,00	1,00	1,00
8.2	0,00	0,98	0,00	1,00
8.3	0,00	1,00	1,00	1,00
8.4	0,00	1,00	1,00	1,00
8.5	0,02	0,99	1,00	1,00
8.6	0,00	0,91	0,00	0,51
8.7	0,00	0,66	0,00	0,58
8.8	0,58	1,00	1,00	1,00
8.9	0,00	1,00	1,00	0,00
8.10	0,00	0,01	0,00	0,62
9.1	1,00	1,00	1,00	1,00
9.2	1,00	1,00	1,00	1,00
9.3	0,74	1,00	1,00	1,00
9.4	0,03	1,00	0,00	0,81
9.5	1,00	1,00	0,98	1,00
9.6	0,01	0,99	1,00	0,85
9.7	0,04	1,00	0,96	1,00
9.8	0,88	1,00	0,28	0,84
9.9	0,72	1,00	0,04	0,66
9.10	0,74	1,00	0,01	1,00
10.1	0,00	1,00	0,00	0,41
10.2	0,99	1,00	0,98	1,00
10.3	1,00	1,00	0,79	1,00
10.4	0,38	1,00	0,70	0,27
10.5	1,00	1,00	0,01	1,00
10.6	0,41	1,00	0,00	0,58
10.7	1,00	1,00	0,09	1,00
10.8	0,97	1,00	0,00	0,02
10.9	0,90	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
10.10	0,83	1,00	0,14	1,00
11.1	1,00	1,00	0,00	0,95
11.2	0,24	1,00	0,00	1,00
11.3	1,00	1,00	0,00	1,00
11.4	0,05	1,00	0,00	0,00
11.5	0,12	0,96	0,00	0,99
11.6	1,00	1,00	0,00	1,00
11.7	1,00	1,00	0,00	0,73
11.8	1,00	1,00	0,00	1,00
11.9	1,00	1,00	0,00	0,94
11.10	1,00	1,00	0,00	1,00
12.1	1,00	1,00	0,00	0,77
12.2	0,96	1,00	0,00	0,99
12.3	1,00	1,00	0,00	0,99
12.4	1,00	1,00	0,00	0,99
12.5	1,00	1,00	0,00	0,99
12.6	0,00	1,00	0,00	0,00
12.7	1,00	1,00	0,53	1,00
12.8	1,00	1,00	0,90	1,00
12.9	1,00	1,00	0,00	0,99
12.10	1,00	1,00	0,00	1,00
13.1	0,90	1,00	0,00	1,00
13.2	0,00	0,95	0,00	0,00
13.3	0,99	1,00	1,00	1,00
13.4	1,00	0,00	0,00	0,00
13.5	0,00	1,00	0,00	0,00
13.6	1,00	1,00	0,00	0,93
13.7	1,00	1,00	0,00	0,35
13.8	1,00	1,00	1,00	1,00
13.9	1,00	1,00	0,07	1,00
13.10	1,00	1,00	0,00	0,23
14.1	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
14.2	0,87	1,00	0,00	1,00
14.3	1,00	1,00	0,00	0,19
14.4	1,00	1,00	0,00	1,00
14.5	1,00	1,00	0,00	1,00
14.6	1,00	1,00	0,00	0,75
14.7	0,56	1,00	0,00	0,00
14.8	0,89	1,00	0,00	0,00
14.9	1,00	1,00	0,00	0,04
14.10	1,00	1,00	0,00	0,46
15.1	0,33	1,00	0,00	0,03
15.2	0,99	1,00	0,00	0,00
15.3	0,97	1,00	0,00	0,00
15.4	0,86	1,00	0,00	0,00
15.5	1,00	1,00	0,00	0,07
15.6	0,16	1,00	0,00	0,44
15.7	1,00	1,00	0,00	0,01
15.8	1,00	1,00	0,00	0,95
15.9	1,00	1,00	1,00	1,00
15.10	1,00	1,00	0,00	1,00
16.1	0,54	1,00	0,00	0,00
16.2	1,00	1,00	0,00	0,00
16.3	1,00	1,00	0,00	0,00
16.4	0,00	1,00	0,00	0,00
16.5	0,00	1,00	0,00	0,38
16.6	1,00	1,00	0,00	0,09
16.7	1,00	1,00	0,00	0,93
16.8	1,00	1,00	0,00	0,24
16.9	0,00	1,00	0,00	0,00
16.10	0,00	1,00	0,00	0,41
17.1	1,00	1,00	0,00	1,00
17.2	1,00	1,00	0,00	0,00
17.3	1,00	1,00	0,00	0,07

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
17.4	1,00	1,00	0,00	0,00
17.5	1,00	1,00	0,00	0,00
17.6	0,73	1,00	0,00	0,00
17.7	0,00	1,00	0,00	0,00
17.8	1,00	1,00	0,00	0,22
17.9	1,00	1,00	0,00	0,32
17.10	0,00	1,00	0,57	0,14
18.1	1,00	1,00	0,01	1,00
18.2	1,00	1,00	0,00	0,98
18.3	1,00	1,00	0,00	0,01
18.4	1,00	1,00	0,00	0,26
18.5	1,00	1,00	0,04	0,00
18.6	1,00	1,00	0,00	0,28
18.7	1,00	1,00	0,00	0,00
18.8	1,00	1,00	0,00	0,00
18.9	1,00	1,00	0,00	0,00
18.10	1,00	1,00	0,00	0,00
20.1	1,00	1,00	0,00	0,97
20.2	1,00	1,00	0,00	0,00
20.3	1,00	1,00	0,00	0,98
20.4	1,00	1,00	0,00	0,00
20.5	1,00	1,00	0,00	0,17
20.6	1,00	1,00	0,00	0,02
20.7	1,00	1,00	0,00	0,90
20.8	1,00	1,00	0,00	0,00
20.9	1,00	1,00	0,00	0,00
20.10	1,00	1,00	0,00	0,00
30.1	1,00	1,00	0,00	0,07
30.2	1,00	1,00	0,00	0,00
30.3	1,00	1,00	0,00	0,00
30.4	1,00	1,00	0,00	0,00
30.5	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
30.6	1,00	1,00	0,00	0,00
30.7	1,00	1,00	0,00	0,00
30.8	1,00	1,00	0,00	0,01
30.9	1,00	1,00	0,00	0,01
30.10	1,00	1,00	0,00	0,80
50.1	1,00	1,00	0,00	0,00
50.2	1,00	1,00	0,00	0,00
50.3	1,00	1,00	0,00	0,00
50.4	1,00	1,00	0,00	0,00
50.5	1,00	1,00	0,00	0,00
50.6	1,00	1,00	0,00	0,00
50.7	1,00	1,00	0,00	0,00
50.8	1,00	1,00	0,00	0,00
50.9	1,00	1,00	0,00	0,00
50.10	1,00	1,00	0,00	0,00

Tabela 25:  $p$ -valores em favor do NSTA para instâncias OP3

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
6.1	0,00	1,00	1,00	1,00
6.2	0,00	1,00	0,01	1,00
6.3	0,00	0,91	0,54	1,00
6.4	0,00	1,00	0,99	1,00
6.5	0,00	1,00	0,99	1,00
6.6	0,00	0,99	1,00	1,00
6.7	0,00	1,00	1,00	1,00
6.8	0,00	1,00	1,00	1,00
6.9	0,00	1,00	0,76	1,00
6.10	0,00	1,00	1,00	1,00
7.1	0,00	1,00	0,23	1,00
7.2	0,88	1,00	1,00	1,00
7.3	0,00	1,00	0,97	1,00
7.4	0,00	1,00	0,99	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
7.5	0,00	1,00	0,01	1,00
7.6	0,00	1,00	0,00	1,00
7.7	0,04	1,00	0,31	1,00
7.8	0,00	0,00	0,00	0,90
7.9	0,00	1,00	1,00	1,00
7.10	0,00	1,00	0,00	1,00
8.1	0,00	1,00	0,00	1,00
8.2	0,00	1,00	0,00	1,00
8.3	0,09	1,00	0,00	1,00
8.4	0,00	1,00	0,00	0,99
8.5	0,02	1,00	0,00	1,00
8.6	0,00	1,00	0,07	1,00
8.7	0,00	1,00	0,00	0,92
8.8	0,00	1,00	0,00	1,00
8.9	0,01	1,00	0,01	1,00
8.10	0,00	0,63	0,00	1,00
9.1	0,01	1,00	0,00	0,92
9.2	0,00	1,00	0,00	0,91
9.3	0,12	1,00	0,00	0,91
9.4	0,18	1,00	0,00	1,00
9.5	0,00	0,89	0,00	1,00
9.6	0,00	1,00	0,00	1,00
9.7	0,01	1,00	0,00	1,00
9.8	0,10	0,96	0,00	1,00
9.9	0,01	1,00	0,00	1,00
9.10	0,00	0,87	0,00	1,00
10.1	0,01	1,00	0,00	0,96
10.2	0,00	1,00	0,00	0,83
10.3	0,99	1,00	0,01	1,00
10.4	0,00	0,25	0,00	1,00
10.5	0,12	1,00	0,00	1,00
10.6	1,00	1,00	0,02	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
10.7	0,00	1,00	0,00	1,00
10.8	1,00	1,00	0,00	0,99
10.9	0,92	1,00	0,00	1,00
10.10	1,00	1,00	0,00	1,00
11.1	0,97	1,00	0,00	1,00
11.2	0,00	1,00	0,00	0,77
11.3	0,93	1,00	0,00	1,00
11.4	0,96	1,00	0,00	0,88
11.5	0,00	0,13	0,00	0,98
11.6	0,00	0,02	0,00	0,96
11.7	0,12	0,29	0,00	1,00
11.8	1,00	1,00	0,00	1,00
11.9	1,00	1,00	0,00	1,00
11.10	1,00	1,00	0,00	1,00
12.1	1,00	1,00	0,00	1,00
12.2	1,00	1,00	0,87	1,00
12.3	1,00	1,00	0,00	1,00
12.4	1,00	1,00	0,40	1,00
12.5	1,00	1,00	1,00	1,00
12.6	1,00	1,00	0,04	1,00
12.7	1,00	1,00	1,00	1,00
12.8	1,00	1,00	0,00	1,00
12.9	1,00	1,00	0,01	1,00
12.10	1,00	1,00	0,02	1,00
13.1	1,00	1,00	0,00	1,00
13.2	1,00	1,00	1,00	1,00
13.3	1,00	1,00	0,03	1,00
13.4	1,00	1,00	0,00	1,00
13.5	0,97	0,96	0,00	1,00
13.6	0,60	0,97	0,00	1,00
13.7	1,00	1,00	0,00	1,00
13.8	1,00	1,00	0,94	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
13.9	1,00	1,00	0,00	1,00
13.10	1,00	1,00	0,00	1,00
14.1	1,00	1,00	0,00	1,00
14.2	0,65	1,00	0,00	1,00
14.3	0,99	1,00	0,00	0,97
14.4	1,00	1,00	0,00	1,00
14.5	1,00	1,00	0,34	1,00
14.6	0,02	0,88	0,00	1,00
14.7	1,00	1,00	0,00	1,00
14.8	1,00	1,00	1,00	1,00
14.9	0,82	1,00	0,00	0,99
14.10	1,00	1,00	0,00	1,00
15.1	1,00	0,98	0,00	1,00
15.2	1,00	1,00	0,00	1,00
15.3	1,00	1,00	0,00	1,00
15.4	1,00	1,00	0,00	1,00
15.5	1,00	1,00	0,00	1,00
15.6	0,40	0,62	0,00	0,67
15.7	1,00	1,00	0,00	1,00
15.8	1,00	1,00	1,00	1,00
15.9	1,00	1,00	0,00	1,00
15.10	1,00	1,00	0,11	1,00
16.1	1,00	0,85	0,00	0,95
16.2	1,00	0,95	0,00	1,00
16.3	1,00	1,00	0,00	1,00
16.4	1,00	0,58	0,00	0,50
16.5	1,00	1,00	0,98	1,00
16.6	1,00	1,00	0,00	1,00
16.7	1,00	1,00	1,00	1,00
16.8	1,00	1,00	0,00	1,00
16.9	1,00	1,00	0,00	1,00
16.10	1,00	1,00	0,00	1,00



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>AT</b>
17.1	1,00	0,38	0,00	1,00
17.2	1,00	0,99	0,00	1,00
17.3	1,00	1,00	0,00	1,00
17.4	1,00	1,00	0,96	1,00
17.5	1,00	1,00	0,00	1,00
17.6	1,00	1,00	0,00	1,00
17.7	1,00	1,00	0,98	1,00
17.8	1,00	1,00	0,00	0,99
17.9	1,00	1,00	1,00	1,00
17.10	1,00	1,00	0,00	1,00
18.1	1,00	1,00	1,00	1,00
18.2	1,00	1,00	0,74	1,00
18.3	1,00	1,00	0,00	1,00
18.4	1,00	0,00	0,00	1,00
18.5	1,00	1,00	0,00	1,00
18.6	0,84	0,00	0,00	0,01
18.7	1,00	0,00	0,00	0,88
18.8	1,00	1,00	1,00	1,00
18.9	1,00	1,00	0,00	1,00
18.10	1,00	1,00	1,00	1,00
20.1	0,00	1,00	1,00	1,00
20.2	0,00	0,96	0,96	0,85
20.3	0,00	0,98	0,98	0,98
20.4	0,00	0,98	0,98	0,98
20.5	0,00	0,60	1,00	1,00
20.6	0,00	0,41	0,05	1,00
20.7	0,00	0,72	0,99	1,00
20.8	0,00	1,00	1,00	1,00
20.9	0,00	0,50	0,50	0,50
20.10	0,01	0,50	0,50	0,50
30.1	1,00	1,00	0,00	1,00
30.2	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>AT</b>
30.3	1,00	1,00	0,00	1,00
30.4	0,93	0,00	0,00	0,90
30.5	1,00	1,00	0,00	1,00
30.6	1,00	1,00	0,02	1,00
30.7	1,00	0,94	0,00	0,01
30.8	1,00	1,00	0,00	1,00
30.9	1,00	1,00	0,00	0,01
30.10	1,00	1,00	0,00	1,00
50.1	1,00	1,00	0,00	1,00
50.2	1,00	1,00	0,00	1,00
50.3	1,00	1,00	0,00	1,00
50.4	1,00	1,00	0,00	1,00
50.5	1,00	0,00	0,00	1,00
50.6	1,00	1,00	0,00	0,34
50.7	1,00	1,00	0,00	0,98
50.8	1,00	1,00	0,00	0,95
50.9	1,00	1,00	0,00	1,00
50.10	1,00	1,00	0,00	1,00

Os  $p$ -valores exibidos nas Tabelas de 22 até 25 revelam indícios estatísticos significativos de que o algoritmo NSTA produz conjuntos de aproximação de melhor qualidade que o NSGA-II, no que se refere ao indicador hipervolume. Quando comparado com as outras abordagens, contudo, os  $p$ -valores apontam para o fato de que o NSTA produz conjuntos de aproximação de qualidade inferior, de acordo com o hipervolume.

Uma das metaheurísticas para a qual o NSTA perde é o AT. Tendo em vista que o NSTA é uma hibridização de NSGA-II com Algoritmos Transgenéticos, esse resultado era, de certa forma, esperado, pois indica a possibilidade de que o algoritmo híbrido compartilhe não apenas as qualidades, mas também os defeitos das duas abordagens hibridizadas. Por outro lado, o comportamento do NSTA em dois contextos específicos dos experimentos relatados também sugere a possibilidade de que, ao mesclar diferentes estratégias de busca por meio de hibridização, surjam resultados melhores que as abordagens originais independentes. Os dois contextos são: (i) instâncias CP de até 25 nós, em que o NSTA ganha do AT em 18 e perde em 15 casos; e (ii) instâncias OP2, em que o NSTA vence o AT para quase todas as instâncias com mais de 14 vértices. Isso demonstra a competitividade

dos métodos híbridos e reforça a necessidade de investigá-la.

As Tabelas 26 a 29 exibem os  $p$ -valores da comparação entre o AT e os outros algoritmos no que se refere ao indicador hipervolume, para os quatro grupos de instâncias adotados. Os  $p$ -valores exibidos são em favor do AT. O rótulo nas colunas indica o algoritmo de comparação e o rótulo nas linhas identifica a instância.

Tabela 26:  $p$ -valores em favor do AT para instâncias CP

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
10.1	0,00	0,50	0,50	0,50
10.2	0,00	0,50	0,50	0,50
10.3	0,02	0,50	0,50	0,50
10.4	0,04	0,50	0,50	0,50
10.5	0,00	0,50	0,50	0,50
10.6	0,09	1,00	0,00	0,00
10.7	0,00	0,00	1,00	0,00
10.8	0,04	1,00	0,06	0,00
10.9	0,38	1,00	0,00	0,00
10.10	0,98	1,00	0,00	0,00
10.11	0,05	1,00	0,00	0,01
10.12	1,00	1,00	0,00	0,00
15.1	0,00	1,00	0,00	0,95
15.2	0,74	1,00	0,00	1,00
15.3	0,00	1,00	0,00	0,71
15.4	1,00	1,00	0,00	1,00
15.5	0,89	0,99	0,00	0,00
15.6	0,18	1,00	0,00	0,00
15.7	0,82	1,00	0,00	0,97
15.8	0,40	1,00	0,00	0,00
15.9	0,78	1,00	0,00	1,00
15.10	1,00	1,00	0,00	1,00
15.11	1,00	1,00	0,00	1,00
15.12	1,00	1,00	0,00	1,00
20.1	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
20.2	1,00	1,00	0,00	1,00
20.3	1,00	1,00	0,00	1,00
20.4	1,00	1,00	0,00	1,00
20.5	1,00	1,00	0,00	0,71
20.6	1,00	1,00	0,00	0,92
20.7	1,00	1,00	0,00	0,74
20.8	1,00	1,00	0,00	0,39
20.9	1,00	1,00	0,00	1,00
20.10	1,00	1,00	0,00	0,02
20.11	1,00	1,00	0,00	1,00
20.12	1,00	1,00	0,00	1,00
25.1	1,00	1,00	0,00	0,41
25.2	1,00	1,00	0,00	1,00
25.3	1,00	1,00	0,00	0,89
25.4	1,00	1,00	0,00	1,00
25.5	1,00	1,00	0,00	0,84
25.6	1,00	1,00	0,00	1,00
25.7	1,00	1,00	0,00	0,31
25.8	1,00	1,00	0,00	0,00
25.9	1,00	1,00	0,00	1,00
25.10	1,00	1,00	0,00	0,00
25.11	1,00	1,00	0,00	0,00
25.12	1,00	1,00	0,00	0,00
30.1	1,00	1,00	0,00	0,00
30.2	1,00	1,00	0,00	0,29
30.3	1,00	1,00	0,00	0,00
30.4	1,00	1,00	0,00	0,00
30.5	1,00	1,00	0,00	0,00
30.6	1,00	1,00	0,00	0,00
30.7	1,00	1,00	0,00	0,00
30.8	1,00	1,00	0,00	0,00
30.9	1,00	1,00	0,00	0,98

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
30.10	1,00	1,00	0,00	1,00
30.11	1,00	1,00	0,00	0,00
30.12	1,00	1,00	0,00	0,31
35.1	1,00	1,00	0,00	0,00
35.2	1,00	1,00	0,00	0,99
35.3	1,00	1,00	0,00	0,00
35.4	1,00	1,00	0,00	0,00
35.5	1,00	1,00	0,00	0,00
35.6	1,00	1,00	0,00	0,00
35.7	1,00	1,00	0,00	0,00
35.8	1,00	1,00	0,00	0,00
35.9	1,00	1,00	0,00	0,00
35.10	1,00	1,00	0,00	0,89
35.11	1,00	1,00	0,00	0,00
35.12	1,00	1,00	0,00	0,00
40.1	1,00	1,00	0,00	0,00
40.2	1,00	1,00	0,00	0,00
40.3	1,00	1,00	0,00	0,00
40.4	1,00	1,00	0,00	0,00
40.5	1,00	1,00	0,00	0,00
40.6	1,00	1,00	0,00	0,97
40.7	1,00	1,00	0,00	0,00
40.8	1,00	1,00	0,00	0,00
40.9	1,00	1,00	0,00	0,00
40.10	1,00	1,00	0,00	0,00
40.11	1,00	1,00	0,00	0,00
40.12	1,00	1,00	0,00	0,00
45.1	1,00	1,00	0,00	0,00
45.2	1,00	1,00	0,00	0,03
45.3	1,00	1,00	0,00	0,00
45.4	1,00	1,00	0,00	0,00
45.5	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
45.6	1,00	1,00	0,00	0,00
45.7	1,00	1,00	0,00	0,00
45.8	1,00	1,00	0,00	0,00
45.9	1,00	1,00	0,00	0,00
45.10	1,00	1,00	0,00	0,97
45.11	1,00	1,00	0,00	0,00
45.12	1,00	1,00	0,00	0,00
50.1	1,00	1,00	0,00	0,00
50.2	1,00	1,00	0,00	0,00
50.3	1,00	1,00	0,00	0,00
50.4	1,00	1,00	0,00	0,00
50.5	1,00	1,00	0,00	0,00
50.6	1,00	1,00	0,00	0,00
50.7	1,00	1,00	0,00	0,00
50.8	1,00	1,00	0,00	0,00
50.9	1,00	1,00	0,00	0,00
50.10	1,00	1,00	0,00	0,00
50.11	1,00	1,00	0,00	0,00
50.12	1,00	1,00	0,00	0,00
75.1	1,00	1,00	0,00	0,00
75.2	1,00	1,00	0,00	0,00
75.3	1,00	1,00	0,00	0,00
75.4	1,00	1,00	0,00	0,00
75.5	1,00	1,00	0,00	0,00
75.6	1,00	1,00	0,00	0,00
75.7	1,00	1,00	0,00	0,00
75.8	1,00	1,00	0,00	0,00
75.9	1,00	1,00	0,00	0,00
75.10	1,00	1,00	0,00	0,00
75.11	1,00	1,00	0,00	0,00
75.12	1,00	1,00	0,00	0,00
100.1	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
100.2	1,00	1,00	0,00	0,00
100.3	1,00	1,00	0,00	0,00
100.4	1,00	1,00	0,00	0,00
100.5	1,00	1,00	0,00	0,00
100.6	1,00	1,00	0,00	0,00
100.7	1,00	1,00	0,00	0,00
100.8	1,00	1,00	0,00	0,00
100.9	1,00	1,00	0,00	0,00
100.10	1,00	1,00	0,00	0,00
100.11	1,00	1,00	0,00	0,00
100.12	1,00	1,00	0,00	0,00
125.1	1,00	1,00	0,00	0,00
125.2	1,00	1,00	0,00	0,00
125.3	1,00	1,00	0,00	0,00
125.4	1,00	1,00	0,00	0,00
125.5	1,00	1,00	0,00	0,00
125.6	1,00	1,00	0,00	0,00
125.7	1,00	1,00	0,00	0,00
125.8	1,00	1,00	0,00	0,00
125.9	1,00	1,00	0,00	0,00
125.10	1,00	1,00	0,00	0,00
125.11	1,00	1,00	0,00	0,00
125.12	1,00	1,00	0,00	0,00
150.1	1,00	1,00	0,00	0,00
150.2	1,00	1,00	0,00	0,00
150.3	1,00	1,00	0,00	0,00
150.4	1,00	1,00	0,00	0,00
150.5	1,00	1,00	0,00	0,00
150.6	1,00	1,00	0,00	0,00
150.7	1,00	1,00	0,00	0,00
150.8	1,00	1,00	0,00	0,00
175.1	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
175.2	1,00	1,00	0,00	0,00
175.3	1,00	1,00	0,00	0,00
175.4	1,00	1,00	0,00	0,00
175.5	1,00	1,00	0,00	0,00
175.6	1,00	1,00	0,00	0,00
175.7	1,00	1,00	0,00	0,00
175.8	1,00	1,00	0,00	0,00

Tabela 27:  $p$ -valores em favor do AT para instâncias OP1

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
6.1	0,00	0,08	0,03	0,00
6.2	0,00	0,00	0,00	0,00
6.3	0,00	0,00	0,00	0,00
6.4	0,00	0,17	0,00	0,00
6.5	0,01	0,50	0,50	0,00
6.6	0,00	0,10	0,00	0,00
6.7	0,00	0,50	0,50	0,00
6.8	0,00	0,50	0,00	0,00
6.9	0,00	0,50	0,50	0,50
6.10	0,00	0,27	0,84	0,01
7.1	0,00	0,00	0,00	0,08
7.2	0,00	0,54	0,00	0,00
7.3	0,00	0,00	0,00	0,00
7.4	0,00	1,00	0,08	0,00
7.5	0,00	1,00	0,00	0,00
7.6	0,00	0,92	0,42	0,00
7.7	0,00	0,92	0,92	0,00
7.8	0,00	0,01	0,00	0,00
7.9	0,00	1,00	0,90	0,00
7.10	0,00	0,98	1,00	0,00
8.1	0,00	1,00	0,99	0,00
8.2	0,00	1,00	0,00	0,00



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
8.3	0,65	1,00	1,00	0,00
8.4	0,01	0,96	0,71	0,00
8.5	0,00	0,95	0,00	0,00
8.6	0,29	1,00	0,00	0,00
8.7	0,00	1,00	0,80	0,00
8.8	0,97	1,00	0,01	0,00
8.9	0,00	0,00	0,00	0,00
8.10	0,00	1,00	0,00	0,00
9.1	0,00	1,00	0,00	0,00
9.2	0,99	1,00	1,00	0,00
9.3	0,81	1,00	0,00	0,00
9.4	0,00	0,12	0,00	0,00
9.5	0,00	0,03	0,00	0,00
9.6	0,00	1,00	0,00	0,00
9.7	0,00	0,99	0,00	0,00
9.8	0,00	0,75	0,00	0,00
9.9	1,00	1,00	1,00	0,00
9.10	0,00	0,00	0,00	0,00
10.1	1,00	1,00	0,30	0,00
10.2	0,00	0,92	0,00	0,00
10.3	1,00	1,00	0,68	0,00
10.4	0,98	1,00	0,00	0,76
10.5	0,01	1,00	0,00	0,06
10.6	0,00	1,00	0,00	0,12
10.7	0,00	1,00	0,00	0,00
10.8	0,59	1,00	0,00	0,00
10.9	1,00	1,00	0,30	0,00
10.10	1,00	1,00	0,00	0,00
11.1	1,00	1,00	0,01	0,37
11.2	0,25	1,00	0,00	0,00
11.3	0,83	1,00	0,00	0,00
11.4	0,00	1,00	0,00	0,27

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
11.5	1,00	1,00	0,00	0,01
11.6	0,00	0,99	0,00	0,02
11.7	0,72	1,00	0,00	0,00
11.8	1,00	1,00	0,00	0,00
11.9	1,00	1,00	0,00	0,00
11.10	1,00	1,00	0,00	0,00
12.1	1,00	1,00	0,00	0,07
12.2	0,89	1,00	0,00	0,00
12.3	1,00	1,00	0,00	1,00
12.4	1,00	1,00	0,00	0,00
12.5	1,00	1,00	0,00	0,00
12.6	0,98	1,00	0,00	0,00
12.7	0,93	1,00	0,00	0,22
12.8	1,00	1,00	0,00	0,64
12.9	1,00	1,00	0,00	0,88
12.10	1,00	1,00	0,00	0,25
13.1	0,58	1,00	0,00	0,00
13.2	1,00	1,00	0,00	0,00
13.3	0,18	1,00	0,00	0,00
13.4	1,00	1,00	0,00	0,99
13.5	1,00	1,00	0,00	1,00
13.6	1,00	1,00	0,00	0,00
13.7	1,00	1,00	0,00	1,00
13.8	1,00	1,00	0,00	0,22
13.9	1,00	1,00	0,00	0,15
13.10	1,00	1,00	0,00	0,00
14.1	1,00	1,00	0,00	1,00
14.2	1,00	1,00	0,00	0,00
14.3	1,00	1,00	0,00	0,95
14.4	0,05	0,94	0,00	1,00
14.5	1,00	1,00	0,00	0,98
14.6	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
14.7	1,00	1,00	0,00	0,17
14.8	1,00	1,00	0,00	0,28
14.9	1,00	1,00	0,00	0,92
14.10	1,00	1,00	0,00	1,00
15.1	1,00	1,00	0,00	0,95
15.2	1,00	1,00	0,00	0,00
15.3	1,00	1,00	0,00	1,00
15.4	1,00	1,00	0,00	1,00
15.5	1,00	1,00	0,00	0,00
15.6	1,00	1,00	0,00	0,21
15.7	1,00	1,00	0,00	0,82
15.8	1,00	1,00	0,00	0,15
15.9	1,00	1,00	0,00	0,69
15.10	1,00	1,00	0,00	1,00
16.1	1,00	1,00	0,00	1,00
16.2	1,00	1,00	0,00	1,00
16.3	1,00	1,00	0,00	0,90
16.4	1,00	1,00	0,00	0,45
16.5	1,00	1,00	0,00	0,86
16.6	1,00	1,00	0,00	0,77
16.7	1,00	1,00	0,00	1,00
16.8	1,00	1,00	0,00	1,00
16.9	1,00	1,00	0,00	1,00
16.10	1,00	1,00	0,00	1,00
17.1	1,00	1,00	0,00	0,16
17.2	1,00	1,00	0,00	1,00
17.3	1,00	1,00	0,00	0,00
17.4	1,00	1,00	0,00	0,93
17.5	1,00	1,00	0,00	0,00
17.6	1,00	1,00	0,00	0,99
17.7	1,00	1,00	0,00	0,99
17.8	1,00	1,00	0,00	0,32

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
17.9	1,00	1,00	0,00	1,00
17.10	1,00	1,00	0,00	0,98
18.1	1,00	1,00	0,00	1,00
18.2	1,00	1,00	0,00	0,00
18.3	1,00	1,00	0,00	0,00
18.4	1,00	1,00	0,00	0,00
18.5	1,00	1,00	0,00	0,00
18.6	1,00	1,00	0,00	1,00
18.7	1,00	1,00	0,00	0,99
18.8	1,00	1,00	0,00	1,00
18.9	1,00	1,00	0,00	0,85
18.10	1,00	1,00	0,00	0,59
20.1	1,00	1,00	0,00	0,00
20.2	1,00	1,00	0,00	0,06
20.3	1,00	1,00	0,00	1,00
20.4	1,00	1,00	0,00	1,00
20.5	1,00	1,00	0,00	0,99
20.6	1,00	1,00	0,00	1,00
20.7	1,00	1,00	0,00	1,00
20.8	1,00	1,00	0,00	0,40
20.9	1,00	1,00	0,00	0,00
20.10	1,00	1,00	0,00	0,00
30.1	1,00	1,00	0,00	0,00
30.2	1,00	1,00	0,00	0,01
30.3	1,00	1,00	0,00	0,00
30.4	1,00	1,00	0,00	0,00
30.5	1,00	1,00	0,00	0,00
30.6	1,00	1,00	0,00	0,00
30.7	1,00	1,00	0,00	1,00
30.8	1,00	1,00	0,00	0,08
30.9	1,00	1,00	0,00	0,00
30.10	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
50.1	1,00	1,00	0,00	0,00
50.2	1,00	1,00	0,00	0,00
50.3	1,00	1,00	0,00	0,00
50.4	1,00	1,00	0,00	0,00
50.5	1,00	1,00	0,00	0,00
50.6	1,00	1,00	0,00	0,00
50.7	1,00	1,00	0,00	0,00
50.8	1,00	1,00	0,00	0,00
50.9	1,00	1,00	0,00	0,00
50.10	1,00	1,00	0,00	0,00

