



**UNIVERSIDADE FEDERAL RURAL DO SEMIÁRIDO  
UNIVERSIDADE DO ESTADO DO RIO GRANDE DO NORTE  
MESTRADO EM CIÊNCIA DA COMPUTAÇÃO**



**ERNANDO GOMES DE SOUSA**

**UM PROCEDIMENTO EXATO E METAHEURÍSTICAS  
MULTIOBJETIVOS PARA O PROBLEMA BIOBJETIVO  
DA ÁRVORE GERADORA DE CUSTO E DIÂMETRO  
MÍNIMOS**

**MOSSORÓ – RN  
FEV / 2014**

**ERNANDO GOMES DE SOUSA**

**UM PROCEDIMENTO EXATO E METAHEURÍSTICAS  
MULTIOBJETIVOS PARA O PROBLEMA BIOBJETIVO  
DA ÁRVORE GERADORA DE CUSTO E DIÂMETRO  
MÍNIMOS**

Dissertação apresentada ao de Programa de Pós-Graduação em Ciência da Computação – associação ampla entre a Universidade do Estado do Rio Grande do Norte e a Universidade Federal Rural do Semiárido, para a obtenção do título de Mestre em Ciência da Computação.

Orientador: Dr. Dario José Aloise - UERN  
Co-orientadora: Dra Andréa Cynthia Santos  
(Université de Technologie de Troyes – UTT,  
França)

**MOSSORÓ – RN  
FEV / 2014**

**Catálogo da Publicação na Fonte.**  
**Universidade do Estado do Rio Grande do Norte.**

Sousa, Ernando Gomes de.

Um procedimento exato e metaheurísticas multiobjetivos para o problema biobjetivo da árvore geradora de custo e diâmetro mínimos. / Ernando Gomes de Sousa. – Mossoró, RN, 2014

71 f.

Orientador(a): Prof. Dr. Dario José Aloise.

Dissertação (Mestrado em Ciência da Computação). Universidade do Estado do Rio Grande do Norte. Universidade Federal Rural do Semi-Árido. Programa de Pós-Graduação em Ciência da Computação.

1. Árvore geradora - Dissertação. 2. Método exato multiobjetivo - Dissertação. 3. Metaheurística multiobjectivo - Dissertação. I. Aloise, Dario José. II. Universidade do Estado do Rio Grande do Norte. III. Título.

UERN/BC

CDD 004

ERNANDO GOMES DE SOUSA

**UM PROCEDIMENTO EXATO E METAHEURÍSTICAS  
MULTIOBJETIVOS PARA O PROBLEMA BIOBJETIVO  
DA ÁRVORE GERADORA DE CUSTO E DIÂMETRO  
MÍNIMOS**

Dissertação apresentada ao Programa  
de Pós-Graduação em Ciência da  
Computação para a obtenção do título  
de Mestre em Ciência da Computação.

APROVADA EM: 14/02/2014.

BANCA EXAMINADORA



Prof. Dr. DARIO JOSÉ ALOISE - UERN  
Orientador - Presidente



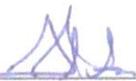
Prof. Dra. ANDREA CYNTHIA SANTOS – Université de Technologie de Troyes –  
UTT, França  
Co-orientadora – Membro Externo



Prof. Dr. CHRISTOPHE DUHAMEL - Université Blaise Pascal – UBP, França  
Membro Externo



Prof. PhD DANIEL ALOISE – UFRN  
Membro Externo



Prof. Dr. CARLOS HEITOR LIBERALINO – UERN  
Membro Interno

## AGRADECIMENTOS

Para concluir este trabalho precisei realizar uma prazerosa caminhada, o qual tive diversas colaborações e por isso os meus agradecimentos:

A Deus por ter me dado força para realizá-lo;

Ao professor Dr. Dario José Aloise pela paciente e sábia orientação, e por ter acreditado em minha pessoa. A Professora Dra. Andréa Cynthia Santos pela sua dedicada, amiga e inteligente co-orientação. Com eles adquiri mais do que conhecimento acadêmico, adquiri conhecimento de vida;

A minha família pelo incentivo, em especial meu pai Salomão Alves de Sousa, minha mãe Enoides Gomes de Sousa e ao meu irmão Davy Gomes de Sousa;

A Cyrlane da Silva Rabelo por todo apoio e envio pelos correios dos meus documentos de inscrição no programa do mestrado;

Aos professores Ms Cleverton Marlon Possani, Ms Fabio Roberto Pillatt e Ms Rodrigo Martins Ferreira (*in memoriam*) pela torcida e por terem me concedido a carta de recomendação para ingresso no programa;

A Diego Rocha Lima por todas as solicitações atendidas durante o desenvolvimento desta pesquisa;

A Rita Santos e Luciano Carlos Costa pelo apoio prestado durante a viagem a França e boas conversas durante o período do mestrado;

A Sra. Rosita Rodrigues Bezerra, secretaria do PPgCC, por toda atenção que me concedeu;

A Íthalo Bruno Grigório de Moura, Fábio Arantes Diniz, João Phellipe Freitas Pinto e Thiago Reis da Silva pela boa receptividade em Mossoró-RN e no mestrado;

Ao pessoal da República IPIZero, a estourada de Mossoró-RN, pelo bons momento de convivência, em especial a Guimarães Silva Ribeiro de Sousa, juntamente com o pessoal de Ipanguaçu-RN, Diógenes Lopes de Figueiredo, João Mauricio Ferreira Aguiar e Francisco Eduardo Silva Alencar;

Ao pessoal do Cajueirense Futebol Clube, em especial Florismário Gomes Sousa, Michel Lima Macedo e Adwilton Lima de Castro;

As pessoas que conheci em Mossoró-RN, em especial a Juliana Almeida de Lira, Karl Hansimuller Alelaf Ferreira e a professora Dr. Angélica Félix de Castro;

Aos amigos de Balsas-MA, em especial a Wellington Cunha de Souza, Belton Moura dos Santos, Allysson de Oliveira Viera, Francisca das Chagas Costa, Jorge Adriano Grampes, Elias Alves da Costa, Washington Luís Ferreira Novaes, Laélia P. Leite Bertoldo, Bruno Leite Bertoldo, Lusitonia da Silva Leite, Marluzio da Silva Leite, Ana Maria Silva de Sousa, Paulo Enoque de Sousa Carvalho;

A CAPES - Coordenação de Aperfeiçoamento de Pessoal de Nível Superior, pelo apoio financeiro que viabilizou a realização deste trabalho;

E a todos que contribuirão direta e indiretamente para a conclusão deste trabalho.

## RESUMO

O problema biobjetivo da Árvore Geradora de Custo e Diâmetro Mínimo (bi-AGCDM) é uma generalização do problema da árvore geradora com restrição de diâmetro. O bi-AGCDM é NP-Difícil e definido em um grafo  $G=(V,E)$ , onde o objetivo é encontrar árvores geradoras de custo e diâmetro mínimos. Problemas de redes de comunicação, de transmissão de energia, de transportes, de computadores, entre outras, podem ser modeladas como um problema bi-AGCDM. Este trabalho apresenta um procedimento exato para encontrar a fronteira de Pareto ótima do bi-AGCDM. Além disso, duas metaheurísticas multiobjetivos são propostas para o problema. A primeira é uma adaptação do *Non-dominated Sorting Genetic Algorithm – II* (NSGA-II) e a segunda é uma versão multiobjetivo do *Particle Swarm Optimization* (PSO). Os experimentos foram realizados com três grupos de instâncias e o desempenho das metaheurísticas foi avaliado através das métricas de quantidade de elementos na fronteira de Pareto, espalhamento das soluções e hipervolume. Os resultados mostram que a adaptação sugerida ao NSGA-II melhora o seu desempenho e o PSO tem excelentes resultados.

**Palavras chave:** problema bi-AGCDM, árvore geradora, método exato multiobjetivo, metaheurística multiobjectivo, otimização combinatória, NP-árduo.

## ABSTRACT

The bi-objective Minimum Diameter-Cost Spanning Tree problem (bi-MDCST) is a generalization of the diameter-constrained minimum spanning tree problem. The bi-MDCST is NP-Hard and set in a graph  $G = (V, E)$ , where the goal is to find cost-diameter minimum spanning trees. Problems of communication networks, power transmission, transportation, computers, among others, can be modeled as a bi-AGCDM problem. This work presents an exact procedure for finding the optimal Pareto front bi-AGCDM. In addition, two multi-objective metaheuristic are proposed for the problem. The first is an adaptation of Nondominated Sorting Genetic Algorithm - II (NSGA-II) and the second is a multi-objective version of the Particle Swarm Optimization (PSO). The experiments were conducted with three groups of instances and the performance of metaheuristics was assessed through measurements of quantity of elements in the Pareto frontier, spacing of solutions and hypervolume. The results show that the change proposed to the NSGA-II improves its performance and the PSO has excellent results.

**Keywords:** bi-MDCST problem, spanning tree, multi-objective exact method, multi-objective metaheuristic, combinatorial optimization, NP-hard.

## LISTA DE FIGURAS

Figura 1 - Exemplo de AGCM e AGDM. Fonte: Santos (2004).....	14
Figura 2 - Grafo esparso com diâmetro 4 e árvore geradora mínima com diâmetro igual a 6. ....	22
Figura 3 – Exemplo de árvore geradora com diâmetro 2 e 3. Fonte: Santos (2004). ...	24
Figura 4 - Um exemplo de pesquisa no método $\epsilon$ -constraint. Fonte: Lemestre, Dhaenens e Talbi (2007). ....	25
Figura 5 - Fases do PPM.....	26
Figura 6 - Exemplo de um grafo com mais AGCM com diâmetros diferentes. ....	27
Figura 7 - Exemplo de grafo que possui duas árvores de diâmetro mínimo. ....	28
Figura 8 - Procedimento geral do NSGA-II. Fonte: Deb <i>et al.</i> (2002). ....	37
Figura 9 - Representação de uma árvore. ....	39
Figura 10 - Exemplo de união de grafos. Fonte: Lima (2012).....	41
Figura 11 - Busca local de troca do nó pai.....	42
Figura 12 - Fluxograma da metaheurística NSGA-II com as estratégias sugeridas. ....	43
Figura 13 - Fluxograma da metaheurística PSO. ....	47
Figura 14 - Fronteira de Pareto ótima para c_v20_a190_d4. ....	53
Figura 15 - Fronteira de Pareto ótima para c_v20_a190_d5. ....	54
Figura 16 - Fronteira de Pareto ótima para p_v40_0.10.....	54
Figura 17 - Fronteira de Pareto ótima para s_v40_a100_d4. ....	54
Figura 18 - Hipervolume. Fonte: Lima (2012).....	58
Figura 19 - Comparação entre o PSO e NSGA-II para a instância s_v40_a100_d6 .....	63
Figura 20 - Comparação entre o PSO e o NSGA-II para a instância sr_v60_a600_d5....	63
Figura 21 - Comparação entre o PSO e o NSGA-II para a instância hc_v25_e120.....	64

## LISTA DE TABELAS

Tabela - 1: Resultados de fronteiras de Pareto para instâncias Hamiltonien cycle e Hamiltonien path.....	55
Tabela - 2: Resultados de fronteiras de Pareto para instâncias Hamiltonien cycle e Hamiltonien path.....	56
Tabela 3 - Calibrando o NSGA-II.....	58
Tabela 4 - Calibrando o PSO.....	59
Tabela 5 - Comparação de resultados entre NSGA-II (A) e NSGA-II (B) para Santos, Lucena e Ribero (2004).....	61
Tabela 6 - Comparação de resultados entre o procedimento exato, NSGA-II (A) e o NSGA-II (B) para Santos, Lucena e Ribeiro (2004).....	61
Tabela 7 - Comparação de resultados entre o procedimento exato, NSGA-II (A) e o NSGA-II (B) para Santos, Lima e Aloise (2013).....	62
Tabela 8 - Comparação de resultados entre o NSGA-II (A) e o NSGA-II (B) para Gouveia e Magnanti (2003).....	62
Tabela 9 - Comparação de resultados entre NSGA-II e PSO para Santos, Lima e Aloise (2012).....	64
Tabela 10 - Comparação de resultados entre o NSGA-II e o PSO para Santos, Lima e Aloise (2013).....	65
Tabela 11 - Comparação de resultados entre NSGA-II e o PSO para Gouveia e Magnanti (2003).....	65

## LISTA DE ALGORITMOS

Algoritmo 1 - Procedimento exato para calcular o bi-AGCDM. ....	31
Algoritmo 2 - Pseudocódigo do NSGA-II.....	38
Algoritmo 3 - Pseudocódigo do PSO. Fonte: Oliveira, Silva e Aloise (2004).....	46

**LISTA DE ABREVIATURAS E SIGLAS**

AGCM	Árvore Geradora de Custo Mínimo
AGDM	Árvore Geradora de Diâmetro Mínimo
bi-AGCDM	Problema Biobjetivo da Árvore Geradora de Custo e Diâmetro Mínimos
AGMRD	Árvore Geradora Mínima com Restrição de Diâmetro
GRASP	<i>Greedy Randomized Adaptive Search Procedure</i>
IBM	<i>International Business Machines</i>
MILP	<i>Mixed Integer Linear Programming</i>
MOEA	<i>Multi-objective Evolutionary Algorithm</i>
NSGA-II	<i>Nondominated Sorting Genetic Algorithm – II</i>
PPM	<i>Parallel Partitioning Method</i>
PSO	<i>Particle Swarm Optimization</i>
TPM	<i>Two-phases method</i>

## SUMÁRIO

1	INTRODUÇÃO.....	14
2	TRABALHOS RELACIONADOS .....	17
3	DEFINIÇÕES, MODELAGENS E PROCEDIMENTO EXATO .....	20
3.1	DEFININDO ESPAÇO DE BUSCA DA SOLUÇÃO PARA O PROBLEMA BI-AGCDM .....	21
3.2	CASO DE ÁRVORES GERADORAS QUE POSSUEM ALGORITMO POLINOMIAL.....	23
3.3	MÉTODOS EXATOS BI OBJETIVO .....	24
3.4	UM PROCEDIMENTO EXATO PARA O BI-AGCDM .....	26
3.5	MODELOS MATEMÁTICOS UTILIZADOS NO PROCEDIMENTO EXATO PARA O BI-AGCDM.....	31
3.6	MODELO MULTIFLUXO PARA O BI-AGCDM .....	33
4	METAHEURÍSTICAS MULTI OBJETIVO PARA O BI-AGCDM .....	35
4.1	NSGA – II .....	35
4.2	NSGA – II PARA O BI-AGCDM .....	39
4.3	METAHEURÍSTICA PSO .....	43
4.4	UTILIZANDO O PSO EM PROBLEMAS DISCRETOS .....	47
4.5	APLICANDO O PSO PARA O BI-AGCDM .....	48
5	TESTES E RESULTADOS.....	52
5.1	RESULTADOS DO PROCEDIMENTO EXATO PARA O BI-AGCDM ....	52
5.2	RESULTADOS DAS METAHEURÍSTICAS.....	57
5.2.1	CALIBRAÇÃO DOS ALGORITMOS.....	58
5.2.2	AVALIAÇÃO DO NSGA-II .....	59
5.2.3	COMPARANDO O NSGA-II E O PSO .....	62
6	CONCLUSÕES E TRABALHOS FUTUROS.....	65
7	REFERÊNCIAS BIBLIOGRÁFICAS.....	67

## 1 INTRODUÇÃO

Problemas do dia a dia que são modelados em rede, como por exemplo, rede de computadores e transportes, a fim de diminuir o custo do projeto e melhorar o desempenho da mesma, geralmente necessitam de uma árvore geradora que tenha seu custo reduzido e a distância entre cada par de nós da árvore a menor possível. Uma árvore geradora  $T$  de um grafo  $G = (V, E)$ , onde  $|V|=n$  e  $|E|=m$ , é aquela que contém todos os vértices de  $G$  e possui exatamente  $n - 1$  arestas. Do conjunto de árvores geradoras de um grafo, aquela que tem o menor custo é conhecida como árvore geradora de custo mínimo (AGCM) e aquela que possui o menor diâmetro é conhecida como árvore geradora de diâmetro mínimo (AGDM). O diâmetro de uma árvore é o número de arestas do maior caminho entre todos os pares de nós da mesma. Este trabalho trata o problema biobjetivo da árvore geradora com custo e diâmetro mínimos (bi-AGCDM).