Tabela 28:  $p$ -valores em favor do AT para instâncias OP2

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
6.1	0,00	0,32	0,71	0,00
6.2	0,00	0,50	0,50	0,02
6.3	0,00	0,50	0,50	0,50
6.4	0,00	0,16	0,50	0,50
6.5	0,00	0,00	0,92	0,01
6.6	0,00	0,00	0,00	0,00
6.7	0,00	0,00	0,06	0,00
6.8	0,00	0,00	0,84	0,00
6.9	0,00	0,50	0,50	0,50
6.10	0,08	0,50	0,50	0,50
7.1	0,00	0,10	1,00	0,00
7.2	0,00	0,00	0,00	0,00
7.3	0,00	0,50	0,50	0,00
7.4	0,00	0,00	0,50	0,00
7.5	0,00	0,85	1,00	0,00
7.6	0,00	0,81	0,00	0,00
7.7	0,00	0,08	0,01	0,50
7.8	0,00	0,74	0,00	0,63
7.9	0,00	0,00	0,02	0,42

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
7.10	0,00	0,99	0,00	0,22
8.1	0,00	0,00	0,05	0,00
8.2	0,00	0,01	0,00	0,00
8.3	0,00	1,00	0,93	0,00
8.4	0,00	0,00	0,01	0,00
8.5	0,00	0,01	0,99	0,00
8.6	0,00	0,94	0,00	0,49
8.7	0,00	0,28	0,02	0,42
8.8	0,00	0,32	0,97	0,00
8.9	0,28	1,00	1,00	1,00
8.10	0,00	0,06	0,00	0,38
9.1	0,00	0,91	0,00	0,00
9.2	0,00	0,00	0,00	0,00
9.3	0,00	0,49	0,00	0,00
9.4	0,00	0,85	0,00	0,19
9.5	0,00	1,00	0,00	0,00
9.6	0,00	0,92	0,96	0,15
9.7	0,00	0,83	0,00	0,00
9.8	0,75	1,00	0,01	0,16
9.9	0,61	1,00	0,02	0,34
9.10	0,02	0,98	0,00	0,00
10.1	0,00	1,00	0,00	0,59
10.2	0,28	1,00	0,00	0,00
10.3	0,93	1,00	0,00	0,00
10.4	0,84	1,00	0,45	0,73
10.5	0,47	1,00	0,00	0,00
10.6	0,66	1,00	0,00	0,42
10.7	0,95	1,00	0,00	0,00
10.8	0,99	1,00	0,00	0,98
10.9	0,00	1,00	0,00	0,00
10.10	0,00	0,34	0,00	0,00
11.1	0,94	1,00	0,00	0,05

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
11.2	0,00	1,00	0,00	0,00
11.3	1,00	1,00	0,00	0,00
11.4	0,99	1,00	0,00	1,00
11.5	0,00	0,48	0,00	0,01
11.6	0,95	1,00	0,00	0,00
11.7	0,99	1,00	0,00	0,27
11.8	1,00	1,00	0,00	0,00
11.9	1,00	1,00	0,00	0,06
11.10	0,83	1,00	0,00	0,00
12.1	1,00	1,00	0,00	0,23
12.2	0,25	1,00	0,00	0,01
12.3	1,00	1,00	0,00	0,01
12.4	1,00	1,00	0,00	0,01
12.5	1,00	1,00	0,00	0,01
12.6	1,00	1,00	1,00	1,00
12.7	0,87	1,00	0,00	0,00
12.8	1,00	1,00	0,01	0,00
12.9	1,00	1,00	0,00	0,01
12.10	1,00	1,00	0,00	0,00
13.1	0,00	1,00	0,00	0,00
13.2	1,00	1,00	0,00	1,00
13.3	0,64	1,00	0,14	0,00
13.4	1,00	1,00	0,17	1,00
13.5	0,67	1,00	0,00	1,00
13.6	0,89	1,00	0,00	0,07
13.7	1,00	1,00	0,00	0,65
13.8	1,00	1,00	0,70	0,00
13.9	0,82	0,99	0,00	0,00
13.10	1,00	1,00	0,00	0,77
14.1	1,00	1,00	0,00	0,00
14.2	0,11	1,00	0,00	0,00
14.3	1,00	1,00	0,00	0,81

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
14.4	0,85	1,00	0,00	0,00
14.5	0,99	1,00	0,00	0,00
14.6	1,00	1,00	0,00	0,25
14.7	1,00	1,00	1,00	1,00
14.8	1,00	1,00	0,98	1,00
14.9	1,00	1,00	0,00	0,96
14.10	1,00	1,00	0,00	0,54
15.1	0,79	1,00	0,00	0,97
15.2	1,00	1,00	0,00	1,00
15.3	1,00	1,00	0,00	1,00
15.4	1,00	1,00	0,00	1,00
15.5	1,00	1,00	0,00	0,93
15.6	0,18	1,00	0,00	0,56
15.7	1,00	1,00	0,00	0,99
15.8	0,92	1,00	0,00	0,05
15.9	1,00	1,00	0,18	0,00
15.10	1,00	1,00	0,00	0,00
16.1	1,00	1,00	0,00	1,00
16.2	1,00	1,00	0,00	1,00
16.3	1,00	1,00	0,91	1,00
16.4	1,00	1,00	0,00	1,00
16.5	0,00	1,00	0,00	0,62
16.6	1,00	1,00	0,00	0,91
16.7	1,00	1,00	0,00	0,07
16.8	1,00	1,00	0,00	0,76
16.9	0,03	1,00	0,00	1,00
16.10	0,00	1,00	0,00	0,59
17.1	1,00	1,00	0,00	0,00
17.2	1,00	1,00	0,00	1,00
17.3	1,00	1,00	0,00	0,93
17.4	1,00	1,00	0,00	1,00
17.5	1,00	1,00	0,00	1,00



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
17.6	1,00	1,00	0,03	1,00
17.7	0,00	1,00	0,08	1,00
17.8	1,00	1,00	0,00	0,78
17.9	1,00	1,00	0,00	0,68
17.10	0,01	1,00	0,85	0,86
18.1	1,00	1,00	0,00	0,00
18.2	1,00	1,00	0,00	0,02
18.3	1,00	1,00	0,00	0,99
18.4	1,00	1,00	0,00	0,74
18.5	1,00	1,00	1,00	1,00
18.6	1,00	1,00	0,00	0,72
18.7	1,00	1,00	0,00	1,00
18.8	1,00	1,00	0,00	1,00
18.9	1,00	1,00	0,87	1,00
18.10	1,00	1,00	0,91	1,00
20.1	1,00	1,00	0,00	0,03
20.2	1,00	1,00	0,00	1,00
20.3	1,00	1,00	0,00	0,02
20.4	1,00	1,00	0,00	1,00
20.5	1,00	1,00	0,00	0,83
20.6	1,00	1,00	0,00	0,98
20.7	1,00	1,00	0,00	0,10
20.8	1,00	1,00	0,00	1,00
20.9	1,00	1,00	0,00	1,00
20.10	1,00	1,00	0,00	1,00
30.1	1,00	1,00	0,00	0,93
30.2	1,00	1,00	0,00	1,00
30.3	1,00	1,00	0,00	1,00
30.4	1,00	1,00	0,00	1,00
30.5	1,00	1,00	0,00	1,00
30.6	1,00	1,00	0,00	1,00
30.7	1,00	1,00	0,00	1,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
30.8	1,00	1,00	0,00	0,99
30.9	1,00	1,00	0,00	0,99
30.10	1,00	1,00	0,00	0,20
50.1	1,00	1,00	0,00	1,00
50.2	1,00	1,00	0,00	1,00
50.3	1,00	1,00	0,00	1,00
50.4	1,00	1,00	0,00	1,00
50.5	1,00	1,00	0,00	1,00
50.6	1,00	1,00	0,00	1,00
50.7	1,00	1,00	0,00	1,00
50.8	1,00	1,00	0,00	1,00
50.9	1,00	1,00	0,00	1,00
50.10	1,00	1,00	0,00	1,00

Tabela 29:  $p$ -valores em favor do AT para instâncias OP3

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
6.1	0,00	0,51	0,84	0,00
6.2	0,00	0,61	0,00	0,00
6.3	0,00	0,01	0,00	0,00
6.4	0,00	0,03	0,03	0,00
6.5	0,00	1,00	0,00	0,00
6.6	0,00	0,00	0,04	0,00
6.7	0,00	0,03	0,00	0,00
6.8	0,00	0,00	0,00	0,00
6.9	0,00	0,22	0,02	0,00
6.10	0,00	0,00	0,16	0,00
7.1	0,00	0,97	0,00	0,00
7.2	0,00	0,13	0,00	0,00
7.3	0,00	0,95	0,00	0,00
7.4	0,00	0,00	0,00	0,00
7.5	0,00	1,00	0,00	0,00
7.6	0,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
7.7	0,00	0,19	0,00	0,00
7.8	0,00	0,00	0,00	0,10
7.9	0,00	0,99	0,62	0,00
7.10	0,00	0,67	0,00	0,00
8.1	0,00	0,98	0,00	0,00
8.2	0,00	0,77	0,00	0,00
8.3	0,00	1,00	0,00	0,00
8.4	0,00	1,00	0,00	0,01
8.5	0,00	1,00	0,00	0,00
8.6	0,00	1,00	0,00	0,00
8.7	0,00	1,00	0,00	0,08
8.8	0,00	0,88	0,00	0,00
8.9	0,00	1,00	0,00	0,00
8.10	0,00	0,13	0,00	0,00
9.1	0,00	1,00	0,00	0,08
9.2	0,00	1,00	0,00	0,09
9.3	0,02	1,00	0,00	0,09
9.4	0,00	1,00	0,00	0,00
9.5	0,00	0,02	0,00	0,00
9.6	0,00	0,99	0,00	0,00
9.7	0,00	1,00	0,00	0,00
9.8	0,00	0,01	0,00	0,00
9.9	0,00	1,00	0,00	0,00
9.10	0,00	0,00	0,00	0,00
10.1	0,00	1,00	0,00	0,04
10.2	0,00	1,00	0,00	0,17
10.3	0,76	1,00	0,00	0,00
10.4	0,00	0,00	0,00	0,00
10.5	0,01	1,00	0,00	0,00
10.6	1,00	1,00	0,00	0,00
10.7	0,00	1,00	0,00	0,00
10.8	1,00	1,00	0,00	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
10.9	0,00	1,00	0,00	0,00
10.10	1,00	1,00	0,00	0,00
11.1	0,00	1,00	0,00	0,00
11.2	0,00	1,00	0,00	0,23
11.3	0,09	1,00	0,00	0,00
11.4	0,86	1,00	0,00	0,12
11.5	0,00	0,00	0,00	0,02
11.6	0,00	0,00	0,00	0,04
11.7	0,00	0,00	0,00	0,00
11.8	0,00	0,49	0,00	0,00
11.9	1,00	1,00	0,00	0,00
11.10	0,96	1,00	0,00	0,00
12.1	0,00	0,02	0,00	0,00
12.2	0,77	1,00	0,00	0,00
12.3	0,99	1,00	0,00	0,00
12.4	0,00	1,00	0,00	0,00
12.5	1,00	1,00	0,00	0,00
12.6	1,00	1,00	0,00	0,00
12.7	0,99	1,00	0,00	0,00
12.8	1,00	0,94	0,00	0,00
12.9	0,73	1,00	0,00	0,00
12.10	1,00	1,00	0,00	0,00
13.1	1,00	1,00	0,00	0,00
13.2	1,00	0,99	0,00	0,00
13.3	1,00	1,00	0,00	0,00
13.4	1,00	1,00	0,00	0,00
13.5	0,15	0,96	0,00	0,00
13.6	0,03	0,13	0,00	0,00
13.7	0,97	0,91	0,00	0,00
13.8	1,00	1,00	0,00	0,00
13.9	0,60	0,36	0,00	0,00
13.10	1,00	1,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
14.1	1,00	1,00	0,00	0,00
14.2	0,10	1,00	0,00	0,00
14.3	0,95	1,00	0,00	0,03
14.4	0,02	0,42	0,00	0,00
14.5	1,00	0,16	0,00	0,00
14.6	0,00	0,00	0,00	0,00
14.7	1,00	1,00	0,00	0,00
14.8	0,00	0,11	0,00	0,00
14.9	0,34	1,00	0,00	0,01
14.10	1,00	1,00	0,00	0,00
15.1	1,00	0,89	0,00	0,00
15.2	1,00	1,00	0,00	0,00
15.3	1,00	1,00	0,00	0,00
15.4	0,93	0,99	0,00	0,00
15.5	1,00	1,00	0,00	0,00
15.6	0,87	0,94	0,00	0,33
15.7	1,00	1,00	0,00	0,00
15.8	1,00	1,00	0,00	0,00
15.9	1,00	1,00	0,00	0,00
15.10	1,00	1,00	0,00	0,00
16.1	1,00	0,87	0,00	0,05
16.2	1,00	0,89	0,00	0,00
16.3	1,00	1,00	0,00	0,00
16.4	1,00	0,99	0,00	0,50
16.5	1,00	1,00	0,00	0,00
16.6	0,69	0,86	0,00	0,00
16.7	1,00	1,00	0,00	0,00
16.8	1,00	1,00	0,00	0,00
16.9	0,00	0,18	0,00	0,00
16.10	1,00	1,00	0,00	0,00
17.1	1,00	0,26	0,00	0,00
17.2	1,00	0,31	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>
17.3	0,20	1,00	0,00	0,00
17.4	1,00	1,00	0,00	0,00
17.5	0,24	0,23	0,00	0,00
17.6	1,00	1,00	0,00	0,00
17.7	1,00	1,00	0,00	0,00
17.8	1,00	1,00	0,00	0,01
17.9	1,00	1,00	0,00	0,00
17.10	1,00	1,00	0,00	0,00
18.1	1,00	1,00	0,00	0,00
18.2	1,00	1,00	0,00	0,00
18.3	1,00	1,00	0,00	0,00
18.4	1,00	0,00	0,00	0,00
18.5	1,00	0,18	0,00	0,00
18.6	1,00	0,01	0,00	0,99
18.7	1,00	0,00	0,00	0,12
18.8	1,00	1,00	0,00	0,00
18.9	1,00	1,00	0,00	0,00
18.10	0,99	0,04	0,00	0,00
20.1	0,00	0,32	0,96	0,00
20.2	0,00	0,84	0,84	0,15
20.3	0,00	0,50	0,50	0,02
20.4	0,00	0,50	0,50	0,02
20.5	0,00	0,00	0,50	0,00
20.6	0,00	0,00	0,00	0,00
20.7	0,00	0,03	0,01	0,00
20.8	0,00	0,03	0,84	0,00
20.9	0,00	0,50	0,50	0,50
20.10	0,01	0,50	0,50	0,50
30.1	1,00	1,00	0,00	0,00
30.2	1,00	1,00	0,00	1,00
30.3	1,00	0,00	0,00	0,00
30.4	0,73	0,00	0,00	0,10

Inst.	PLS	Tabu	NSGAII	NSTA
30.5	1,00	0,00	0,00	0,00
30.6	1,00	1,00	0,00	0,00
30.7	1,00	0,96	0,00	0,99
30.8	1,00	1,00	0,00	0,00
30.9	1,00	1,00	0,00	0,99
30.10	1,00	0,84	0,00	0,00
50.1	1,00	1,00	0,00	0,00
50.2	1,00	0,90	0,00	0,00
50.3	1,00	1,00	0,00	0,00
50.4	1,00	1,00	0,00	0,00
50.5	1,00	0,00	0,00	0,00
50.6	1,00	1,00	0,00	0,66
50.7	1,00	1,00	0,00	0,02
50.8	1,00	1,00	0,00	0,05
50.9	1,00	1,00	0,00	0,00
50.10	1,00	1,00	0,00	0,00

Os  $p$ -valores exibidos nas Tabelas de 26 até 29 indicam, de modo geral, que o algoritmo transgenético obtém melhores resultados que o NSGA-II e que o NSTA, conforme já explicitado anteriormente. O teste estatístico em função dos valores de hipervolume também indica que, de modo geral, os algoritmos PLS e Tabu tendem a fornecer conjuntos de aproximação com qualidade melhor que os produzidos pelo algoritmo transgenético. Vale lembrar que há dois contextos em que o algoritmo NSTA se mostra competitivo em relação ao AT, conforme menção anterior. Ainda assim, considerando todos os bancos de instâncias adotados, não resta dúvidas de que o AT é melhor que o NSTA para o indicador hipervolume.

As Tabelas 30 a 35 exibem os  $p$ -valores da comparação entre a busca Tabu e os outros algoritmos no que se refere ao indicador hipervolume, para os quatro grupos de instâncias adotados. Os  $p$ -valores exibidos são em favor da busca Tabu. O rótulo nas colunas indica o algoritmo de comparação e o rótulo nas linhas identifica a instância.

Tabela 30:  $p$ -valores em favor da Busca Tabu para instâncias CP

Inst.	PLS	NSGAI	NSTA	AT
10.1	0,00	0,50	0,50	0,50
10.2	0,00	0,50	0,50	0,50
10.3	0,02	0,50	0,50	0,50
10.4	0,04	0,50	0,50	0,50
10.5	0,00	0,50	0,50	0,50
10.6	0,00	0,00	0,00	0,00
10.7	0,00	1,00	0,01	1,00
10.8	0,00	0,00	0,00	0,00
10.9	0,01	0,00	0,00	0,00
10.10	0,00	0,00	0,00	0,00
10.11	0,00	0,00	0,00	0,00
10.12	0,00	0,00	0,00	0,00
15.1	0,00	0,00	0,00	0,00
15.2	0,00	0,00	0,00	0,00
15.3	0,00	0,00	0,00	0,00
15.4	0,00	0,00	0,00	0,00
15.5	0,07	0,00	0,00	0,01
15.6	0,00	0,00	0,00	0,00
15.7	0,00	0,00	0,00	0,00
15.8	0,00	0,00	0,00	0,00
15.9	0,00	0,00	0,00	0,00
15.10	0,00	0,00	0,00	0,00
15.11	0,02	0,00	0,00	0,00
15.12	0,00	0,00	0,00	0,00
20.1	0,07	0,00	0,29	0,00
20.2	0,08	0,00	0,76	0,00
20.3	0,02	0,00	0,16	0,00
20.4	0,09	0,00	0,00	0,00
20.5	0,02	0,00	0,00	0,00
20.6	0,00	0,00	0,00	0,00
20.7	0,03	0,00	0,00	0,00



Inst.	PLS	NSGAI	NSTA	AT
20.8	1,00	0,00	0,00	0,00
20.9	0,01	0,00	0,00	0,00
20.10	0,11	0,00	0,00	0,00
20.11	0,08	0,00	0,00	0,00
20.12	0,52	0,00	0,00	0,00
25.1	0,00	0,00	0,00	0,00
25.2	0,33	0,00	0,00	0,00
25.3	0,08	0,00	0,00	0,00
25.4	0,99	0,00	0,00	0,00
25.5	0,16	0,00	0,00	0,00
25.6	0,01	0,00	0,00	0,00
25.7	0,97	0,00	0,00	0,00
25.8	1,00	0,00	0,00	0,00
25.9	0,98	0,00	0,00	0,00
25.10	0,99	0,00	0,00	0,00
25.11	0,95	0,00	0,00	0,00
25.12	0,99	0,00	0,00	0,00
30.1	1,00	0,00	0,00	0,00
30.2	1,00	0,00	0,00	0,00
30.3	1,00	0,00	0,00	0,00
30.4	1,00	0,00	0,00	0,00
30.5	0,00	0,00	0,00	0,00
30.6	0,03	0,00	0,00	0,00
30.7	0,99	0,00	0,00	0,00
30.8	1,00	0,00	0,00	0,00
30.9	0,89	0,00	0,00	0,00
30.10	0,82	0,00	0,00	0,00
30.11	1,00	0,00	0,00	0,00
30.12	1,00	0,00	0,00	0,00
35.1	1,00	0,00	0,00	0,00
35.2	1,00	0,00	0,00	0,00
35.3	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
35.4	1,00	0,00	0,00	0,00
35.5	1,00	0,00	0,00	0,00
35.6	1,00	0,00	0,00	0,00
35.7	1,00	0,00	0,00	0,00
35.8	1,00	0,00	0,00	0,00
35.9	1,00	0,00	0,00	0,00
35.10	1,00	0,00	0,00	0,00
35.11	1,00	0,00	0,00	0,00
35.12	1,00	0,00	0,00	0,00
40.1	1,00	0,00	0,00	0,00
40.2	1,00	0,00	0,00	0,00
40.3	1,00	0,00	0,00	0,00
40.4	1,00	0,00	0,00	0,00
40.5	0,98	0,00	0,00	0,00
40.6	1,00	0,00	0,00	0,00
40.7	1,00	0,00	0,00	0,00
40.8	1,00	0,00	0,00	0,00
40.9	1,00	0,00	0,00	0,00
40.10	1,00	0,00	0,00	0,00
40.11	1,00	0,00	0,00	0,00
40.12	1,00	0,00	0,00	0,00
45.1	1,00	0,00	0,00	0,00
45.2	1,00	0,00	0,00	0,00
45.3	1,00	0,00	0,00	0,00
45.4	1,00	0,00	0,00	0,00
45.5	1,00	0,00	0,00	0,00
45.6	1,00	0,00	0,00	0,00
45.7	1,00	0,00	0,00	0,00
45.8	1,00	0,00	0,00	0,00
45.9	1,00	0,00	0,00	0,00
45.10	1,00	0,00	0,00	0,00
45.11	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
45.12	1,00	0,00	0,00	0,00
50.1	1,00	0,00	0,00	0,00
50.2	1,00	0,00	0,00	0,00
50.3	1,00	0,00	0,00	0,00
50.4	1,00	0,00	0,00	0,00
50.5	1,00	0,00	0,00	0,00
50.6	1,00	0,00	0,00	0,00
50.7	1,00	0,00	0,00	0,00
50.8	1,00	0,00	0,00	0,00
50.9	1,00	0,00	0,00	0,00
50.10	1,00	0,00	0,00	0,00
50.11	1,00	0,00	0,00	0,00
50.12	1,00	0,00	0,00	0,00
75.1	1,00	0,00	0,00	0,00
75.2	1,00	0,00	0,00	0,00
75.3	1,00	0,00	0,00	0,00
75.4	1,00	0,00	0,00	0,00
75.5	1,00	0,00	0,00	0,00
75.6	1,00	0,00	0,00	0,00
75.7	1,00	0,00	0,00	0,00
75.8	1,00	0,00	0,00	0,00
75.9	1,00	0,00	0,00	0,00
75.10	1,00	0,00	0,00	0,00
75.11	1,00	0,00	0,00	0,00
75.12	1,00	0,00	0,00	0,00
100.1	1,00	0,00	0,00	0,00
100.2	1,00	0,00	0,00	0,00
100.3	1,00	0,00	0,00	0,00
100.4	1,00	0,00	0,00	0,00
100.5	1,00	0,00	0,00	0,00
100.6	1,00	0,00	0,00	0,00
100.7	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
100.8	1,00	0,00	0,00	0,00
100.9	1,00	0,00	0,00	0,00
100.10	1,00	0,00	0,00	0,00
100.11	1,00	0,00	0,00	0,00
100.12	1,00	0,00	0,00	0,00
125.1	1,00	0,00	0,00	0,00
125.2	1,00	0,00	0,00	0,00
125.3	1,00	0,00	0,00	0,00
125.4	1,00	0,00	0,00	0,00
125.5	1,00	0,00	0,00	0,00
125.6	1,00	0,00	0,00	0,00
125.7	1,00	0,00	0,00	0,00
125.8	1,00	0,00	0,00	0,00
125.9	1,00	0,00	0,00	0,00
125.10	1,00	0,00	0,00	0,00
125.11	1,00	0,00	0,00	0,00
125.12	1,00	0,00	0,00	0,00
150.1	1,00	0,00	0,00	0,00
150.2	1,00	0,00	0,00	0,00
150.3	1,00	0,00	0,00	0,00
150.4	1,00	0,00	0,00	0,00
150.5	1,00	0,00	0,00	0,00
150.6	1,00	0,00	0,00	0,00
150.7	1,00	0,00	0,00	0,00
150.8	1,00	0,00	0,00	0,00
175.1	1,00	0,00	0,00	0,00
175.2	1,00	0,00	0,00	0,00
175.3	1,00	0,00	0,00	0,00
175.4	1,00	0,00	0,00	0,00
175.5	1,00	0,00	0,00	0,00
175.6	0,95	0,00	0,00	0,00
175.7	0,49	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
175.8	0,72	0,00	0,00	0,00

Tabela 31:  $p$ -valores em favor da Busca Tabu para instâncias OP1

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,39	0,00	0,92
6.2	0,00	0,19	0,00	1,00
6.3	0,00	0,98	0,22	1,00
6.4	0,00	0,00	0,01	0,83
6.5	0,01	0,50	0,00	0,50
6.6	0,00	0,01	0,02	0,90
6.7	0,00	0,50	0,00	0,50
6.8	0,00	0,00	0,00	0,50
6.9	0,00	0,50	0,50	0,50
6.10	0,00	0,92	0,05	0,73
7.1	0,00	0,00	0,99	1,00
7.2	0,00	0,00	0,00	0,46
7.3	0,00	0,00	0,06	1,00
7.4	0,00	0,00	0,00	0,00
7.5	0,00	0,00	0,00	0,00
7.6	0,00	0,02	0,00	0,08
7.7	0,00	0,50	0,00	0,08
7.8	0,00	0,00	0,41	0,99
7.9	0,00	0,00	0,00	0,00
7.10	0,00	0,20	0,02	0,02
8.1	0,00	0,13	0,00	0,00
8.2	0,00	0,00	0,00	0,00
8.3	0,00	0,70	0,00	0,00
8.4	0,00	0,25	0,00	0,04
8.5	0,00	0,00	0,00	0,05
8.6	0,00	0,00	0,00	0,00
8.7	0,00	0,00	0,00	0,00
8.8	0,00	0,00	0,00	0,00

Inst.	PLS	NSGAI	NSTA	AT
8.9	0,00	0,63	0,85	1,00
8.10	0,00	0,00	0,00	0,00
9.1	0,00	0,00	0,00	0,00
9.2	0,00	0,00	0,00	0,00
9.3	0,00	0,00	0,00	0,00
9.4	0,00	0,20	0,00	0,88
9.5	0,05	0,00	0,00	0,97
9.6	0,00	0,00	0,00	0,00
9.7	0,00	0,00	0,00	0,01
9.8	0,00	0,00	0,00	0,25
9.9	0,00	0,00	0,00	0,00
9.10	0,00	0,08	1,00	1,00
10.1	0,00	0,00	0,00	0,00
10.2	0,00	0,00	0,00	0,08
10.3	0,40	0,00	0,00	0,00
10.4	0,00	0,00	0,00	0,00
10.5	0,00	0,00	0,00	0,00
10.6	0,00	0,00	0,00	0,00
10.7	0,00	0,00	0,00	0,00
10.8	0,00	0,00	0,00	0,00
10.9	0,00	0,00	0,00	0,00
10.10	0,00	0,00	0,00	0,00
11.1	0,77	0,00	0,00	0,00
11.2	0,00	0,00	0,00	0,00
11.3	0,00	0,00	0,00	0,00
11.4	0,00	0,00	0,00	0,00
11.5	0,23	0,00	0,00	0,00
11.6	0,00	0,00	0,00	0,01
11.7	0,00	0,00	0,00	0,00
11.8	0,00	0,00	0,00	0,00
11.9	0,00	0,00	0,00	0,00
11.10	0,00	0,00	0,00	0,00

Inst.	PLS	NSGAI	NSTA	AT
12.1	0,00	0,00	0,00	0,00
12.2	0,00	0,00	0,00	0,00
12.3	0,06	0,00	0,00	0,00
12.4	0,13	0,00	0,00	0,00
12.5	0,00	0,00	0,00	0,00
12.6	0,00	0,00	0,00	0,00
12.7	0,01	0,00	0,00	0,00
12.8	0,01	0,00	0,00	0,00
12.9	0,00	0,00	0,00	0,00
12.10	0,00	0,00	0,00	0,00
13.1	0,00	0,00	0,00	0,00
13.2	0,00	0,00	0,00	0,00
13.3	0,00	0,00	0,00	0,00
13.4	0,61	0,00	0,00	0,00
13.5	0,00	0,00	0,00	0,00
13.6	0,07	0,00	0,00	0,00
13.7	0,00	0,00	0,00	0,00
13.8	0,00	0,00	0,00	0,00
13.9	0,01	0,00	0,00	0,00
13.10	0,15	0,00	0,00	0,00
14.1	1,00	0,00	0,00	0,00
14.2	0,99	0,00	0,00	0,00
14.3	0,00	0,00	0,00	0,00
14.4	0,01	0,00	0,48	0,06
14.5	0,94	0,00	0,00	0,00
14.6	0,77	0,00	0,00	0,00
14.7	0,11	0,00	0,00	0,00
14.8	0,99	0,00	0,00	0,00
14.9	0,14	0,00	0,00	0,00
14.10	0,01	0,00	0,00	0,00
15.1	0,20	0,00	0,00	0,00
15.2	0,09	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
15.3	0,00	0,00	0,00	0,00
15.4	0,03	0,00	0,00	0,00
15.5	0,00	0,00	0,00	0,00
15.6	0,78	0,00	0,00	0,00
15.7	0,00	0,00	0,00	0,00
15.8	0,10	0,00	0,00	0,00
15.9	0,00	0,00	0,00	0,00
15.10	0,00	0,00	0,00	0,00
16.1	0,38	0,00	0,00	0,00
16.2	0,01	0,00	0,00	0,00
16.3	0,04	0,00	0,00	0,00
16.4	1,00	0,00	0,00	0,00
16.5	0,09	0,00	0,00	0,00
16.6	0,04	0,00	0,00	0,00
16.7	0,00	0,00	0,00	0,00
16.8	0,88	0,00	0,00	0,00
16.9	0,04	0,00	0,00	0,00
16.10	0,02	0,00	0,00	0,00
17.1	0,44	0,00	0,00	0,00
17.2	0,60	0,00	0,00	0,00
17.3	0,52	0,00	0,00	0,00
17.4	0,68	0,00	0,00	0,00
17.5	0,33	0,00	0,00	0,00
17.6	0,00	0,00	0,00	0,00
17.7	0,16	0,00	0,00	0,00
17.8	0,49	0,00	0,00	0,00
17.9	0,99	0,00	0,00	0,00
17.10	0,00	0,00	0,00	0,00
18.1	1,00	0,00	0,00	0,00
18.2	0,37	0,00	0,00	0,00
18.3	0,08	0,00	0,00	0,00
18.4	0,84	0,00	0,00	0,00



<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
18.5	0,99	0,00	0,00	0,00
18.6	0,32	0,00	0,00	0,00
18.7	0,00	0,00	0,00	0,00
18.8	0,37	0,00	0,00	0,00
18.9	0,99	0,00	0,00	0,00
18.10	0,00	0,00	0,00	0,00
20.1	0,23	0,00	0,00	0,00
20.2	0,12	0,00	0,00	0,00
20.3	0,24	0,00	0,00	0,00
20.4	0,87	0,00	0,00	0,00
20.5	0,00	0,00	0,00	0,00
20.6	0,37	0,00	0,00	0,00
20.7	0,00	0,00	0,00	0,00
20.8	0,96	0,00	0,00	0,00
20.9	0,65	0,00	0,00	0,00
20.10	0,66	0,00	0,00	0,00
30.1	1,00	0,00	0,00	0,00
30.2	1,00	0,00	0,00	0,00
30.3	1,00	0,00	0,00	0,00
30.4	1,00	0,00	0,00	0,00
30.5	0,38	0,00	0,00	0,00
30.6	1,00	0,00	0,00	0,00
30.7	1,00	0,00	0,00	0,00
30.8	1,00	0,00	0,00	0,00
30.9	1,00	0,00	0,00	0,00
30.10	0,98	0,00	0,00	0,00
50.1	1,00	0,00	0,00	0,00
50.2	1,00	0,00	0,00	0,00
50.3	1,00	0,00	0,00	0,00
50.4	1,00	0,00	0,00	0,00
50.5	1,00	0,00	0,00	0,00
50.6	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
50.7	1,00	0,00	0,00	0,00
50.8	1,00	0,00	0,00	0,00
50.9	1,00	0,00	0,00	0,00
50.10	1,00	0,00	0,00	0,00

Tabela 32:  $p$ -valores em favor da Busca Tabu para instâncias OP2

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,84	0,00	0,68
6.2	0,00	0,50	0,02	0,50
6.3	0,00	0,50	0,50	0,50
6.4	0,00	0,84	0,84	0,84
6.5	0,00	1,00	0,51	1,00
6.6	0,00	0,04	0,05	1,00
6.7	0,00	0,98	0,07	1,00
6.8	0,00	1,00	0,20	1,00
6.9	0,00	0,50	0,50	0,50
6.10	0,08	0,50	0,50	0,50
7.1	0,00	1,00	0,00	0,90
7.2	0,00	0,04	0,82	1,00
7.3	0,00	0,50	0,00	0,50
7.4	0,00	1,00	0,00	1,00
7.5	0,00	1,00	0,00	0,15
7.6	0,00	0,00	0,00	0,19
7.7	0,00	0,05	0,92	0,92
7.8	0,00	0,00	0,38	0,26
7.9	0,00	1,00	1,00	1,00
7.10	0,00	0,00	0,00	0,01
8.1	0,00	1,00	0,00	1,00
8.2	0,00	0,00	0,02	0,99
8.3	0,00	0,27	0,00	0,00
8.4	0,00	1,00	0,00	1,00
8.5	0,00	1,00	0,01	0,99

Inst.	PLS	NSGAI	NSTA	AT
8.6	0,00	0,00	0,09	0,06
8.7	0,00	0,00	0,34	0,72
8.8	0,00	1,00	0,00	0,68
8.9	0,00	0,23	0,00	0,00
8.10	0,00	0,00	0,99	0,94
9.1	0,00	0,00	0,00	0,09
9.2	0,17	1,00	0,00	1,00
9.3	0,00	0,00	0,00	0,51
9.4	0,00	0,00	0,00	0,15
9.5	0,00	0,00	0,00	0,00
9.6	0,00	0,16	0,01	0,08
9.7	0,00	0,00	0,00	0,17
9.8	0,00	0,00	0,00	0,00
9.9	0,00	0,00	0,00	0,00
9.10	0,00	0,00	0,00	0,02
10.1	0,00	0,00	0,00	0,00
10.2	0,00	0,00	0,00	0,00
10.3	0,00	0,00	0,00	0,00
10.4	0,00	0,00	0,00	0,00
10.5	0,00	0,00	0,00	0,00
10.6	0,00	0,00	0,00	0,00
10.7	0,00	0,00	0,00	0,00
10.8	0,00	0,00	0,00	0,00
10.9	0,00	0,00	0,00	0,00
10.10	0,00	0,00	0,00	0,66
11.1	0,00	0,00	0,00	0,00
11.2	0,00	0,00	0,00	0,00
11.3	0,03	0,00	0,00	0,00
11.4	0,00	0,00	0,00	0,00
11.5	0,00	0,00	0,04	0,52
11.6	0,00	0,00	0,00	0,00
11.7	0,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
11.8	0,05	0,00	0,00	0,00
11.9	0,23	0,00	0,00	0,00
11.10	0,00	0,00	0,00	0,00
12.1	1,00	0,00	0,00	0,00
12.2	0,00	0,00	0,00	0,00
12.3	0,59	0,00	0,00	0,00
12.4	0,00	0,00	0,00	0,00
12.5	0,00	0,00	0,00	0,00
12.6	0,00	0,00	0,00	0,00
12.7	0,00	0,00	0,00	0,00
12.8	0,00	0,00	0,00	0,00
12.9	0,04	0,00	0,00	0,00
12.10	0,00	0,00	0,00	0,00
13.1	0,00	0,00	0,00	0,00
13.2	0,00	0,00	0,05	0,00
13.3	0,00	0,00	0,00	0,00
13.4	1,00	0,00	1,00	0,00
13.5	0,00	0,00	0,00	0,00
13.6	0,00	0,00	0,00	0,00
13.7	0,00	0,00	0,00	0,00
13.8	0,00	0,00	0,00	0,00
13.9	0,04	0,00	0,00	0,01
13.10	0,00	0,00	0,00	0,00
14.1	0,00	0,00	0,00	0,00
14.2	0,00	0,00	0,00	0,00
14.3	0,54	0,00	0,00	0,00
14.4	0,00	0,00	0,00	0,00
14.5	0,01	0,00	0,00	0,00
14.6	0,00	0,00	0,00	0,00
14.7	0,00	0,00	0,00	0,00
14.8	0,00	0,00	0,00	0,00
14.9	0,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
14.10	0,29	0,00	0,00	0,00
15.1	0,00	0,00	0,00	0,00
15.2	0,00	0,00	0,00	0,00
15.3	0,00	0,00	0,00	0,00
15.4	0,00	0,00	0,00	0,00
15.5	0,11	0,00	0,00	0,00
15.6	0,00	0,00	0,00	0,00
15.7	0,00	0,00	0,00	0,00
15.8	0,00	0,00	0,00	0,00
15.9	0,00	0,00	0,00	0,00
15.10	0,00	0,00	0,00	0,00
16.1	0,00	0,00	0,00	0,00
16.2	0,00	0,00	0,00	0,00
16.3	0,00	0,00	0,00	0,00
16.4	0,00	0,00	0,00	0,00
16.5	0,00	0,00	0,00	0,00
16.6	0,00	0,00	0,00	0,00
16.7	0,00	0,00	0,00	0,00
16.8	0,00	0,00	0,00	0,00
16.9	0,00	0,00	0,00	0,00
16.10	0,00	0,00	0,00	0,00
17.1	0,00	0,00	0,00	0,00
17.2	0,00	0,00	0,00	0,00
17.3	0,00	0,00	0,00	0,00
17.4	0,00	0,00	0,00	0,00
17.5	0,00	0,00	0,00	0,00
17.6	0,00	0,00	0,00	0,00
17.7	0,00	0,00	0,00	0,00
17.8	0,00	0,00	0,00	0,00
17.9	0,00	0,00	0,00	0,00
17.10	0,00	0,00	0,00	0,00
18.1	0,89	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
18.2	0,00	0,00	0,00	0,00
18.3	0,00	0,00	0,00	0,00
18.4	0,00	0,00	0,00	0,00
18.5	0,00	0,00	0,00	0,00
18.6	0,00	0,00	0,00	0,00
18.7	0,00	0,00	0,00	0,00
18.8	0,00	0,00	0,00	0,00
18.9	0,00	0,00	0,00	0,00
18.10	0,00	0,00	0,00	0,00
20.1	0,00	0,00	0,00	0,00
20.2	0,00	0,00	0,00	0,00
20.3	0,00	0,00	0,00	0,00
20.4	0,00	0,00	0,00	0,00
20.5	0,00	0,00	0,00	0,00
20.6	0,00	0,00	0,00	0,00
20.7	0,00	0,00	0,00	0,00
20.8	0,02	0,00	0,00	0,00
20.9	0,00	0,00	0,00	0,00
20.10	0,00	0,00	0,00	0,00
30.1	0,79	0,00	0,00	0,00
30.2	0,45	0,00	0,00	0,00
30.3	0,04	0,00	0,00	0,00
30.4	0,49	0,00	0,00	0,00
30.5	1,00	0,00	0,00	0,00
30.6	0,00	0,00	0,00	0,00
30.7	0,26	0,00	0,00	0,00
30.8	0,04	0,00	0,00	0,00
30.9	0,00	0,00	0,00	0,00
30.10	1,00	0,00	0,00	0,00
50.1	1,00	0,00	0,00	0,00
50.2	1,00	0,00	0,00	0,00
50.3	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
50.4	0,98	0,00	0,00	0,00
50.5	0,99	0,00	0,00	0,00
50.6	0,93	0,00	0,00	0,00
50.7	1,00	0,00	0,00	0,00
50.8	0,68	0,00	0,00	0,00
50.9	0,99	0,00	0,00	0,00
50.10	1,00	0,00	0,00	0,00

Tabela 33:  $p$ -valores em favor da Busca Tabu para instâncias OP3

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,84	0,00	0,49
6.2	0,00	0,00	0,00	0,39
6.3	0,00	0,33	0,09	0,99
6.4	0,00	0,54	0,00	0,97
6.5	0,00	0,00	0,00	0,00
6.6	0,00	0,98	0,01	1,00
6.7	0,00	0,00	0,00	0,97
6.8	0,00	0,65	0,00	1,00
6.9	0,00	0,03	0,00	0,78
6.10	0,00	0,97	0,00	1,00
7.1	0,00	0,00	0,00	0,03
7.2	0,00	0,00	0,00	0,87
7.3	0,00	0,00	0,00	0,05
7.4	0,00	0,07	0,00	1,00
7.5	0,00	0,00	0,00	0,00
7.6	0,00	0,00	0,00	0,00
7.7	0,00	0,00	0,00	0,81
7.8	0,00	0,00	1,00	1,00
7.9	0,00	0,04	0,00	0,01
7.10	0,00	0,00	0,00	0,33
8.1	0,00	0,00	0,00	0,02
8.2	0,00	0,00	0,00	0,23

Inst.	PLS	NSGAI	NSTA	AT
8.3	0,00	0,00	0,00	0,00
8.4	0,00	0,00	0,00	0,00
8.5	0,00	0,00	0,00	0,00
8.6	0,00	0,00	0,00	0,00
8.7	0,00	0,00	0,00	0,00
8.8	0,00	0,00	0,00	0,12
8.9	0,00	0,00	0,00	0,00
8.10	0,00	0,00	0,37	0,87
9.1	0,00	0,00	0,00	0,00
9.2	0,00	0,00	0,00	0,00
9.3	0,00	0,00	0,00	0,00
9.4	0,00	0,00	0,00	0,00
9.5	0,00	0,00	0,11	0,98
9.6	0,00	0,00	0,00	0,01
9.7	0,00	0,00	0,00	0,00
9.8	0,01	0,00	0,04	0,99
9.9	0,00	0,00	0,00	0,00
9.10	0,00	0,00	0,13	1,00
10.1	0,00	0,00	0,00	0,00
10.2	0,00	0,00	0,00	0,00
10.3	0,00	0,00	0,00	0,00
10.4	0,00	0,00	0,75	1,00
10.5	0,00	0,00	0,00	0,00
10.6	0,06	0,00	0,00	0,00
10.7	0,00	0,00	0,00	0,00
10.8	0,01	0,00	0,00	0,00
10.9	0,00	0,00	0,00	0,00
10.10	0,24	0,00	0,00	0,00
11.1	0,00	0,00	0,00	0,00
11.2	0,00	0,00	0,00	0,00
11.3	0,00	0,00	0,00	0,00
11.4	0,00	0,00	0,00	0,00



Inst.	PLS	NSGAI	NSTA	AT
11.5	0,00	0,00	0,87	1,00
11.6	0,00	0,00	0,98	1,00
11.7	0,25	0,00	0,71	1,00
11.8	0,06	0,00	0,00	0,51
11.9	0,10	0,00	0,00	0,00
11.10	0,00	0,00	0,00	0,00
12.1	0,00	0,00	0,00	0,98
12.2	0,00	0,00	0,00	0,00
12.3	0,08	0,00	0,00	0,00
12.4	0,00	0,00	0,00	0,00
12.5	0,00	0,00	0,00	0,00
12.6	0,77	0,00	0,00	0,00
12.7	0,00	0,00	0,00	0,00
12.8	0,98	0,00	0,00	0,06
12.9	0,00	0,00	0,00	0,00
12.10	0,15	0,00	0,00	0,00
13.1	0,00	0,00	0,00	0,00
13.2	0,04	0,00	0,00	0,01
13.3	0,00	0,00	0,00	0,00
13.4	0,14	0,00	0,00	0,00
13.5	0,13	0,00	0,04	0,04
13.6	0,12	0,00	0,03	0,87
13.7	0,15	0,00	0,00	0,09
13.8	0,02	0,00	0,00	0,00
13.9	0,72	0,00	0,00	0,64
13.10	0,27	0,00	0,00	0,00
14.1	0,22	0,00	0,00	0,00
14.2	0,00	0,00	0,00	0,00
14.3	0,14	0,00	0,00	0,00
14.4	0,01	0,00	0,00	0,58
14.5	0,97	0,00	0,00	0,84
14.6	0,01	0,00	0,12	1,00

Inst.	PLS	NSGAI	NSTA	AT
14.7	0,07	0,00	0,00	0,00
14.8	0,36	0,12	0,00	0,89
14.9	0,00	0,00	0,00	0,00
14.10	0,88	0,00	0,00	0,00
15.1	0,63	0,00	0,02	0,11
15.2	0,08	0,00	0,00	0,00
15.3	0,60	0,00	0,00	0,00
15.4	0,13	0,00	0,00	0,01
15.5	1,00	0,00	0,00	0,00
15.6	0,12	0,00	0,38	0,06
15.7	0,20	0,00	0,00	0,00
15.8	0,34	0,00	0,00	0,00
15.9	0,00	0,00	0,00	0,00
15.10	1,00	0,00	0,00	0,00
16.1	1,00	0,00	0,15	0,13
16.2	0,78	0,00	0,05	0,11
16.3	0,84	0,00	0,00	0,00
16.4	0,95	0,00	0,42	0,01
16.5	0,83	0,00	0,00	0,00
16.6	0,23	0,00	0,00	0,14
16.7	0,83	0,00	0,00	0,00
16.8	0,19	0,00	0,00	0,00
16.9	0,01	0,00	0,00	0,82
16.10	0,91	0,00	0,00	0,00
17.1	1,00	0,02	0,62	0,74
17.2	1,00	0,01	0,01	0,69
17.3	0,01	0,00	0,00	0,00
17.4	1,00	0,00	0,00	0,00
17.5	0,49	0,00	0,00	0,77
17.6	1,00	0,00	0,00	0,00
17.7	0,88	0,00	0,00	0,00
17.8	0,99	0,00	0,00	0,00

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
17.9	1,00	0,00	0,00	0,00
17.10	0,87	0,00	0,00	0,00
18.1	0,99	0,00	0,00	0,00
18.2	1,00	0,00	0,00	0,00
18.3	0,92	0,00	0,00	0,00
18.4	1,00	0,05	1,00	1,00
18.5	1,00	0,00	0,00	0,82
18.6	1,00	0,00	1,00	0,99
18.7	1,00	0,46	1,00	1,00
18.8	1,00	0,00	0,00	0,00
18.9	0,18	0,00	0,00	0,00
18.10	1,00	0,00	0,00	0,96
20.1	0,00	0,98	0,00	0,68
20.2	0,00	0,50	0,04	0,16
20.3	0,00	0,50	0,02	0,50
20.4	0,00	0,50	0,02	0,50
20.5	0,00	1,00	0,40	1,00
20.6	0,00	0,09	0,59	1,00
20.7	0,00	0,79	0,28	0,97
20.8	0,00	0,99	0,00	0,97
20.9	0,00	0,50	0,50	0,50
20.10	0,01	0,50	0,50	0,50
30.1	1,00	0,00	0,00	0,00
30.2	1,00	0,00	0,00	0,00
30.3	1,00	0,00	0,00	1,00
30.4	1,00	0,00	1,00	1,00
30.5	1,00	0,00	0,00	1,00
30.6	1,00	0,00	0,00	0,00
30.7	1,00	0,00	0,06	0,04
30.8	1,00	0,00	0,00	0,00
30.9	0,97	0,00	0,00	0,00
30.10	1,00	0,00	0,00	0,16

<b>Inst.</b>	<b>PLS</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
50.1	1,00	0,00	0,00	0,00
50.2	1,00	0,00	0,00	0,10
50.3	1,00	0,00	0,00	0,00
50.4	1,00	0,00	0,00	0,00
50.5	1,00	0,00	1,00	1,00
50.6	1,00	0,00	0,00	0,00
50.7	1,00	0,00	0,00	0,00
50.8	1,00	0,00	0,00	0,00
50.9	1,00	0,00	0,00	0,00
50.10	1,00	0,00	0,00	0,00

O algoritmo de busca Tabu obtém excelentes resultados quando comparado, com base no indicador de qualidade hipervolume, às outras abordagens propostas nesta Tese. Tal fato pode ser observado por meio dos  $p$ -valores apresentados nas Tabelas de 30 até 33. Apenas uma heurística consegue ganhar da busca Tabu: a busca PLS.

Conforme pode ser confirmado também nas Tabelas de 34 até 37 (as quais exibem os  $p$ -valores, em favor de PLS, da comparação entre a busca PLS e os outros algoritmos no que se refere ao indicador hipervolume, para os quatro grupos de instâncias adotados), dentre os algoritmos desenvolvidos para esta Tese, a busca PLS é o que apresenta melhores resultados no quesito qualidade do conjunto de aproximação gerado, levando em conta o indicador hipervolume. No entanto, a melhoria na qualidade dos conjuntos de aproximação gerados tem um preço, o tempo computacional bem superior aos de outras abordagens. Testes com instâncias ainda maiores seriam interessantes para verificar se a qualidade dos conjuntos gerados se mantém com o aumento do tamanho da instância, como o tempo computacional cresce em função do tamanho da instância e como a referida qualidade ficaria no caso de se restringir o tempo de execução.

Tabela 34:  $p$ -valores em favor da busca PLS para instâncias CP

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
10.1	1,00	1,00	1,00	1,00
10.2	1,00	1,00	1,00	1,00
10.3	0,98	0,98	0,98	0,98
10.4	0,96	0,96	0,96	0,96

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
10.5	1,00	1,00	1,00	1,00
10.6	1,00	0,44	0,02	0,91
10.7	1,00	1,00	0,98	1,00
10.8	1,00	0,65	0,00	0,96
10.9	0,99	0,00	0,00	0,62
10.10	1,00	0,00	0,00	0,02
10.11	1,00	0,00	0,55	0,95
10.12	1,00	0,00	0,00	0,00
15.1	1,00	0,00	1,00	1,00
15.2	1,00	0,00	1,00	0,26
15.3	1,00	0,00	1,00	1,00
15.4	1,00	0,00	0,69	0,00
15.5	0,93	0,00	0,00	0,11
15.6	1,00	0,00	0,48	0,82
15.7	1,00	0,00	0,79	0,18
15.8	1,00	0,00	0,29	0,60
15.9	1,00	0,00	0,95	0,22
15.10	1,00	0,00	0,00	0,00
15.11	0,98	0,00	0,00	0,00
15.12	1,00	0,00	0,00	0,00
20.1	0,93	0,00	0,98	0,00
20.2	0,92	0,00	0,98	0,00
20.3	0,98	0,00	1,00	0,00
20.4	0,91	0,00	0,05	0,00
20.5	0,98	0,00	0,00	0,00
20.6	1,00	0,00	0,00	0,00
20.7	0,97	0,00	0,00	0,00
20.8	0,00	0,00	0,00	0,00
20.9	0,99	0,00	0,00	0,00
20.10	0,89	0,00	0,00	0,00
20.11	0,92	0,00	0,00	0,00
20.12	0,48	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>NSTA</b>	<b>AT</b>
25.1	1,00	0,00	0,00	0,00
25.2	0,67	0,00	0,00	0,00
25.3	0,92	0,00	0,00	0,00
25.4	0,01	0,00	0,00	0,00
25.5	0,84	0,00	0,00	0,00
25.6	0,99	0,00	0,00	0,00
25.7	0,03	0,00	0,00	0,00
25.8	0,00	0,00	0,00	0,00
25.9	0,02	0,00	0,00	0,00
25.10	0,01	0,00	0,00	0,00
25.11	0,05	0,00	0,00	0,00
25.12	0,01	0,00	0,00	0,00
30.1	0,00	0,00	0,00	0,00
30.2	0,00	0,00	0,00	0,00
30.3	0,00	0,00	0,00	0,00
30.4	0,00	0,00	0,00	0,00
30.5	1,00	0,00	0,00	0,00
30.6	0,97	0,00	0,00	0,00
30.7	0,01	0,00	0,00	0,00
30.8	0,00	0,00	0,00	0,00
30.9	0,11	0,00	0,00	0,00
30.10	0,18	0,00	0,00	0,00
30.11	0,00	0,00	0,00	0,00
30.12	0,00	0,00	0,00	0,00
35.1	0,00	0,00	0,00	0,00
35.2	0,00	0,00	0,00	0,00
35.3	0,00	0,00	0,00	0,00
35.4	0,00	0,00	0,00	0,00
35.5	0,00	0,00	0,00	0,00
35.6	0,00	0,00	0,00	0,00
35.7	0,00	0,00	0,00	0,00
35.8	0,00	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
35.9	0,00	0,00	0,00	0,00
35.10	0,00	0,00	0,00	0,00
35.11	0,00	0,00	0,00	0,00
35.12	0,00	0,00	0,00	0,00
40.1	0,00	0,00	0,00	0,00
40.2	0,00	0,00	0,00	0,00
40.3	0,00	0,00	0,00	0,00
40.4	0,00	0,00	0,00	0,00
40.5	0,02	0,00	0,00	0,00
40.6	0,00	0,00	0,00	0,00
40.7	0,00	0,00	0,00	0,00
40.8	0,00	0,00	0,00	0,00
40.9	0,00	0,00	0,00	0,00
40.10	0,00	0,00	0,00	0,00
40.11	0,00	0,00	0,00	0,00
40.12	0,00	0,00	0,00	0,00
45.1	0,00	0,00	0,00	0,00
45.2	0,00	0,00	0,00	0,00
45.3	0,00	0,00	0,00	0,00
45.4	0,00	0,00	0,00	0,00
45.5	0,00	0,00	0,00	0,00
45.6	0,00	0,00	0,00	0,00
45.7	0,00	0,00	0,00	0,00
45.8	0,00	0,00	0,00	0,00
45.9	0,00	0,00	0,00	0,00
45.10	0,00	0,00	0,00	0,00
45.11	0,00	0,00	0,00	0,00
45.12	0,00	0,00	0,00	0,00
50.1	0,00	0,00	0,00	0,00
50.2	0,00	0,00	0,00	0,00
50.3	0,00	0,00	0,00	0,00
50.4	0,00	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSAI</b>	<b>AT</b>
50.5	0,00	0,00	0,00	0,00
50.6	0,00	0,00	0,00	0,00
50.7	0,00	0,00	0,00	0,00
50.8	0,00	0,00	0,00	0,00
50.9	0,00	0,00	0,00	0,00
50.10	0,00	0,00	0,00	0,00
50.11	0,00	0,00	0,00	0,00
50.12	0,00	0,00	0,00	0,00
75.1	0,00	0,00	0,00	0,00
75.2	0,00	0,00	0,00	0,00
75.3	0,00	0,00	0,00	0,00
75.4	0,00	0,00	0,00	0,00
75.5	0,00	0,00	0,00	0,00
75.6	0,00	0,00	0,00	0,00
75.7	0,00	0,00	0,00	0,00
75.8	0,00	0,00	0,00	0,00
75.9	0,00	0,00	0,00	0,00
75.10	0,00	0,00	0,00	0,00
75.11	0,00	0,00	0,00	0,00
75.12	0,00	0,00	0,00	0,00
100.1	0,00	0,00	0,00	0,00
100.2	0,00	0,00	0,00	0,00
100.3	0,00	0,00	0,00	0,00
100.4	0,00	0,00	0,00	0,00
100.5	0,00	0,00	0,00	0,00
100.6	0,00	0,00	0,00	0,00
100.7	0,00	0,00	0,00	0,00
100.8	0,00	0,00	0,00	0,00
100.9	0,00	0,00	0,00	0,00
100.10	0,00	0,00	0,00	0,00
100.11	0,00	0,00	0,00	0,00
100.12	0,00	0,00	0,00	0,00



<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSGAII</b>	<b>NSTA</b>	<b>AT</b>
125.1	0,00	0,00	0,00	0,00	0,00
125.2	0,00	0,00	0,00	0,00	0,00
125.3	0,00	0,00	0,00	0,00	0,00
125.4	0,00	0,00	0,00	0,00	0,00
125.5	0,00	0,00	0,00	0,00	0,00
125.6	0,00	0,00	0,00	0,00	0,00
125.7	0,00	0,00	0,00	0,00	0,00
125.8	0,00	0,00	0,00	0,00	0,00
125.9	0,00	0,00	0,00	0,00	0,00
125.10	0,00	0,00	0,00	0,00	0,00
125.11	0,00	0,00	0,00	0,00	0,00
125.12	0,00	0,00	0,00	0,00	0,00
150.1	0,00	0,00	0,00	0,00	0,00
150.2	0,00	0,00	0,00	0,00	0,00
150.3	0,00	0,00	0,00	0,00	0,00
150.4	0,00	0,00	0,00	0,00	0,00
150.5	0,00	0,00	0,00	0,00	0,00
150.6	0,00	0,00	0,00	0,00	0,00
150.7	0,00	0,00	0,00	0,00	0,00
150.8	0,00	0,00	0,00	0,00	0,00
175.1	0,00	0,00	0,00	0,00	0,00
175.2	0,00	0,00	0,00	0,00	0,00
175.3	0,00	0,00	0,00	0,00	0,00
175.4	0,00	0,00	0,00	0,00	0,00
175.5	0,00	0,00	0,00	0,00	0,00
175.6	0,05	0,00	0,00	0,00	0,00
175.7	0,51	0,00	0,00	0,00	0,00
175.8	0,28	0,00	0,00	0,00	0,00

Tabela 35:  $p$ -valores em favor da busca PLS para instâncias OP1

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSGAII</b>	<b>NSTA</b>	<b>AT</b>
6.1	1,00	1,00	1,00	1,00	1,00
6.2	1,00	1,00	1,00	1,00	1,00

Inst.	Tabu	NSGAI	NSTA	AT
6.3	1,00	1,00	0,97	1,00
6.4	1,00	1,00	1,00	1,00
6.5	0,99	0,99	0,16	0,99
6.6	1,00	1,00	1,00	1,00
6.7	1,00	1,00	0,31	1,00
6.8	1,00	1,00	0,41	1,00
6.9	1,00	1,00	1,00	1,00
6.10	1,00	1,00	1,00	1,00
7.1	1,00	0,99	1,00	1,00
7.2	1,00	1,00	0,95	1,00
7.3	1,00	1,00	1,00	1,00
7.4	1,00	0,99	0,57	1,00
7.5	1,00	0,51	0,20	1,00
7.6	1,00	1,00	1,00	1,00
7.7	1,00	1,00	1,00	1,00
7.8	1,00	0,05	1,00	1,00
7.9	1,00	1,00	0,85	1,00
7.10	1,00	1,00	1,00	1,00
8.1	1,00	1,00	0,01	1,00
8.2	1,00	0,00	0,07	1,00
8.3	1,00	1,00	0,00	0,35
8.4	1,00	1,00	0,00	0,99
8.5	1,00	1,00	0,99	1,00
8.6	1,00	0,00	0,02	0,71
8.7	1,00	1,00	0,63	1,00
8.8	1,00	0,00	0,00	0,03
8.9	1,00	1,00	1,00	1,00
8.10	1,00	0,07	1,00	1,00
9.1	1,00	0,97	1,00	1,00
9.2	1,00	0,48	0,00	0,01
9.3	1,00	0,00	0,00	0,19
9.4	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
9.5	0,95	0,13	0,00	1,00
9.6	1,00	0,12	0,00	1,00
9.7	1,00	0,00	0,05	1,00
9.8	1,00	1,00	1,00	1,00
9.9	1,00	0,02	0,00	0,00
9.10	1,00	1,00	1,00	1,00
10.1	1,00	0,00	0,00	0,00
10.2	1,00	0,00	1,00	1,00
10.3	0,60	0,00	0,00	0,00
10.4	1,00	0,00	0,03	0,02
10.5	1,00	0,00	0,99	0,99
10.6	1,00	0,00	1,00	1,00
10.7	1,00	0,01	0,45	1,00
10.8	1,00	0,00	0,00	0,41
10.9	1,00	0,00	0,00	0,00
10.10	1,00	0,00	0,00	0,00
11.1	0,23	0,00	0,00	0,00
11.2	1,00	0,00	0,03	0,75
11.3	1,00	0,00	0,00	0,17
11.4	1,00	0,00	1,00	1,00
11.5	0,77	0,00	0,00	0,00
11.6	1,00	0,00	1,00	1,00
11.7	1,00	0,00	0,00	0,28
11.8	1,00	0,00	0,00	0,00
11.9	1,00	0,00	0,00	0,00
11.10	1,00	0,00	0,00	0,00
12.1	1,00	0,00	0,00	0,00
12.2	1,00	0,00	0,00	0,11
12.3	0,94	0,00	0,00	0,00
12.4	0,87	0,00	0,00	0,00
12.5	1,00	0,00	0,00	0,00
12.6	1,00	0,00	0,00	0,02

Inst.	Tabu	NSGAI	NSTA	AT
12.7	0,99	0,00	0,02	0,07
12.8	0,99	0,00	0,00	0,00
12.9	1,00	0,00	0,00	0,00
12.10	1,00	0,00	0,00	0,00
13.1	1,00	0,00	0,34	0,42
13.2	1,00	0,00	0,00	0,00
13.3	1,00	0,00	0,06	0,82
13.4	0,39	0,00	0,00	0,00
13.5	1,00	0,00	0,00	0,00
13.6	0,93	0,00	0,00	0,00
13.7	1,00	0,00	0,00	0,00
13.8	1,00	0,00	0,00	0,00
13.9	0,99	0,00	0,00	0,00
13.10	0,85	0,00	0,00	0,00
14.1	0,00	0,00	0,00	0,00
14.2	0,01	0,00	0,00	0,00
14.3	1,00	0,00	0,00	0,00
14.4	0,99	0,00	1,00	0,95
14.5	0,06	0,00	0,00	0,00
14.6	0,23	0,00	0,00	0,00
14.7	0,89	0,00	0,00	0,00
14.8	0,01	0,00	0,00	0,00
14.9	0,86	0,00	0,00	0,00
14.10	0,99	0,00	0,00	0,00
15.1	0,80	0,00	0,00	0,00
15.2	0,91	0,00	0,00	0,00
15.3	1,00	0,00	0,00	0,00
15.4	0,97	0,00	0,00	0,00
15.5	1,00	0,00	0,00	0,00
15.6	0,22	0,00	0,00	0,00
15.7	1,00	0,00	0,00	0,00
15.8	0,90	0,00	0,00	0,00