Como exemplo de AGCM e AGDM, considere o grafo da Figura - 1(a). Dentre todas as possibilidades de árvores geradoras presentes neste grafo, a AGCM é ilustrada na Figura - 1(b) com custo 7 e diâmetro 3, e uma possível AGDM é ilustrada na Figura - 1(c) com custo 19 e diâmetro 2.

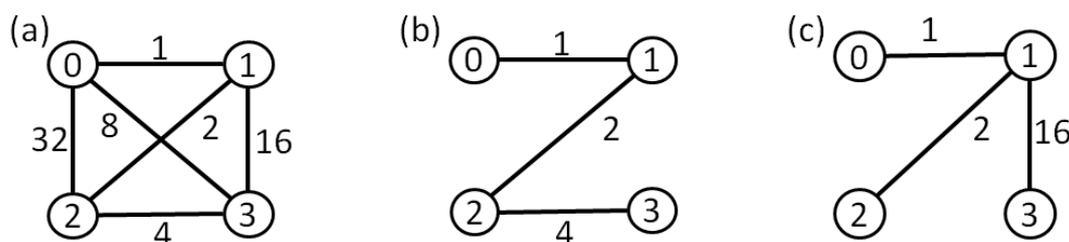


Figura 1 - Exemplo de AGCM e AGDM. Fonte: Santos (2004).

Para calcular a AGCM de um grafo existem diversos algoritmos polinomiais na literatura, tais como os clássicos de Prim e Kruskal, com complexidade no pior caso, respectivamente, de  $O(m \log n)$  e  $O((m + n) \log n)$  (Cormen *et al.*, 1990). Outro algoritmo polinomial que também calcula uma AGCM de  $G$  é baseado na busca em profundidade e consiste em eliminar todos os ciclos de um grafo para encontrá-la, sendo a complexidade deste algoritmo no pior caso é  $O(k \cdot m)$ , onde  $k$  é o número de arestas retornadas na busca em profundidade, ou seja,  $k$

$= m - (n-1)$  (Salehi-Fathabadi e Ahrabian, 1995). Algoritmos eficientes para calcular a AGDM são apresentados em Bui *et al.* (2004).

O bi-AGCDM é NP-Difícil (Ho *et al.*, 1991) e é definido em um grafo não orientado e conexo  $G = (V, E)$ . Neste grafo, um custo  $c_{ij} \geq 0$  está associado a cada aresta  $(i, j) \in E$ . Nele, existe uma árvore geradora  $T$  de  $G$  com um único caminho  $P_{ij}$  ligando qualquer par de nós  $(i, j) \in V$ , onde  $i \neq j$ . Neste caminho,  $d_{ij}$  corresponde ao número de arestas de  $P_{ij}$  e o diâmetro  $D$  de  $T$  é definido como  $D = \max\{d_{ij} : i, j \in V\}$ . Considerando como o primeiro objetivo deste problema a árvore geradora de custo mínimo, e como o segundo objetivo a árvore geradora de diâmetro mínimo, o bi-AGCDM consiste em encontrar uma árvore geradora com diâmetro e custo mínimos (Sousa, Santos e Aloise, 2013).

O problema bi-AGCDM modela diversos problemas. Projetos de rede de computadores e até mesmo rede de distribuição de energia elétrica podem ser modelados como tal. Para estes casos, com o intuito de garantir um custo acessível ao projeto, as estações (vértices) devem comunicar-se entre si a um custo mínimo. Além disso, o diâmetro mínimo da árvore a ser montada é associado à qualidade de serviço (QoS) de comunicação da rede e objetiva diminuir seu retardo de transmissão de dados para aumentar sua confiabilidade. Linhas de trens e metrô também compreendem o problema bi-AGCDM. A infraestrutura forma uma árvore geradora onde os custos das arestas correspondem ao custo de instalação da rede. A árvore com diâmetro mínimo representa o menor caminho em número de arestas entre as estações dos trens e metrô (Lima, Santos e Aloise, 2012).

Para tratar o problema bi-AGCDM foi desenvolvido um procedimento exato que calcula a fronteira de Pareto ótima. Além disto, a metaheurística NSGA-II, o qual já foi proposta por Lima, Santos e Aloise (2012) para o mesmo problema, foi reproduzida neste trabalho com um acréscimo de melhoria. A metaheurística *Particle Swarm Optimization* (PSO) também foi aplicada ao problema e apresentou melhores resultados que o NSGA-II. Por fim, uma comparação entre essas metaheurísticas é realizada por meio das métricas de quantidade de elementos na fronteira de Pareto ótima, hipervolume e *spacing*.

Esta dissertação está dividida em 7 capítulos como se segue: o capítulo 2 é dedicado a revisão bibliográfica do problema bi-AGCDM. No capítulo 3 são apresentadas definições, modelagens e um procedimento exato empregado no problema. No capítulo 4 são apresentadas duas metaheurísticas aplicadas ao bi-AGCDM. No capítulo 5 são exibidos testes e resultados computacionais com os algoritmos desenvolvidos e no capítulo 6 são apresentadas conclusões e trabalhos futuros.

## 2 TRABALHOS RELACIONADOS

O bi-AGCDM é uma generalização do problema da árvore geradora mínima com restrição de diâmetro (AGMRD). Este problema consiste em encontrar uma árvore geradora com custo total mínimo para um diâmetro específico (Santos, 2004) e é NP-difícil quando o diâmetro da árvore geradora a ser buscada é maior do que 3 (Garey e Johnson, 1979). Uma vasta literatura para o AGMRD é disponível: algoritmos exatos são encontrados em T. F. Noronha (2010), Gouveia *et al.* (2011); heurísticas e metaheurísticas foram propostas por Lucena *et al.* (2010), Gruber e Raidl (2005), Requejo e Santos (2009); formulações matemáticas e inequações válidas foram desenvolvidas em Achuthan *et al.* (1994), Gouveia e Magnanti (2003) e Santos *et al.* (2004).

Existem diversos trabalhos na literatura dedicados ao estudo de problemas biobjetivos de árvores geradoras de custo mínimo considerando duas funções de custo, tais como, o algoritmo genético proposto por Zhou e Gen (1999), o *Greedy Randomized Adaptive Search Procedure* (GRASP) apresentado em Arroyo, Vieira e Vianna (2008), o algoritmo de enumeração sugerido por Steiner e Radzik (2008) e o algoritmo *Branch-and-Bound* descrito por Sourd e Spanjaard (2008).

Alguns poucos trabalhos na literatura foram desenvolvidos para resolver o bi-AGCDM. Um estudo teórico sobre algoritmos aproximativos para os problemas de bi-AGCDM e AGCM com minimização de grau (quantidade de arestas incidentes a cada nó de  $G$ ) é apresentado em Marathe *et al.* (1998), sem resultados computacionais. Algoritmos genéticos multiobjetivos para o bi-AGCDM foram propostos por Saha e Kumar (2011) e Santos, Lima e Aloise (2013).

Saha e Kumar (2011) propõem o *Multi-objective Evolutionary Algorithm* (MOEA) para o problema bi-AGCDM. Esta metaheurística é populacional e seus indivíduos são evoluídos através de um histograma de *ranking*. Em cada iteração, através da classificação dos indivíduos, os piores são descartados e novos são gerados a partir de dois indivíduos elite. Um novo indivíduo é gerado

a partir de um operador de recombinação e a metaheurística ainda utiliza dois operadores simples de mutação, chamados de mutação da substituição gulosa de uma aresta e mutação da modificação de uma aresta.

O operador de recombinação cria um novo indivíduo a partir das arestas comuns entre duas soluções selecionadas. Posteriormente, para completar a solução que está sendo criada, as melhores arestas dos dois indivíduos selecionados que ainda não fazem parte da solução são escolhidas. O operador de mutação da substituição gulosa de uma aresta deleta uma aresta randomicamente de uma solução, criando duas subárvores, e insere na solução outra aresta de menor custo para unir as duas subárvores criadas. O operador de mutação da modificação de uma aresta deleta randomicamente uma aresta de uma solução, tornando-a inviável, e logo em seguida utiliza um operador de recombinação entre esta solução inviável e um indivíduo elite da população para se obter uma nova solução. As instâncias de testes utilizadas neste trabalho variam entre 50 e 100 nós.

Uma formulação matemática utilizando multfluxo foi proposta por Santos, Lima e Aloise (2012) para o bi-AGCDM. Neste problema, a minimização do custo está relacionado a escolha das arestas de menores valores e a minimização do diâmetro está diretamente ligada a estrutura dos caminhos entre os nós da árvore geradora. Considerando a minimização do diâmetro, os modelos matemáticos são adaptados para poder contabilizar a quantidade de arestas nos caminhos da árvore.

Santos, Lima e Aloise (2012) utilizaram seu modelo através da otimização em duas fases. Neste tipo de otimização, os objetivos são trabalhados por ordem de prioridade e o diâmetro foi tratado como objetivo principal e o custo como objetivo secundário.

Apesar do modelo proposto por Santos, Lima e Aloise (2012) encontrar soluções ótimas, sua utilização para algumas instâncias é inviável devido a grande quantidade de tempo necessário para processá-la. Para contornar tal problema, foi aplicado duas metaheurísticas para o problema bi-. As duas metaheurísticas aplicadas foram os algoritmos evolucionários NSGA-II e o MOEA. Seus resultados foram comparados através das métricas de quantidade

de elementos na fronteira de Pareto, espalhamento das soluções e hipervolume. O NSGA-II é explicado em detalhes na seção 4.1 e o MOEA foi explanado no quarto parágrafo anterior a este. Para os testes realizados foram utilizados 4 grupos de instâncias. Destes, 3 grupos de instâncias são provenientes dos trabalhos de Lucena *et al.* (2010), Raidl e Julstrom (2003), Gouveia e Magnanti (2003). O quarto grupo de instâncias foi criado no desenvolvimento do trabalho. Nos testes realizados o NSGA-II obteve melhor desempenho e as instâncias testadas variaram de 10 a 250 nós, sendo algumas de grafo completo e outras de grafo esparso.

### 3 DEFINIÇÕES, MODELAGENS E PROCEDIMENTO EXATO

Um problema de otimização combinatória multiobjetivo (MOOP, do inglês *Multi-Objective Optimization Problem*), como é caso do problema bi-AGCDM, possui um conjunto de funções objetivos a serem otimizadas e um conjunto de restrições que devem ser satisfeitas para que uma solução viável ao problema seja encontrada.

Problemas do tipo MOOP possuem funções objetivos que geralmente são conflitantes entre si. Uma função objetivo  $f_1$  é conflitante com outra função  $f_2$  quando não é possível melhorar o valor de  $f_1$  sem piorar o valor da função  $f_2$ . Para esses casos, não existe uma única solução para um problema a ser resolvido, mas sim um conjunto de soluções que respeitam um critério de dominância.

Suponha que uma solução  $x$  para um tal problema possa ser descrita por um vetor de decisão  $(x_1, x_2, x_3, \dots, x_n)$  em um espaço de decisão  $X$ . Este vetor de decisão representa os valores dados para as variáveis do problema. Uma função de custo  $f: X \rightarrow Y$  atribui para  $x$  um vetor objetivo  $(f_1(x), f_2(x), f_3(x), \dots, f_k(x))$  em um espaço objetivo  $Y$ . Neste contexto, o problema de otimização multiobjetivo consiste em encontrar as soluções no espaço de decisão de otimização (minimização ou maximização) para  $k$  objetivos (designa-se  $K$  como o conjunto dos  $k$  objetivos considerados) (Dhaenens, Lemestre e Talbi, 2010).

Dentre todas as soluções do problema a serem encontradas, o conjunto das soluções que não são dominadas por nenhuma outra recebe o nome de fronteira de Pareto ótima. Esse conjunto de soluções que forma a fronteira de Pareto ótima é a solução de problemas multiobjetivos e é definida por Dhaenens, Lemestre e Talbi (2010) como se segue:

**Definição 1:** Em um problema de minimização, uma solução  $x$  domina uma solução  $x'$  se e somente se, respeita as desigualdades descritas em (1):

$$\begin{cases} \forall k \in \mathcal{K}, & f_k(x) \leq f_k(x'), \\ \exists k \in \mathcal{K}, & f_k(x) < f_k(x'). \end{cases} \quad (1)$$

**Definição 2.** Uma solução é Pareto ótima se ela não é dominada por nenhuma outra solução do conjunto viável. Uma solução deste tipo é chamada suportada.