Inst.	Tabu	NSGAI	NSTA	AT
15.9	1,00	0,00	0,00	0,00
15.10	1,00	0,00	0,00	0,00
16.1	0,62	0,00	0,00	0,00
16.2	0,99	0,00	0,00	0,00
16.3	0,96	0,00	0,00	0,00
16.4	0,00	0,00	0,00	0,00
16.5	0,91	0,00	0,00	0,00
16.6	0,96	0,00	0,00	0,00
16.7	1,00	0,00	0,16	0,00
16.8	0,12	0,00	0,00	0,00
16.9	0,96	0,00	0,00	0,00
16.10	0,98	0,00	0,01	0,00
17.1	0,56	0,00	0,00	0,00
17.2	0,40	0,00	0,00	0,00
17.3	0,48	0,00	0,00	0,00
17.4	0,32	0,00	0,00	0,00
17.5	0,67	0,00	0,00	0,00
17.6	1,00	0,00	0,00	0,00
17.7	0,84	0,00	0,00	0,00
17.8	0,51	0,00	0,00	0,00
17.9	0,01	0,00	0,00	0,00
17.10	1,00	0,00	0,00	0,00
18.1	0,00	0,00	0,00	0,00
18.2	0,63	0,00	0,00	0,00
18.3	0,92	0,00	0,00	0,00
18.4	0,16	0,00	0,00	0,00
18.5	0,01	0,00	0,00	0,00
18.6	0,68	0,00	0,00	0,00
18.7	1,00	0,00	0,00	0,00
18.8	0,63	0,00	0,00	0,00
18.9	0,01	0,00	0,00	0,00
18.10	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
20.1	0,77	0,00	0,00	0,00
20.2	0,88	0,00	0,00	0,00
20.3	0,76	0,00	0,00	0,00
20.4	0,13	0,00	0,00	0,00
20.5	1,00	0,00	0,00	0,00
20.6	0,63	0,00	0,00	0,00
20.7	1,00	0,00	0,00	0,00
20.8	0,04	0,00	0,00	0,00
20.9	0,35	0,00	0,00	0,00
20.10	0,34	0,00	0,00	0,00
30.1	0,00	0,00	0,00	0,00
30.2	0,00	0,00	0,00	0,00
30.3	0,00	0,00	0,00	0,00
30.4	0,00	0,00	0,00	0,00
30.5	0,62	0,00	0,00	0,00
30.6	0,00	0,00	0,00	0,00
30.7	0,00	0,00	0,00	0,00
30.8	0,00	0,00	0,00	0,00
30.9	0,00	0,00	0,00	0,00
30.10	0,02	0,00	0,00	0,00
50.1	0,00	0,00	0,00	0,00
50.2	0,00	0,00	0,00	0,00
50.3	0,00	0,00	0,00	0,00
50.4	0,00	0,00	0,00	0,00
50.5	0,00	0,00	0,00	0,00
50.6	0,00	0,00	0,00	0,00
50.7	0,00	0,00	0,00	0,00
50.8	0,00	0,00	0,00	0,00
50.9	0,00	0,00	0,00	0,00
50.10	0,00	0,00	0,00	0,00

Tabela 36:  $p$ -valores em favor da busca PLS para instâncias OP2

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	1,00	1,00	1,00	1,00
6.2	1,00	1,00	1,00	1,00
6.3	1,00	1,00	1,00	1,00
6.4	1,00	1,00	1,00	1,00
6.5	1,00	1,00	1,00	1,00
6.6	1,00	1,00	1,00	1,00
6.7	1,00	1,00	1,00	1,00
6.8	1,00	1,00	1,00	1,00
6.9	1,00	1,00	1,00	1,00
6.10	0,92	0,92	0,92	0,92
7.1	1,00	1,00	1,00	1,00
7.2	1,00	1,00	1,00	1,00
7.3	1,00	1,00	1,00	1,00
7.4	1,00	1,00	1,00	1,00
7.5	1,00	1,00	1,00	1,00
7.6	1,00	1,00	1,00	1,00
7.7	1,00	1,00	1,00	1,00
7.8	1,00	1,00	1,00	1,00
7.9	1,00	1,00	1,00	1,00
7.10	1,00	1,00	1,00	1,00
8.1	1,00	1,00	1,00	1,00
8.2	1,00	1,00	1,00	1,00
8.3	1,00	1,00	1,00	1,00
8.4	1,00	1,00	1,00	1,00
8.5	1,00	1,00	0,98	1,00
8.6	1,00	0,96	1,00	1,00
8.7	1,00	1,00	1,00	1,00
8.8	1,00	1,00	0,42	1,00
8.9	1,00	1,00	1,00	0,72
8.10	1,00	0,34	1,00	1,00
9.1	1,00	0,01	0,00	1,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
9.2	0,83	0,99	0,00	1,00
9.3	1,00	1,00	0,26	1,00
9.4	1,00	0,00	0,97	1,00
9.5	1,00	0,01	0,00	1,00
9.6	1,00	1,00	0,99	1,00
9.7	1,00	1,00	0,96	1,00
9.8	1,00	0,01	0,12	0,25
9.9	1,00	0,04	0,28	0,39
9.10	1,00	0,00	0,26	0,98
10.1	1,00	1,00	1,00	1,00
10.2	1,00	0,10	0,01	0,72
10.3	1,00	0,00	0,00	0,07
10.4	1,00	0,17	0,62	0,16
10.5	1,00	0,00	0,00	0,53
10.6	1,00	0,00	0,59	0,34
10.7	1,00	0,00	0,00	0,05
10.8	1,00	0,00	0,03	0,01
10.9	1,00	0,00	0,10	1,00
10.10	1,00	0,72	0,17	1,00
11.1	1,00	0,00	0,00	0,06
11.2	1,00	0,01	0,76	1,00
11.3	0,97	0,00	0,00	0,00
11.4	1,00	0,00	0,95	0,01
11.5	1,00	0,00	0,88	1,00
11.6	1,00	0,00	0,00	0,05
11.7	1,00	0,00	0,00	0,01
11.8	0,95	0,00	0,00	0,00
11.9	0,77	0,00	0,00	0,00
11.10	1,00	0,00	0,00	0,17
12.1	0,00	0,00	0,00	0,00
12.2	1,00	0,00	0,04	0,75
12.3	0,41	0,00	0,00	0,00



<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
12.4	1,00	0,00	0,00	0,00
12.5	1,00	0,00	0,00	0,00
12.6	1,00	0,27	1,00	0,00
12.7	1,00	0,00	0,00	0,13
12.8	1,00	0,00	0,00	0,00
12.9	0,96	0,00	0,00	0,00
12.10	1,00	0,00	0,00	0,00
13.1	1,00	0,00	0,10	1,00
13.2	1,00	0,00	1,00	0,00
13.3	1,00	0,21	0,01	0,36
13.4	0,00	0,00	0,00	0,00
13.5	1,00	0,05	1,00	0,33
13.6	1,00	0,00	0,00	0,11
13.7	1,00	0,00	0,00	0,00
13.8	1,00	0,00	0,00	0,00
13.9	0,96	0,00	0,00	0,18
13.10	1,00	0,00	0,00	0,00
14.1	1,00	0,00	0,00	0,00
14.2	1,00	0,00	0,13	0,89
14.3	0,46	0,00	0,00	0,00
14.4	1,00	0,00	0,00	0,15
14.5	0,99	0,00	0,00	0,01
14.6	1,00	0,00	0,00	0,00
14.7	1,00	0,00	0,44	0,00
14.8	1,00	0,00	0,11	0,00
14.9	1,00	0,00	0,00	0,00
14.10	0,71	0,00	0,00	0,00
15.1	1,00	0,00	0,67	0,21
15.2	1,00	0,00	0,01	0,00
15.3	1,00	0,00	0,03	0,00
15.4	1,00	0,00	0,14	0,00
15.5	0,89	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
15.6	1,00	0,00	0,84	0,82
15.7	1,00	0,00	0,00	0,00
15.8	1,00	0,00	0,00	0,08
15.9	1,00	0,00	0,00	0,00
15.10	1,00	0,00	0,00	0,00
16.1	1,00	0,00	0,46	0,00
16.2	1,00	0,00	0,00	0,00
16.3	1,00	0,00	0,00	0,00
16.4	1,00	0,00	1,00	0,00
16.5	1,00	0,05	1,00	1,00
16.6	1,00	0,00	0,00	0,00
16.7	1,00	0,00	0,00	0,00
16.8	1,00	0,00	0,00	0,00
16.9	1,00	0,37	1,00	0,97
16.10	1,00	0,96	1,00	1,00
17.1	1,00	0,00	0,00	0,00
17.2	1,00	0,00	0,00	0,00
17.3	1,00	0,00	0,00	0,00
17.4	1,00	0,00	0,00	0,00
17.5	1,00	0,00	0,00	0,00
17.6	1,00	0,00	0,27	0,00
17.7	1,00	1,00	1,00	1,00
17.8	1,00	0,00	0,00	0,00
17.9	1,00	0,00	0,00	0,00
17.10	1,00	1,00	1,00	0,99
18.1	0,11	0,00	0,00	0,00
18.2	1,00	0,00	0,00	0,00
18.3	1,00	0,00	0,00	0,00
18.4	1,00	0,00	0,00	0,00
18.5	1,00	0,00	0,00	0,00
18.6	1,00	0,00	0,00	0,00
18.7	1,00	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
18.8	1,00	0,00	0,00	0,00
18.9	1,00	0,00	0,00	0,00
18.10	1,00	0,00	0,00	0,00
20.1	1,00	0,00	0,00	0,00
20.2	1,00	0,00	0,00	0,00
20.3	1,00	0,00	0,00	0,00
20.4	1,00	0,00	0,00	0,00
20.5	1,00	0,00	0,00	0,00
20.6	1,00	0,00	0,00	0,00
20.7	1,00	0,00	0,00	0,00
20.8	0,98	0,00	0,00	0,00
20.9	1,00	0,00	0,00	0,00
20.10	1,00	0,00	0,00	0,00
30.1	0,21	0,00	0,00	0,00
30.2	0,55	0,00	0,00	0,00
30.3	0,96	0,00	0,00	0,00
30.4	0,51	0,00	0,00	0,00
30.5	0,00	0,00	0,00	0,00
30.6	1,00	0,00	0,00	0,00
30.7	0,74	0,00	0,00	0,00
30.8	0,96	0,00	0,00	0,00
30.9	1,00	0,00	0,00	0,00
30.10	0,00	0,00	0,00	0,00
50.1	0,00	0,00	0,00	0,00
50.2	0,00	0,00	0,00	0,00
50.3	0,00	0,00	0,00	0,00
50.4	0,02	0,00	0,00	0,00
50.5	0,01	0,00	0,00	0,00
50.6	0,07	0,00	0,00	0,00
50.7	0,00	0,00	0,00	0,00
50.8	0,32	0,00	0,00	0,00
50.9	0,01	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
50.10	0,00	0,00	0,00	0,00

Tabela 37:  $p$ -valores em favor da busca PLS para instâncias OP3

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	1,00	1,00	1,00	1,00
6.2	1,00	1,00	1,00	1,00
6.3	1,00	1,00	1,00	1,00
6.4	1,00	1,00	1,00	1,00
6.5	1,00	1,00	1,00	1,00
6.6	1,00	1,00	1,00	1,00
6.7	1,00	1,00	1,00	1,00
6.8	1,00	1,00	1,00	1,00
6.9	1,00	1,00	1,00	1,00
6.10	1,00	1,00	1,00	1,00
7.1	1,00	1,00	1,00	1,00
7.2	1,00	1,00	0,12	1,00
7.3	1,00	1,00	1,00	1,00
7.4	1,00	1,00	1,00	1,00
7.5	1,00	1,00	1,00	1,00
7.6	1,00	1,00	1,00	1,00
7.7	1,00	0,91	0,96	1,00
7.8	1,00	1,00	1,00	1,00
7.9	1,00	1,00	1,00	1,00
7.10	1,00	0,68	1,00	1,00
8.1	1,00	0,65	1,00	1,00
8.2	1,00	0,43	1,00	1,00
8.3	1,00	0,09	0,91	1,00
8.4	1,00	0,93	1,00	1,00
8.5	1,00	0,06	0,98	1,00
8.6	1,00	1,00	1,00	1,00
8.7	1,00	0,89	1,00	1,00
8.8	1,00	1,00	1,00	1,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
8.9	1,00	0,74	0,99	1,00
8.10	1,00	0,02	1,00	1,00
9.1	1,00	0,00	0,99	1,00
9.2	1,00	0,01	1,00	1,00
9.3	1,00	0,00	0,88	0,98
9.4	1,00	0,06	0,82	1,00
9.5	1,00	0,43	1,00	1,00
9.6	1,00	0,01	1,00	1,00
9.7	1,00	0,00	0,99	1,00
9.8	0,99	0,00	0,90	1,00
9.9	1,00	0,34	0,99	1,00
9.10	1,00	0,00	1,00	1,00
10.1	1,00	0,00	0,99	1,00
10.2	1,00	1,00	1,00	1,00
10.3	1,00	0,00	0,01	0,24
10.4	1,00	0,01	1,00	1,00
10.5	1,00	0,00	0,88	0,99
10.6	0,94	0,00	0,00	0,00
10.7	1,00	0,00	1,00	1,00
10.8	0,99	0,00	0,00	0,00
10.9	1,00	0,00	0,08	1,00
10.10	0,76	0,00	0,00	0,00
11.1	1,00	0,00	0,03	1,00
11.2	1,00	0,01	1,00	1,00
11.3	1,00	0,00	0,07	0,91
11.4	1,00	0,00	0,04	0,14
11.5	1,00	0,00	1,00	1,00
11.6	1,00	0,00	1,00	1,00
11.7	0,75	0,00	0,88	1,00
11.8	0,94	0,00	0,00	1,00
11.9	0,90	0,00	0,00	0,00
11.10	1,00	0,00	0,00	0,04

Inst.	Tabu	NSGAII	NSTA	AT
12.1	1,00	0,00	0,00	1,00
12.2	1,00	0,00	0,00	0,23
12.3	0,92	0,00	0,00	0,01
12.4	1,00	0,00	0,00	1,00
12.5	1,00	0,00	0,00	0,00
12.6	0,23	0,00	0,00	0,00
12.7	1,00	0,00	0,00	0,01
12.8	0,02	0,00	0,00	0,00
12.9	1,00	0,00	0,00	0,27
12.10	0,85	0,00	0,00	0,00
13.1	1,00	0,00	0,00	0,00
13.2	0,96	0,00	0,00	0,00
13.3	1,00	0,00	0,00	0,00
13.4	0,86	0,00	0,00	0,00
13.5	0,87	0,00	0,03	0,85
13.6	0,88	0,00	0,40	0,97
13.7	0,85	0,00	0,00	0,03
13.8	0,98	0,00	0,00	0,00
13.9	0,28	0,00	0,00	0,40
13.10	0,73	0,00	0,00	0,00
14.1	0,78	0,00	0,00	0,00
14.2	1,00	0,00	0,35	0,90
14.3	0,86	0,00	0,01	0,05
14.4	0,99	0,00	0,00	0,98
14.5	0,03	0,00	0,00	0,00
14.6	0,99	0,05	0,98	1,00
14.7	0,93	0,00	0,00	0,00
14.8	0,64	0,00	0,00	1,00
14.9	1,00	0,00	0,18	0,66
14.10	0,12	0,00	0,00	0,00
15.1	0,37	0,00	0,00	0,00
15.2	0,92	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
15.3	0,40	0,00	0,00	0,00
15.4	0,87	0,00	0,00	0,07
15.5	0,00	0,00	0,00	0,00
15.6	0,88	0,00	0,60	0,13
15.7	0,80	0,00	0,00	0,00
15.8	0,66	0,00	0,00	0,00
15.9	1,00	0,00	0,00	0,00
15.10	0,00	0,00	0,00	0,00
16.1	0,00	0,00	0,00	0,00
16.2	0,22	0,00	0,00	0,00
16.3	0,16	0,00	0,00	0,00
16.4	0,05	0,00	0,00	0,00
16.5	0,17	0,00	0,00	0,00
16.6	0,77	0,00	0,00	0,31
16.7	0,17	0,00	0,00	0,00
16.8	0,81	0,00	0,00	0,00
16.9	0,99	0,00	0,00	1,00
16.10	0,09	0,00	0,00	0,00
17.1	0,00	0,00	0,00	0,00
17.2	0,00	0,00	0,00	0,00
17.3	0,99	0,00	0,00	0,80
17.4	0,00	0,00	0,00	0,00
17.5	0,51	0,00	0,00	0,76
17.6	0,00	0,00	0,00	0,00
17.7	0,12	0,00	0,00	0,00
17.8	0,01	0,00	0,00	0,00
17.9	0,00	0,00	0,00	0,00
17.10	0,13	0,00	0,00	0,00
18.1	0,01	0,00	0,00	0,00
18.2	0,00	0,00	0,00	0,00
18.3	0,08	0,00	0,00	0,00
18.4	0,00	0,00	0,00	0,00

<b>Inst.</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
18.5	0,00	0,00	0,00	0,00
18.6	0,00	0,00	0,16	0,00
18.7	0,00	0,00	0,00	0,00
18.8	0,00	0,00	0,00	0,00
18.9	0,82	0,00	0,00	0,00
18.10	0,00	0,00	0,00	0,01
20.1	1,00	1,00	1,00	1,00
20.2	1,00	1,00	1,00	1,00
20.3	1,00	1,00	1,00	1,00
20.4	1,00	1,00	1,00	1,00
20.5	1,00	1,00	1,00	1,00
20.6	1,00	1,00	1,00	1,00
20.7	1,00	1,00	1,00	1,00
20.8	1,00	1,00	1,00	1,00
20.9	1,00	1,00	1,00	1,00
20.10	0,99	0,99	0,99	0,99
30.1	0,00	0,00	0,00	0,00
30.2	0,00	0,00	0,00	0,00
30.3	0,00	0,00	0,00	0,00
30.4	0,00	0,00	0,07	0,27
30.5	0,00	0,00	0,00	0,00
30.6	0,00	0,00	0,00	0,00
30.7	0,00	0,00	0,00	0,00
30.8	0,00	0,00	0,00	0,00
30.9	0,03	0,00	0,00	0,00
30.10	0,00	0,00	0,00	0,00
50.1	0,00	0,00	0,00	0,00
50.2	0,00	0,00	0,00	0,00
50.3	0,00	0,00	0,00	0,00
50.4	0,00	0,00	0,00	0,00
50.5	0,00	0,00	0,00	0,00
50.6	0,00	0,00	0,00	0,00



<b>Inst.</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>NSTA</b>	<b>AT</b>
50.7	0,00	0,00	0,00	0,00
50.8	0,00	0,00	0,00	0,00
50.9	0,00	0,00	0,00	0,00
50.10	0,00	0,00	0,00	0,00

As tabelas 38, 39, 40 e 41 contêm os tempos médios de execução(em segundos) dos algoritmos (meta)heurísticos apresentados nesta Tese para os grupos de instâncias: CP, OP1, OP2 e OP3. Vale lembrar que o tempo de execução de todos os algoritmos apresentados no Capítulo 5 foi limitado a 150s, ou seja, 2,5min.

Tabela 38: Tempos computacionais médios (s) para as instâncias CP

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAII</b>	<b>NSTA</b>	<b>AT</b>
10.1	0,00	0,00	0,03	0,02	0,01
10.2	0,00	0,00	0,03	0,02	0,01
10.3	0,00	0,00	0,03	0,02	0,01
10.4	0,00	0,00	0,03	0,02	0,01
10.5	0,00	0,01	0,04	0,03	0,01
10.6	0,00	0,01	0,04	0,03	0,01
10.7	0,00	0,01	0,04	0,03	0,01
10.8	0,00	0,01	0,04	0,03	0,01
10.9	0,00	0,02	0,05	0,03	0,01
10.10	0,00	0,02	0,04	0,03	0,01
10.11	0,00	0,02	0,04	0,03	0,01
10.12	0,00	0,02	0,04	0,03	0,01
15.1	0,00	0,02	0,07	0,05	0,02
15.2	0,00	0,02	0,07	0,05	0,02
15.3	0,00	0,02	0,07	0,05	0,02
15.4	0,00	0,02	0,07	0,05	0,02
15.5	0,01	0,04	0,08	0,06	0,02
15.6	0,01	0,04	0,08	0,06	0,02
15.7	0,00	0,04	0,07	0,06	0,02
15.8	0,01	0,04	0,07	0,06	0,02

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
15.9	0,01	0,06	0,08	0,06	0,02
15.10	0,01	0,06	0,08	0,06	0,02
15.11	0,02	0,06	0,08	0,06	0,02
15.12	0,02	0,06	0,08	0,06	0,02
20.1	0,01	0,07	0,21	0,17	0,05
20.2	0,01	0,07	0,21	0,17	0,05
20.3	0,01	0,07	0,20	0,16	0,05
20.4	0,02	0,08	0,19	0,16	0,05
20.5	0,03	0,14	0,23	0,18	0,06
20.6	0,03	0,14	0,23	0,19	0,06
20.7	0,04	0,14	0,22	0,17	0,06
20.8	0,05	0,15	0,22	0,18	0,06
20.9	0,05	0,21	0,24	0,18	0,07
20.10	0,06	0,21	0,24	0,20	0,07
20.11	0,08	0,22	0,23	0,17	0,07
20.12	0,09	0,22	0,23	0,19	0,07
25.1	0,02	0,12	0,23	0,20	0,07
25.2	0,03	0,12	0,23	0,20	0,07
25.3	0,03	0,12	0,22	0,18	0,07
25.4	0,04	0,12	0,22	0,20	0,07
25.5	0,07	0,22	0,27	0,22	0,09
25.6	0,07	0,22	0,26	0,22	0,09
25.7	0,11	0,23	0,26	0,22	0,09
25.8	0,13	0,23	0,25	0,22	0,09
25.9	0,12	0,34	0,32	0,28	0,12
25.10	0,14	0,34	0,31	0,26	0,12
25.11	0,21	0,35	0,30	0,27	0,12
25.12	0,25	0,36	0,30	0,27	0,12
30.1	0,07	0,18	0,24	0,23	0,10
30.2	0,09	0,18	0,24	0,23	0,10
30.3	0,09	0,19	0,24	0,23	0,10
30.4	0,11	0,19	0,24	0,23	0,10

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
30.5	0,15	0,35	0,32	0,29	0,14
30.6	0,18	0,35	0,31	0,30	0,14
30.7	0,23	0,35	0,31	0,28	0,14
30.8	0,28	0,35	0,31	0,28	0,14
30.9	0,25	0,57	0,41	0,34	0,20
30.10	0,32	0,57	0,41	0,35	0,20
30.11	0,36	0,56	0,39	0,36	0,20
30.12	0,53	0,57	0,38	0,34	0,20
35.1	0,17	0,25	0,29	0,27	0,13
35.2	0,21	0,25	0,29	0,26	0,13
35.3	0,26	0,26	0,28	0,26	0,13
35.4	0,29	0,27	0,28	0,26	0,13
35.5	0,30	0,53	0,39	0,37	0,20
35.6	0,38	0,52	0,39	0,37	0,20
35.7	0,55	0,53	0,38	0,36	0,20
35.8	0,75	0,54	0,38	0,36	0,21
35.9	0,50	0,86	0,54	0,49	0,32
35.10	0,62	0,87	0,53	0,48	0,32
35.11	0,89	0,81	0,51	0,47	0,32
35.12	1,16	0,83	0,51	0,47	0,32
40.1	0,27	0,63	0,72	0,66	0,26
40.2	0,35	0,64	0,70	0,64	0,26
40.3	0,42	0,65	0,69	0,65	0,26
40.4	0,61	0,67	0,69	0,66	0,26
40.5	0,64	1,32	0,98	0,89	0,40
40.6	0,83	1,31	0,95	0,84	0,40
40.7	1,01	1,33	0,91	0,84	0,40
40.8	1,44	1,36	0,91	0,84	0,40
40.9	0,89	2,02	1,25	1,14	0,61
40.10	1,19	2,01	1,23	1,11	0,61
40.11	1,80	2,01	1,13	1,03	0,61
40.12	2,52	2,05	1,12	1,04	0,62

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
45.1	0,65	0,82	0,78	0,72	0,33
45.2	0,70	0,82	0,77	0,69	0,33
45.3	0,86	0,84	0,75	0,70	0,33
45.4	1,15	0,86	0,75	0,69	0,33
45.5	1,25	1,79	1,07	0,94	0,55
45.6	1,61	1,77	1,05	0,96	0,55
45.7	2,22	1,77	1,01	0,94	0,55
45.8	2,95	1,78	1,01	0,94	0,55
45.9	1,66	2,71	1,47	1,37	0,88
45.10	2,29	2,69	1,46	1,39	0,88
45.11	3,24	2,71	1,40	1,32	0,88
45.12	4,35	2,73	1,40	1,33	0,89
50.1	1,02	1,05	0,87	0,82	0,41
50.2	1,23	1,05	0,86	0,82	0,41
50.3	1,48	1,05	0,84	0,82	0,41
50.4	1,91	1,09	0,84	0,82	0,41
50.5	1,82	2,28	1,29	1,24	0,74
50.6	2,18	2,24	1,27	1,22	0,74
50.7	3,37	2,30	1,23	1,18	0,75
50.8	4,83	2,32	1,23	1,17	0,75
50.9	3,34	3,90	1,88	1,75	1,24
50.10	4,37	3,87	1,87	1,76	1,25
50.11	7,18	3,83	1,77	1,70	1,25
50.12	9,49	3,88	1,78	1,70	1,26
75.1	19,38	2,98	1,61	1,62	1,14
75.2	17,65	2,97	1,58	1,62	1,14
75.3	16,33	2,99	1,59	1,60	1,14
75.4	20,16	3,07	1,55	1,62	1,14
75.5	52,17	7,18	3,36	3,30	2,77
75.6	48,66	7,20	3,42	3,33	2,79
75.7	62,91	7,28	3,42	3,28	2,77
75.8	55,02	7,09	3,45	3,31	2,80

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
75.9	88,35	12,91	6,11	5,92	5,27
75.10	78,22	12,39	6,10	5,98	5,33
75.11	85,01	13,13	6,05	5,94	5,28
75.12	94,66	12,51	6,10	5,96	5,35
100.1	94,91	6,62	3,08	3,25	2,64
100.2	97,56	6,74	3,08	3,23	2,66
100.3	124,75	6,58	3,11	3,23	2,65
100.4	95,25	6,62	3,11	3,23	2,66
100.5	150,20	17,23	8,50	8,46	7,69
100.6	150,19	17,06	8,60	8,53	7,75
100.7	150,01	17,29	8,54	8,51	7,70
100.8	149,99	17,09	8,62	8,56	7,82
100.9	149,98	30,75	17,03	16,78	15,61
100.10	149,99	30,78	17,23	16,91	15,80
100.11	149,98	31,27	17,13	16,79	15,73
100.12	149,99	31,29	17,27	16,95	15,89
125.1	150,00	13,13	6,09	6,31	5,61
125.2	149,93	12,71	6,14	6,35	5,67
125.3	150,04	12,73	6,09	6,31	5,62
125.4	150,05	13,28	6,11	6,37	5,67
125.5	150,00	34,80	19,48	19,31	18,17
125.6	150,01	34,77	19,59	19,48	18,34
125.7	149,98	35,89	19,55	19,40	18,30
125.8	149,98	34,57	19,68	19,56	18,43
125.9	149,96	65,21	40,73	39,96	38,10
125.10	149,95	65,46	41,03	40,32	38,46
125.11	149,96	65,42	40,91	40,08	38,24
125.12	149,96	64,72	41,33	40,69	38,88
150.1	149,88	22,29	11,28	11,61	10,79
150.2	149,93	22,18	11,57	11,76	10,82
150.3	150,01	22,26	11,41	11,65	10,83
150.4	149,58	21,66	11,52	11,78	10,86

Inst.	PLS	Tabu	NSGAI	NSTA	AT
150.5	149,99	63,81	39,70	39,03	37,00
150.6	149,98	64,19	39,72	39,35	37,37
150.7	149,94	65,04	39,56	39,16	37,24
150.8	149,95	64,08	40,11	39,66	37,56
175.1	149,99	35,83	19,78	20,09	18,90
175.2	149,99	36,24	20,01	20,19	19,01
175.3	149,99	35,38	19,87	20,20	18,90
175.4	149,99	35,45	19,99	20,29	19,16
175.5	149,95	109,38	73,11	71,87	68,82
175.6	149,92	110,80	73,89	72,90	69,64
175.7	149,92	110,63	71,83	70,62	66,58
175.8	149,93	104,07	68,94	68,05	66,71

Tabela 39: Tempos computacionais médios (s) para as instâncias OP1

Inst.	PLS	Tabu	NSGAI	NSTA	AT
6.1	0,00	0,00	0,03	0,02	0,00
6.2	0,00	0,00	0,03	0,02	0,00
6.3	0,00	0,00	0,04	0,02	0,00
6.4	0,00	0,00	0,03	0,03	0,00
6.5	0,00	0,00	0,04	0,02	0,00
6.6	0,00	0,00	0,04	0,02	0,00
6.7	0,00	0,00	0,04	0,03	0,00
6.8	0,00	0,00	0,03	0,02	0,00
6.9	0,00	0,00	0,04	0,02	0,00
6.10	0,00	0,00	0,03	0,02	0,00
7.1	0,00	0,01	0,04	0,03	0,01
7.2	0,00	0,00	0,04	0,03	0,00
7.3	0,00	0,00	0,04	0,03	0,00
7.4	0,00	0,00	0,04	0,03	0,00
7.5	0,00	0,01	0,04	0,02	0,00
7.6	0,00	0,00	0,04	0,03	0,01
7.7	0,00	0,00	0,04	0,03	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
7.8	0,00	0,00	0,04	0,03	0,00
7.9	0,00	0,01	0,03	0,02	0,00
7.10	0,00	0,01	0,03	0,03	0,00
8.1	0,00	0,01	0,04	0,03	0,01
8.2	0,00	0,01	0,04	0,03	0,01
8.3	0,00	0,01	0,04	0,03	0,01
8.4	0,00	0,01	0,04	0,03	0,01
8.5	0,00	0,01	0,04	0,03	0,01
8.6	0,00	0,01	0,04	0,03	0,01
8.7	0,00	0,01	0,04	0,03	0,01
8.8	0,00	0,01	0,04	0,03	0,01
8.9	0,00	0,01	0,04	0,03	0,01
8.10	0,00	0,01	0,04	0,03	0,01
9.1	0,00	0,01	0,04	0,03	0,01
9.2	0,00	0,01	0,04	0,03	0,01
9.3	0,00	0,01	0,04	0,03	0,01
9.4	0,00	0,01	0,04	0,03	0,01
9.5	0,00	0,01	0,04	0,03	0,01
9.6	0,00	0,01	0,04	0,03	0,01
9.7	0,00	0,01	0,04	0,03	0,01
9.8	0,00	0,01	0,04	0,03	0,01
9.9	0,00	0,01	0,04	0,03	0,01
9.10	0,00	0,01	0,04	0,03	0,01
10.1	0,00	0,01	0,04	0,03	0,01
10.2	0,00	0,01	0,04	0,03	0,01
10.3	0,00	0,01	0,04	0,03	0,01
10.4	0,00	0,01	0,04	0,03	0,01
10.5	0,00	0,01	0,04	0,03	0,01
10.6	0,00	0,01	0,04	0,03	0,01
10.7	0,00	0,01	0,04	0,03	0,01
10.8	0,00	0,02	0,04	0,03	0,01
10.9	0,00	0,01	0,04	0,03	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
10.10	0,00	0,01	0,04	0,03	0,01
11.1	0,01	0,02	0,05	0,03	0,01
11.2	0,01	0,02	0,04	0,03	0,01
11.3	0,01	0,02	0,04	0,03	0,01
11.4	0,00	0,02	0,04	0,03	0,01
11.5	0,01	0,02	0,04	0,03	0,01
11.6	0,00	0,02	0,04	0,03	0,01
11.7	0,00	0,02	0,05	0,03	0,01
11.8	0,01	0,02	0,04	0,03	0,01
11.9	0,00	0,02	0,04	0,04	0,01
11.10	0,01	0,02	0,04	0,04	0,01
12.1	0,01	0,02	0,05	0,04	0,01
12.2	0,01	0,02	0,05	0,04	0,01
12.3	0,01	0,02	0,05	0,04	0,01
12.4	0,01	0,02	0,05	0,04	0,01
12.5	0,01	0,02	0,05	0,04	0,01
12.6	0,01	0,02	0,05	0,03	0,01
12.7	0,01	0,02	0,05	0,04	0,01
12.8	0,01	0,02	0,05	0,04	0,01
12.9	0,00	0,02	0,05	0,04	0,01
12.10	0,01	0,02	0,05	0,04	0,01
13.1	0,01	0,02	0,05	0,03	0,01
13.2	0,01	0,02	0,05	0,04	0,01
13.3	0,01	0,02	0,05	0,04	0,01
13.4	0,01	0,02	0,05	0,04	0,01
13.5	0,01	0,02	0,05	0,04	0,01
13.6	0,01	0,02	0,05	0,03	0,01
13.7	0,01	0,02	0,05	0,04	0,01
13.8	0,01	0,02	0,05	0,04	0,01
13.9	0,01	0,02	0,05	0,04	0,01
13.10	0,01	0,02	0,05	0,04	0,01
14.1	0,01	0,03	0,05	0,04	0,02



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
14.2	0,01	0,03	0,05	0,04	0,02
14.3	0,01	0,03	0,05	0,04	0,02
14.4	0,01	0,03	0,05	0,04	0,02
14.5	0,01	0,03	0,05	0,04	0,02
14.6	0,01	0,03	0,05	0,04	0,02
14.7	0,00	0,03	0,05	0,04	0,02
14.8	0,01	0,03	0,05	0,04	0,02
14.9	0,01	0,03	0,05	0,04	0,02
14.10	0,01	0,03	0,05	0,04	0,02
15.1	0,01	0,05	0,07	0,06	0,02
15.2	0,01	0,05	0,07	0,06	0,02
15.3	0,01	0,05	0,07	0,06	0,02
15.4	0,01	0,05	0,07	0,06	0,02
15.5	0,01	0,05	0,07	0,06	0,02
15.6	0,01	0,05	0,07	0,06	0,02
15.7	0,01	0,05	0,07	0,06	0,02
15.8	0,02	0,05	0,07	0,06	0,02
15.9	0,01	0,05	0,07	0,06	0,02
15.10	0,01	0,05	0,07	0,06	0,02
16.1	0,02	0,06	0,07	0,07	0,02
16.2	0,02	0,06	0,07	0,06	0,02
16.3	0,02	0,06	0,08	0,06	0,02
16.4	0,02	0,06	0,08	0,06	0,02
16.5	0,01	0,06	0,08	0,07	0,02
16.6	0,02	0,06	0,07	0,06	0,02
16.7	0,02	0,06	0,08	0,06	0,02
16.8	0,01	0,06	0,07	0,06	0,02
16.9	0,02	0,06	0,08	0,06	0,02
16.10	0,03	0,06	0,08	0,06	0,02
17.1	0,03	0,07	0,08	0,06	0,03
17.2	0,03	0,07	0,08	0,06	0,03
17.3	0,03	0,07	0,08	0,07	0,03

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
17.4	0,02	0,07	0,08	0,07	0,03
17.5	0,02	0,07	0,08	0,07	0,03
17.6	0,02	0,07	0,08	0,06	0,03
17.7	0,03	0,07	0,08	0,06	0,03
17.8	0,03	0,07	0,08	0,07	0,03
17.9	0,03	0,07	0,08	0,07	0,03
17.10	0,03	0,07	0,08	0,06	0,03
18.1	0,03	0,08	0,08	0,07	0,03
18.2	0,03	0,08	0,08	0,07	0,03
18.3	0,03	0,08	0,08	0,07	0,03
18.4	0,03	0,08	0,08	0,07	0,03
18.5	0,03	0,08	0,08	0,07	0,03
18.6	0,04	0,08	0,08	0,07	0,03
18.7	0,03	0,08	0,08	0,07	0,03
18.8	0,03	0,08	0,08	0,07	0,03
18.9	0,04	0,08	0,08	0,07	0,03
18.10	0,02	0,08	0,08	0,07	0,03
20.1	0,05	0,20	0,23	0,20	0,06
20.2	0,04	0,20	0,22	0,18	0,06
20.3	0,07	0,19	0,23	0,19	0,06
20.4	0,06	0,19	0,22	0,16	0,06
20.5	0,04	0,18	0,22	0,18	0,06
20.6	0,06	0,19	0,23	0,19	0,06
20.7	0,06	0,18	0,22	0,18	0,06
20.8	0,06	0,20	0,22	0,18	0,06
20.9	0,06	0,20	0,23	0,18	0,06
20.10	0,05	0,18	0,23	0,20	0,06
30.1	0,32	0,49	0,31	0,28	0,13
30.2	0,46	0,48	0,31	0,27	0,13
30.3	0,46	0,51	0,30	0,28	0,13
30.4	0,38	0,47	0,31	0,28	0,13
30.5	0,42	0,50	0,31	0,27	0,13

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
30.6	0,45	0,51	0,30	0,27	0,13
30.7	0,42	0,51	0,31	0,26	0,13
30.8	0,32	0,46	0,31	0,27	0,13
30.9	0,33	0,44	0,31	0,29	0,13
30.10	0,27	0,47	0,31	0,29	0,13
50.1	6,03	3,17	1,22	1,11	0,70
50.2	5,42	2,99	1,21	1,12	0,70
50.3	7,10	3,31	1,21	1,11	0,70
50.4	5,82	3,10	1,21	1,13	0,70
50.5	7,00	3,33	1,22	1,16	0,70
50.6	6,34	3,19	1,22	1,12	0,70
50.7	6,47	3,16	1,23	1,12	0,70
50.8	6,87	3,00	1,20	1,09	0,70
50.9	6,46	3,14	1,23	1,12	0,71
50.10	5,72	3,34	1,24	1,14	0,70

Tabela 40: Tempos computacionais médios (s) para as instâncias OP2

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,01	0,03	0,02	0,00
6.2	0,00	0,01	0,03	0,02	0,00
6.3	0,00	0,01	0,03	0,02	0,00
6.4	0,00	0,01	0,03	0,02	0,00
6.5	0,00	0,01	0,03	0,02	0,00
6.6	0,00	0,00	0,04	0,02	0,00
6.7	0,00	0,01	0,03	0,02	0,00
6.8	0,00	0,01	0,03	0,02	0,00
6.9	0,00	0,01	0,04	0,02	0,00
6.10	0,00	0,01	0,04	0,02	0,00
7.1	0,00	0,01	0,04	0,02	0,01
7.2	0,00	0,01	0,03	0,02	0,01
7.3	0,00	0,01	0,04	0,02	0,00
7.4	0,00	0,01	0,03	0,02	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
7.5	0,00	0,01	0,03	0,03	0,00
7.6	0,00	0,01	0,03	0,02	0,01
7.7	0,00	0,01	0,03	0,02	0,01
7.8	0,00	0,01	0,04	0,03	0,00
7.9	0,00	0,01	0,03	0,02	0,01
7.10	0,00	0,01	0,04	0,02	0,00
8.1	0,00	0,01	0,04	0,03	0,01
8.2	0,00	0,01	0,04	0,03	0,01
8.3	0,00	0,01	0,04	0,03	0,01
8.4	0,00	0,01	0,04	0,03	0,01
8.5	0,00	0,01	0,04	0,03	0,01
8.6	0,00	0,01	0,04	0,03	0,01
8.7	0,00	0,01	0,04	0,03	0,01
8.8	0,00	0,01	0,04	0,02	0,01
8.9	0,00	0,01	0,04	0,03	0,01
8.10	0,00	0,01	0,04	0,02	0,01
9.1	0,00	0,01	0,04	0,03	0,01
9.2	0,00	0,01	0,04	0,03	0,01
9.3	0,00	0,01	0,04	0,03	0,01
9.4	0,00	0,01	0,04	0,03	0,01
9.5	0,00	0,01	0,04	0,03	0,01
9.6	0,00	0,01	0,04	0,03	0,01
9.7	0,00	0,01	0,04	0,03	0,01
9.8	0,00	0,01	0,04	0,03	0,01
9.9	0,00	0,01	0,04	0,03	0,01
9.10	0,00	0,01	0,04	0,03	0,01
10.1	0,01	0,01	0,04	0,03	0,01
10.2	0,01	0,01	0,04	0,03	0,01
10.3	0,00	0,01	0,04	0,03	0,01
10.4	0,00	0,01	0,04	0,03	0,01
10.5	0,01	0,01	0,04	0,03	0,01
10.6	0,00	0,01	0,04	0,03	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
10.7	0,00	0,01	0,04	0,03	0,01
10.8	0,00	0,01	0,04	0,03	0,01
10.9	0,00	0,01	0,04	0,03	0,01
10.10	0,00	0,01	0,04	0,03	0,01
11.1	0,01	0,02	0,04	0,03	0,01
11.2	0,00	0,02	0,05	0,03	0,01
11.3	0,01	0,02	0,04	0,03	0,01
11.4	0,01	0,02	0,04	0,03	0,01
11.5	0,00	0,02	0,04	0,03	0,01
11.6	0,00	0,02	0,05	0,03	0,01
11.7	0,01	0,02	0,04	0,03	0,01
11.8	0,00	0,02	0,04	0,03	0,01
11.9	0,00	0,02	0,04	0,03	0,01
11.10	0,00	0,02	0,04	0,03	0,01
12.1	0,01	0,02	0,05	0,04	0,02
12.2	0,00	0,02	0,05	0,03	0,01
12.3	0,01	0,02	0,05	0,04	0,01
12.4	0,01	0,02	0,05	0,04	0,01
12.5	0,00	0,02	0,05	0,03	0,01
12.6	0,01	0,02	0,04	0,03	0,02
12.7	0,00	0,02	0,05	0,03	0,01
12.8	0,00	0,02	0,05	0,03	0,02
12.9	0,00	0,02	0,05	0,03	0,01
12.10	0,00	0,02	0,05	0,03	0,01
13.1	0,01	0,02	0,04	0,04	0,02
13.2	0,01	0,02	0,05	0,04	0,02
13.3	0,01	0,02	0,05	0,03	0,02
13.4	0,02	0,03	0,05	0,04	0,02
13.5	0,01	0,02	0,04	0,03	0,02
13.6	0,01	0,02	0,05	0,03	0,02
13.7	0,01	0,02	0,05	0,04	0,02
13.8	0,01	0,02	0,05	0,03	0,02

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
13.9	0,00	0,02	0,05	0,03	0,02
13.10	0,01	0,02	0,05	0,04	0,02
14.1	0,01	0,02	0,05	0,04	0,02
14.2	0,01	0,02	0,05	0,04	0,02
14.3	0,01	0,03	0,05	0,04	0,02
14.4	0,01	0,02	0,05	0,04	0,02
14.5	0,00	0,02	0,05	0,04	0,02
14.6	0,01	0,02	0,05	0,04	0,02
14.7	0,01	0,03	0,05	0,04	0,02
14.8	0,01	0,02	0,05	0,04	0,02
14.9	0,01	0,03	0,05	0,04	0,02
14.10	0,01	0,03	0,05	0,04	0,02
15.1	0,02	0,05	0,07	0,05	0,03
15.2	0,01	0,05	0,07	0,06	0,02
15.3	0,02	0,04	0,07	0,06	0,02
15.4	0,02	0,05	0,07	0,05	0,02
15.5	0,01	0,04	0,08	0,05	0,02
15.6	0,01	0,05	0,07	0,05	0,02
15.7	0,01	0,05	0,07	0,05	0,02
15.8	0,01	0,04	0,07	0,05	0,02
15.9	0,01	0,04	0,07	0,05	0,02
15.10	0,02	0,05	0,07	0,06	0,02
16.1	0,02	0,05	0,07	0,06	0,03
16.2	0,03	0,05	0,07	0,06	0,03
16.3	0,02	0,05	0,08	0,06	0,03
16.4	0,01	0,05	0,07	0,06	0,03
16.5	0,01	0,05	0,07	0,05	0,03
16.6	0,02	0,06	0,07	0,05	0,03
16.7	0,02	0,05	0,08	0,06	0,03
16.8	0,02	0,06	0,08	0,05	0,03
16.9	0,02	0,05	0,07	0,06	0,03
16.10	0,01	0,05	0,07	0,06	0,03

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
17.1	0,02	0,05	0,08	0,06	0,03
17.2	0,03	0,06	0,08	0,06	0,03
17.3	0,03	0,06	0,08	0,06	0,03
17.4	0,02	0,06	0,08	0,06	0,03
17.5	0,03	0,06	0,08	0,06	0,03
17.6	0,03	0,06	0,08	0,06	0,03
17.7	0,02	0,06	0,08	0,06	0,03
17.8	0,03	0,06	0,08	0,06	0,03
17.9	0,03	0,06	0,08	0,06	0,03
17.10	0,02	0,06	0,08	0,06	0,03
18.1	0,02	0,06	0,08	0,06	0,03
18.2	0,03	0,07	0,08	0,06	0,03
18.3	0,03	0,07	0,08	0,07	0,03
18.4	0,02	0,06	0,08	0,06	0,03
18.5	0,03	0,07	0,08	0,06	0,03
18.6	0,05	0,08	0,08	0,06	0,03
18.7	0,04	0,07	0,08	0,06	0,03
18.8	0,05	0,07	0,08	0,06	0,03
18.9	0,03	0,06	0,08	0,06	0,03
18.10	0,04	0,07	0,08	0,06	0,03
20.1	0,07	0,17	0,22	0,17	0,07
20.2	0,06	0,16	0,22	0,16	0,07
20.3	0,06	0,18	0,22	0,16	0,07
20.4	0,07	0,18	0,21	0,16	0,07
20.5	0,05	0,16	0,22	0,16	0,07
20.6	0,06	0,15	0,22	0,16	0,07
20.7	0,05	0,15	0,22	0,16	0,07
20.8	0,07	0,16	0,22	0,16	0,06
20.9	0,08	0,17	0,21	0,16	0,06
20.10	0,05	0,15	0,22	0,16	0,07
30.1	0,42	0,42	0,31	0,26	0,15
30.2	0,41	0,39	0,30	0,25	0,15

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
30.3	0,53	0,39	0,30	0,25	0,15
30.4	0,35	0,35	0,30	0,25	0,14
30.5	0,35	0,38	0,31	0,26	0,15
30.6	0,45	0,38	0,30	0,26	0,14
30.7	0,38	0,35	0,31	0,25	0,15
30.8	0,34	0,35	0,32	0,26	0,15
30.9	0,31	0,35	0,31	0,26	0,15
30.10	0,34	0,36	0,31	0,26	0,15
50.1	5,50	2,27	1,20	1,08	0,75
50.2	6,04	2,19	1,21	1,08	0,75
50.3	5,38	2,06	1,24	1,10	0,75
50.4	4,81	2,36	1,22	1,10	0,75
50.5	5,95	2,33	1,22	1,08	0,75
50.6	5,80	2,04	1,25	1,11	0,75
50.7	5,06	2,33	1,22	1,10	0,74
50.8	3,55	2,14	1,23	1,10	0,75
50.9	7,53	2,34	1,23	1,11	0,75
50.10	4,11	2,17	1,25	1,11	0,75

Tabela 41: Tempos computacionais médios (s) para as instâncias OP3

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
6.1	0,00	0,01	0,03	0,02	0,00
6.2	0,00	0,00	0,04	0,03	0,00
6.3	0,00	0,01	0,04	0,03	0,00
6.4	0,00	0,00	0,04	0,03	0,00
6.5	0,00	0,00	0,04	0,03	0,00
6.6	0,00	0,01	0,04	0,02	0,00
6.7	0,00	0,01	0,04	0,03	0,00
6.8	0,00	0,01	0,04	0,02	0,00
6.9	0,00	0,01	0,04	0,03	0,00
6.10	0,00	0,01	0,04	0,03	0,00
7.1	0,00	0,01	0,04	0,03	0,00



<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
7.2	0,00	0,01	0,04	0,03	0,01
7.3	0,00	0,01	0,04	0,03	0,01
7.4	0,00	0,01	0,04	0,03	0,01
7.5	0,00	0,01	0,04	0,03	0,01
7.6	0,00	0,01	0,04	0,03	0,01
7.7	0,00	0,01	0,04	0,03	0,00
7.8	0,00	0,01	0,04	0,03	0,01
7.9	0,00	0,01	0,04	0,03	0,00
7.10	0,00	0,01	0,04	0,03	0,00
8.1	0,00	0,01	0,04	0,03	0,01
8.2	0,00	0,01	0,04	0,03	0,01
8.3	0,00	0,01	0,04	0,03	0,01
8.4	0,00	0,01	0,04	0,03	0,01
8.5	0,00	0,01	0,04	0,03	0,01
8.6	0,00	0,01	0,04	0,03	0,01
8.7	0,00	0,01	0,04	0,03	0,01
8.8	0,00	0,01	0,04	0,03	0,01
8.9	0,00	0,01	0,04	0,03	0,01
8.10	0,00	0,01	0,04	0,03	0,01
9.1	0,00	0,01	0,04	0,03	0,01
9.2	0,00	0,01	0,04	0,03	0,01
9.3	0,00	0,01	0,04	0,04	0,01
9.4	0,00	0,01	0,05	0,03	0,01
9.5	0,00	0,01	0,05	0,03	0,01
9.6	0,00	0,01	0,04	0,03	0,01
9.7	0,00	0,01	0,04	0,03	0,01
9.8	0,00	0,01	0,04	0,03	0,01
9.9	0,00	0,01	0,04	0,03	0,01
9.10	0,00	0,01	0,04	0,04	0,01
10.1	0,00	0,02	0,05	0,03	0,01
10.2	0,00	0,01	0,05	0,03	0,01
10.3	0,00	0,02	0,05	0,04	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
10.4	0,00	0,01	0,05	0,04	0,01
10.5	0,00	0,01	0,05	0,04	0,01
10.6	0,00	0,01	0,05	0,04	0,01
10.7	0,00	0,01	0,05	0,03	0,01
10.8	0,00	0,01	0,05	0,04	0,01
10.9	0,00	0,01	0,05	0,04	0,01
10.10	0,00	0,01	0,05	0,03	0,01
11.1	0,00	0,02	0,05	0,04	0,01
11.2	0,00	0,02	0,05	0,04	0,01
11.3	0,00	0,02	0,05	0,04	0,01
11.4	0,00	0,02	0,05	0,04	0,01
11.5	0,00	0,02	0,05	0,04	0,01
11.6	0,00	0,02	0,05	0,04	0,01
11.7	0,00	0,01	0,05	0,04	0,01
11.8	0,00	0,02	0,05	0,04	0,01
11.9	0,00	0,02	0,05	0,04	0,01
11.10	0,00	0,02	0,05	0,04	0,01
12.1	0,00	0,02	0,05	0,04	0,01
12.2	0,00	0,02	0,05	0,04	0,01
12.3	0,00	0,02	0,05	0,04	0,01
12.4	0,00	0,02	0,05	0,05	0,01
12.5	0,00	0,02	0,05	0,04	0,01
12.6	0,00	0,02	0,05	0,04	0,01
12.7	0,00	0,02	0,05	0,04	0,01
12.8	0,00	0,02	0,05	0,04	0,01
12.9	0,00	0,02	0,05	0,04	0,01
12.10	0,00	0,02	0,05	0,04	0,01
13.1	0,00	0,02	0,05	0,04	0,01
13.2	0,00	0,02	0,05	0,04	0,01
13.3	0,00	0,02	0,05	0,04	0,01
13.4	0,00	0,02	0,05	0,05	0,01
13.5	0,00	0,02	0,05	0,05	0,01

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
13.6	0,00	0,02	0,05	0,04	0,01
13.7	0,00	0,02	0,05	0,04	0,01
13.8	0,00	0,02	0,05	0,05	0,01
13.9	0,00	0,02	0,05	0,04	0,01
13.10	0,00	0,02	0,05	0,04	0,01
14.1	0,00	0,03	0,05	0,04	0,02
14.2	0,01	0,03	0,05	0,05	0,02
14.3	0,00	0,02	0,05	0,05	0,02
14.4	0,00	0,02	0,05	0,05	0,02
14.5	0,00	0,03	0,05	0,05	0,02
14.6	0,00	0,03	0,05	0,05	0,02
14.7	0,01	0,03	0,05	0,05	0,02
14.8	0,00	0,03	0,05	0,05	0,02
14.9	0,00	0,02	0,05	0,05	0,02
14.10	0,01	0,02	0,05	0,05	0,02
15.1	0,01	0,05	0,08	0,07	0,02
15.2	0,01	0,05	0,08	0,07	0,02
15.3	0,01	0,05	0,08	0,07	0,02
15.4	0,01	0,05	0,08	0,07	0,02
15.5	0,01	0,05	0,08	0,07	0,02
15.6	0,00	0,05	0,08	0,07	0,02
15.7	0,01	0,05	0,08	0,07	0,02
15.8	0,00	0,05	0,08	0,07	0,02
15.9	0,01	0,05	0,08	0,07	0,02
15.10	0,01	0,05	0,08	0,07	0,02
16.1	0,01	0,06	0,09	0,07	0,02
16.2	0,01	0,06	0,09	0,07	0,02
16.3	0,01	0,06	0,08	0,07	0,02
16.4	0,01	0,06	0,09	0,07	0,02
16.5	0,01	0,06	0,08	0,07	0,02
16.6	0,01	0,05	0,08	0,08	0,02
16.7	0,01	0,06	0,08	0,07	0,02

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
16.8	0,01	0,06	0,08	0,08	0,02
16.9	0,01	0,06	0,08	0,08	0,02
16.10	0,01	0,06	0,08	0,07	0,02
17.1	0,01	0,07	0,09	0,08	0,02
17.2	0,01	0,06	0,09	0,07	0,02
17.3	0,01	0,07	0,09	0,08	0,02
17.4	0,01	0,07	0,09	0,07	0,02
17.5	0,01	0,08	0,09	0,07	0,02
17.6	0,02	0,06	0,09	0,08	0,02
17.7	0,01	0,06	0,09	0,08	0,03
17.8	0,01	0,06	0,09	0,07	0,03
17.9	0,02	0,07	0,09	0,08	0,02
17.10	0,01	0,06	0,09	0,07	0,03
18.1	0,02	0,08	0,09	0,08	0,03
18.2	0,01	0,08	0,09	0,08	0,03
18.3	0,02	0,08	0,09	0,08	0,03
18.4	0,01	0,08	0,09	0,08	0,03
18.5	0,02	0,07	0,09	0,08	0,03
18.6	0,01	0,07	0,09	0,08	0,03
18.7	0,02	0,06	0,09	0,07	0,03
18.8	0,01	0,08	0,09	0,08	0,03
18.9	0,01	0,08	0,09	0,08	0,03
18.10	0,01	0,06	0,09	0,08	0,03
20.1	0,00	0,00	0,03	0,02	0,00
20.2	0,00	0,00	0,03	0,02	0,00
20.3	0,00	0,00	0,03	0,02	0,00
20.4	0,00	0,00	0,03	0,02	0,00
20.5	0,00	0,00	0,03	0,02	0,00
20.6	0,00	0,00	0,04	0,02	0,00
20.7	0,00	0,00	0,03	0,02	0,00
20.8	0,00	0,00	0,03	0,02	0,00
20.9	0,00	0,00	0,04	0,02	0,00

<b>Inst.</b>	<b>PLS</b>	<b>Tabu</b>	<b>NSGAI</b>	<b>NSTA</b>	<b>AT</b>
20.10	0,00	0,00	0,04	0,02	0,00
30.1	0,21	0,42	0,35	0,30	0,13
30.2	0,64	0,49	0,29	0,26	0,14
30.3	0,17	0,43	0,35	0,31	0,13
30.4	0,09	0,52	0,34	0,29	0,13
30.5	0,11	0,42	0,34	0,31	0,13
30.6	0,12	0,41	0,34	0,31	0,13
30.7	0,17	0,49	0,34	0,29	0,13
30.8	0,14	0,45	0,34	0,30	0,13
30.9	0,19	0,44	0,34	0,28	0,13
30.10	0,16	0,46	0,34	0,29	0,13
50.1	2,22	2,57	1,37	1,25	0,70
50.2	1,68	3,57	1,35	1,13	0,70
50.3	1,65	2,46	1,35	1,25	0,70
50.4	1,25	3,05	1,36	1,21	0,69
50.5	1,54	3,27	1,36	1,22	0,70
50.6	2,46	3,27	1,37	1,23	0,70
50.7	1,39	2,56	1,36	1,20	0,70
50.8	2,09	2,81	1,35	1,16	0,70
50.9	1,61	2,39	1,36	1,24	0,70
50.10	1,39	3,13	1,36	1,19	0,70

É interessante notar que os tempos do AT são os menores entre as abordagens apresentadas e ainda obtém soluções de boa qualidade quando comparado com as outras abordagens. Os tempos dos algoritmos NSGA-II e NSTA são relativamente próximos. O tempo da Busca Tabu já é significativamente superior. A busca local PLS corresponde ao algoritmo implementado com maior custo em termos de tempo computacional médio. Com base nessas informações, novos experimentos computacionais podem ser delineados, fazendo um estudo comparativo entre as abordagens desenvolvidas ao longo do tempo. Nesse sentido, poder-se-ia observar aspectos de convergência dos algoritmos e compará-los em um mesmo tempo de execução para verificar se, em condições iguais de recursos de processamento (tempo) o desempenho relativo entre os algoritmos se mantém.

Considerando não apenas o critério de qualidade do conjunto de aproximação gerado, mas também o tempo gasto para executar os algoritmos, pode-se, pois, ordenar os algorit-

mos desenvolvidos, do melhor para o pior, como segue: Tabu, Transgenético, PLS, NSTA, NSGA-II.

## 7 Considerações finais

Esta tese apresentou um novo problema de Otimização Combinatória, denominado Problema Biobjetivo da Árvore Geradora Quadrática em Adjacência de Arestas. Tal problema é uma versão com múltiplos objetivos do problema da Árvore Geradora Mínima Quadrática em Adjacência de Arestas, proposto por Xu (1984). O problema da AGQA-bi se relaciona com outros problemas clássicos de otimização, tais como a Árvore Geradora Bi-objetivo e o Problema de Alocação Quadrática. Para conhecer melhor as características do problema, a literatura da AGMQ e da AGMQA foi revisada. Fundamentos da Otimização Multiobjetivo também foram descritos.

Nesta tese, apresentamos ainda sete algoritmos para o problema da AGQA-bi, dos quais dois são exatos e cinco, heurísticos. Os algoritmos exatos implementados foram um *backtracking* e um *branch-and-bound*. Os experimentos demonstraram a eficiência do *branch-and-bound*, que obteve um tempo computacional significativamente menor que o tempo do *backtracking*, como já era esperado.

Os algoritmos heurísticos implementados, por sua vez, foram uma busca local *Pareto Local Search*, um NSGA-II, um NSTA, um Transgenético e um Busca Tabu. Em termos de exclusivamente da qualidade de solução, o algoritmo heurístico que produziu melhores resultados foi a busca PLS, seguida pela busca Tabu, algoritmo transgenético, NSTA e NSGA-II. Com relação ao tempo de execução, a abordagem via algoritmos transgenéticos foi a mais rápida. O NSGA-II e o NSTA apresentaram tempos de execução relativamente próximos, o que os coloca em pé de igualdade neste quesito. A busca Tabu apresenta tempos computacionais superiores aos das metaheurísticas já mencionadas, mas, como já explicitado, dentre elas, é a que gerou os conjuntos de aproximação com maior qualidade de solução. Já as buscas locais PLS apresentaram tempo computacional bem superior ao das outras abordagens, tendo sua execução interrompida ao final de 150s (2,5min). Nesse sentido, se considerarmos tanto a qualidade de solução quanto o tempo de computação, constatamos que a melhor abordagem foi a busca Tabu, seguida pelo algoritmo Transgenético e pela busca local PLS. Por fim, seguem a abordagem NSTA e NSGA-II.

Nesse sentido, consideramos os resultados bastante promissores, os objetivos da pesquisa forma alcançados. Conseguimos desenvolver todos os algoritmos propostos, bem como mostrar a relevância de se desenvolver abordagens heurísticas para a solução do problema-alvo dessa investigação.

A hibridização NSTA obteve os resultados esperados, conseguindo aproveitar o poder dos algoritmos transgenéticos, o que a fez ser melhor classificada que o NSGA-II. Por outro lado, a abordagem NSTA também incorpora características do NSGA-II, cuja implementação proposta foi a que obteve resultados menos significativos à solução do problema da AGQA-bi. Por isso, no geral, o NSTA também não conseguiu superar o transgenético. Além disso, algumas anomalias mostram que a hibridização de heurísticas tem potencial e deve ser explorada: para um grupo específico de casos de teste, a abordagem híbrida superou tanto o NSGA-II quanto o Transgenético.

Confirmamos que a abordagem Tabu se mostra bastante apropriada para a solução do problema da AGQA-bi, assim como ocorre com o problema de AQ, que também apresenta natureza quadrática. São exemplos os trabalhos: Drezner (2005), James, Rego e Glover (2009a), James, Rego e Glover (2009b), Misevicius (2005), Paul (2010, 2011), Fescioglu-Unver e Kokar (2011) e Hussin e Stützle (2014). Além disso, em se tratando de abordagens evolucionárias, o algoritmo transgenético foi superior, o que também confirma a adequação dessa metaheurística, que já apresenta excelentes resultados no caso do problema da AG-bi, por exemplo em: Monteiro (2011), MonteiroGoldbarg10, MonteiroGoldbarg09.



## 8 Trabalhos Futuros

Esta tese, por si só, já apresenta diversas contribuições para as pesquisas em Otimização Combinatória. Um novo problema com múltiplos objetivos, sem trabalhos publicados na literatura, foi investigado e já foram desenvolvidos 7 algoritmos para este problema, dentre abordagens exatas e (meta)heurísticas. Uma revisão da literatura para a versão mono-objetivo do problema-alvo dessa pesquisa também foi realizada. Mesmo assim, o problema da AGQA-bi constitui um novo campo de pesquisas da Otimização Multiobjetivo, ainda pouco explorado, que ainda pode ser investigado. Dentre as principais possíveis contribuições para enriquecer a Tese e sugestões de trabalhos futuros sobre o tema, destacam-se:

- Verificar a possibilidade de adicionar restrições ao problema da AGQA-bi, tais como restrição de grau e de capacidade, da mesma maneira que ocorre com a AGM.
- Estudar o caso em que os intercustos possam assumir valores negativos ou qual o impacto dessa consideração no que se refere aos algoritmos apresentados nesta Tese.
- Investigar o Problema Biobjetivo do Caminho Mais Curto Em Adjacência de Arestas, no qual deseja-se encontrar o caminho de menor custo linear e menor intercusto. Este problema encontra aplicação no traçado de rotas em que penalidades de conversão precisam ser minimizadas.
- Desenvolvimento de algoritmo exato baseado na estratégia *k-best* existente para o problema da AG-bi.
- Comparação do algoritmo *k-best* com as estratégias exatas propostas nesta Tese.
- Análise, no todo ou em partes, da complexidade dos algoritmos implementados.
- Estudo de estruturas de dados especiais para lidar com árvores e com testes de conectividade em grafos, com o intuito de verificar se a adoção de tais estruturas poderia reduzir o tempo computacional dos algoritmos.
- Hibridização entre os algoritmos exatos e heurísticos propostos.