Para a identificação das soluções pertencentes a fronteira de Pareto ótima do problema bi-AGCDM pode-se utilizar heurísticas (em particular metaheurísticas), como por exemplo, a PSO e a NSGA-II, ou métodos exatos, como por exemplo, o *Parallel Partitioning Method* (PPM),  $\varepsilon$ -constraint ou *two-phases method* (TPM).

### 3.1 DEFININDO ESPAÇO DE BUSCA DA SOLUÇÃO PARA O PROBLEMA BI-AGCDM

Definir o espaço de busca da solução para o problema bi-AGCDM significa identificar um subespaço da região de solução que possui todas as árvores geradoras da fronteira de Pareto ótima. Isto é interessante porque analisar todas as possíveis árvores geradoras de um grafo é inviável. Em grafos completos, por exemplo, a quantidade de árvores geradoras presentes no mesmo é definida segundo o Teorema de Caley por  $n^{n-2}$  (Kropiwiec, 2002). No caso do grafo completo que possui somente 4 vértices, por exemplo, existem 16 árvores geradoras diferentes. Num grafo completo com 10 vértices existem  $10^8$  árvores geradoras.

Na definição do espaço de busca, inicialmente é calculada a árvore geradora de custo mínimo e o diâmetro da mesma. Sabe-se que na fronteira de Pareto ótimo não haverá uma árvore geradora de custo menor, pois o seu custo é o mínimo, e diâmetro maior, porque qualquer solução que mantenha esse custo e aumente seu diâmetro é considerada um solução dominada de acordo com as definições de dominância da fronteira de Pareto.



largura é iniciada a partir vértice  $y$ . O valor da distância do vértice mais afastado de  $y$  determinará o diâmetro da árvore.

Para calcular o diâmetro de um grafo o procedimento é similar ao usado em árvore. Inicialmente é escolhido um vértice por onde a busca em largura deve começar, como por exemplo, o vértice  $x$ . O algoritmo é iniciado pelo vértice  $x$  e ao final de sua execução as arestas utilizadas no percorrimento do grafo formaram uma árvore. Em seguida, guardam-se os nós folhas da árvore gerada, por exemplo, os vértices  $z$  e  $y$ , e outras busca em largura no grafo são realizadas a partir de cada um desses nós folhas. Por fim, o diâmetro do grafo é definido pela maior distância encontrada para os nós  $z$  e  $y$  em relação aos demais nós do grafo.

### **3.2 CASO DE ÁRVORES GERADORAS QUE POSSUEM ALGORITMO POLINOMIAL**

Identificar árvores geradoras de custo mínimo com um diâmetro específico em um grafo  $G=(V,E)$  tem algoritmo polinomial em 4 casos. O primeiro caso corresponde em encontrar à AGCM de  $G$ , a qual poder ser calculada pelo algoritmo de Prim, por exemplo. O diâmetro da AGCM é calculado com o algoritmo de busca em largura. O segundo caso ocorre quando  $D = 2$ . Neste caso, o procedimento constrói, a cada iteração, uma árvore geradora considerando um vértice  $i \in G$  como central. A partir do vértice central os outros vértices  $j \in V \setminus \{i\}$  são conectados a ele através de uma aresta  $(i, j)$ . Em seguida, basta comparar as árvores geradoras formada a partir de  $G$  e guardar a de menor custo. Este procedimento tem complexidade computacional no pior caso  $O(n^2)$ . Uma solução de árvore com  $D = 2$  tem a forma de estrela como está ilustrado na Figura - 3(a) (Santos, 2004).

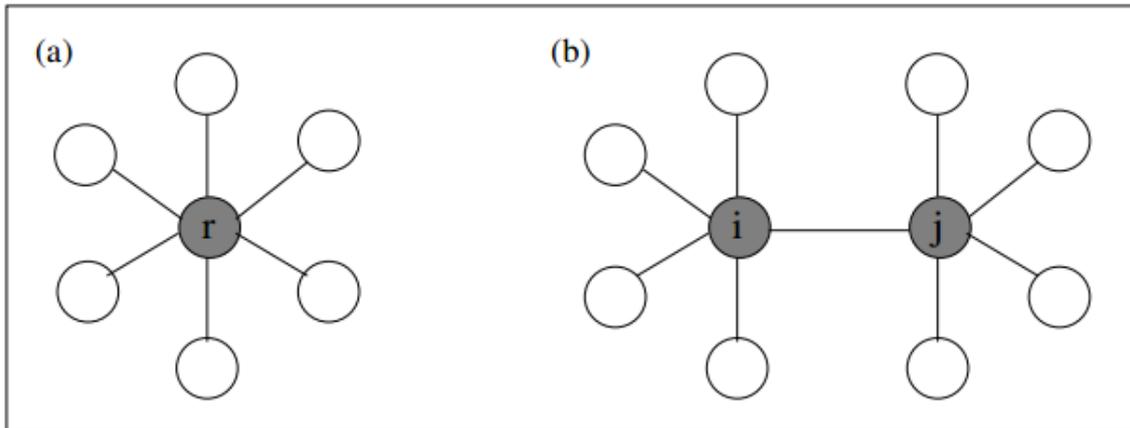


Figura 3 – Exemplo de árvore geradora com diâmetro 2 e 3. Fonte: Santos (2004).

O terceiro caso ocorre quando  $D = 3$ . A cada iteração uma aresta  $[i, j] \in E$  é tomada como aresta central da árvore. Posteriormente os vértices  $k \in V \setminus \{i, j\}$ , diferente dos presentes na aresta central, são conectados na mesma, conforme ilustrado na Figura - 3(b). As conexões a serem realizadas são as de menor custo possível. O algoritmo para calcular uma árvore geradora mínima com  $D = 3$  tem complexidade no pior caso  $O(m.n)$ . O quarto caso ocorre quando todos os custos das arestas são iguais, não influenciando na construção da árvore. Neste caso, localizar a árvore geradora com qualquer diâmetro, dentro do seu espaço de busca, tem complexidade polinomial (Santos, 2004).

### 3.3 MÉTODOS EXATOS BI OBJETIVO

Métodos exatos são algoritmos que geram a solução ótima de um problema. Alguns deles são comumente empregados para calcular a fronteira de Pareto ótima em problemas biobjetivo, como por exemplo, o  $\epsilon$ -constraint, o *Two-phases method* (TPM) e o *Parallel Partitioning Method* (PPM).

O método  $\epsilon$ -constraint é baseado no princípio de enumeração proposto por Haimes e Ladson (1971). A ideia geral deste procedimento é integrar um dos objetivos no conjunto de restrições e otimizar o outro. Assim sendo, um problema restrito de otimização  $P'$  é definido como  $\{\min f_1(x) \mid f_2(x) \leq \epsilon; x \in X\}$  Seu funcionamento consiste em inicialmente procurar o melhor valor para uma

função  $f_1$ . Posteriormente, o melhor valor para  $f_2$  é procurado usando como limite da busca o valor encontrado para  $f_1$ , conforme ilustrado na Figura - 4(a). Este busca é realizada sucessivamente até que não seja localizado nenhum outro valor, Figura – 4(b) (Sousa, Santos e Aloise, 2013).

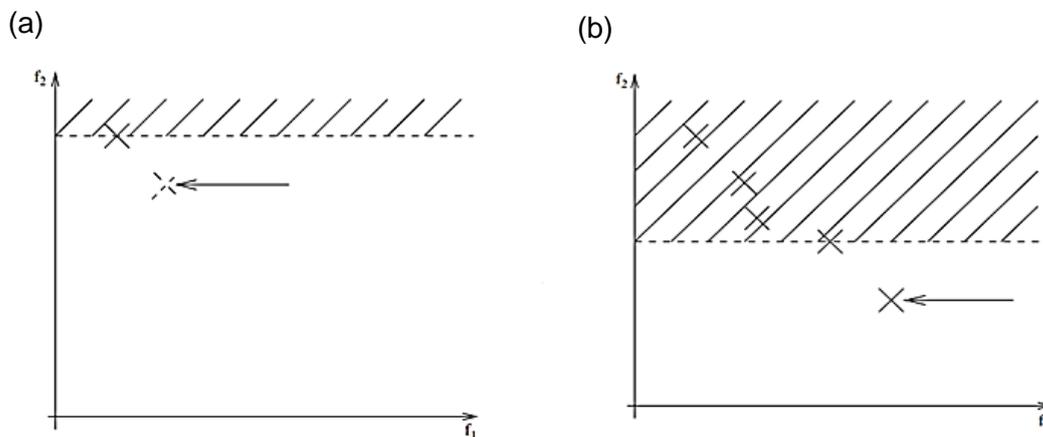


Figura 4 - Um exemplo de pesquisa no método  $\epsilon$ -constraint. Fonte: Lemestre, Dhaenens e Talbi (2007).

O TPM foi proposto por Ulungu e Teghem (1995) e utiliza o conceito de agregação das funções objetivos da seguinte forma: as funções objetivos são devidamente normalizadas e tratadas como uma combinação linear (princípio de agregação) dada por  $\alpha \cdot f_1(x) + (1 - \alpha) f_2(x)$ , onde  $\alpha \in [0,1]$ . Se  $\alpha = 1$ , o objetivo  $f_1(x)$  é otimizado, enquanto que se  $\alpha = 0$ ,  $f_2(x)$  é otimizado. Este método possui um esquema geral e pode ser adaptado para vários problemas. O procedimento para o TPM é realizado em duas fases. Na primeira fase, soluções suportadas são calculadas utilizando o princípio de agregação. Ele inicia buscando as soluções extremas da fronteira de Pareto. Em seguida, entre duas soluções uma busca é realizada com a finalidade de encontrar novas soluções. Quando o procedimento não encontra mais soluções suportadas, a próxima fase é iniciada. A segunda fase é dedicada a localizar os pontos não suportados da fronteira de Pareto no espaço entre cada par de soluções.

O algoritmo PPM, proposto por Lemestre, Dhaenens e Talbi (2007), opera em três fases e é inspirado no  $\epsilon$ -constraint e no TPM. A primeira fase consiste em definir o limite inferior e superior do espaço de busca. Afim de acelerar a procura, cada extremo é encontrado sob uma ordem lexicográfica,

ou seja,  $f_1$  é otimizado primeiro sem considerar  $f_2$ . Posteriormente,  $f_2$  é otimizado sem considerar  $f_1$ . Essas duas soluções são os limites do espaço de busca de soluções, pois eles indicam o menor e o maior valor de diâmetro para o conjunto de soluções viáveis, Figura - 5 (a).

A segunda fase do algoritmo PPM consiste em dividir igualmente o espaço de solução em vários subconjuntos bem distribuídos, como na Figura - 5 (b). As soluções extremas definidas no primeiro estágio são usadas como base para fazer a divisão. Todas as divisões realizadas são associadas a um objetivo e para cada uma tem-se uma solução computada. Essa solução tem o melhor valor para o objetivo não usado para a divisão. O terceiro estágio consiste em localizar todas as soluções não encontradas durante os estágios anteriores. Para cada duas soluções adjacentes uma pesquisa é realizada em busca de uma nova solução, Figura - 5 (c), (Lemestre, Dhaenens e Talbi, 2007).

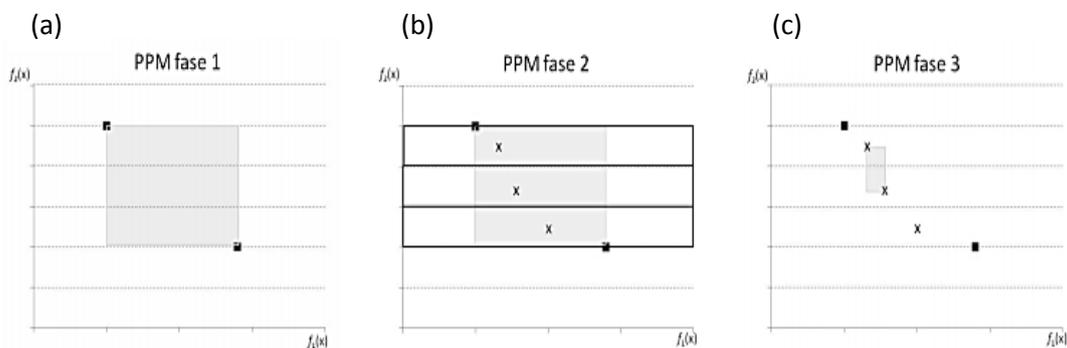


Figura 5 - Fases do PPM.

### 3.4 UM PROCEDIMENTO EXATO PARA O BI-AGCDM

Este trabalho utiliza um procedimento exato para encontrar a fronteira Pareto ótima. O procedimento proposto referencia o custo e o diâmetro como o primeiro e o segundo objetivo, respectivamente. Inicialmente, ele usa a primeira

fase do PPM. Assim sendo, o limite do espaço de busca da solução é definido de acordo com a seção 3.1. Se o grafo  $G$  considerado possui algumas arestas com custos idênticos, diferentes AGCM de  $G$  podem existir. A Figura 6 exemplifica tal situação. Um grafo é dado na Figura - 6(a) e duas AGCM deste grafo são apresentadas na Figura - 6(b) e (c), com custos iguais a 37 e diâmetros iguais a  $D = 5$  e  $D = 7$ , respectivamente. A solução da Figura - 6(c) é dominada pela solução da Figura - 6(b) com relação ao objetivo da minimização do diâmetro.

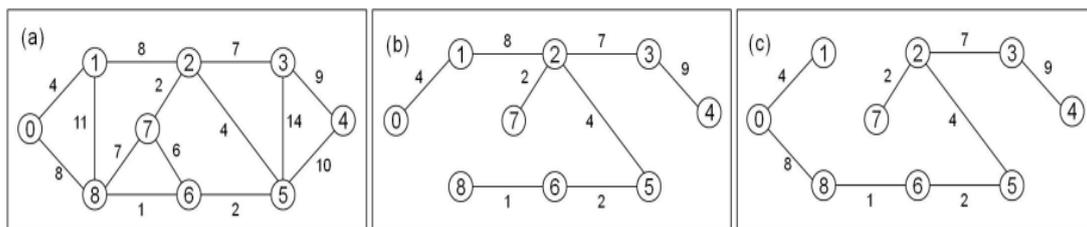


Figura 6 - Exemplo de um grafo com mais AGCM com diâmetros diferentes.

Se o grafo fornecido tem diâmetros  $D = 2$  ou  $D = 3$  pode-se calcular a árvore geradora de diâmetro mínimo conforme explicado na seção 3.2. Para os casos em que o diâmetro  $D \geq 4$ , o problema passa a ser NP-difícil e modelos matemáticos para calculá-los são apresentados posteriormente.