- Emprego de teste estatístico mais adequado para a comparação entre os diversos algoritmos, por exemplo, a variante Mack-Skillings do teste de Friedman.
- Comparação dos algoritmos heurísticos implementados com algoritmos exatos, de forma a conhecer (i) qual porcentagem da fronteira ótima de Pareto é encontrada pelos algoritmos e (ii) quantas soluções não-suportadas são geradas.
- Adensar a análise multiobjetivo dos resultados, exemplificando os resultados com fronteiras de Pareto e com os valores resultantes da aplicação dos indicadores de qualidade.
- Ilustrar o funcionamento dos algoritmos por meio de um caso de teste real.

# Referências

- ADRA, S. F.; FLEMING, P. J. Diversity management in evolutionary many-objective optimization. *IEEE Transactions On Evolutionary Computing*, v. 15, p. 183–195, 2011.
- AGGARWAL, V.; ANEJA, Y.; NAIR, K. Minimal spanning tree subject to a side constraint. *Computers & Operations Research*, v. 9, p. 287–296, 1982.
- ALMEIDA, C. P. de et al. An experimental analysis of evolutionary heuristics for the biobjective traveling purchaser problem. *Annals of Operations Research*, v. 199, p. 305–341, 2012.
- ARROYO, J. E. C. *Heurísticas e metaheurísticas para otimização combinatória multiobjetivo*. Tese (Doutorado) — Faculdade de Engenharia Elétrica e de Computação, Universidade Estadual de Campinas, 2002.
- ASSAD, A.; XU, W. On lower bounds for a class of quadratic 0, 1 programs. *Operations Research Letters*, v. 4, p. 175–180, 1985.
- ASSAD, A.; XU, W. The quadratic minimum spanning tree problem. *Naval Research Logistics*, v. 39, p. 399–417, 1992.
- BANDYOPADHYAY, S. Modified nsga-ii for a bi-objective job sequencing problem. *Intelligent Information Management*, v. 4, p. 319–329, 2012.
- BAZLAMAÇCI, C. F.; HINDI, K. S. Minimum-weight spanning tree algorithms: a survey and empirical study. *Computers and Operations Research*, v. 28, p. 767–785, 2001.
- BEHE, M. *Darwin's Black Box: The biochemical to evolution*. New York: The Free Press, 1996.
- BRASSARD, G.; BRATLEY, P. *Fundamentals of algorithms*. [S.l.]: Prentice Hall, 1996.
- BURKARD, R. E.; KARISCH, S. E.; RENDL, F. Qaplib - a quadratic assignment problem library. *Journal of Global Optimization*, v. 10, p. 391–403, 1997.
- CAMPELO, R. E.; MACULAN, N. *Algoritmos e heurísticas: desenvolvimento e avaliação de performance*. [S.l.]: EDUFF, 1994.
- CAPRA, F. *The turning point: science, society and the rising culture*. New York: Bantam Books, 1983.
- CHANG, H. et al. Energy dispatching analysis of lightning surges on underground cables in a cable connection station. *Energy Conversion and Management*, v. 52, p. 693–702, 2011.

- CLICKCIÊNCIA-UFSCAR. *Um passo além: a formulação da teoria que atualizou Darwin*. jan. 2009. Disponível em: <[http://www.clickciencia.ufscar.br/portal/edicao15/materia5\\_detalhe.php](http://www.clickciencia.ufscar.br/portal/edicao15/materia5_detalhe.php)>. Acesso em 10 de jan. 2011.
- COELHO, A. *Curso de genética dos microorganismos: notas de aula*. out. 2010. Disponível em: <<http://b200.nce.ufrj.br/genetica/graduacao.html>>. Acesso em out. 2010.
- CORDONE, R.; PASSERI, G. Heuristic and exact approaches to the quadratic minimum spanning tree problem. In: *Proceedings of Seventh Cologne - Twente Workshop on Graphs and Combinatorial Optimization*. Gargnano: Elsevier Science Ltd., 2008. p. 52–55.
- CORDONE, R.; PASSERI, G. *The quadratic minimum spanning tree problem (QMSTP)*. set. 2011. Disponível em: <<http://homes.dsi.unimi.it/~cordone/research/qmst.html>>. Acesso em 03 de out. 2011.
- CORDONE, R.; PASSERI, G. Solving the quadratic minimum spanning tree problem. *Applied Mathematics and Computation*, v. 218, p. 11597–11612, 2012.
- DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE Transactions on Evolutionary Computation*, v. 6, p. 182–197, 2002.
- DREZNER, Z. The extended concentric tabu for the quadratic assignment problem. *European Journal of Operational Research*, v. 160, n. 2, p. 416–422, 2005.
- EHRGOTT, M.; GANDIBLEUX, X. A survey and annotated bibliography of multiobjective combinatorial optimization. *OR Spektrum*, v. 22, p. 425–460, 2000.
- FALLAH-MEHDIPOUR, E. et al. Extraction of decision alternatives in construction management projects: application and adaptation of nsga-ii and mopso. *Expert Systems with Applications*, v. 39, p. 2794–2803, 2012.
- FESCIOGLU-UNVER, N.; KOKAR, M. M. Self controlling tabu search algorithm for the quadratic assignment problem. *Computers & Industrial Engineering*, v. 60, p. 310–319, 2011.
- GAO, J.; LU, M. Fuzzy quadratic minimum spanning tree problem. *Applied Mathematics and Computation*, v. 164, p. 773–788, 2005.
- GAO, J.; LU, M.; LIU, L. Chance-constrained programming for fuzzy quadratic minimum spanning tree problem. In: *IEEE International Conference on Fuzzy Systems*. Budapest: IEEE Press, 2004. p. 983–987.
- GENDREAU, M.; POTVIN, J. *Handbook of Metaheuristics*. 2nd. ed. [S.l.]: Springer Publishing Company, Incorporated, 2010.
- GILMORE, P. C. Optimal and suboptimal algorithms for the quadratic assignment problem. *SIAM Journal of Applied Mathematics*, v. 10, p. 305–313, 1962.
- GLOVER, F. Future paths for integer programming and links to artificial intelligence. *Comput. Oper. Res.*, v. 3, p. 533–549, 1986.

- GOLDBARG, E. F. G.; GOLDBARG, M. C. Transgenetic algorithm: a new endosymbiotic approach for evolutionary algorithms. In: ABRAHAM, A. et al. (Ed.). *Foundations of Computational Intelligence volume 3*. [S.l.]: Springer Berlin Heidelberg, 2009, (Studies in computational intelligence, v. 203). p. 425–460.
- GOLDBARG, E. F. G.; GOLDBARG, M. C.; SCHMIDT, C. C. A hybrid transgenetic algorithm for the prize collecting steiner tree problem. *Journal of Universal Computer Science*, v. 14, p. 2491–2511, 2008.
- GOLDBARG, M. C.; BAGI, L. B.; GOLDBARG, E. F. G. Transgenetic algorithm for the traveling purchaser problem. *European Journal of Operational Research*, v. 199, p. 36–45, 2009.
- GOLDBARG, M. C.; GOLDBARG, E. F. G. Computação natural: modelos e algoritmos. In: *ERI 2007 - XIV Escola Regional de Informática*. Guarapuava: SBC, 2007. p. 09–54.
- GOLDBARG, M. C.; LUNA, H. P. L. *Otimização combinatória e programação linear: modelos e algoritmos*. 2a. ed. Rio de Janeiro: Campus, 2005.
- GOLDBERG, D. E. *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading: Addison-Wesley, 1989.
- GOULD, S. J.; ELDREDGE, N. Punctuated equilibrium comes of age. *Nature*, v. 366, p. 223–227, 1993.
- GUTIN, G.; PUNNEN, A. P. *Traveling salesman problem and its variations*. Dordrecht: Kluwer Academic Publishers, 2002.
- HAKAMI, S. L. Steiner's problem in graphs and its implications. *Networks*, v. 1, p. 113–133, 1971.
- HANSEN, M. P. Tabu search for multiobjective optimization: Mots. In: *Proceedings of the Thirteenth International Conference on Multiple Criteria Decision Making*. Cape Town: Springer-Verlag, 1997. p. 6–10.
- HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. Ann Arbor: University of Michigan Press, 1975.
- HUANG, B.; BUCKLEY, B.; KECHADI, T.-M. Multi-objective feature selection by using nsga-ii for customer churn prediction in telecommunications. *Expert Systems with Applications*, v. 37, p. 3638–3646, 2010.
- HUMPHREYS, J. Lamarck and the general theory of evolution. *Journal of Biological Education*, v. 30, p. 295–303, 1996.
- HUSSIN, M. S.; STÜTZLE, T. Tabu search vs. simulated annealing as a function of the size of quadratic assignment problem instances. *Computers & Operations Research*, v. 43, p. 286–291, 2014.
- JAMES, T.; REGO, C.; GLOVER, F. A cooperative parallel tabu search algorithm for the quadratic assignment problem. *European Journal of Operational Research*, v. 195, n. 3, p. 810–826, 2009.

- JAMES, T.; REGO, C.; GLOVER, F. Multistart tabu search and diversification strategies for the quadratic assignment problem. *Trans. Sys. Man Cyber. Part A*, v. 39, n. 3, p. 579–596, 2009.
- JAN, G. E. et al. Transportation network navigation with turn penalties. In: *IEEE International Conference on Advanced Intelligent Mechatronics*. Singapore: IEEE Press, 2009. p. 1224–1229.
- JEYADEVI, S. et al. Solving multiobjective optimal reactive power dispatch using modified nsga-ii. *Electrical Power and Energy Systems*, v. 33, p. 219–228, 2011.
- JOTHI, R.; RAGHAVACHARI, B. Approximation algorithms for the capacitated minimum spanning tree problem and its variants in network design. *ACM Transactions on Algorithms*, v. 1, p. 265–282, 2005.
- KNOWLES, J. D. *Local-search and hybrid evolutionary algorithms for Pareto optimization*. Tese (Doutorado) — Department of Computer Science, University of Reading, Reading, UK, 2005.
- KNOWLES, J. D.; CORNE, D. W. A comparison of encodings and algorithms for multiobjective spanning tree problems. In: *Proceedings of the 2001 Congress on Evolutionary Computation, CEC01*. Seoul: IEEE Press, 2001. p. 544–551.
- KNOWLES, J. D.; THIELE, L.; ZITZLER, E. *A tutorial on the performance assessment of stochastic multiobjective optimizers*. [S.l.], 2006.
- KRUSKAL, J. B. On the shortest spanning subtree of a graph and the traveling salesman problem. *Proceedings of the American Mathematical Society*, American Mathematical Society, v. 7, p. 48–50, 1956.
- LACERDA, E. G. M.; CARVALHO, A. C. P. L. Introdução aos algoritmos genéticos. In: GALVÃO, C. O.; VALENÇA, M. J. S. (Ed.). *Sistemas inteligentes: aplicações a recursos hídricos e ciências ambientais*. [S.l.]: Universidade/UFRGS, 1999, (Coleção ABRH de Recursos Hídricos, v. 7). p. 99–150.
- LAWLER, E. L. The quadratic assignment problem. *Management science*, v. 19, p. 586–590, 1963.
- LIU, B. *Theory and practice of uncertain programming*. Heidelberg: Physica-Verlag, 2002.
- LIU, B. *Uncertainty theory: an introduction to its axiomatic foundations*. Heidelberg: Springer-Verlag, 2004.
- LOPES, S. *Bio*. 1ed. ed. São Paulo: Saraiva, 2004.
- LUO, J.; ZHANG, X. New method for constructing phylogenetic tree based on 3d graphical representation. In: *ICBBE 2007 The 1st International Conference on Bioinformatics and Biomedical Engineering*. Wuhan: IEEE Press, 2007. p. 318–321.
- MARGULIS, L. *Symbiosis as a source of evolutionary innovation: speciation and morphogenesis*. [S.l.]: The MIT Press, 1991.

- MARGULIS, L. *Symbiosis in cell evolution: microbial communities in the archean and proterozoic eons*. [S.l.]: W. H. Freeman, 2002.
- MISEVICIUS, A. A tabu search algorithm for the quadratic assignment problem. *Computational Optimization and Applications*, v. 30, p. 95–111, 2005.
- MONTEIRO, S. M. D. *Algoritmos transgenéticos aplicados ao problema da árvore geradora biobjetivo*. Natal: [s.n.], 2011.
- MONTEIRO, S. M. D.; GOLDBARG, E. F. G.; GOLDBARG, M. C. A plasmid based transgenetic algorithm for the biobjective minimum spanning tree problem. In: COTTA, C.; COWLING, P. (Ed.). *Evolutionary Computation in Combinatorial Optimization*. [S.l.]: Springer Berlin Heidelberg, 2009, (Lecture Notes in Computer Science, v. 5482). p. 49–60.
- MONTEIRO, S. M. D.; GOLDBARG, E. F. G.; GOLDBARG, M. C. A new transgenetic approach for the biobjective spanning tree problem. In: *Proceedings of IEEE CEC 2010 Congress on Evolutionary Computation*. Barcelona: IEEE Press, 2010. p. 519–526.
- MURUGAN, P.; KANNAN, S.; BASKAR, S. Nsga-ii algorithm for multi-objective generation expansion planning problem. *Electric Power Systems Research*, v. 79, p. 622–628, 2009.
- PALUBECKIS, G.; RUBLIAUSKAS, D.; TARGAMADZÉ, A. Metaheuristic approaches for the quadratic minimum spanning tree problem. *Information Technology and Control*, v. 39, p. 257–268, 2010.
- PAQUETE, L.; STÜTZLE, T. *Stochastic Local Search Algorithms for Multiobjective Combinatorial Optimization: a review*. [S.l.], 2006.
- PAUL, G. Comparative performance of tabu search and simulated annealing heuristics for the quadratic assignment problem. *Operations Research Letters*, v. 38, p. 577–581, 2010.
- PAUL, G. An efficient implementation of the robust tabu search heuristic for sparse quadratic assignment problems. *European Journal of Operational Research*, v. 209, p. 215–218, 2011.
- PICCIOTTO, S. *How to encode a tree*. Tese (Doutorado) — Department of Mathematics, University of California, San Diego, CA, US, 1999.
- PRIM, R. C. Shortest connection networks and some generalizations. *Bell System Technical Journal*, v. 36, p. 1389–1401, 1957.
- RAIDL, G. R. An efficient evolutionary algorithm for the degree-constrained minimum spanning tree problem. In: *Proceedings of the 2000 Congress on Evolutionary Computation, CEC*. La Jolla: IEEE Press, 2000. p. 104–111.
- RAIDL, G. R.; JULSTROM, B. A. Edge sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, v. 7, p. 225–239, 2003.
- REEVES, C. R. *Modern heuristic techniques for combinatorial problems*. [S.l.]: John Wiley & Sons, 1993.

- ROCHA, D. A. M.; GOLDBARG, E. F. G.; GOLDBARG, M. C. A new evolutionary algorithm for the bi-objective minimum spanning tree. In: *ISDA '07 Seventh International Conference on Intelligent Systems Design and Applications*. Rio de Janeiro: IEEE Press, 2007. p. 735–740.
- SCHMIDT, C. C. *Uma abordagem através de algoritmos transgenéticos para o problema da configuração do traçado de uma rede de distribuição de gás natural*. Natal: [s.n.], 2007.
- SEDGEWICK, R. *Algorithms*. EUA: Addison-Wesley, 1983.
- SERAFINI, P. Some considerations about computational complexity for multi objective combinatorial problems. In: JAHN, J.; KRABS, W. (Ed.). *Recent Advances and Historical Development of Vector Optimization*. [S.l.]: Springer Berlin Heidelberg, 1987, (Lecture Notes in Economics and Mathematical Systems, v. 294). p. 222–232.
- SOAK, S. M.; CORNE, D. W.; AHN, B. H. The edge-window-decoder representation for tree based problems. *IEEE Transactions on Evolutionary Computation*, v. 10, p. 124–144, 2006.
- SOUZA, G. R. de. *Uma abordagem por nuvem de partículas para problemas de otimização combinatória*. Natal: [s.n.], 2006.
- SUNDAR, S.; SINGH, A. A swarm intelligence approach to the quadratic minimum spanning tree problem. *Information Sciences*, v. 180, p. 3182–3191, 2010.
- SWAMY, M. N. S.; THULASIRAMAN, K. *Graphs, networks and algorithms*. [S.l.]: Wiley, 1981.
- WHILE, L. A new analysis of the lebmeasure algorithm for calculating hypervolume. In: COELLO, C. A. C.; AGUIRRE, A. H.; ZITZLER, E. (Ed.). *Evolutionary Computation in Combinatorial Optimization*. [S.l.]: Springer Berlin Heidelberg, 2005, (Lecture Notes in Computer Science, v. 3410). p. 326–340.
- WHILE, L. et al. Heuristics for optimising the calculation of hypervolume for multi-objective optimisation problems. In: *Proceeding of the IEEE Congress on Evolutionary Computation, CEC, 2005*. Edinburgh: IEEE Press, 2005. p. 192–199.
- XU, W. *Quadratic minimum spanning tree problems and related topics*. Tese (Doutorado) — University of Maryland, 1984.
- YU, J.; SHI, R.; SHEU, H. A linearization method for the quadratic minimum spanning tree problem. *International Journal of Fuzzy Systems*, v. 10, p. 287–291, 2008.
- ZHOU, D. et al. Ant-mst: an ant-based minimum spanning tree for gene expression data clustering. *Lecture Notes in Computer Science*, v. 4774, p. 198–205, 2007.
- ZHOU, G.; GEN, M. An effective genetic algorithm to the quadratic minimum spanning tree problem. *Computers & OR*, v. 25, p. 229–237, 1998.
- ZITZLER, E. *Evolutionary algorithms for multiobjective optimization: methods and applications*. Tese (Doutorado) — Swiss Federal Institute of Technology, Zurich, Switzerland, 1999.



ZITZLER, E. et al. Performance assessment of multiobjective optimizers: An analysis and review. *IEEE Transactions on Evolutionary Computation*, v. 7, p. 117–132, 2003.

ÖNCAN, T.; PUNNEN, A. P. The quadratic minimum spanning tree problem: a lower bounding procedure and an efficient search algorithm. *Computers and Operations Research*, v. 37, p. 1762–1773, 2010.

## APÊNDICE A – Conceitos Fundamentais em Grafos

O Apêndice A desta tese diz respeito ao estabelecimento de conceitos fundamentais sobre grafos, que possibilitam a correta definição dos problemas de otimização tratados. Tais conceitos estão todos ancorados em Swamy e Thulasiraman (1981).

Um grafo  $G = (V, E)$  é constituído por dois conjuntos: um conjunto finito  $V$  de elementos denominados vértices e um conjunto finito  $E$  de elementos chamados arestas.  $n = |V|$  e  $m = |E|$  correspondem, respectivamente, ao número de vértices em  $V$  e ao número de arestas em  $E$ . Representamos os elementos em  $V$  e em  $E$ , respectivamente, pelos símbolos  $v_1, \dots, v_n$  e  $e_1, \dots, e_m$ . Cada aresta é identificada por um par de vértices, ou seja,  $e_k = (v_i, v_j)$ ,  $0 < i, j \leq n$ , para todo  $0 < k \leq m$ . Os vértices  $v_i$  e  $v_j$  são ditos vértices terminais da aresta  $e_k$ .

Um grafo  $G = (V, E)$  é denominado direcionado ou orientado se as arestas de  $G$  constituem pares ordenados de vértices. Caso contrário, o grafo é dito não-direcionado ou não-orientado. Nesta tese, lida-se apenas com grafos não-orientados, a menos que comentário em contrário.

Se duas arestas possuem os mesmos vértices terminais, são ditas arestas paralelas. Por outro lado, se os dois vértices terminais de uma mesma aresta são iguais, tal aresta caracteriza um laço próprio (*self-loop*) no respectivo vértice. Um grafo  $G$  que não possui arestas paralelas nem laços próprios é denominado um grafo simples. A menos que menção explícita em contrário, este trabalho considera grafos simples.

Uma representação pictórica de um grafo pode ser realizada por meio de pontos e linhas. Os vértices são pontos e as arestas são linhas que conectam os pontos correspondentes aos respectivos vértices terminais.

Uma aresta é dita incidente em seus vértices terminais. Dois vértices são adjacentes se são vértices terminais de alguma aresta. Se duas arestas têm um vértice em comum,

as arestas também são denominadas adjacentes. O grau (ou valência) de um vértice é determinado pelo número de arestas incidentes a ele.

Um grafo  $G' = (V', E')$  é um subgrafo do grafo  $G = (V, E)$  se  $V'$  e  $E'$  são, respectivamente, subconjuntos de  $V$  e de  $E$  de tal forma que uma aresta  $(v_i, v_j)$  somente pode estar em  $E'$  se  $v_i$  e  $v_j$  estão em  $V'$ .  $G'$  é dito um subgrafo próprio de  $G$  se ou  $V'$  é subconjunto próprio de  $V$  ou  $E'$  é um subconjunto próprio de  $E$ . Se  $V'$  for igual a  $V$ , então  $G'$  é um subgrafo gerador de  $G$ .

Diz-se que um grafo  $G = (V, E)$  é completo se existe uma aresta conectando cada par de vértices diferentes em  $V$ . Neste caso, para um grafo simples, o número de arestas  $m$  é igual  $\frac{n(n-1)}{2}$ .

Um passeio em um grafo  $G = (V, E)$  é uma sequência alternada finita de vértices e arestas  $v_0, e_1, v_1, e_2, \dots, v_{k-1}, e_k, v_k$ , que começa e termina com vértices de forma que  $v_{i-1}$  e  $v_i$  são os vértices terminais da aresta  $e_i$ ,  $0 < i \leq k$ . Os vértices  $v_0$  e  $v_k$  são denominados vértices terminais do passeio. Os demais, são considerados vértices internos. Em um passeio, não há restrição proibindo que um mesmo vértice ou aresta apareça mais de uma vez na sequência. Quando  $v_0$  é igual a  $v_k$ , então o passeio é dito fechado. Caso contrário, é aberto.

Um passeio é denominado cadeia se todas as arestas são distintas. Uma cadeia fechada (cujos vértices terminais são iguais) é um ciclo ou circuito se todos os vértices internos são também distintos. Nesse sentido, um grafo é dito acíclico se não possui nenhum ciclo. Caminho, por sua vez, é uma cadeia aberta em que todos os vértices são distintos.

Um grafo  $G = (V, E)$  é conexo se existe um caminho entre cada par de vértices em  $V$ . Neste caso, os caminhos são da forma  $v_i, \dots, v_j$ , para todo  $v_i, v_j \in V$ , e diz-se que os vértices  $v_i$  e  $v_j$  estão conectados.

Uma árvore é um grafo acíclico e conexo. Conforme Swamy e Thulasiraman (1981), as seguintes afirmações são equivalentes para um grafo  $G$  com  $n$  vértices e  $m$  arestas:

- (i)  $G$  é uma árvore.
- (ii) Existe exatamente um caminho entre quaisquer dois vértices de  $G$ .
- (iii)  $G$  é conexo e  $m = n - 1$ .
- (iv)  $G$  é acíclico e  $m = n - 1$ .

(v) $G$  é acíclico e se dois vértices quaisquer não-adjacentes em  $G$  forem conectados por uma aresta, então o grafo resultante terá exatamente um ciclo.

Uma árvore de um grafo  $G$  é um subgrafo acíclico e conexo de  $G$ . Uma árvore geradora de um grafo  $G$  é, pois, uma árvore de  $G$  que abrange todos os vértices do grafo. Denomina-se subárvore um subgrafo conexo de uma árvore. Cayley mostrou que o número de árvores geradoras em um grafo completo é  $n^{n-2}$ .

Quando se associa a cada aresta  $e_i$  de um grafo  $G$  um valor  $w_i$ ,  $0 < i \leq m$ , diz-se que o grafo é ponderado. Para grafos ponderados, uma árvore geradora possui um custo, equivalente à soma dos valores associados a cada uma das arestas que pertencem à árvore. Sendo assim, a árvore geradora mínima de um grafo  $G$  é a árvore geradora de  $G$  que apresenta menor custo.

## APÊNDICE B – Otimização Multiobjetivo

Na otimização mono-objetivo, busca-se uma única solução que minimiza (ou maximiza) uma função-objetivo e que satisfaz restrições de igualdade e de desigualdade associadas ao problema. Entretanto, no mundo real, problemas com apenas um único e bem definido objetivo são exceção, não regra (KNOWLES, 2005).

Em problemas relacionados ao projeto de redes de distribuição (de água, energia, gás, sinal de TV, etc), por exemplo, poder-se-ia levar em conta não somente os custos de instalação, mas também questões relacionadas à segurança, aos impactos ambientais, à confiabilidade da rede, dentre outros critérios. O problema de determinar o trajeto mais econômico para entregar produtos a clientes pode levar em conta, além do custo com combustível, fatores como tempo de entrega, distância e tráfego (SOUZA, 2006). Em ambos os casos, a abordagem multiobjetivo (ou multicritério) é uma representação mais realista do problema. Ehrgott e Gandibleux (2000) apresentam uma revisão a respeito de problemas multicritério.

### B.1 Definição de um Problema de Otimização Multi-objetivo

Em um Problema de Otimização Multiobjetivo (POMO) deseja-se encontrar o vetor de variáveis de decisão que satisfaz certas restrições e otimiza um vetor-função cujos elementos representam as funções-objetivo (SOUZA, 2006).

Os objetivos de um POMO não são independentes, podem até mesmo ser conflitantes. Desse modo, a solução composta pelo ótimo de cada função-objetivo individualmente em geral não resulta em uma solução viável. Em outras palavras, uma solução admissível que otimiza um dos critérios não otimiza, em geral, os outros critérios, devido à condição de simultaneidade de otimização dos objetivos ou devido à existência de conflito entre os mesmos (KNOWLES, 2005).

O problema geral de minimização multiobjetivo (sem restrições) pode ser enunciado como expresso por B.1, em que  $X$  é o conjunto de todas as possíveis soluções.

$$\text{Minimizar } f(x) = (f(x_1), \dots, f(x_k)), \text{ em que } x \in X \quad (\text{B.1})$$

Nos problemas que consideram apenas um único objetivo, as soluções-candidatas podem ser dispostas em uma seqüência (ordem) de acordo com seus custos ou utilidades. A solução ótima é aquela que ocupa a primeira posição da seqüência e possui o menor valor, se o problema for de minimização, ou o maior valor, caso contrário.

No contexto dos POMOs, a ordenação de soluções é geralmente impossível de ser obtida, a menos que se apliquem técnicas que façam considerações ou que imponham regras que reflitam, por exemplo, a importância de cada objetivo. Para tanto, utilizam-se métodos de escalarização dos vetores-solução, possibilitando uma ordenação total, da qual se pode extrair a “melhor” solução.

Todavia, o emprego destas técnicas nem sempre está em consonância com os objetivos do tomador de decisões e pode representar uma visão muito simplória das relações entre os objetivos ou da relevância dos mesmos. Pode ocorrer, ainda, que o estabelecimento de prioridades entre os objetivos simplesmente não seja possível ou desejável, sendo a escolha da solução adotada exclusivamente delegada ao tomador de decisões (KNOWLES, 2005).

Por isso, em geral, POMOs são resolvidos considerando-se a Otimização de Pareto, que deixa para o tomador de decisões a escolha da melhor solução. Assim, não há apenas uma única solução para o problema e a palavra “minimizar” deve ser entendida no contexto da eficiência, ou Pareto otimalidade, considerando as relações de dominância de Pareto.

## B.2 Dominância de Pareto

A seguir, serão apresentados os conceitos envolvidos na dominância de Pareto, de acordo com Zitzler et al. (2003). Para tanto, considere  $x, y \in X$  e a função-vetor  $f(x)$ , da Equação B.1. Sem perda de generalidade, ao longo deste trabalho, os conceitos serão apresentados considerando problemas de minimização.

Diz-se que  $x$  domina fracamente  $y$ , o que se escreve  $x \succeq y$ , se, e somente se,  $\forall i, i = 1, \dots, k, f(x_i) \leq f(y_i)$ . Isto é,  $x$  domina fracamente  $y$  se  $x$  não for pior que  $y$  para nenhum objetivo.

Se além de  $x$  dominar fracamente  $y$ ,  $\exists i$ , tal que  $f(x_i) < f(y_i)$ ,  $i = 1, \dots, k$ , diz-se, então, que  $x$  domina  $y$ , o que se escreve  $x \succ y$ . Ou seja,  $x$  domina  $y$  se  $x$  não é pior que  $y$  em nenhum objetivo e é melhor que  $y$  para pelo menos um objetivo. Em outras palavras, ao se passar de  $y$  para  $x$ , existe a melhoria de pelo menos um dos critérios, permanecendo inalterados os restantes.

Por outro lado,  $x$  domina estritamente (ou fortemente)  $y$ , o que é representado por  $x \succ \succ y$ , caso  $\forall i, i = 1, \dots, k, f(x_i) < f(y_i)$ . Assim, a dominância estrita de  $x$  sobre  $y$  ocorre quando  $x$  é melhor que  $y$  em todos os objetivos. Observe que, se uma solução é estritamente (ou fortemente) dominada, ela também é dominada e é fracamente dominada. Entretanto, a afirmação inversa é falsa.

Diante da formulação matemática apresentada para a dominância de Pareto, podem existir  $x, y \in X$  tal que nem  $x$  domina fracamente  $y$  nem  $y$  domina fracamente  $x$ . Neste caso,  $x$  e  $y$  são ditas soluções incomparáveis, ou não-dominadas, e representam uma relação de compromisso entre os objetivos. Isso ocorre sempre que, ao passar de  $x$  para  $y$  ou de  $y$  para  $x$ , a melhoria de um critério, necessariamente, implicar a piora de outro.

O conjunto de soluções não-dominadas de todo o conjunto de soluções viáveis constitui o chamado conjunto de Pareto ótimo, ou conjunto de soluções ótimas. Assim, seja  $X^* \subseteq X$  o conjunto de soluções Pareto ótimas, uma solução  $x^* \in X^*$  se não há  $x \in X$  tal que  $x \succ x^*$ .

Neste contexto, a otimização multiobjetivo busca encontrar o conjunto de Pareto ótimo ou um conjunto de soluções incomparáveis que seja bem próximo ao conjunto ótimo de Pareto. Assim, o tomador de decisões poderá optar pela solução de melhor compromisso de acordo com o contexto específico em que o problema estiver inserido no momento.

### B.3 Soluções Eficientes e Soluções Não-dominadas

As soluções incomparáveis de todo o conjunto de soluções possíveis são também chamadas de soluções eficientes. O conceito de soluções eficientes é utilizado para pontos no espaço de decisão. Ehrgott e Gandibleux (2000) definem ainda o conceito de não-dominância, empregando-o para a respectiva imagem no espaço das funções-objetivo. Nesse sentido, a expressão “solução não-dominada”, em geral, se refere à imagem de uma solução eficiente, segundo os autores.

Matematicamente, tem-se (SOUZA, 2006):

- Uma solução admissível  $x$  ( $x \in X$ ) é eficiente se, e somente se, não existe outra solução admissível  $y$  ( $y \in X$ ) tal que  $y \succ x$  e  $f(y) \neq f(x)$ , em que  $f = (f_1, f_2, \dots, f_k)$ . Caso contrário,  $x$  é ineficiente. Ao conjunto das soluções eficientes, dá-se o nome de fronteira eficiente (no espaço de funções-objetivo) ou conjunto de soluções não-inferiores.
- Considere  $Z$  a região admissível no espaço das funções-objetivo (o conjunto de todas as imagens dos pontos em  $X$ ), definido da seguinte forma:  $Z = \{z \in R_k : z = (f_1(x), f_2(x), \dots, f_k(x)), x \in X\}$ .
- Seja  $z_1 \in Z$ . Então  $z_1$  é não-dominado se, e somente se, não existe um  $z_2 \in Z$ , tal que  $z_2 \leq z_1$  e  $z_2 \neq z_1$ . Caso contrário,  $z_1$  é um vetor de critérios dominado.

Entretanto, como visto na seção anterior, Zitzler et al. (2003) estabelece que duas soluções são não-dominadas entre si quando nenhuma delas domina fracamente a outra. A definição de Zitzler et al. (2003) é mais abrangente que a de Ehrgott e Gandibleux (2000), pois considera uma dominância relativa entre soluções (ou entre conjuntos de aproximação, como descrito adiante). Isso significa que duas soluções podem ser não-dominadas entre si e não fazerem parte da fronteira de Pareto. Este é o conceito para não-dominância adotado ao longo deste trabalho. Desse modo, ao mencionar o termo “soluções não-dominadas”, estaremos tratando de um conjunto de soluções para o qual não se pode provar (ou simplesmente não se considera) a relação de continência no conjunto de Pareto ótimo. Ao tratar de soluções especificamente pertencentes ao conjunto de Pareto ótimo, por sua vez, o termo utilizado será sempre soluções eficientes.

Resolver um problema multicritério é encontrar o conjunto de soluções eficientes, ou um conjunto de soluções não-dominadas que se aproxima o máximo possível da fronteira de Pareto. Assim, o tomador de decisão só precisa se preocupar com o conjunto de soluções obtido, descartando as outras soluções admissíveis. Com base nas soluções não-dominadas obtidas, pode-se, então, traçar a fronteira de Pareto, ou curva minimal, que exhibe os valores das soluções obtidas no espaço de objetivos (SOUZA, 2006).

A otimização combinatória, por natureza, lida com problemas discretos, não contínuos, embora os objetivos sejam funções lineares. Ehrgott e Gandibleux (2000) defendem que, uma consequência desse fato é que, ao se tentar determinar o conjunto de todas as soluções eficientes (vetores não-dominados no espaço de objetivos), não é suficiente agregar os objetivos por meio de somas ponderadas.

Assim, o conjunto de soluções eficientes é dividido em duas classes: soluções eficientes



suportadas e não-suportadas. As soluções eficientes suportadas podem ser obtidas se o problema for resolvido para somas ponderadas de seus objetivos, isto é, por meio de vetores de escalarização.

Mais formalmente (EHRGOTT; GANDIBLEUX, 2000):

$$\text{minimizar } \sum_{i=1,\dots,k} \alpha_i f(x_i) \quad (\text{B.2})$$

$$\text{em que } \sum_{i=1,\dots,k} \alpha_i, \alpha_i > 0, i = 1, \dots, k \quad (\text{B.3})$$

As soluções eficientes não-suportadas são aquelas que não são ótimas para nenhuma soma ponderada de seus objetivos. Este conjunto de soluções representa um maior desafio para os pesquisadores que lidam com problemas multiobjetivo, pois contribui significativamente para a dificuldade de resolução computacional do problema.

As noções de solução eficiente suportada e não-suportada podem ser melhor compreendidas a partir da Figura 7.

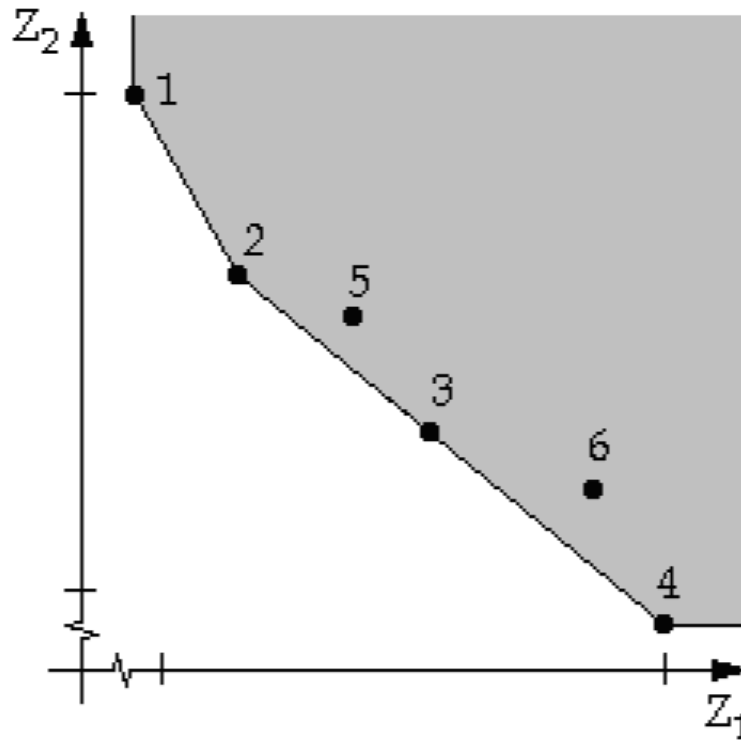


Figura 7: Soluções eficientes suportadas e não-suportadas (SOUZA, 2006).

Na Figura 7, mostram-se pontos no espaço de objetivos. As soluções correspondentes aos pontos 1, 2, 3 e 4 são eficientes suportadas, pois suas imagens podem ser obtidas a

partir de uma ponderação dos objetivos, isto é, situam-se na fronteira da envoltória convexa (contorno convexo) definida pelas soluções na fronteira de Pareto. Neste caso, 1, 2 e 4 são soluções eficientes suportadas extremas (pois suas imagens são vértices da envoltória convexa) e 3, por estar exatamente sobre a reta que une 2 e 4, equivale a uma solução suportada não-extrema (formada pela combinação convexa das soluções correspondentes aos pontos 2 e 4). As soluções correspondentes aos pontos 5 e 6, por sua vez, são soluções eficientes não-suportadas, haja vista estarem no interior do invólucro convexo, o que significa que não podem ser obtidas a partir de uma soma ponderada dos objetivos.

## B.4 Comparação entre Algoritmos Multiobjetivos

Para efetuar a comparação entre dois algoritmos, vários fatores devem ser levados em conta, tais como a qualidade das soluções obtidas, o tempo de processamento despendido, o ajuste de parâmetros, etc. Em otimização mono-objetivo, a qualidade de uma solução pode ser avaliada simplesmente pela diferença relativa entre os valores da solução heurística e da solução ótima.

Como já mencionamos neste trabalho, em geral, um algoritmo multiobjetivo resulta em um conjunto de soluções incomparáveis que, espera-se, seja o mais próximo possível do conjunto Pareto-ótimo. Uma vez que cada solução é um vetor, a saída do algoritmo é composta por um conjunto de vetores, os quais, como anteriormente discutido, não podem ser ordenados.

Nesse sentido, de acordo com Arroyo (2002), não há uma métrica simples e natural que seja capaz de aferir a qualidade de um conjunto aproximado em relação ao conjunto Pareto-ótimo. Assim, uma questão crucial na otimização multiobjetivo é de que forma realizar a comparação entre algoritmos.

O primeiro passo para efetuar a comparação entre conjuntos de aproximação (ou seja, os conjuntos de soluções não-dominadas obtidos pelos algoritmos como aproximação da Fronteira de Pareto) é estender os conceitos de dominância de Pareto para os referidos conjuntos. Para tanto, considere  $A$  e  $B$  dois conjuntos de aproximação, bem como adote  $z^1$  para representar soluções em  $A$  e  $z^2$  para representar soluções em  $B$ .

Diz-se que  $A$  domina  $B$  ( $A \succ B$ ), se toda solução  $z^2$  em  $B$  for dominada por pelo menos uma solução  $z^1$  em  $A$ . Analogamente,  $A$  domina estritamente  $B$  ( $A \succ\!\succ B$ ) se toda solução  $z^2$  em  $B$  for estritamente dominada por pelo menos uma solução  $z^1$  em  $A$  e  $A$  domina fracamente  $B$  ( $A \succeq B$ ) se toda solução  $z^2$  de  $B$  for fracamente dominada por pelo

menos uma solução  $z^1$  de  $A$ . Diz-se que  $A$  é incomparável com  $B$  ( $A||B$ ) no caso em que nem  $A$  domina fracamente  $B$  nem o contrário ocorre.

Ao se estender a dominância de Pareto para os conjuntos de aproximação, surgem duas novas situações. Pode ser que  $A \succeq B$  e  $B \succeq A$ , mesmo que  $A$  e  $B$  sejam conjuntos diferentes. Neste caso,  $A$  e  $B$  são considerados indiferentes ( $A \sim B$ ). Por outro lado, se todo  $z^2 \in B$  é fracamente dominado por pelo menos um  $z^1 \in A$  e  $A$  não é indiferente a  $B$  ( $A \not\sim B$ ), então  $A$  é considerado melhor que  $B$ , o que se escreve  $A \succ B$ .

Desse modo, a partir dos conceitos de dominância de Pareto, têm-se quatro situações de interesse (KNOWLES; THIELE; ZITZLER, 2006):

- (i)  $A$  é melhor que  $B$ ;
- (ii)  $B$  é melhor que  $A$ ;
- (iii)  $A$  e  $B$  são incomparáveis; e
- (iv)  $A$  e  $B$  são indiferentes.

Isso é tudo o que se pode obter acerca da qualidade dos conjuntos de aproximação se nenhuma informação privilegiada com relação às preferências do tomador de decisões estiver disponível. Entretanto, muitas vezes é necessário saber, também, o quão melhor um conjunto de aproximação é em relação ao outro (nos casos (i) e (ii)) e se um conjunto de aproximação é melhor que o outro considerando alguns aspectos de interesse (no caso (iii)). Com a finalidade de obter tais medições, fez-se necessário o desenvolvimento de métricas específicas.

As métricas desenvolvidas visam estabelecer uma ordem total entre os conjuntos de aproximação e podem ser divididas em duas classes, as Pareto compatíveis e as Pareto não-compatíveis (KNOWLES; THIELE; ZITZLER, 2006).

Uma métrica é Pareto compatível se a ordem que ela estabelece é coerente com o que a dominância de Pareto estabelece, independentemente dos conjuntos de aproximação em questão. Por exemplo, uma métrica Pareto compatível não pode inferir que um conjunto de aproximação  $A$  é pior que um conjunto de aproximação  $B$  se, pela dominância de Pareto,  $A$  dominar  $B$ .

Por outro lado, uma métrica é dita Pareto não-compatível se, para alguma comparação entre os conjuntos de aproximação  $A$  e  $B$ , a métrica inferir que  $A$  é preferível a  $B$  quando a dominância de Pareto mostrar que  $B$  é preferível a  $A$ .

Este trabalho utiliza indicadores de qualidade Pareto compatíveis, no caso, a métrica  $S$  e o  $\epsilon$ -binário. A métrica  $S$  tem se consolidado como a métrica mais indicada para comparação entre algoritmos multiobjetivo. Por outro lado, existem resultados na literatura que apontam para o fato de, caso os indicadores  $S$  e  $\epsilon$ -binário apresentem resultados contraditórios, então, para o caso de teste em questão, os algoritmos não podem ser comparados.

### B.4.1 Métrica $S$

A métrica  $S$ , proposta por Zitzler (1999), infere que um conjunto de aproximação  $A$  é preferível a (ou melhor que) um conjunto de aproximação  $B$  se  $A$  apresenta um maior hiper-volume (ou hiper-área, no caso de dois objetivos) que  $B$  em relação ao mesmo ponto de referência. Tal hiper-volume (ou hiper-área) representa o volume (ou a área) do espaço de objetivos que é pelo menos fracamente dominado pelo respectivo conjunto de aproximação. Na Figura 8, a hiper-área de  $B$  é maior que a de  $A$  e, por isso,  $B$  é considerado preferível a  $A$ .

O ponto de referência ( $Z_{ref}$  na Figura 8) adotado para limitar o espaço de objetivos deve ser pelo menos fracamente dominado por todos os pontos dos conjuntos de aproximação sob avaliação. Em geral, é composto pelo pior valor obtido para cada um dos objetivos, dentre todos os conjuntos de referência.

Seja  $S(X)$  o valor da métrica  $S$  para o conjunto de aproximação  $X$ . Para avaliar a qualidade de um conjunto de aproximação  $A$  com relação a um conjunto de aproximação  $B$  faz-se (SOUZA, 2006):

$$\frac{S(A) - S(B)}{S(B)} \quad (\text{B.4})$$

O uso da métrica  $S$  tem várias vantagens, dentre elas a de que não é necessário conhecer o conjunto Pareto-ótimo para usá-la. Entretanto, While (2005) e While et al. (2005) mostram que o tempo computacional gasto para calcular o hiper-volume pode ser muito alto, uma vez que é exponencial com relação ao número de objetivos e polinomial com relação ao número de pontos no conjunto de aproximação.

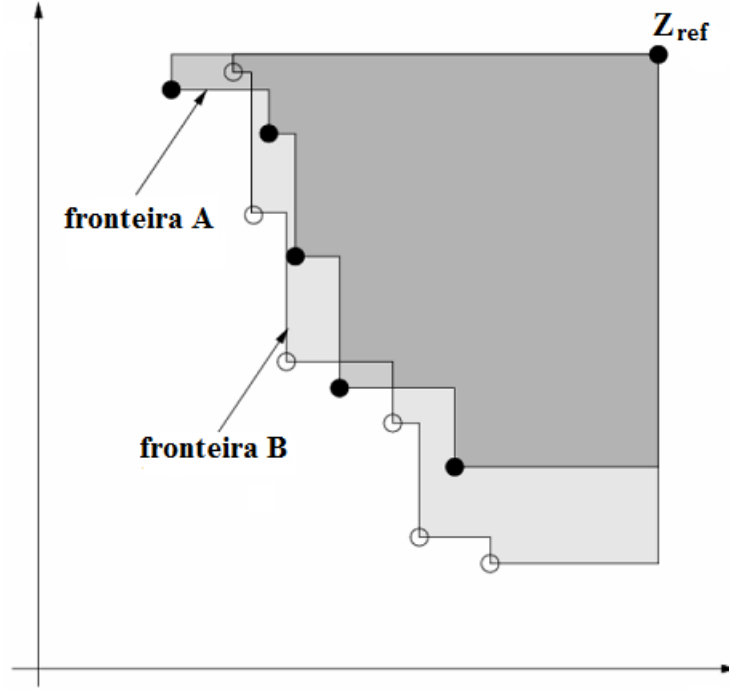


Figura 8: Métrica S para os conjuntos de aproximação A e B (SOUZA, 2006).

### B.4.2 $\epsilon$ -binário

Os indicadores épsilon foram introduzidos por Zitzler et al. (2003) e envolvem uma versão aditiva e multiplicativa. Sejam  $A$  e  $B$  dois conjuntos de aproximação, o indicador  $\epsilon$ -binário multiplicativo,  $I_\epsilon(A, B)$ , fornece o menor fator  $\epsilon$  pelo qual cada ponto de  $B$  pode ser multiplicado de forma que o conjunto de aproximação gerado seja fracamente dominado por  $A$ . O  $\epsilon$ -binário aditivo,  $I_{\epsilon+}(A, B)$  é definido de forma análoga, fornecendo o menor valor  $\epsilon$  com o qual cada ponto de  $B$  pode ser somado de forma que o conjunto de aproximação gerado seja fracamente dominado por  $A$ . Logo:

$$I_\epsilon(A, B) = \inf_{\epsilon \in \mathbb{R}} \left\{ \forall Z^2 \in B, \exists z^1 \in A | z^1 \succeq_\epsilon z^2 \right\} \quad (\text{B.5})$$

$$I_{\epsilon+}(A, B) = \inf_{\epsilon \in \mathbb{R}} \left\{ \forall Z^2 \in B, \exists z^1 \in A | z^1 \succeq_{\epsilon+} z^2 \right\} \quad (\text{B.6})$$

No caso do indicador  $\epsilon$ -binário multiplicativo, diz-se que uma solução  $z^2$  é  $\epsilon$ -dominada por uma solução  $z^1$  ( $z^1 \succeq_\epsilon z^2$ ) se, e somente se,  $\forall i | 1 \leq i \leq k, z_i^1 \leq \epsilon \cdot z_i^2$ , ou seja, se, para todos os objetivos, o valor de  $z^1$  não for pior que  $\epsilon$  vezes o correspondente valor em  $z^2$ . Analogamente, para o  $\epsilon$ -binário aditivo, diz-se que uma solução  $z^2$  é  $\epsilon$ -aditivo dominada por uma solução  $z^1$ ,  $z^1 \succeq_{\epsilon+} z^2$ , se, e somente se,  $\forall i | 1 \leq i \leq k, z_i^1 \leq \epsilon + z_i^2$ .

Na prática, o valor de  $\epsilon$  pode ser calculado para os casos multiplicativo e aditivo como descrito pelas Expressões B.7 e B.8, respectivamente (ZITZLER et al., 2003).

$$I_{\epsilon}(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq k} \left( \frac{z_i^1}{z_i^2} \right) \quad (\text{B.7})$$

$$I_{\epsilon+}(A, B) = \max_{z^2 \in B} \min_{z^1 \in A} \max_{1 \leq i \leq k} (z_i^1 - z_i^2) \quad (\text{B.8})$$

Nas Expressões B.7 e B.8, a maximização mais interna serve para calcular o maior  $\epsilon$  da comparação de uma solução de  $A$  com outra de  $B$  no que se refere a cada um dos objetivos. A minimização considera o valor de  $\epsilon$  obtido por cada solução em  $A$  com relação a uma mesma solução em  $B$ . Neste caso, seleciona-se o menor valor. Como cada solução em  $B$  tem que ser  $\epsilon$ -dominada, a maximização mais externa seleciona o maior  $\epsilon$  dentre os obtidos pela minimização para cada solução em  $B$ .

A Tabela 42, extraída de (SOUZA, 2006), mostra o significado da aplicação dos indicadores  $I_{\epsilon}$  e  $I_{\epsilon+}$  em dois conjuntos aproximados  $A$  e  $B$ , de acordo com a dominância de Pareto.

Tabela 42: Interpretação dos resultados fornecidos pelos indicadores binários.

Compatível com respeito à relação:						
	$\succ\succ$	$\succ$	$\triangleright$	$\succeq$	$=$	$\parallel$
$I_{\epsilon}$	$I_{\epsilon}(A, B) < 1$	-	$I_{\epsilon}(A, B) \leq 1$ $I_{\epsilon}(B, A) > 1$	$I_{\epsilon}(A, B) \leq 1$	$I_{\epsilon}(A, B) = 1$ $I_{\epsilon}(B, A) = 1$	$I_{\epsilon}(A, B) > 1$ $I_{\epsilon}(B, A) > 1$
$I_{\epsilon+}$	$I_{\epsilon}(A, B) < 0$	-	$I_{\epsilon}(A, B) \leq 0$ $I_{\epsilon}(B, A) > 0$	$I_{\epsilon}(A, B) \leq 0$	$I_{\epsilon}(A, B) = 0$ $I_{\epsilon}(B, A) = 0$	$I_{\epsilon}(A, B) > 0$ $I_{\epsilon}(B, A) > 0$

Pelas equações apresentadas e pelo disposto na Tabela 42 percebe-se que, se  $I_{\epsilon}(A, B) < 1$  (ou, analogamente,  $I_{\epsilon+}(A, B) < 0$ ),  $A$  tem que piorar para ser  $\epsilon$ -dominado por  $B$  e, por isso, consegue-se inferir que  $A \succ\succ B$ . Se nenhum  $\epsilon$  precisa ser adicionado ao conjunto  $A$  ou o  $\epsilon$  adicionado piora  $A$  para que  $A$  seja  $\epsilon$ -dominado por  $B$  e se o  $\epsilon$  melhora  $B$  para ser  $\epsilon$ -dominado por  $A$ , então  $A$  é considerado melhor que  $B$ .

## APÊNDICE C – Algoritmos Evolucionários

Este Apêndice revisa as teorias evolucionárias que inspiraram os algoritmos genéticos e transgenéticos, bem como detalha aspectos teóricos acerca dessas metaheurísticas. As teorias de Lamarck e de Darwin são discutidas nas Seções C.1 e C.2. A Teoria Serial da Endossimbiose, a transferência horizontal de genes e o equilíbrio pontuado ganham destaque nas Seções C.3, C.4 e C.5, respectivamente. Enfim, as Seções C.6 e C.7 apresentam, respectivamente, os Algoritmos Genéticos e a Transgenética Computacional, algoritmos evolucionários empregados nesta tese.

### C.1 Teoria de Lamarck

A teoria de Lamarck não influenciou diretamente os algoritmos genéticos e transgenéticos, entretanto, será mencionada nesta seção devido a sua relevância para os estudos sobre evolução. Tal teoria, apresentada nos livros “Filosofia Zoológica” e “História Natural dos Invertebrados”, é baseada em duas leis fundamentais: a lei do uso ou desuso e a lei da transmissão dos caracteres adquiridos. A lei do uso ou desuso defende que o uso/desuso de determinadas partes do organismo é responsável pelo desenvolvimento/atrofia destas partes. Por sua vez, a lei da transmissão dos caracteres adquiridos enfatiza que as alterações provocadas pelo uso/desuso são transmitidas aos descendentes (LOPES, 2004).

De acordo com Humphreys (1996), dois fatores cruciais motivaram Lamarck a definir sua teoria com base nessas duas leis. O primeiro deles era a crença de que a evolução era conduzida por uma tendência de herança que aumentasse os seres em tamanho e complexidade ao longo do tempo geológico. Ele também acreditava que, de outra forma, a evolução mais suave de formas mais avançadas de vida estaria sujeita aos caprichos da mudança ambiental. As mudanças ambientais criariam novas necessidades e demandariam novos comportamentos, o que teria um impacto no uso ou desuso de estruturas e levaria ao desenvolvimento ou atrofia das mesmas. Nesse sentido, Lamarck acreditava que os fatores essenciais para a adaptação dos indivíduos eram o ambiente e o modo de vida.

A teoria evolucionária Lamarckiana foi rapidamente refutada. Inicialmente, porque a idéia de evolução ainda não era bem aceita à época. Em seguida, experimentos refutaram a herança dos caracteres adquiridos. O biólogo alemão Weissman, no período de 1868 a 1876, por exemplo, cortou a cauda de ratos durante várias gerações, e os seus filhotes continuavam a nascer com cauda. Assim, Weissman provou que a característica “ausência de cauda” adquirida pelos ratos não foi transmitida a outras gerações. Ora, uma pessoa que pratica atividade física terá musculatura mais desenvolvida, mas essa condição não é transmitida aos seus descendentes (LOPES, 2004).

Embora principalmente lembrado por sua crença errônea na herança de caracteres adquiridos, Lamarck deve ser reconhecido como responsável pela primeira teoria científica da evolução geral, na qual se considera que todas as espécies vivas e fósseis surgiram por descendência com modificações a partir de organismos simples originais. Outra importante contribuição de Lamarck foi o desenvolvimento do conceito de adaptação ao meio. Para propor sua teoria evolucionária, Lamarck baseou-se na sua ampla experiência nas ciências naturais, e ao fazê-lo criou uma ciência nova, que ele chamou de “Biologia” (HUMPHREYS, 1996).

## C.2 Teoria da Seleção Natural

Conforme Lopes (2004), a teoria Darwiniana, proposta na obra “On the Origin of Species by Means of Natural Selection”, em 1859, é baseada nos seguintes princípios:

- A população tende a crescer de forma exponencial, tendo em vista que todos os indivíduos têm potencial reprodutivo.
- O número de indivíduos de uma espécie se mantém mais ou menos constante ao longo das gerações.
- O aumento da disponibilidade de recursos naturais não acompanha o crescimento populacional.
- Os indivíduos de uma mesma espécie não são idênticos. Portanto, apresentam variação nos seus caracteres.
- Parte dessa variação pode ser transmitida aos descendentes.
- Parte dessa variação ocorre em caracteres que afetam as chances de sobrevivência e reprodução dos organismos.



Em outras palavras, se a quantidade de recursos disponíveis não é suficiente para a quantidade de indivíduos da população, há uma competição natural por estes recursos. Há organismos que apresentam variações genéticas mais favoráveis às condições do ambiente onde vivem. Estes indivíduos, portanto, têm mais chance de sobreviver à seleção natural e também de deixar descendentes. A eterna competição entre os indivíduos em um cenário de luta pela sobrevivência seria, então, o motor da evolução, permitindo que apenas os mais aptos sobrevivam. Assim, a seleção natural atua mantendo ou melhorando o grau de adaptação dos indivíduos ao meio ao longo das gerações (LOPES, 2004).

As novas informações advindas de estudos em evolução e genética agregaram informações às idéias lançadas na obra de Darwin. Esse fato conduziu ao surgimento de uma nova formulação para os estudos evolucionários - a Teoria Sintética da Evolução, ou Neodarwinismo (CLICKCIÊNCIA-UFSCAR, 2009).

Apesar de ser considerada fundamentalmente uma atualização da teoria evolucionista, a teoria sintética trouxe algumas modificações conceituais para os estudos sobre evolução. As principais diferenças entre as teorias mencionadas podem ser resumidas como a seguir.

A teoria sintética admite a mudança genética aleatória como um mecanismo da evolução importante, juntamente com o mecanismo da seleção natural; assume que os caracteres são herdados através dos genes e que as variações da população devem-se à presença de múltiplas variações de um gene; e, finalmente, supõe que a especiação (surgimento de novas espécies) se deve, normalmente, a pequenas mudanças aleatórias na informação genética (CLICKCIÊNCIA-UFSCAR, 2009).

A teoria clássica, em sua forma original, reconhecia, mas ainda de forma embrionária, que fatores aleatórios estavam associados aos processos de mutação. Além disso, apenas após os estudos de Mendel sobre genética foi possível explicar os mecanismos de herança, por exemplo, da cor dos olhos e do cabelo. E quanto à especiação, Darwin considerava o fator de separação geográfica como preponderante nesse processo. Nesse sentido, diferentes ambientes possuiriam diferentes fatores mutagênicos e aleatórios que mudam o DNA de uma forma específica em uma e em outra população. Além disso, as mudanças seriam filtradas pelo mecanismo de seleção natural, que define os indivíduos mais ou menos adaptados ao meio (CLICKCIÊNCIA-UFSCAR, 2009).

Diante das diferenças expostas, percebe-se que o Neodarwinismo é fruto do enriquecimento da Teoria Darwiniana com base nas informações que ainda não eram conhecidas pelos pesquisadores e cientistas à época de Darwin.

À luz da Teoria Sintética da Evolução (Neodarwinismo), as principais forças evolucionárias são: mutação e recombinação (que tendem a aumentar a variabilidade genética), migração, deriva (ou oscilação) genética e seleção natural (que atuam sobre a variabilidade genética já estabelecida).

A mutação é considerada uma fonte primária de variabilidade genética, uma vez que consiste na alteração de um gene e, pois, pode dar origem a novas informações genéticas. Esse processo ocorre por acaso. A seleção natural se encarrega de manter ou não o resultado da mutação com base na adaptação. O processo de recombinação genética, por sua vez, é resultado da reprodução sexuada e promove a reorganização de informações genéticas já existentes nos cromossomos. A interação entre os genes é fator essencial para a evolução, uma vez que a atuação de um gene depende dos genes que interagem com ele e com o meio (LOPES, 2004).

O fenômeno da migração consiste na entrada e saída de indivíduos em uma população, o que pode introduzir novos genes em uma população. Outra consequência da migração é a diminuição das diferenças genéticas entre as populações. A deriva genética ou oscilação genética ocorre quando há uma mudança na frequência dos genes que se deve ao acaso e não à seleção natural. Catástrofes naturais como grandes incêndios ou enchentes podem modificar o meio e atingir determinada população de forma que esse processo de modificação genética tenha uma probabilidade de ocorrência maior (LOPES, 2004).

Por fim, a seleção natural é o principal fator evolucionário que atua sobre a variabilidade genética da população. Pode-se dizer que a evolução é consequência da atuação da seleção natural. Esse fenômeno consiste na seleção de indivíduos que possuem características genéticas mais favoráveis (adaptadas) a determinada condição ecológica, eliminando aqueles desvantajosos para as mesmas condições. Nesse contexto, o conceito de adaptação está intimamente relacionado à capacidade de sobrevivência no meio. Quanto mais intensa a seleção natural, menor a variabilidade genética da população, pois apenas um grupo restrito de diferentes perfis genéticos é selecionado. A seleção natural tem atuação constante sobre todas as populações, ainda que com caráter estabilizador das informações genéticas (LOPES, 2004).

A teoria Neodarwiniana sobre a evolução das espécies tem sido aceita no meio acadêmico. Entretanto, as discussões acerca deste assunto nunca foram mitigadas e há autores que contestam veementemente a referida teoria. Behe (1996) chega a dizer que “[...] há razões convincentes, com base na estrutura dos próprios sistemas [*biológicos*] para pensar que uma explicação darwiniana dos mecanismos da vida será sempre enganosa”.

Uma das grandes críticas feitas à Teoria de Darwin diz respeito ao gradualismo. De acordo com Lopes (2004), a teoria da evolução de Darwin pressupõe e defende que a especiação é um evento lento e gradual que ocorre pelo acúmulo de pequenas mudanças ao longo do tempo. Isto é, a evolução darwiniana exige que as variações individuais e funcionais sejam sempre de pequena amplitude, o que pode ser sintetizado pela famosa frase “a natureza não dá saltos”. Outra questão se refere ao fato de a evolução sempre ocorrer positivamente, na direção de uma situação melhor, pelo acúmulo de adaptações úteis. Essa suposição fica evidente por meio do próprio mecanismo de seleção natural, que determina que apenas os mais aptos e com modificações que promovam melhor adaptação irão sobreviver.

Todavia, o conhecimento atual sobre as estruturas moleculares permitiu a Behe (1996) contestar a teoria Darwiniana. Behe defende que o real trabalho da vida não ocorre no nível do animal ou órgão como um todo, e que as partes mais importantes da vida são muito pequenas para serem vistas. A vida é vivida nos detalhes, e é nas moléculas que a vida é controlada em detalhes.

Fundamentando-se em conhecimentos de bioquímica das estruturas moleculares, Behe (1996) mostra que a vida é baseada em máquinas, máquinas de moléculas, as quais realizam as mais diversas tarefas, tais como “ligar” e “desligar” células, matando-as ou fazendo-as crescer; e capturar e armazenar energia dos fótons. Em suma, Behe explica que cada processo celular é controlado por máquinas moleculares altamente sofisticadas. Tudo isso se constitui em um “maquinário da vida altamente complexo”. A questão apontada é: pode a teoria de Darwin explicar tudo sobre a evolução da vida, inclusive os aspectos moleculares conhecidos atualmente?