Assim como a AGCM, diferentes árvores de diâmetro mínimo podem existir em um grafo  $G$ . Um exemplo de tal fato é apresentado na Figura - 7. Para o grafo  $G$  da Figura - 7(a), cujo seu diâmetro é igual a 4, duas árvores de diâmetro mínimo são fornecidas nas Figuras - 7(b) e (c), sendo a primeira com custo igual a 47 e a segunda com custo igual a 54. Desse modo, a solução da Figura - 7(b) domina a solução da Figura - 7(c) por causa do valor da função de minimização do custo.

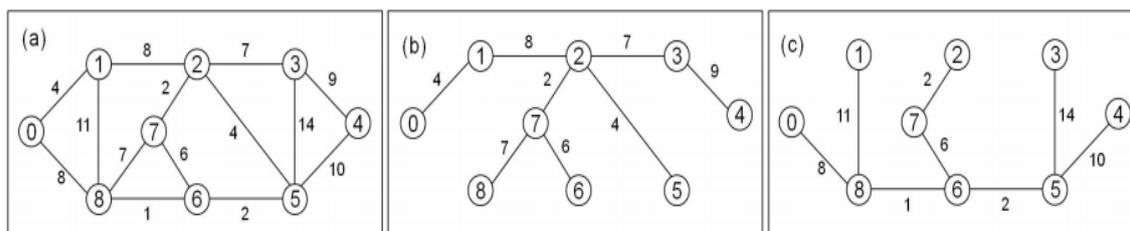


Figura 7 - Exemplo de grafo que possui duas árvores de diâmetro mínimo.

O segundo passo do procedimento proposto para o bi-AGCDM consiste em particionar o espaço de busca em retângulos idênticos como na segunda fase do PPM. Para o bi-AGCDM, esta divisão é realizada utilizando a função objetivo do diâmetro, pois corresponde a valores inteiros. Considerando os pontos extremos encontrados na primeira fase, divide-se o espaço de busca em retângulos com a altura 1, de modo que cada retângulo contenha uma solução. O processo para encontrar uma solução em cada partição baseia-se no método  $\varepsilon$ -constraint.

Encontrar uma AGCM com restrição de diâmetro  $3 < D < n - 1$  corresponde a tratar um problema NP-Difícil. Nestes casos, provar a otimalidade está condicionado e limitado pela instância fornecida. As formulações matemáticas utilizadas neste trabalho para encontrar árvores geradoras com diâmetro maior ou igual a 4, descritas a seguir, correspondem à versão melhorada dos modelos de Achuthan *et al.* (1994), proposta em Santos *et al.* (2004). Embora os modelos utilizados tenham uma relaxação linear mais fraca que os modelos baseados em multifluxo, de Gouveia e Magnanti (2003), softwares de programação linear mista trabalham eficientemente sob tais modelos. Uma possível razão é a quantidade de restrições e variáveis que nos modelos multifluxo são da ordem de  $O(n^4)$ , enquanto que nos modelos propostos por Achuthan *et al.* (1994) são significativamente menores, como por exemplo, no modelo que calcula a árvore geradora como diâmetro ímpar tem-se  $O(m+n)$  restrições,  $O(m+n)$  variáveis inteiras e  $O(n)$  variáveis contínuas.

Os modelos utilizados buscam AGMRD com um diâmetro menor ou igual a um valor  $D$  fornecido, onde  $D \geq 4$ . Uma vantagem do emprego dos modelos utilizados neste trabalho é que se a solução  $d$  retornada por esses modelos é menor que  $D$ , ou seja,  $d < D$ , isso implica que uma árvore geradora com

diâmetro  $D$  é dominada pela solução com diâmetro  $d$ . Sendo assim, árvores geradoras com diâmetro  $d + 1$  até  $D$  não precisam ser mais calculadas, pois elas são dominadas por  $d$ , ocasionando assim uma economia de tempo de cálculo da solução.

O pseudocódigo do procedimento proposto é dado no Algoritmo 1. As variáveis  $UB$ ,  $LB$ ,  $F^*$  e  $d$  representam, respectivamente, o limite superior de diâmetro, limite inferior de diâmetro, o conjunto de soluções Pareto e uma variável auxiliar que determina o diâmetro alvo da busca de uma AGMRD. Além destas,  $T_k$  é o  $k$ -ésimo elemento do conjunto de Pareto ótimo.

As linhas de 1 a 7 são os passos de inicialização do algoritmo. Um contador de soluções é inicializado na linha 1. Em seguida, a AGCM de  $G$  é computada na linha 2 usando o algoritmo de Prim e esta solução,  $T_k$ , é inserida no conjunto de soluções Pareto ótimo  $F^*$ , na linha 3. Logo após, as variáveis  $k$ ,  $UB$ ,  $LB$  e  $d$  são respectivamente atualizadas da linha 4 até a 7. Na linha 8, o procedimento verifica se existe mais de uma solução na fronteira de Pareto para a instância que está sendo tratada, pois se  $UB$  e  $LB$  forem iguais só existe uma solução no espaço de busca da solução. Em caso positivo, uma iteração do laço da linha 9 a 27 é executada para cada partição do espaço de busca, iniciando pelo limite superior, até atingir o limite inferior.

Quando  $d > 3$ , um programa comercial de programação linear mista é aplicado na linha 11. Se uma solução existe para o diâmetro correspondente,  $F^*$  e  $k$  são respectivamente atualizados na linha 13 e 14. Os modelos matemáticos procuram uma AGCM com menor diâmetro ou igual ao valor da variável  $d$ . Desta forma, o objetivo da linha 15 até a 17 é averiguar se uma solução com o diâmetro  $d$  foi encontrada ou não. Caso não,  $d$  é atualizado na linha 16 com um valor abaixo do atual. Da linha 20 até 25 o algoritmo trata os casos em que AGMRD são polinomiais ( $d = 2$  e  $d = 3$ ). Sempre que uma solução é encontrada (linha 21), o conjunto Pareto  $F^*$  e  $k$  são respectivamente atualizados nas linhas 22 e 23.

Finalmente, um procedimento *Cleaning* (linha 26) verifica quais soluções da fronteira Pareto são dominadas e as elimina. O procedimento termina depois que todo o espaço entre os pontos extremos que correspondem a

AGCDM e a AGDM são explorados. A linha 30 retorna a fronteira Pareto ótima. O espaço de busca é implicitamente decomposto. De fato, o procedimento computa soluções entre o limite superior e inferior do espaço de busca, considerando um diâmetro para cada partição.

```

    Leia:  $G=(V,E)$ 
1.  $k \leftarrow 0$ 
2.  $T_k \leftarrow \text{computa\_AGCM}$ ;
3.  $F^* \leftarrow F^* \cup \{T_k\}$ 
4.  $k \leftarrow k + 1$ ;
5.  $UB \leftarrow \text{busca\_diametro}(T_k)$ ;
6.  $LB \leftarrow \text{busca\_diametro}(G)$ ;
7.  $d \leftarrow UB$ ;
8. If( $UB < LB$ ) faça
9.     Enquanto( $d \geq LB$ ) faça
10.        Se( $d > 3$ )
11.             $T_k \leftarrow \text{rode\_MILP\_solver}(G, d)$ ;
12.            se( $T_k < \emptyset$ ) faça
13.                 $F^* \leftarrow F^* \cup \{T_k\}$ ;
14.                 $k \leftarrow k + 1$ ;
15.                If( $\text{busca\_diametro}(T_k) < d$ ) faça
16.                     $d \leftarrow \text{diâmetro de } T_k$ ;
17.                Fim se
18.            Fim se
19.        Senão
20.             $T_k \leftarrow \text{rode\_procedimento\_polinomial}(G, d)$ ;
21.            Se( $T_k < \emptyset$ ) faça
22.                 $F^* \leftarrow F^* \cup \{T_k\}$ 
23.                 $k \leftarrow k + 1$ ;
24.            Fim se
25.        Fim senão
26.    Cleaning ( $T_k, F^*$ );

```

27.  $d \leftarrow d - 1;$
28. Fim enquanto
29. Fim se
30. Retorne  $F^*$ ;

**Algoritmo 1 - Procedimento exato para calcular o bi-AGCDM.**

### 3.5 MODELOS MATEMÁTICOS UTILIZADOS NO PROCEDIMENTO EXATO PARA O BI-AGCDM

No procedimento exato aplicado ao bi-AGCDM foram aplicados 2 modelos matemáticos: um para o caso em que o diâmetro a ser tratado é par e outro para o caso em que o diâmetro tratado é ímpar. Esses modelos utilizam a propriedade definida por Handler (1978) e um grafo  $G=(V,E)$ , onde  $|V|=n$  e  $|E|=m$ . Essa propriedade determina que se o diâmetro de uma árvore geradora  $T$  de  $G$  é par, existe um vértice central  $i \in V$  tal que nenhum outro vértice de  $T$  tenha uma altura maior do que  $L = D/2$  arestas de distância  $i$ . A variável  $L$  é a altura do vértice na árvore. A altura de um vértice  $x$  é a sua distância em número de arestas em relação ao nó raiz da árvore. Quando o diâmetro de uma árvore geradora  $T$  de  $G$  é ímpar, existe uma aresta central  $(i, j) \in T$  tal que nenhum vértice de  $T$  tenha altura maior que  $L = (D-1)/2$ .

Em ambas formulações, um nó artificial  $r$  é introduzido no grafo  $G$  com custo  $c_{ri} = 0, \forall i \in V$ . Um novo grafo  $G'$  é obtido de  $G$  através da introdução de um nó artificial  $r$  e da orientação das arestas de  $G$  da seguinte forma: para cada aresta  $(i, j) \in E$ , com  $i < j$ , dois arcos são criados em  $G'$ ,  $(i, j)$  e  $(j, i)$ , com custos  $c_{ij} = c_{ji}$ . Assim,  $G' = (V', A')$ , onde  $|V'| = n'$  e  $|A'| = m'$ , o conjunto de vértices é definido por  $V' = V \cup \{r\}$  e o conjunto de arestas é dado por  $A' = A \cup \{(r, I), \dots, (r, V)\}$ . A orientação dos arcos é realizada para controlar a construção da AGCM a partir do nó artificial, mas são desconsideradas após a obtenção da solução final.

O modelo para o caso par é dado nas equações de (2) a (7) e utiliza variáveis binárias  $x_{ij}$  que determinam se um arco  $(i, j)$  é selecionado para participar da AGCM ( $x_{ij} = 1$ ), ou não ( $x_{ij} = 0$ ). Além disto, variáveis  $u_i$  são associadas aos vértices  $i \in V'$ . Estas variáveis determinam uma ordem de visita na solução para cada vértice (ordem topológica), tendo  $u_r = 0$ . Assim, o vértice central  $l$  da AGCM tem  $u_l = 1$ . A função objetivo é dada em (2). A equação (3) assegura que o vértice artificial  $r$  conecta-se a exatamente um vértice em  $V$ , o vértice central. A restrição (4) define que um único arco deve ser incidente a cada vértice de  $V$ . As restrições (5) correspondem a uma adaptação das clássicas restrições MTZ de Miller *et al.* (1960), com o procedimento de *lifting* proposto por Desrochers e Laporte (1991). Estas restrições definem uma ordem topológica para cada nó na árvore, assim como eliminam subciclos. As variáveis são definidas de (6) a (7). Este modelo possui  $O(n'+m')$  variáveis e  $O(n+m')$  restrições.

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} \quad (2)$$

$$\sum_{j \in V} x_{rj} = 1 \quad (3)$$

$$\sum_{(i,j) \in A'} x_{ij} = 1 \quad \forall j \in V \quad (4)$$

$$u_i - u_j + (L + 1)x_{ij} + (L - 1)x_{ji} \leq L \quad \forall (i, j) \in A \quad (5)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A' \quad (6)$$

$$0 \leq u_i \leq L + 1 \quad \forall i \in V' \quad (7)$$

O modelo para o caso ímpar é apresentado nas equações de (8) a (16) e utiliza variáveis  $x_{ij}$  e  $u_{ij}$  como no caso par. Além disto, possui variáveis binárias  $z_{ij}$ ,  $\forall (i, j) \in E$  com  $i < j$  que são responsáveis pela seleção da aresta central. Esta aresta é selecionada quando  $z_{ij} = 1$ , em caso contrário  $z_{ij} = 0$ . Nesta formulação, o vértice artificial  $r$  se conecta a exatamente dois vértices  $p, q \in V$ , o que determina implicitamente a seleção de uma aresta central  $(p, q) \in E$ . Após a obtenção da solução final, além das orientações dos arcos serem

desconsideradas, as conexões  $(r, p)$  e  $(r, q)$  também são desconsideradas e a aresta central é  $(p, q)$ .

A função objetivo é fornecida em (8) e inclui o termo  $\sum_{(i,j) \in E} c_{ij} z_{ij}$  para contabilização do custo da aresta central na solução. A equação (9) estabelece que o vértice artificial  $r$  conecta-se a exatamente dois vértices de  $V$ . As restrições (10) determinam que um único arco é incidente a cada vértice de  $V$ . A equação (11) assegura que somente uma aresta central existe na solução final. As desigualdades não lineares em (12) relacionam as variáveis de decisão  $z_{ij}$  e  $x_{ij}$  e asseguram que o vértice artificial se conecta a exatamente dois nós de  $V$ . Estas restrições podem ser facilmente linearizadas como proposto em Santos *et al.* (2004). As restrições MTZ com *lifting*, definidas em (13), são similares ao caso par. O domínio das variáveis é determinado de (14) a (16). Este modelo possui  $O(n'+m')$  variáveis e  $O(n+m')$  restrições.

$$\min \sum_{(i,j) \in A} c_{ij} x_{ij} + \sum_{[i,j] \in E} c_{ij} z_{ij} \quad (8)$$

$$\sum_{j \in V} x_{rj} = 2 \quad (9)$$

$$\sum_{(i,j) \in A'} x_{ij} = 1 \quad \forall j \in V \quad (10)$$

$$\sum_{[i,j] \in E} z_{ij} = 1 \quad (11)$$

$$z_{ij} \geq x_{ri} \cdot x_{rj} \quad \forall [i, j] \in E \quad (12)$$

$$u_i - u_j + (L + 1)x_{ij} + (L - 1)x_{ji} \leq L \quad \forall (i, j) \in A \quad (13)$$

$$0 \leq u_i \leq L + 1 \quad \forall i \in V' \quad (14)$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \in A' \quad (15)$$

$$z_{ij} \in \{0, 1\} \quad \forall [i, j] \in E \quad (16)$$

### 3.6 MODELO MULTIFLUXO PARA O BI-AGCDM

Santos, Lima e Aloise (2012) apresenta uma formulação matemática, (17) - (27), para o problema bi-AGCDM, o qual é inspirada no trabalho de

Golveia e Magnanti (2003) para a AGMRD e faz uso de um grafo não direcionado  $G=(V,E)$ , com  $|V|=n$  e  $|E|=m$ . Sendo  $c_{ij}$  a variável de decisão na escolha da aresta  $(i,j)$  que, as variáveis de fluxo direcionadas especificam se um caminho de  $p \in V$  até  $q \in V$ , com  $i \neq q$  e  $j \neq p$ , passam pela aresta  $(i, j)$ . As variáveis  $d_{pq}$  especificam o número de arestas no caminho de  $p$  até  $q$ .

$$z_1(x) = \min \sum_{(i,j) \in E} c_{ij} \cdot x_{ij} \quad (17)$$

$$z_2 = \min D \quad st \quad (18)$$

$$\sum_{(i,j) \in E} x_{ij} = |V| - 1 \quad (19)$$

$$\sum_{j \in V} y_{ij}^{pq} - \sum_{j \in V} y_{ji}^{pq} = \begin{cases} 1, & \text{if } i = p \\ 0, & \text{if } i \neq q \text{ e } i \neq p \\ -1, & \text{if } i = q \end{cases} \quad \forall i \in V, \forall p, q \in V \quad (20)$$

$$y_{ij}^{pq} + y_{ji}^{pq} \leq x_{ij} \quad \forall (i,j) \in E; \forall p, q \in V \quad (21)$$

$$\sum_{(i,j) \in E} (y_{ij}^{pq} + y_{ji}^{pq}) \leq d_{pq} \quad \forall p, q \in V \quad (22)$$

$$D \geq d_{pq} \quad \forall p, q \in V \quad (23)$$

$$y_{ij}^{pq} \in \{0,1\} \quad \forall (i,j) \in E; \forall p, q \in V, i \neq q, j \neq p \quad (24)$$

$$x_{ij} \in \{0,1\} \quad \forall (i,j) \in E \quad (25)$$

$$d_{pq} \geq 1 \quad \forall p, q \in V \quad (26)$$

$$D > 1 \quad (27)$$

Os dois objetivos são dados nas equações (17) e (18) e eles visam respectivamente minimizar o custo total e o diâmetro. A restrição (19) garante

que a árvore geradora tem no máximo  $n - 1$  arestas. As restrições (20) são as clássicas restrições das conservações de multifluxo. As inequações (21) garantem que nenhum fluxo passa através da aresta  $(i, j)$  sempre que a aresta  $(i, j)$  não pertence à solução, ou seja,  $x_{ij} = 0$ . A restrição (22) calcula o número de arestas no caminho de  $p$  até  $q$ . Restrições (23) juntamente com o objetivo (18) minimizam o diâmetro. As variáveis são definidas de (24) à (27).

## 4 METAHEURÍSTICAS MULTIOBJETIVO PARA O BI-AGCDM

Metaheurísticas é um conjunto de conceitos que definem métodos heurísticos aplicáveis a um extenso conjunto de problemas. Elas podem ser classificadas como mono-objetivas ou multiobjetivas. A diferença entre esses dois tipos reside apenas na quantidade de funções consideradas por cada um. No caso da otimização mono-objetiva, as metaheurísticas realizam o processo de busca seguindo apenas uma função de satisfação. Em técnicas de otimização multiobjetiva, como o nome sugere, a busca deve considerar mais de uma função ao mesmo tempo (Freitas *et al.*, 2010).

Neste trabalho, duas metaheurísticas com abordagem multiobjetivo são propostas para o problema bi-AGCDM. A primeira é a NSGA-II, sendo uma reprodução do trabalho de Lima, Santos e Aloise (2012) com uma adaptação, e a segunda é PSO.

### 4.1 NSGA – II

O *Non-dominated Sorting Genetic Algorithm – II* (NSGA-II) é uma metaheurística baseada em algoritmos genéticos e destinada a resolver problemas biobjetivos. Sua estratégia possui complexidade assintótica no pior caso de  $O(n^2.m)$ , onde  $n$  é o número de nós e  $m$  o número de arestas. A ideia principal desta metaheurística consisti em evoluir suas soluções através do cruzamento de indivíduos elite. Em cada iteração, os indivíduos de uma população são classificados em conjunto elite e não elite. Para tanto, os indivíduos são organizados em fronteiras através do uso de dois

procedimentos: o primeiro ordena as soluções em fronteiras por ordem de dominância (chamado de *ranking*) e o segundo estima a densidade de soluções ao redor de cada indivíduo (chamado de *crowding distance*).

As fronteiras são camadas no espaço de soluções identificadas por uma dominância. Para identificação de qual fronteira uma solução pertence, verifica-se a quantidade de soluções que a dominam. Seja  $x_p$  o contador de dominância para um indivíduo  $p$ ,  $x_p$  recebe 0(zero) caso o elemento  $p$  não seja dominado por nenhum indivíduo da população,  $x_p$  recebe 1 caso o elemento  $p$  seja dominado por uma solução da população, e assim por diante. Por fim, os elementos que pertencem a primeira fronteira são aqueles que tem  $x_p = 0$ , os elementos que integram a segunda fronteira são aqueles quem tem  $x_p = 1$ , e assim consecutivamente (Lima, 2012).

No pior caso são criadas  $N$  fronteiras, onde cada indivíduo da população pertence a uma fronteira. Em um problema de minimização, a fronteira mais próxima do eixo do plano cartesiano possui as soluções que não são dominadas por nenhuma outra da população.

O procedimento geral do NSGA-II é ilustrado na Figura 8. Seja  $R$  uma população de tamanho  $N$  e  $k$  a iteração corrente da execução do algoritmo,  $R_k$  é dada por  $R_k = P_k \cup Q_k$ , onde  $P_k$  é igual a  $N/2$  e  $Q_k$  é igual a  $N/2$ .  $P_k$  é o conjunto da população elite e  $Q_k$  o conjunto da população não elite. Na próxima iteração do algoritmo,  $k + 1$ , os indivíduos do conjunto  $P$  da interação  $k$  são mantidos e os indivíduos do conjunto  $Q$  são descartados. Para que a população  $k + 1$  tenha uma população de tamanho  $N$  novos indivíduos são gerados a partir do cruzamento de dois indivíduos do conjunto  $P_k$ . Após a geração dos novos indivíduos, a população da iteração  $k + 1$  é classificada em indivíduos elite e não elite, e todo o processo é repetido até que seja satisfeita uma condição de parada, (Deb *et al.*, 2012).

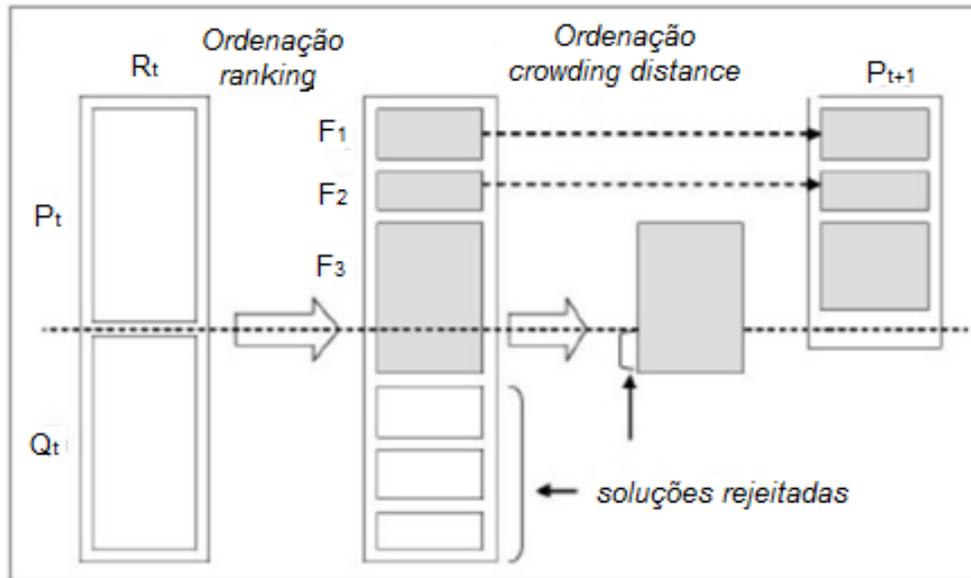


Figura 8 - Procedimento geral do NSGA-II. Fonte: Deb et al. (2002).

O procedimento *ranking* é aplicado na população  $R_t$  para classificar seus indivíduos nas fronteiras  $\{F_1, F_2, F_3, \dots, F_j\}$ . O objetivo do *crowding distance* é definir a distância entre os pontos de vizinhança para cada indivíduo de uma determinada fronteira. Assim sendo, quanto maior for o valor do *crowding distance* para um indivíduo, maior é a necessidade de melhorar o resultado desse indivíduo para aperfeiçoar a fronteira a que ele pertence (Deb et al., 2002).

O cálculo do *crowding distance* é dado na equação (28), onde  $f_k^{suc(j)}$  e o  $f_k^{pred(j)}$  correspondem respectivamente aos valores do sucessor e predecessor do indivíduo  $j$  na iteração  $k$ , e  $f_k^{max}$  e  $f_k^{min}$  correspondem aos valores máximo e mínimo da função objetivo na iteração  $m$  (Deb et al., 2002).

$$d_j = \sum_{m=1}^M \frac{f_k^{suc(j)} - f_k^{pred(j)}}{f_k^{max} - f_k^{min}} \quad (28)$$

O *crowding distance* contribui para fazer a população convergir em direção a  $F_1$ , uniformizando a fronteira de Pareto. Após o cálculo do *ranking* e

*crowding distance*, os indivíduos de uma fronteira são ordenados utilizando o operador *crowded* definido da seguinte forma: dados dois indivíduos  $i$  e  $j$ ,  $i$  domina  $j$  se  $(c_i < c_j \text{ e } d_i < d_j)$  ou  $(c_i = c_j \text{ e } d_i < d_j)$  ou  $(c_i < c_j \text{ e } d_i = d_j)$ , onde a variável  $c$  representa o custo e a variável  $d$  representa o diâmetro, ambos de um indivíduo.

O pseudocódigo do NSGA-II é apresentado no Algoritmo 2 e sua implementação é bastante simples. Na linha 1 é iniciado a população de acordo com alguma estratégia. Da linha 2 a 8 a população do algoritmo será evoluída para encontrar soluções viáveis do problema a ser resolvido. A linha 3 classifica a população de uma determinada iteração através do procedimento de *ranking*. Em seguida, na linha 4, é aplicado o *crowding distance* para que seja verificado a densidade de cada solução. Na linha 5, a população é classificada em elite e não elite. Posteriormente, na linha 6, os indivíduos da população não elite são descartados e na linha 7 novos indivíduos são gerados para completar a população. Por fim, a linha 9 apresenta as soluções do problema que está sendo tratado, as quais são os elementos da fronteira  $F_1$  da população da última iteração.

Leia:  $G=(V,E)$

1. Gerar população inicial
2. Para  $k = 1$  até número de iterações
3.     Aplicar o procedimento de *ranking* em todos os indivíduos
4.     Aplicar o *crowding distance* em todos os indivíduos
5.     Utilizando o *ranking* e o operador de *crowded*, classificar a população em elite e não elite
6.     Descartar população não elite
7.     Gerar novos indivíduos até a população conter novamente  $N$  indivíduos
8. Fim para
9. Apresentar os elementos da fronteira  $F_1$  da população da última iteração

Algoritmo 2 - Pseudocódigo do NSGA-II.

## 4.2 NSGA – II PARA O BI-AGCDM

No NSGA-II desenvolvido para o bi-AGCDM, um indivíduo (solução) representa uma árvore geradora com um custo e diâmetro associados. Existem diversas maneiras de representar (codificar) uma árvore em metaheurísticas. Alguns autores avaliam a eficiência de diferentes representações, como os trabalhos de Raild e Julstron (2003) e Carrano *et al.* (2007). Neste trabalho, uma solução é representada por um vetor, de forma que cada posição do vetor e o valor contido na mesma representa uma aresta da solução, sendo o nó predecessor da aresta uma posição do vetor e o nó sucessor o valor contido nesta posição. A Figura - 9 ilustra um exemplo da codificação utilizada. Na Figura - 9(a) temos um exemplo de árvore geradora e na Figura - 9(b) sua respectiva representação no vetor. A posição do vetor que representa a raiz da árvore contém em seu interior o valor de -1. As primeiras  $n$  primeiras posições representam as arestas da árvore, a posição  $n + 1$  representa o custo da árvore e a posição  $n + 2$  representa o diâmetro da mesma.

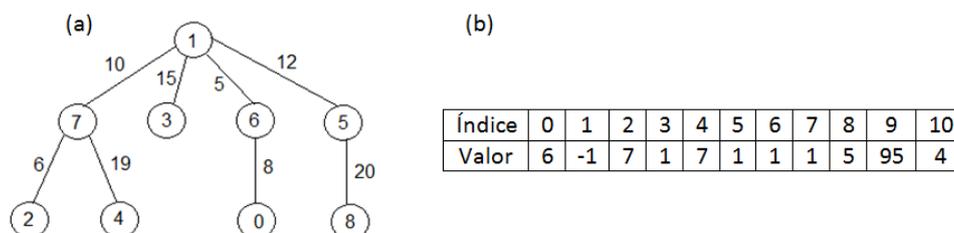


Figura 9 - Representação de uma árvore.

A primeira etapa da metaheurística consiste em determinar o espaço de busca da solução do problema, de acordo com a seção 3.1. Uma vez definido os limites do espaço de busca, cada partícula a ser gerada na população inicial é ligada a um diâmetro entre o menor e o maior, incluindo ambos. Desta forma, um grupo de indivíduos estará relacionado a um diâmetro, outro grupo de indivíduos com outro diâmetro e assim por diante. No final, tem-se a garantia de que todos os elementos da fronteira de Pareto foram tratados. No NSGA-II aplicado ao bi-AGCDM a população de indivíduos tem tamanho igual a  $2n$ .