Segundo Behe, toda essa estrutura molecular de alta complexidade constitui os chamados Sistemas de Complexidade Irredutível (SCI). Os SCIs são sistemas compostos de mecanismos bem sincronizados e interativos, os quais contribuem para uma funcionalidade básica, de forma que a remoção de qualquer uma das partes de um SCI faz com que o sistema efetivamente deixe de funcionar. Os SCIs estão presentes na estrutura de várias células, sendo abundantes na natureza. Exemplos de SCI incluem a estrutura da visão humana bem como um simples flagelo em uma bactéria, que a permite se locomover.

Conforme Behe (1996), o problema da teoria de Darwin com os SCIs é que a formação de um SCI representa um alto investimento biológico, que demora gerações para se concretizar. Ao longo das gerações intermediárias em que o SCI ainda não está completo, o sistema não funciona, o que significa um acúmulo de adaptações inúteis (algo impossí-

vel de ser aceito pelo modelo evolucionário de Darwin). Com a formação do sistema e o correto funcionamento do SIC, um grande salto positivo ocorre na adaptação, fato que o gradualismo da Teoria de Darwin não é suficiente para explicar. Assim, a formação de um SIC, supostamente, ocorre de forma alheia ao processo de seleção natural. Behe afirma ainda que os estudos para tentar explicar esses fenômenos por meio da Teoria de Darwin parecem não dar resultados, haja vista não haver publicações científicas relatando avanços nesse sentido.

Tendo em vista o reducionismo da Teoria (Neo)Darwiniana a um contexto macroevolucionário, os estudiosos da evolução passaram a admitir que outros fatores guiam a evolução, e não apenas a mutação e a herança de genes em um processo de seleção natural.

### C.3 Teoria Serial da Endossimbiose

O modelo de evolução proposto por Darwin foi baseado na observação de seres vivos e animais durante suas viagens, em especial uma viagem de volta ao mundo que durou 5 anos (LOPES, 2004). À época, não se tinha o conhecimento de hoje acerca dos seres microscópicos, tais como bactérias, fungos, parasitas, etc; nem a bioquímica da vida era tão evoluída. Entretanto, os microorganismos são os principais constituintes da biomassa da Terra. Estes seres também são os que habitam o planeta Terra há mais tempo e, pois, conseguiram sobreviver às mais severas condições. Nesse sentido, é fundamental que as teorias sobre evolução consigam fornecer explicações para a evolução considerando um contexto microscópico.

Como já descrito na seção anterior, a teoria evolucionária de Darwin tem dificuldades para explicar a evolução da maioria destes seres, haja vista a pouca influência dos mecanismos de reprodução e mutação no contexto dos seres microscópicos e da vida molecular.

Fritjof Capra defende que:

Na história da vida na Terra, a co-evolução entre o macrocosmo e o microcosmo é de particular importância. Abordagens convencionais da origem da vida, em geral, descrevem o surgimento de formas mais avançadas de vida na microevolução, negligenciando aspectos macroevolutivos. Entretanto, esses são dois aspectos complementares de um mesmo processo evolucionário. De uma perspectiva microscópica, a vida cria as condições macroscópicas para a futura evolução. De uma perspectiva macroscópica, a biosfera se encarrega de criar sua própria vida microscópica. O desdobramento da complexidade não decorre da adaptação dos organismos a um dado meio, mas sim da co-evolução de organismo e

ambiente em todos os níveis do sistema (CAPRA, 1983).

Nessa perspectiva, emerge a Teoria Serial da Endossimbiose, defendida por Margulis (1991, 2002), que explica fenômenos em um contexto de aplicação mais amplo que a perspectiva darwiniana de evolução.

A simbiose é uma relação mutuamente vantajosa entre dois ou mais organismos vivos de espécies diferentes. Na relação simbiótica, os organismos agem ativamente em conjunto para proveito mútuo, o que pode acarretar em especializações funcionais de cada espécie envolvida. Os organismos atuam juntos em prol da sobrevivência de ambos. Nesse sentido, o paradigma da cooperação é preponderante sobre o panorama da competição e da sobrevivência do mais forte.

A endossimbiose, por sua vez, diz respeito às relações biológicas nas quais um organismo vive no interior de sua célula hospedeira em uma relação de simbiose. A natureza é abundante em exemplos de endossimbiose, como, por exemplo, as bactérias que habitam o sistema digestivo dos seres humanos.

Margulis (1991) mostra que uma relação endossimbiótica que traz grandes benefícios para ambos os organismos pode ser de tal relevância para a sobrevivência de ambos que pode culminar na completa integração entre ambos. Neste momento, não é mais possível distinguir entre os dois seres. Surge um novo organismo da fusão dos anteriormente mencionados e as características “herdadas” de ambos os organismos são transmitidas aos descendentes. Por este aspecto, Margulis (2002) ressalta que a evolução endossimbiótica possui um componente “lamarkista” ao viabilizar a herança de conjuntos genômicos adquiridos.

A autora afirma ainda que “a evolução do genoma no longo prazo é muito mais influenciada por interações extra-intracelulares do que pela mutação ou seleção natural”. Tais interações ocorrem por meio da endossimbiose e da transferência horizontal de genes. Nas interações extracelulares por endossimbiose, o endossimbionte está dentro do organismo do hospedeiro, porém reside externamente às células do mesmo. No caso de interações intracelulares de natureza endossimbiótica, o endossimbionte reside no interior de uma célula do hospedeiro.

A Teoria Serial da Endossimbiose consegue fornecer explicação para vários eventos importantes para a evolução da vida no planeta Terra e para os quais a evolução (Neo)Darwiniana não apresenta justificativa plausível. A comunidade científica já aceita a tese de que o surgimento dos organismos eucariontes (célula com núcleo, DNA centra-

lizado no interior do núcleo) se deu a partir da absorção de um procarionte (célula sem núcleo, DNA disperso no citoplasma) por outro procarionte. De relações endossimbióticas, também teriam surgido várias estruturas intra-celulares, tais como as mitocôndrias (presente nas células de animais e vegetais, responsáveis pela quebra de proteína e liberação de energia) e os cloroplastos (presentes nas células de vegetais, responsáveis pela síntese de proteína).

Na endossimbiose, a competição não representa a única via promotora do aperfeiçoamento, tampouco a sobrevivência e reprodução dos mais aptos. A abordagem enfatiza a cooperação que pode ser traduzida por ações coordenadas e troca de informações genéticas (SCHMIDT, 2007).

A endossimbiose permeia a evolução de seres microscópicos e macroscópicos, uma vez que indivíduos desses dois contextos são personagens de relações endossimbióticas intensas. O ser humano pode ser considerado uma colônia ambulante de bactérias. Nossa digestão, por exemplo, é resultado de interações com bactérias que vivem no interior do nosso organismo. Outras bactérias nos auxiliam na defesa contra microorganismos prejudiciais ao nosso corpo. Como mostrado pelos exemplos da evolução celular no tocante ao surgimento de estruturas internas das células (tais como as mitocôndrias e os cloroplastos), a teoria serial da endossimbiose também é capaz de explicar fenômenos no contexto intracelular/microscópico. Sendo assim, essa teoria é uma abordagem sobre a evolução mais abrangente que a proposta de Darwin.

Este paradigma da evolução, proposto por Margulis, foi inicialmente rejeitado por falta de comprovação científica. Contudo, à medida que se obtém melhores informações sobre a evolução microbial, esta abordagem se torna mais aceita.

Segundo Goldbarg et al.:

A comprovação da proposta de Margulis tem sido corroborada por sucessivas descobertas de mecanismos evolucionários naturais voltados para o compartilhamento direto de DNA entre microorganismos / células (Hadfield & Axton, 1999). Tais mecanismos são muito primitivos e foram constituídos no sentido de permitirem a ocorrência de alterações permanentes no código genético de células e microorganismos, facilitando a emergência de saltos de adequação (GOLDBARG; GOLDBARG; SCHMIDT, 2008).

Nesse sentido, a Teoria Serial da Endossimbiose não se prende ao gradualismo de Darwin e contribui para a evolução das espécies em todos os níveis do sistema, tanto no microscópico quanto no macroscópico.

## C.4 Transferência Horizontal de Genes

A transferência horizontal de genes é um conjunto de processos que promove alteração permanente do código genético a partir de interações extra-intracelulares. Existem vários “veículos de transferências de genes”, a saber: plasmídios, plasmídios recombinados, transposons, dentre outros.

Os plasmídios são partículas genéticas móveis (anéis de DNA) que podem se replicar independentemente do cromossomo. Eles são compostos por DNA e podem ser entendidos como “mini-cromossomos”.

Os plasmídios podem ocorrer naturalmente, como, por exemplo, em bactérias e, mais raramente, em organismos eucariontes. Entretanto, os plasmídios também podem ser construídos com DNA criado artificialmente, por meio do emprego de técnicas de engenharia genética. Goldbarg e Goldbarg (2009) definem que se o DNA artificial que dá origem ao plasmídio advém de mais de uma fonte, este plasmídio é chamado recombinado.

Em 1950, Barbara McClintock observou a mudança de posição de genes do milho. Entretanto, sua pesquisa só teve o devido reconhecimento muitos anos depois, o que resultou nos estudos sobre transposons.

Elementos de transposição, ou transposons, são regiões de DNA que podem se transferir de uma região para outra do genoma, deixando ou não uma cópia no local antigo onde estavam. A transposição pode ser para o mesmo cromossomo, para outro cromossomo ou para um plasmídio (COELHO, 2010). Para que a transposição ocorra, a enzima transposase deve estar presente. Geralmente, esta enzima é codificada pelo próprio elemento de transposição.

De acordo com o mecanismo de transposição, os transposons podem ser divididos em duas classes: retrotransposons e transposons DNA. Os retrotransposons atuam se copiando e colando de volta no genoma em múltiplos lugares. O mecanismo utilizado pelos transposons DNA se assemelha mais a um recortar-e-colar que a um copiar-e-colar (GOLDBARG; GOLDBARG, 2009).

A transferência horizontal de genes entre endossimbiontes e hospedeiro tem um papel relevante na endossimbiose, uma vez que assume características próprias (GOLDBARG; GOLDBARG, 2009). No complexo processo de transferência endossimbiótica de genes, há sistemas os quais permitem a identificação e aquisição de genes e outros que promovem a incorporação do DNA às células receptoras. É necessária, também, a existência de

processos que permitam o transporte da informação genética pelo meio celular sem que tal informação seja destruída ou corrompida.

De acordo com Goldbarg e Goldbarg:

O cenário descrito acima provê uma alternativa instigante para o conceito de evolução baseado na lógica do “DNA egoísta” e abre a possibilidade de utilizar agentes ou operações inspirados nos vetores celulares intra, inter e extra-celulares com o objetivo de guiar e acelerar a busca realizada por um algoritmo evolucionário. Esta é a característica na qual os algoritmos transgenéticos se inspiram. É importante notar que os mecanismos acima descritos também são formas de selecionar, compartilhar e misturar informações genéticas que sejam independentes de reprodução (GOLDBARG; GOLDBARG, 2008:5).

Com base nesta concepção será abordada a transgenética computacional na Seção C.7

## C.5 Equilíbrio Pontuado

Outra teoria que tem influência sobre os algoritmos transgenéticos é o Equilíbrio Pontuado, proposto por Gould e Eldredge em 1972.

O equilíbrio pontuado contrapõe o gradualismo de Darwin. Segundo o gradualismo, a especiação é proveniente de um processo lento e gradual (passo a passo) que ocorre pelo acúmulo de pequenas mudanças ao longo do tempo (LOPES, 2004). Essa evolução lenta e gradual, entretanto, não foi observada nos registros fósseis coletados até o presente momento. À época, Darwin justificou que essa lacuna nos registros fósseis se devia ao fato de não haver ainda muitos fósseis coletados e estudados. Porém, a quantidade de registros fósseis atualmente é vasta e o que se observa são súbitas aparições de espécies fósseis seguidas de longo período sem que esses seres sofressem mudanças (GOULD; ELDREDGE, 1993).

A proposta de Gould e Eldredge é, pois, que tanto a aparição repentina de várias espécies quanto os períodos sem mudança existem de fato e não são resultado de falha de registro fóssil. Assim, o processo de evolução se daria em um cenário de equilíbrio constante em que ocorrem mudanças rápidas e pontuais, por isso o nome Equilíbrio Pontuado (LOPES, 2004).



## C.6 Algoritmos Genéticos

Os algoritmos genéticos são uma metaheurística evolucionária inspirada na Teoria de Darwin para a evolução de seres de reprodução sexuada. Sendo assim, essa abordagem mimetiza os princípios da seleção natural (que implica a sobrevivência do mais apto) e do gradualismo. Holland (1975) e Goldberg (1989) definiram e popularizaram essa metaheurística.

Nos algoritmos genéticos, uma possível solução é codificada em um indivíduo, o qual é representado unicamente por seu cromossomo. Cada posição do cromossomo é um gene. No caso do problema da AGQA-bi, por exemplo, um cromossomo poderia ser uma representação, por conjunto de arestas (RAIDL; JULSTROM, 2003), de uma árvore geradora do grafo de entrada. Nesse sentido, cada gene seria uma aresta.

O primeiro passo de um algoritmo genético (e de um algoritmo evolucionário, em geral) é a criação de uma população, no caso, de cromossomos. A ideia central da metaheurística é aplicar, no contexto computacional, os princípios da evolução darwiniana à população de cromossomos inicializada. Nesse sentido, ao longo das iterações, operadores genéticos são aplicados aos indivíduos da população gerando novos indivíduos. A nova população precisa, então, ser avaliada de acordo com a sua adequação ao meio (representada pelo valor de *fitness*). Os cromossomos mais fortes (com melhor adequação) terão maior probabilidade de permanecer na população (ficarem vivos) e passarem para a próxima iteração do algoritmo (GENDREAU; POTVIN, 2010).

Perceba que uma iteração do algoritmo corresponde, na verdade, a uma geração, de acordo com a mimetização biológica. Além disso, faz-se necessário um mecanismo para avaliar a adequação dos indivíduos. Em geral, tal mecanismo consiste na própria função-objetivo do problema, com adaptações apenas para considerar se o problema é de minimização ou de maximização, uma vez que a adequação, ou *fitness*, deve ser sempre maximizada. Também se faz necessário estabelecer um procedimento para mimetizar a seleção natural, ou seja, um procedimento que determine, a partir do *fitness* dos indivíduos, quais deles conseguirão passar para a próxima geração (iteração). Os operadores genéticos mencionados são analogias à mutação e *crossover* (cruzamento).

A operação de cruzamento consiste em combinar a informação dos cromossomos pais para gerar cromossomos filhos, ou descendentes. Isso implica a seleção de indivíduos da população para serem os pais de uma operação de cruzamento. Em geral, essa operação envolve dois cromossomos pais, embora tal quantidade possa variar. O número de

descendentes pode variar de acordo com a implementação da operação.

Um exemplo de operador de cruzamento para o caso da AGQA-bi gera uma máscara aleatória de valores *booleanos* cujo tamanho corresponde ao número de arestas de uma árvore geradora do grafo de entrada. Se o valor *booleano* em uma determinada posição  $i$  for 0, a aresta da posição  $i$  do primeiro descendente virá do primeiro pai e, a do segundo descendente, do segundo pai. Caso contrário, a aresta  $i$  será obtida do segundo pai para o primeiro descendente e do primeiro pai para o segundo descendente. A aresta somente permanecerá em um dos filhos se não gerar ciclo, ou seja, se não quebrar a restrição de árvore. Se, ao final da operação, a árvore geradora de cada um dos filhos ainda não estiver completa, arestas aleatórias podem ser selecionadas para completar o descendente. Conforme já mencionado, o operador assim definido pode gerar até dois descendentes.

A operação de mutação envolve modificar a informação armazenada em um dos cromossomos. No caso do problema da AGQA-bi, um operador de mutação simples é substituir uma aresta que está no cromossomo (árvore) por uma aresta aleatória que não está no cromossomo e que não gera ciclo.

Nos algoritmos genéticos, o processo de seleção natural é aplicado de forma mais clara em dois momentos: a seleção de pais para reprodução (apenas os mais aptos conseguem se reproduzir e transmitir seus genes aos descendentes) e a seleção para renovação da população, ou seja, a seleção dos indivíduos que passarão para a próxima geração.

No primeiro caso, os métodos mais comuns são classificação, roleta e torneio. No segundo caso, considerando que o tamanho da população não varie (o que é mais comum, mas nem sempre é verdade), dois processos distintos de seleção se destacam, a saber: (i) apenas os descendentes passam para a próxima iteração do algoritmo; (ii) os melhores indivíduos dentro do conjunto união entre a população atual e seus descendentes irão constituir a população da próxima geração. Em Gendreau e Potvin (2010) e Lacerda e Carvalho (1999), vários operadores de cruzamento e mutação são explicados em detalhes, como também os procedimentos para seleção.

O pseudocódigo do algoritmo AG ilustra o funcionamento de um algoritmo genético clássico, de acordo com os comentários tecidos a esse respeito, conforme Lacerda e Carvalho (1999). Considere que  $S(t)$  é a população de cromossomos na iteração  $t$ .

---

**Algoritmo 8** Algoritmo AG
 

---

```

 $t \leftarrow 0$ 
Inicializar  $S(t)$ 
Avaliar  $S(t)$ 
enquanto o critério de parada não for satisfeito faça
   $t \leftarrow t + 1$ 
  Selecionar  $S(t)$  a partir de  $S(t - 1)$ 
  Aplicar crossover sobre  $S(t)$ 
  Aplicar mutação sobre  $S(t)$ 
  Avaliar  $S(t)$ 
fim enquanto

```

---

## C.7 Algoritmos Transgenéticos

Algoritmos transgenéticos são técnicas de computação evolucionária baseadas em processos de vida em que a cooperação é a principal estratégia de evolução (GOLDBARG; BAGI; GOLDBARG, 2009). Os Algoritmos Transgenéticos realizam a pesquisa no espaço de soluções de problemas de otimização por meio de um contexto computacional inspirado na troca de informações entre uma célula hospedeira e uma população de endossimbiontes. Ou seja, a busca realizada nos algoritmos transgenéticos mimetiza as interações intracelulares ocorridas via endossimbiose e transferência horizontal de genes. Segundo Goldberg e Goldberg (2009), três contextos de informação são considerados nos Algoritmos Transgenéticos:

- Soluções, representadas pelos cromossomos, também chamados cromossomos endossimbiontes;
- Informações sobre o problema em questão e informações sobre a busca evolucionária que o algoritmo realiza, chamadas informações do hospedeiro;
- Entidades que são utilizadas para modificar as soluções-candidatas, chamadas vetores transgenéticos.

Os algoritmos transgenéticos mimetizam a evolução de endossimbiontes em uma célula hospedeira em consonância com o paradigma da evolução proposto por Lynn Margulis em sua Teoria Serial da Endossimbiose. Nesse contexto, os endossimbiontes são interpretados como os indivíduos da população de soluções-candidatas, as quais constituem a população de cromossomos endossimbiontes.

As informações sobre o problema e sobre a busca evolucionária constituem a informação genética do hospedeiro. A informação é dita *a priori* se existir independentemente da

busca que o algoritmo realiza. Este tipo de informação é, em geral, inerente ao problema ou à instância que está sendo tratada, tais como: limites superiores e inferiores, soluções heurísticas e resultados de análise estatística da estrutura do problema. Informações *a posteriori* são obtidas durante a execução do algoritmo e estão relacionadas à busca. Por exemplo, novas e melhores soluções ou soluções parciais podem emergir da população de cromossomos. Informações *a priori* e *a posteriori* são armazenadas em um repositório chamado repositório do hospedeiro.

A evolução artificial promovida pelo algoritmo transgenético é guiada pela otimização da população de soluções-candidatas. De acordo com o exposto na Seção C.3, um novo indivíduo pode emergir da fusão de dois ou mais indivíduos em função de uma relação endossimbiótica de grande relevância para a sobrevivência destes seres. A partir deste momento, o código genético a ser transmitido aos descendentes advém dos organismos que se fundiram. Na abordagem proposta pela transgenética computacional, esta interação entre a informação genética dos endossimbiontes e a informação genética do hospedeiro é o único mecanismo utilizado para promover a evolução e, conseqüentemente, a busca evolucionária. Assim, as informações do repositório do hospedeiro são utilizadas pelos vetores transgenéticos para modificar as soluções na população de endossimbiontes.

Os vetores transgenéticos são inspirados em mecanismos de transferência horizontal de genes e são nomeados como tais. Sua função é modificar as soluções representadas nos cromossomos endossimbiontes, gerando a variação necessária para efetuar a busca. Os vetores transgenéticos são as únicas ferramentas que realizam as tarefas de intensificação e diversificação nos algoritmos transgenéticos (GOLDBARG; GOLDBARG, 2009). Os vetores transgenéticos, em geral, implementados são os plasmídios, os plasmídios recombinados e os transposons.

Um vetor transgenético  $\lambda = (I, \Phi)$  é composto por uma cadeia de informação  $I$  e por um método de manipulação de cromossomos  $\Phi = (p_1, \dots, p_s), p_j, j = 1, \dots, s$ . Tais procedimentos definem a ação do vetor, ou seja, a forma com que cada vetor vai atuar sobre os endossimbiontes com base em sua cadeia de informação. Esses procedimentos são pré-definidos de forma genérica. Assim, a depender da natureza do problema, diversas implementações de cada procedimento podem ser propostas e, obviamente, terem resultados distintos. A Tabela C.1 lista os procedimentos que compõem os métodos de manipulação de um vetor transgenético, de acordo com Goldbarg e Goldbarg (2009).

A cadeia de informação  $I$  de um vetor transgenético pode conter (a) fragmentos de DNA obtidos do repositório de informações genéticas ou (b) regras abstratas de rearranjo

genético.

Em consonância com a inspiração biológica, há quatro tipos de vetores transgenéticos: plasmídios, plasmídios recombinados, vírus e transposons. Neste trabalho, enfatizam-se os vetores utilizados nos algoritmos desenvolvidos, a saber: plasmídios, plasmídios recombinados e transposons.

Tabela 43: Procedimentos dos vetores transgenéticos.

Procedimento	Nome	Descrição
$p_1$	Ataque	Verifica, de acordo com um dado critério, se o cromossomo $C$ é suscetível, ou não, à manipulação pelo vetor transgenético.
$p_2$	Transcrição	Define como a cadeia de informação de $\lambda$ é transcrita em $C$ . A transcrição é executada apenas se o procedimento $p_1$ retorna “verdadeiro”.
$p_3$	Bloqueio	Estabelece um período de tempo (isto é, um número de iterações) durante o qual a informação transcrita não pode ser alterada em $C$ .
$p_4$	Identificação	Identifica as posições em $C$ que vão ser utilizadas para limitar a operação de $\lambda$ .

No caso dos plasmídios e plasmídios recombinados, a cadeia de informação é uma sequência de genes que representa uma solução parcial. Plasmídios e plasmídios recombinados diferem apenas quanto à maneira com que suas cadeias de informação são obtidas. Plasmídios obtêm sua cadeia de informação de uma fonte do repositório do hospedeiro. A cadeia de informação de um plasmídio recombinado pode ser obtida da combinação de duas ou mais fontes do repositório do hospedeiro, pode ser gerada por uma heurística ou método exato, ou ainda, pode ser o resultado da combinação de informações obtidas do repositório do hospedeiro com informações obtidas por algum método (GOLDBARG; GOLDBARG, 2009).

O método de manipulação dos plasmídios, recombinados ou não, é composto pelos procedimentos de *ataque* ( $p_1$ ) e *transcrição* ( $p_2$ ). O tamanho de um plasmídio é obtido pelo número de genes na sua cadeia de informação.

A cadeia de informação de um transposon apresenta regras para rearranjo de genes no cromossomo do endossimbionte. Por exemplo, estas regras podem refletir uma ação do transposon no sentido de permutar os genes de um determinado intervalo do cromossomo. Os procedimentos de manipulação utilizados são *ataque* ( $p_1$ ), *transcrição* ( $p_2$ ) e *identificação* ( $p_4$ ) (GOLDBARG; GOLDBARG, 2009). No âmbito deste trabalho, não são

considerados transposons que lidam com mais de um cromossomo.

O equilíbrio pontuado emerge pelo modo pelo qual ocorre a seleção dos agentes transgenéticos para atuar sobre a população. Se o algoritmo transgenético incorpora o equilíbrio pontuado, cada agente tem uma probabilidade inicial de ser escolhido para manipular a população e, de tempos em tempos, tais probabilidades são alteradas. Nesse sentido, o período sem mudanças corresponde à fase de equilíbrio e o momento em que as probabilidades são alteradas correspondem às modificações pontuais e rápidas.

O algoritmo TC (Transgenética Computacional) exibido a seguir apresenta um *framework* genérico para a abordagem da transgenética computacional. Uma população inicial de cromossomos é iniciada e avaliada. O repositório do hospedeiro é inicializado com informações *a priori* e, se há alguma informação interessante na população, então ela também é incluída no referido repositório. Os passos de 3 a 8 são repetidos até que um critério de parada seja satisfeito. No passo 4, os vetores transgenéticos são criados e, no passo 5, um conjunto de cromossomos é escolhido para ser manipulado. Se a ação dos vetores transgenéticos nos cromossomos produz novas informações, então o repositório do hospedeiro é atualizado no passo 7.

---

**Algoritmo 9** Algoritmo TC

---

Gerar e avaliar a população de cromossomos

Inicializar o repositório do hospedeiro(RH)

**repetir**

    Gere vetores transgenéticos

    Selecione cromossomos para manipulação

    Manipule os cromossomos selecionados

    Atualize RH

**até que** um critério de parada seja satisfeito

---

Goldbarg e Goldbarg (2009) defendem que, em um algoritmo transgenético, vários tipos de informações são combinadas, por meio de contextos de informação interdependentes, porém autônomos. Assim, o aninhamento destes contextos resulta em um comportamento complexo. Em virtude desta característica, a abordagem transgenética se caracteriza pelo gerenciamento de informações. Nesse sentido, se o problema é capaz de fornecer informação de qualidade, esta metaheurística tem grande probabilidade de obter resultados competitivos quando comparada a outras abordagens.

## APÊNDICE D – Busca Tabu

### D.1 Breve histórico

A Busca Tabu foi proposta inicialmente por Glover (1986), a fim de permitir que métodos de Busca Local pudessem ir além de ótimos locais. A Busca Tabu emprega elementos como a memória de curto prazo, usada para evitar a reversão de movimentos recentes, e a memória de longo prazo, usada reforçar componentes atrativos. O princípio básico da Busca Tabu é realizar uma busca local até que um ótimo local seja encontrado. Quando isso acontece, ela é capaz de escapar deles permitindo movimentos não aprimorantes; a volta à soluções já visitadas é impedida pelo uso das memórias, chamadas de listas tabu, que armazenam o histórico recente de busca, uma idéia que pode ser associada aos conceitos de Inteligência Artificial. É Importante ressaltar também que Glover considerava a Busca Tabu uma metaheurística, isto é, uma estratégia geral para guiar e controlar heurísticas internas especificamente projetadas para os problemas em questão (GENDREAU; POTVIN, 2010).

### D.2 Espaço de busca e estrutura de vizinhança

Uma Busca Tabu simples pode ser vista como uma combinação de Busca Local e memória de curto prazo. Dessa forma, dois dos elementos básicos de uma Busca Tabu são as definições de espaço de busca e estrutura de vizinhança.

O espaço de busca de uma Busca Local ou de uma Busca Tabu é simplesmente o espaço de todas as soluções que podem ser visitadas durante a busca. Para o problema da AGQA-bi, o espaço de busca é constituído por todas as árvores geradoras do grafo de entrada.

O conceito de estrutura de vizinhança está intimamente relacionado ao conceito espaço de busca. Em cada iteração de uma Busca Local ou Busca Tabu, as transformações locais

que podem ser aplicadas à solução corrente,  $S$ , definem um conjunto de soluções vizinhas no espaço de busca,  $N(S)$ . De maneira formal,  $N(S)$  é um subconjunto do espaço de busca constituído de todas as soluções obtidas pela aplicação de uma única transformação sobre  $S$ . Normalmente, existem muito mais estruturas de vizinhança possíveis do que definições de espaço de busca distintas para um dado problema. A escolha do espaço de busca e da estrutura de vizinhança é um passo crítico no projeto de um algoritmo de Busca Tabu (GENDREAU; POTVIN, 2010).

### D.3 Lista Tabu

A lista tabu é o principal elemento diferenciador, quando comparamos a Busca Tabu com a Busca Local. Como já mencionado, a lista tabu é empregada para prevenir a volta a soluções já visitadas quando estamos tentando escapar de ótimos locais por meio de movimentos não aprimorantes. A lista tabu, portanto, impede que movimentos tabu (proibidos) revertam alterações recentes. Os movimentos tabu permanecem na lista somente por um determinado período conhecido como tabu *tenure*. A lista tabu ajuda também na exploração de partes do espaço de busca ainda pouco exploradas, favorecendo exploração mais extensiva.

A lista tabu forma a memória de curto prazo da busca e, normalmente, somente apenas uma pequena quantidade de informações é armazenada em cada entrada da lista. Existem várias possibilidades para as informações específicas que são armazenadas. Até mesmo soluções completas poderiam eventualmente ser armazenadas, embora essa estratégia raramente seja empregada, em virtude do elevado espaço de armazenamento necessário e do alto custo computacional envolvido. Comumente, as informações armazenadas são as poucas últimas transformações realizadas na solução corrente, proibindo a reversão das mesmas.

### D.4 Critério de Aspiração

Apesar de serem essenciais para a Busca Tabu, as listas tabus, muitas vezes, são muito restritivas: elas podem proibir movimentos atrativos, mesmo quando não há risco de movimentos cíclicos, ou podem levar a uma estagnação do processo de busca. É, portanto, necessário permitir a revogação de movimentos tabu, de acordo com critérios algorítmicos, conhecidos como critérios de aspiração. Um dos critérios de aspiração mais simples,



encontrando em praticamente todas as implementações de Busca Tabu, é permitir um movimento tabu, desde que a solução resultante seja melhor do que a melhor solução encontrada até o momento. No caso multiobjetivo, isso seria equivalente a permitir um movimento tabu se a solução em questão fosse não dominada.

## D.5 Uma Busca Tabu simples

O pseudocódigo de uma Busca Tabu mono-objetiva é apresentado a seguir:

---

### Algoritmo 10 Algoritmo BuscaTabu

---

```

Construa uma solução inicial  $S_0$ ;
 $S \leftarrow S_0$ ;
 $f^* \leftarrow f(S_0)$ ;
 $S^* \leftarrow S_0$ ;
 $T \leftarrow \{\}$ ;
enquanto critério de parada não for satisfeito faça
  Selecione  $S$  em  $\operatorname{argmin}_{S' \in N'(S)} [f(S')]$ ;
  se  $f(S) < f^*$  então
     $f^* \leftarrow f(S)$ ;
     $S^* \leftarrow S$ ;
  fim se
  Armazena o movimento corrente em  $T$ ;
fim enquanto

```

---

## D.6 Critério de terminação

O critério de parada não foi especificado no pseudocódigo acima, já que ele pode ser escolhido de diferentes maneiras. Os critérios de parada mais comuns empregados em Buscas Tabu são:

- Número fixo de iterações ou de tempo computacional;
- Número fixo de iterações sem melhoria;
- Quando o objetivo atinge um valor pré-determinado.