A quantidade de indivíduos de uma população a serem ligados com um diâmetro é determinada pela variável  $e$  definida na equação (29). Nesta equação,  $2n$  representa o tamanho da população,  $s$  representa o diâmetro superior e  $i$  o diâmetro inferior do espaço de busca. Os indivíduos com diâmetro 2, 3 e o diâmetro da AGCM são excluídos da população do NSGA-II porque eles podem ser calculados com algoritmos polinomiais definidos na seção 3.2.

$$e = 2n / (s - i) \quad (29)$$

Cada indivíduo da população é construído através do algoritmo de Prim aleatorizado. Na versão original do Prim, em cada iteração o nó selecionado para fazer parte da solução é aquele que se conecta com o menor custo a solução parcial. Na versão aleatorizada, um sorteio é feito para selecionar o nó que entrará na solução. Cada solução a ser construída obedece um diâmetro específico determinado a priori. Inicialmente, todo nó sorteado será unido a um nó folha da solução parcial até que a mesma tenha seu diâmetro igual ao valor pré-estabelecido. A partir desse ponto, para completar a árvore geradora que está sendo montada, todos os nós que ainda não fazem parte dela serão unidos a solução parcial através da aresta de menor custo possível, a fim de diminuir o custo total da árvore geradora que está sendo montada.

Em alguns casos não é possível construir uma árvore geradora com um diâmetro estabelecido ao espaço de busca, como já foi explicado na seção 3.1. Cada árvore é iniciada a partir de um nó grafo. Se a partir de um nó não foi possível montar uma árvore geradora com um diâmetro  $d$  específico, outro nó da instância é escolhido para o algoritmo iniciar uma nova construção de árvore. Este processo é repetido sempre selecionando como nó inicial da árvore a ser montada um que ainda não tenha sido escolhido. Por fim, caso todos os nós do grafo tenham sido utilizados e nenhuma árvore com o diâmetro  $d$  tenha sido gerada, o indivíduo da população que representaria a árvore a ser montada é ignorado e outro indivíduo da população é considerado.

No decorrer do algoritmo, os indivíduos a serem construídos em uma iteração serão originados do cruzamento de outros dois. Embora a versão original do NSGA-II sugira que o cruzamento deve ser realizado entre dois indivíduos do conjunto elite, em testes realizados durante esta pesquisa observou-se que o NSGA-II para o problema bi-AGCDM apresenta melhores resultados cruzando um indivíduo da população elite e outro da população não elite. Os cruzamentos foram realizados de acordo com o sugerido por Carrano *et al.* (2007). O mesmo propõe realizar a união de duas árvores (indivíduos) em um grafo suporte. Em seguida, calcula-se o novo indivíduo a partir do grafo suporte. Este cruzamento é interessante porque sempre gera soluções viáveis. Por exemplo, dados duas soluções como mostrado na Figura - 10(a) e (b), a união que forma o grafo de suporte é dado na Figura - 10(c). Nesse cruzamento, um indivíduo é gerado através do algoritmo de Prim aleatorizado e outro através do prim original. O indivíduo que tiver o menor custo será inserida na população.

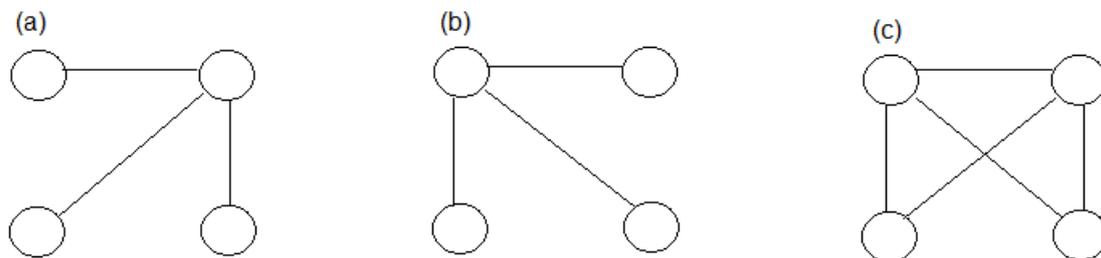


Figura 10 - Exemplo de união de grafos. Fonte: Lima (2012).

Para melhorar os resultados encontrados pelo o NSGA-II foram aplicadas três buscas locais em cada indivíduo gerado. A primeira busca local foi a 1-opt. Nela, uma aresta da solução corrente é substituída por outra fora da solução, se melhorar o custo total da solução e o diâmetro permanecer o mesmo, a alteração é aceita.

Na segunda busca local, o atual nó raiz da árvore é substituído por outro que possa assumir suas ligações, tendo essas um custo total menor do que as do atual nó raiz. Em seguida, o antigo nó raiz é ligado novamente a solução com a aresta de menor custo possível. Esta estratégia tem boa influência no bi-

AGCDM porque geralmente a solução montada tem diâmetro menor que a solução inicial, sendo que, árvores geradoras com diâmetro e custo pequenos são difíceis de serem encontradas. Se a árvore geradora obtida por esta busca local tiver um diâmetro diferente do que é alvo desta busca local, tenta-se substituir a solução gerada por outra da população não elite que tenha o mesmo diâmetro e custo maior.

Para entender melhor a busca local citada no parágrafo anterior, analise a figura 11. Na árvore inicial, figura - 11(b), tem-se como nó raiz da árvore o nó 3. Analisando o grafo que deu origem a esta árvore, figura - 11(a), busca-se um nó do grafo que contenha as mesmas ligações que o nó 3 da árvore contida na figura - 11(b). Essas ligações devem ter um custo menor do que todas as ligações do atual nó raiz da árvore, que no caso tem o custo 30. Na figura - 11(a), o nó 6 possui as mesmas ligações do que o nó 3 e tem o custo total 14. Sendo assim, as ligações do nó 3 na figura - 11(b) são desfeitas e nó 6 passa a ser o novo nós pai. Para que o nó 3 volte a solução ele será ligado com a árvore através de sua aresta de menor custo, cujo é a aresta de peso 10. O nó 7 que ficou a deriva quando o nó 6 passou a ser pai da árvore também deve ser unido a solução com sua aresta de menor custo, que no caso é a de custo 5. Por fim, a árvore resultante desta busca local está representada na figura - 11(c), com diâmetro igual a 5, sendo o diâmetro a árvore inicial, figura - 11(b), igual a 6.

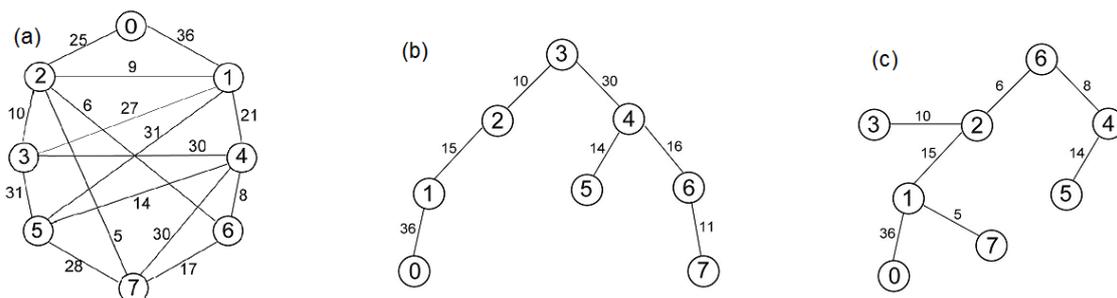


Figura 11 - Busca local de troca do nó pai

A terceira busca local é a mais simples. A partir de uma solução abordada são inseridas em um grafo suporte todas as arestas da instância que não fazem parte desta solução. De posse do grafo suporte, é gerado um novo

indivíduo através do algoritmo de Prim aleatorizado e outro através do de Prim original. A solução resultante de menor custo é inserida na população.

Na figura 12 é apresentado o fluxograma do NSGA-II aplicado ao problema bi-AGCDM com busca local.

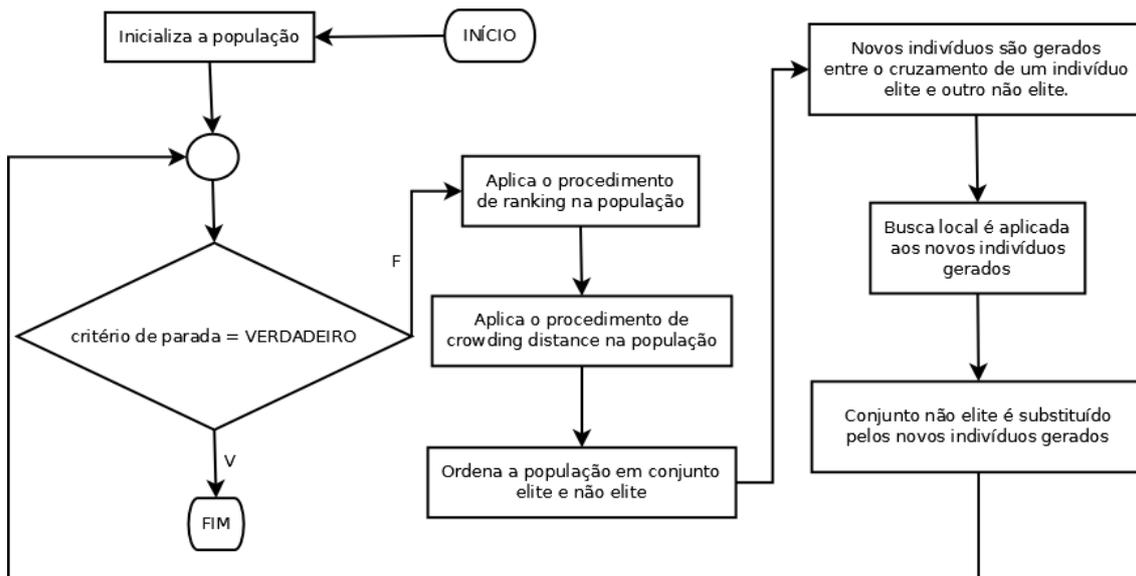


Figura 12 - Fluxograma da metaheurística NSGA-II com as estratégias sugeridas.

### 4.3 METAHEURÍSTICA PSO

O PSO, em português Otimização por Nuvem de Partículas, é uma técnica de computação estocástica baseada em dinâmica de populações que foi desenvolvida em 1995 por James Kennedy, um psicólogo social, e por Russell Eberhart, um engenheiro elétrico. Esta técnica implementa a metáfora do aprendizado social, onde considera que um indivíduo pode aprender tanto a partir de sua própria experiência quanto a partir do conhecimento obtido por outros membros do grupo. Ela já foi amplamente utilizada na resolução de problemas contínuos não lineares e ultimamente tem sido bastante utilizada para resolver problemas discretos com eficiência (Kennedy e Eberhart, 1995).

O PSO é a simulação do voo de um bando de pássaros, onde a procura de cada pássaro por uma melhor posição é a busca de uma solução “ótima” para um problema. No PSO, um pássaro é identificado como partícula e um bando de pássaros é identificado como enxame de partículas. O conjunto das possíveis posições a ser alcançada por uma partícula corresponde ao espaço de busca da solução do problema.

O PSO é iniciado com um enxame de partículas aleatório. Uma partícula possui sua velocidade e ela determinará sua mudança de posição durante o voo. Cada partícula mantém um rastro de seu percurso no espaço de busca, sendo que o movimento de cada uma é associado a melhor posição que ela já alcançou e com a melhor posição obtida dentre todas as partículas do enxame. A melhor posição que cada partícula já alcançou é chamada de *pbest*. A melhor posição obtida entre todas as partículas do enxame é identificada como *gbest*. O princípio da metaheurística PSO consiste, em cada iteração, influenciar o movimento de cada partícula em direção posição *pbest* e *gbest* (Eberhart e Shi,1998). Cada partícula é tratada como um ponto em um espaço *d*-dimensional.

Ao longo das iterações do algoritmo, as partículas realizam seu movimento através de um balanceamento entre aprendizado individual e conhecimento social, de acordo com as equações (30) e (31). Na equação (30), *V* é a velocidade que irá determinar o tamanho do movimento de uma partícula, na iteração *k*. A variável *W* é o peso de inércia que será aplicado à velocidade *V* da partícula *i* na iteração *k*. A inércia de uma partícula descreve o estado de agitação da mesma e o tamanho de sua aceleração. A variável *W* começa com o valor 1, no início da execução do algoritmo, e decresce linearmente até o valor 0(zero), final da execução do algoritmo. O peso de inércia *W* é empregado para controlar o impacto da velocidade anterior na velocidade atual, influenciando as habilidades de exploração global e local das partículas. Um peso de inércia maior facilita a exploração global, enquanto um peso de inércia menor tende a facilitar exploração local. As variáveis *C*<sub>1</sub> e *C*<sub>2</sub> são constantes de aceleração, onde  $C_1 \geq 0$ ,  $C_1 \in R$  e corresponde a componente de aprendizagem individual e  $C_2 \geq 0$ ,  $C_2 \in R$  e está relacionada a componente de aprendizagem social. As variáveis *Z*<sub>1</sub>, *Z*<sub>2</sub>  $\in [0, 1]$ .

$$V_i = W * V_i + C_1 * Z_1 * (Pb_i - X_i) + C_2 * Z_2 * (Gb - X_i) \quad (30)$$

$$X_i = X_i + V_i \quad (31)$$

As equações (30) e (31) são aplicadas para problemas discretos e contínuos. Em problemas contínuos, as partículas movimentam-se no espaço de busca através de uma soma vetorial de sua localização e velocidade, sendo a velocidade um número escalar. Em problemas discretos, a velocidade é uma lista de transposições (*swaps*) dos valores contidos nas posições do vetor que representam a localização de uma partícula (Oliveira, Silva e Aloise, 2004).

O pseudocódigo da metaheurística PSO para o problema bi-AGCDM é apresentado no Algoritmo 3. Nele,  $X_i^k$  é uma posição atual da partícula  $i$  na iteração  $k$ ,  $V_i^k$  é a velocidade da partícula  $i$  na iteração  $k$ ,  $pbest_i$  é a melhor posição da partícula  $i$ ,  $gbest$  é um vetor de posições. Devido o problema bi-AGCDM geralmente possuir mais de uma solução para uma instância tratada,  $gbest$  precisa ser um vetor onde cada posição do mesmo é relacionada a uma elemento do conjunto de soluções viáveis. Na linha 2 é definido o espaço de busca da solução, conforme a seção 3.1. A linha 3 inicializa o vetor  $gbest$  como vazio. Da linha 4 até a 10 o enxame de partículas é inicializado. Na linha 4 é definido em qual posição a partícula  $i$  é inicializada. Em seguida, na linha 5 é criada a velocidade inicial da partícula  $i$ . Na linha 6, a posição atribuída para a partícula  $i$  é guardada como sua melhor posição até o momento. Posteriormente, na linha 7, verifica-se com qual elemento do conjunto de soluções viáveis a partícula  $i$  é relacionada. Logo após, na linha 8, faz-se a verificação se a posição criada para a partícula  $i$  tem uma melhor posição referente ao elemento do conjunto de soluções viáveis que ela se relaciona. Caso sim, na linha 9, atribui-se a posição da partícula  $i$  a este elemento.

A linha 12 controla as iterações do algoritmo. A linha 13 controla a abordagem de todas as partículas em uma iteração. Para cada partícula, a linha 14 atualiza sua velocidade e posição. Se a posição  $X_i^k$  da partícula  $i$  na iteração  $k$  for a melhor posição encontrada até o momento para  $i$ , a condição da linha 15 é satisfeita e essa posição é salva na linha 16. Em seguida, na linha 17, verifica-se com qual elemento do conjunto de soluções viáveis a partícula  $i$

tratada é relacionada. Na linha 18 faz-se a verificação se a posição  $X_i^k$  da partícula  $i$  na iteração  $k$  tem uma melhor posição referente ao elemento do conjunto de soluções viáveis que ela se relaciona. Caso sim, na linha 19, atribui-se a posição da partícula  $i$  a este elemento.

Por fim, quando todas as iterações do laço na linha 12 tiverem sido executadas o algoritmo retorna na linha 20 a solução do problema, o qual é o vetor  $gbest$ .

1. Leia:  $G=(V,E)$
2. Definir espaço de solução da instância
3.  $gbest[ ] \leftarrow \emptyset$ ;
4. Para  $i = 1$  até  $|enxame|$
5.     Inicializar  $X_i$  e  $V_i$  aleatoriamente
6.      $pbest_i \leftarrow X_i$
7.      $y \leftarrow$  posição do  $gbest$  que se relaciona com  $X_i$ ;
8.     Se  $(f(X_i) < f(gbest[y]))$  então
9.          $gbest[y] \leftarrow X_i$
10.     Fim se
11. Fim para
12. Para  $k = 1$  até número de iterações
13.     Para  $i = 1$  até  $|enxame|$
14.         Atualizar  $X_i^k$  e  $V_i^k$  de acordo com as equações de atualizações
15.         Se  $(f(X_i^k) < f(pbest))$  então
16.              $pbest \leftarrow X_i^k$
17.          $y \leftarrow$  posição do  $gbest$  que se relaciona com  $X_i^k$
18.         Se  $f(X_i^k) < f(gbest[y])$  então
19.              $gbest[y] \leftarrow X_i^k$
20.         Fim se
21.     Fim se
22. Fim para
23. Fim para
24. Retorne  $gbest[ ]$

Na figura 13 está o fluxograma que representa o funcionamento da metaheurística PSO para problemas em geral.

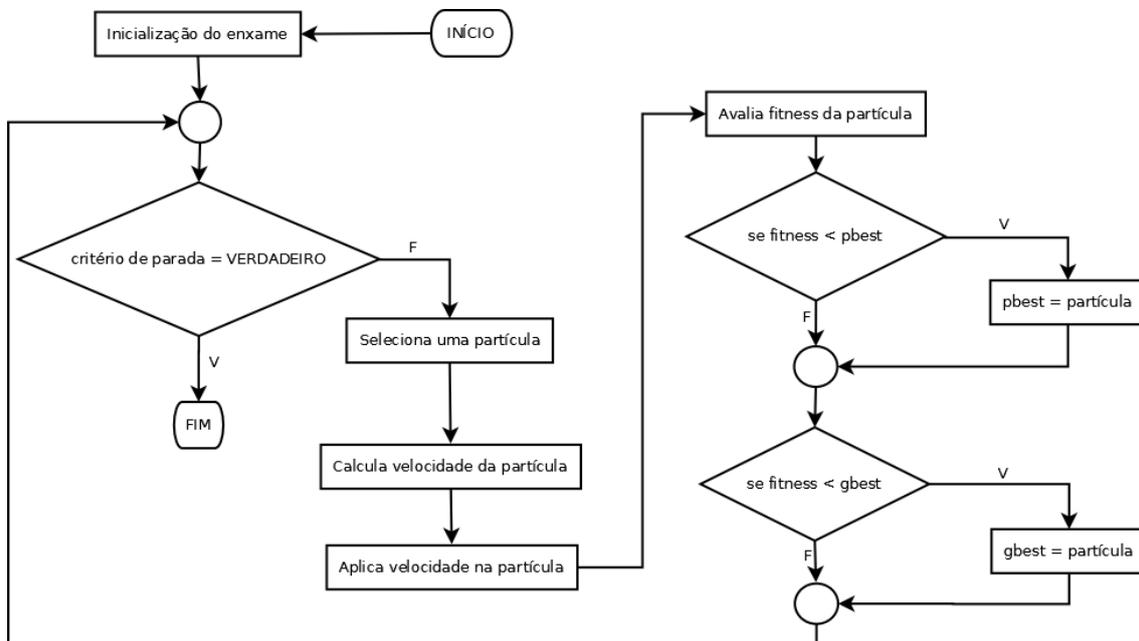


Figura 13 - Fluxograma da metaheurística PSO.

#### 4.4 UTILIZANDO O PSO EM PROBLEMAS DISCRETOS

Devido o problema bi-AGCDM ser um problema discreto, esta dissertação dedica uma subseção ao PSO aplicado a este tipo de problema. No PSO, o tamanho da velocidade de uma partícula varia de acordo com a execução do algoritmo e pode até ser nula, onde o tamanho da lista de transposição é igual a zero.

Duas ou mais velocidades são consideradas iguais se quando aplicadas a uma mesma partícula o mesmo resultado é obtido. A movimentação de uma partícula  $P$  ocorre com a aplicação de velocidade  $V$ , sendo  $P'$  a nova posição alcançada. Supondo uma partícula em determinadas posições  $A$  e  $B$ , obtém-se a velocidade  $V$  aplicada em  $A$  para se alcançar  $B$ , subtraindo uma posição da outra,  $V = A - B$ , onde o sinal de  $-$  indicará a lista de transposição (velocidade) resultado da equação  $B = V + A$ . Considere as duas posições  $A$  e  $B$  presentes

em (32) para seja verificado qual a velocidade que foi aplicada em  $A$  para se obter  $B$ .

$$A: ( 1 2 3 4 5 ) \qquad B: ( 2 3 1 5 4 ) \qquad (32)$$

Verifica-se inicialmente que foi aplicada uma transposição entre a primeira e terceira posição no vetor  $A$  para se obter  $B$ , a qual se obtém  $B' = B + V(1,3)$ , onde  $B' = ( 3 2 1 4 5 )$ . Em seguida, verifica-se que existe uma transposição entre as posições um e dois dos vetores  $B'$  e  $B$ , a qual se tem  $B'' = B' + V(1,2)$ , onde  $B'' = ( 2 3 1 5 4 )$ . Por fim, observa-se que ainda existe a transposição da posição quatro e cinco nos vetores  $B''$  e  $B$ , a qual se obtém  $B''' = B'' + V(4,5)$ , onde  $B''' = ( 2 3 1 5 4 )$ . Logo, verifica-se que a velocidade aplicada em  $A$  para se obter  $B$  é  $V = [(1,3), (1,2), (4,5)]$ .

A adição entre duas velocidades consiste na concatenação de listas de transposições. Além disso, pode existir a possibilidade de multiplicação de uma velocidade por uma constante. A multiplicação de uma constante por uma velocidade consiste em quatro casos:

1. Se  $C = 0$ , então  $C * V =$  nulo;
2. Se  $C \in ]0,1]$ , então  $C * V$  deve ser truncado;
3. Se  $C > 1$ , então aplica-se a velocidade  $V$  na partícula  $C$  vezes para a parte inteira, mais o caso 2, para que a parte fracionária seja aplicada;
4.  $C < 0$  não é definido;

#### 4.5 APLICANDO O PSO PARA O BI-AGCDM

No PSO desenvolvido para o bi-AGCDM, uma solução (partícula) é caracterizada por uma estrutura de árvore geradora com um custo e diâmetro associados. Neste trabalho, uma partícula é representada por um vetor, de forma que cada posição do vetor e o valor contido na mesma representa uma

aresta da partícula, sendo o nó predecessor da aresta uma posição do vetor e o nó sucessor o valor contido nesta posição. Uma ilustração dessa representação está disponível na figura 9. A posição do vetor que representa a raiz da árvore contém em seu interior valor -1. Sempre a penúltima posição do vetor guarda o custo da árvore e a última posição o diâmetro da mesma. No PSO aplicado ao bi-AGCDM, o enxame de partículas tem tamanho igual a  $2n$ .

A primeira etapa do algoritmo consiste em determinar o espaço de busca da solução do problema, o qual é definido de acordo com a seção 3.1. Uma vez definido os limites do espaço de busca, cada partícula a ser gerada na população inicial é ligada a um diâmetro entre o menor e o maior, incluindo ambos. Desta forma, um grupo de indivíduos estará relacionado com um diâmetro, outro grupo de indivíduos com outro diâmetro e assim por diante. No final, tem-se a garantia de que todos os elementos da fronteira de Pareto foram tratados.

A quantidade de elementos a serem ligados com um diâmetro do espaço de busca é determinada pela variável  $\epsilon$  definida na equação (29), explicada na seção 4.2. As partículas com diâmetro 2, 3 e o diâmetro da AGCM são excluídas do enxame do PSO porque eles podem ser calculadas com algoritmos polinomiais definidos na seção 3.2.

Cada partícula do enxame é construída através do algoritmo de Prim aleatorizado, como foi explicado na seção 4.2.

Em alguns casos não é possível construir uma árvore geradora com um diâmetro pertencente ao espaço de busca, de acordo com a seção 3.1. Cada árvore é iniciada a partir de um nó do grafo. Se a partir de um nó não foi possível montar uma árvore geradora com um diâmetro  $x$  específico, outro nó é escolhido para o algoritmo iniciar a construção desta árvore. Este processo é repetido sempre selecionando como nó inicial da árvore a ser montada um que ainda não tenha sido escolhido. Por fim, caso todos os nós do grafo tenham sido utilizados e nenhuma árvore com o diâmetro  $x$  especificado tenha sido gerada, a partícula do enxame que representaria a árvore a ser montada é ignorada e outra partícula do enxame é considerada.

No PSO original, a velocidade de uma partícula é uma lista de transposições. No caso do problema bi-AGCDM, em grafos esparsos, a troca dos valores contidos nas posições vetor pode gerar arestas que não existem na instância que está sendo tratada. Além disso, tanto em grafos esparsos como em completos, a lista de transposições pode inserir na solução as arestas de maiores custos do grafo, as quais certamente não fazem parte de nenhuma solução da fronteira de Pareto. Por tudo isso, a velocidade de uma partícula para o problema bi-AGCDM é considerada como uma lista de arestas, a qual é formada a partir de um conjunto restrito.

Antes que seja montada a velocidade de uma partícula, todas as arestas da instância são ordenadas por ordem crescente de custo. No caso de grafos completos, somente metade das arestas, as de menores custos, é utilizada para selecionar quais formarão uma velocidade. Em um grafo completo de 25 nós, por exemplo, existem 300 arestas e somente 24 fazem parte de uma solução. Dificilmente aquelas de maiores custos desse grafo farão parte uma solução da fronteira de Pareto.

Em grafos esparsos, para determinar o conjunto restrito de arestas a compor uma velocidade é calculado quantas arestas a instância teria caso ela fosse um grafo completo. Se a quantidade de arestas do grafo esparsos for menor que a metade do valor calculado, todas as arestas do grafo poderão ser selecionadas para montar a velocidade. Se o número de arestas ultrapassa metade do valor calculado, somente aquelas que estão posicionadas abaixo da metade do valor calculado é que formam o conjunto restrito de arestas.

Uma característica do PSO é sempre salvar a velocidade que foi aplicada em uma partícula na iteração  $k$ , para que parte dessa velocidade seja aproveitada na iteração  $k + 1$ . No entanto, se a lista de arestas (velocidade) a ser aplicada em uma partícula for idêntica, ou quase idêntica, em todas as iterações da metaheurística, a mudança de posição das partículas será mínima. Para evitar isto, no PSO aplicado ao bi-AGCDM, em cada iteração do algoritmo uma nova velocidade é montada.

Para melhorar os resultados encontrados pelo o PSO original, foram aplicadas três buscas locais em cada partícula que teve sua posição

atualizada. A primeira busca local foi a 1-opt. Nela, uma aresta da solução corrente é substituída por outra fora da solução, se melhorar o custo total da solução e o diâmetro permanecer o mesmo, a alteração é aceita.

Na segunda busca local, o atual nó raiz da árvore é substituído por outro que possa assumir suas ligações, tendo essas um custo total menor do que as do atual nó raiz. Em seguida, o antigo nó raiz é ligado novamente a solução com a aresta de menor custo possível. Esta estratégia tem boa influência no bi-AGCDM porque geralmente a solução montada diminui de diâmetro e árvores geradoras com diâmetro e custo pequenos são difíceis de serem encontradas. Se a árvore geradora obtida por esta busca local conter diâmetro diferente ao da árvore alvo desta busca local, tenta-se substituir a solução gerada por outra do enxame de partículas com mesmo diâmetro e custo maior.

A terceira busca local é a mais simples. A partir de uma solução abordada é inserido em um grafo suporte todas as arestas da instância que não fazem parte desta solução. De posse do grafo suporte, é gerado um novo indivíduo através do algoritmo de Prim aleatorizado e outro através do Prim original. A solução resultante de menor custo e com mesmo diâmetro é inserida na população.

## 5 TESTES E RESULTADOS

Os algoritmos utilizados neste trabalho foram desenvolvidos na linguagem de programação C ANSI e por meio do compilador Dev-C++ 4.9.9.2. A execução dos mesmos foram realizadas em uma máquina com o sistema operacional Windows 7, 64 bits, com 4Gb de memória RAM e processador Intel core i5 com 2,53GHz.

### 5.1 RESULTADOS DO PROCEDIMENTO EXATO PARA O BI-AGCDM

Um número de 20 instâncias de Santos, Lucena e Ribero (2004) e 23 instâncias de Santos, Lima e Aloise (2012) são usadas em experimentos computacionais para o procedimento exato proposto. Cada instância e o seu resultado estão na Tabela 1 e 2, respectivamente.

Nas Tabelas 1 e 2 cada linha corresponde a uma instância. A coluna “Q” indica a quantidade de pontos encontrados na fronteira de Pareto. Para cada solução dominada foi inserido o símbolo “-”. Além disso, se não existe no grafo uma solução com um determinado diâmetro que esteja dentro do espaço de solução, a mesma é indicada com o símbolo “\*”. Os espaços em branco correspondem a elementos que não fazem parte da fronteira de Pareto. Finalmente, a coluna tempo(s) mostra os tempos totais em segundos para achar todos os pontos ótimos da fronteira de Pareto.

Os resultados apresentados na Tabela 1 indicam que, exceto a instância *s\_v40\_a100\_d4*, o espaço de soluções contém no máximo 13 soluções (13 diâmetros alvos). Em termos de tempo de processamento, encontrar todas as soluções da fronteira de Pareto ótima para estas instâncias de testes consumiu no máximo 50 segundos para as instâncias de grafos completos com 10 e 15 nós. O procedimento precisou de menos de 40 minutos para as instâncias completas de 20 nós e aproximadamente 13 horas para a instância com 40 nós. Observou-se também que as instâncias de diâmetros pequenos, por

exemplo,  $D = 4$  e  $D = 5$ , consomem a maior parte do tempo de processamento. Em geral, o problema torna-se mais fácil a medida que se aproxima do diâmetro da AGCM. Além disto, instâncias com arestas de custos idênticos são mais difíceis de serem resolvidas. Possivelmente, a resolução de tais problemas implica uma combinatória mais significativa na árvore de *Branch-and-Bound*, como é o caso da instância  $s\_v40\_a100\_d4$ .

A Tabela 2 apresenta resultados para instâncias de grafos esparsos para o segundo conjunto de testes. O tempo de execução de suas instâncias são menores do que as presentes na Tabela 1. O número de partições para ser inspecionado são de 9 até 14 para as instâncias *Halmitonien cycle* e *Hamiltonien path*. Devido ao fato de que os grafos são esparsos, não é surpreendente que algumas soluções com diâmetros específicos não existem. Para esse conjunto de teste, o maior número de diâmetros sem soluções varia de 1 a 7. Assim, não há espaço para melhorar o limite inferior.

Os gráficos das figuras 14, 15, 16 e 17 ilustram as fronteiras de Pareto ótimas para algumas instâncias. A instância  $c\_v20\_a190\_d4$ , Figura - 13, possui pontos dominados na fronteira de Pareto, com  $D = 11$  e  $D = 13$ . A instância  $c\_v20\_a190\_d5$ , Figura - 14, possui a solução da AGCM de diâmetro  $D = 13$  dominada pela solução com  $D = 12$ . A instância  $p\_v40\_0.10$ , Figura – 15, possui como limite inferior do seu espaço de busca o  $D = 5$ , mas não possui árvore geradora com  $D = 5$  e  $D = 6$ . Finalmente, observa-se ainda que na maioria dos casos, a função de custo das soluções com os menores diâmetros tem valores bastante elevados em relação aos outros diâmetros, gerando um espaçamento significativo na fronteira de Pareto ótima nessas regiões do espaço de busca.

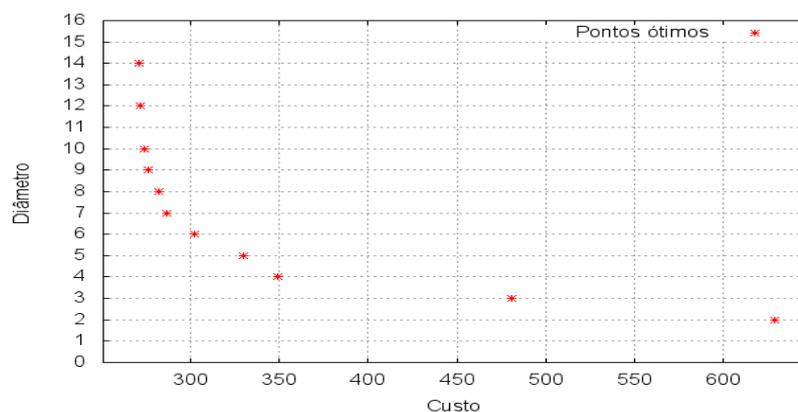


Figura 14 - Fronteira de Pareto ótima para  $c\_v20\_a190\_d4$ .

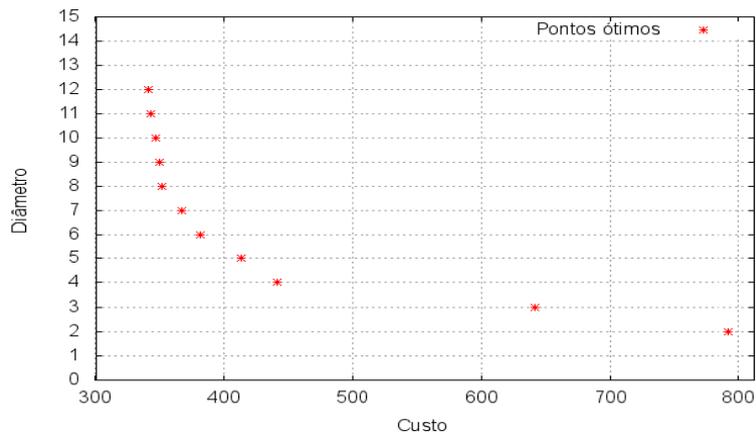


Figura 15 - Fronteira de Pareto ótima para c\_v20\_a190\_d5.

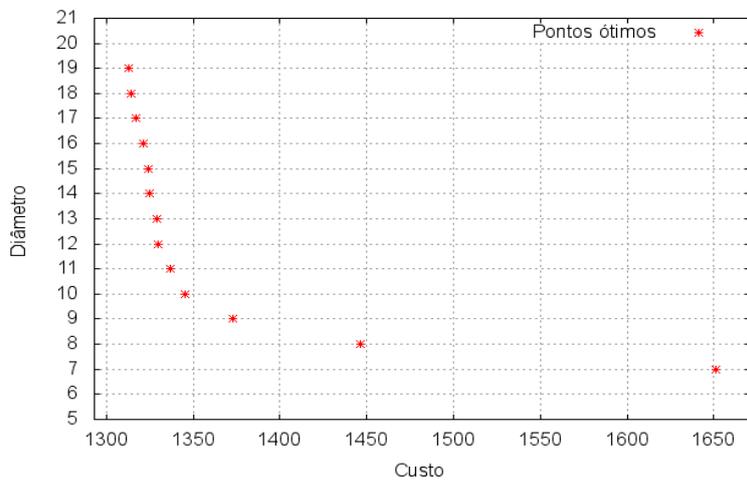


Figura 16 - Fronteira de Pareto ótima para p\_v40\_0.10.

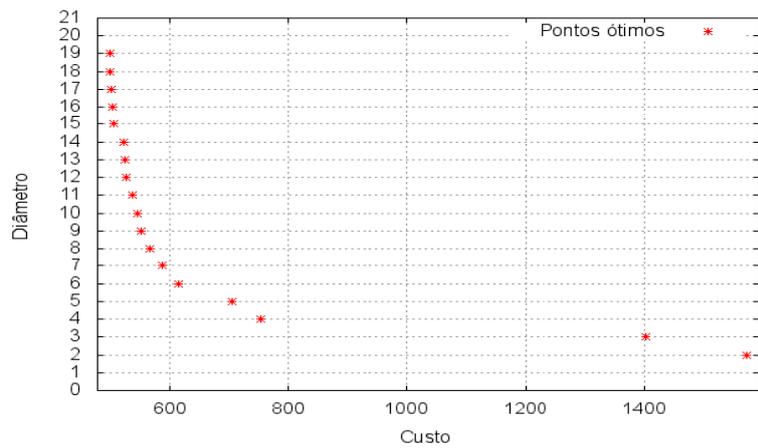


Figura 17 - Fronteira de Pareto ótima para s\_v40\_a100\_d4.

Instâncias	Q	Diâmetros																			tempo(s)
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	
C_V10_A45_D4	8	392	279	252	234	218	210	208	206												0,70
C_V10_A45_D5	5	382	330	247	230	218															0,31
C_V10_A45_D6	5	360	275	235	226	221															0,26
C_V10_A45_D7	8	447	292	250	226	211	203	199	198												0,72
C_V10_A45_D8	6	389	297	243	236	233	232	-													0,30
C_V10_A45_D9	7	483	347	284	270	263	256	254													0,55
C_V15_A105_D4	11	681	482	346	331	314	303	295	290	286	-	281	277								25,16
C_V15_A105_D8	7	491	390	291	274	256	247	240													23,81
C_V20_A190_D4	11	629	481	349	330	302	287	282	276	274	-	272	-	271							520,53
C_V20_A190_D5	11	792	641	441	414	382	367	352	350	347	343	341									199,01
C_V20_A190_D6	14	681	498	343	316	298	287	278	276	272	269	267	265	263	261						387,10
C_V20_A190_D7	8	716	546	414	376	345	333	321	316												248,32
C_V20_A190_D8	13	789	592	421	382	353	344	331	323	317	311	304	299	295							2241,32
C_V20_A190_D9	12	822	599	418	390	356	345	333	327	324	-	-	321	316	313						1014,45
S_V20_A50_D4	10	693	601	442	410	369	357	340	334	-	332										11,47
S_V20_A50_D5	9	807	694	417	381	347	330	322	318	315											14,15
S_V20_A50_D6	10	755	617	395	366	329	323	321	309	303	300										36,35
S_V20_A50_D7	9	707	598	404	386	370	362	358	357	356											22,86
S_V20_A50_D8	10	822	679	434	411	379	373	366	362	359	357										49,60
S_V40_A100_D4	19	1574	1403	755	706	615	589	567	552	546	538	528	525	523	508	505	503	501	500	-	46690,46

Tabela – 1: Resultados de fronteiras de Pareto para instâncias Santos, Lucena e Ribero (2004).

Intâncias	Q	Diâmetros																		tempo (s)		
		2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19		20	
c_v10_0.3	2			*	434	420															0.13	
c_v10_0.4	3			305	275	269																0.09
c_v15_0.2	4			*	*	497	453	*	451	443												0.54
c_v15_0.3	7			635	537	469	435	418	415													1.87
c_v15_0.4	6			465	407	391	378	376	375	-												1.37
c_v20_0.2	6			*	980	844	813	794	787	776												2.68
c_v20_0.3	5			1001	618	521	478	470	468	467												19.38
c_v20_0.4	7			614	536	493	480	474	468	467												13.92
c_v25_0.2	8			*	940	741	682	641	624	611	606	599										61.23
c_v25_0.3	5			731	648	556	535	-	529													18.08
c_v25_0.4	8			569	519	481	471	461	455	451	-	447										18.08
p_v25_0.09	2							*	1323	1259	1223	1215	1190	*	1187	1186	1146	1144				1.47
p_v25_0.10	7							*	*	*	*	1858	1847	-	1844	-	1841	-	-	1838		2.05
p_v30_0.08	5							*	*	*	*	1407	1276	1250	1237	1234	1217					3.34
p_v30_0.09	6						*	*	*	*	*	1315	1249	1227	1189	1180	1168	1160	-	1140		8.41
p_v30_0.10	8						*	*	*	*	*	1677	1653	1635	-	1617	1611	-	-	1608	1604	39.08
p_v35_0.08	7						*	*	*	*	*	1702	1568	1513								23.4
p_v35_0.09	3					*	*	*	1243	1179	1166	1135	1132	1112								447.77
p_v35_0.10	7						*	*	*	*	*	1889	1646	1640	1591	1566	1553					17.52
p_v40_0.08	6					*	*	*	1613	1520	1471	1468	-	1467								697.58
p_v40_0.09	6				*	*	*	*	1651	1446	1373	1345	1337	1330	1329	1325	1324	1321	1317	1314	1313	484.66
p_v40_0.10	13				*	*	*	*	1651	1446	1373	1345	1337	1330	1329	1325	1324	1321	1317	1314	1313	1787.55

Tabela – 2: Resultados de fronteiras de Pareto para instâncias *Hamiltonien cycle* e *Hamiltonien path* (Santos, Lima e Aloise, 2013).

## 5.2 RESULTADOS DAS METAHEURÍSTICAS

Três grupos de instâncias foram utilizados nos experimentos com as metaheurísticas, os quais são provenientes dos trabalhos de Gouveia e Magnanti (2003), Santos, Lucena e Ribero (2004) e Santos, Lima e Aloise (2013).

Para medir a qualidade e eficiência dos algoritmos desenvolvidos foram utilizadas métricas de avaliação. Devido os valores de diâmetro e custo serem de escalas diferentes, todos os valores foram normalizados entre 0 (zero) e 1 para facilitar a análise dos resultados.

Uma das métricas é denominada de Q, onde indica a quantidade de elementos da fronteira de Pareto ótima. A segunda métrica, chamada em inglês de *Spacing* (S), verifica o espalhamento das soluções do conjunto viável. Ela foi proposta por Veldhuizen e Lamont (2000) e é obtida através do cálculo que tem como base a distância entre os pontos da fronteira de Pareto. Sua fórmula é detalhada em (33):

$$S \triangleq \sqrt{\frac{1}{n-1} \sum_{i=1}^n (\bar{d} - d_i)^2} \quad (33)$$

onde:  $d_i = \min_j (|f_1^i(\vec{x}) - f_1^j(\vec{x})| + |f_2^i(\vec{x}) - f_2^j(\vec{x})|)$ ,  $i, j=1, \dots, n$ , e  $\bar{d}$  é a média de todos os  $d_i$ , e  $n$  é o número de valores na fronteira de Pareto. O valor 0 (zero) nesta métrica indica que todos as soluções na fronteira de Pareto estão equidistantes.

A métrica H, Zitzler *et al.* (2003), representa o Hipervolume da área coberta pelas soluções presentes na fronteira de Pareto em relação a um ponto  $w$  específico. O ponto  $w$  é a pior solução para a instância que está sendo tratada. Neste trabalho definimos  $w$  como sendo uma solução formada pelos piores valores em termos de custo e diâmetro. Quanto maior o hipervolume, melhor a fronteira de Pareto encontrada. Um exemplo do hipervolume para uma solução é mostrado na Figura - 18:

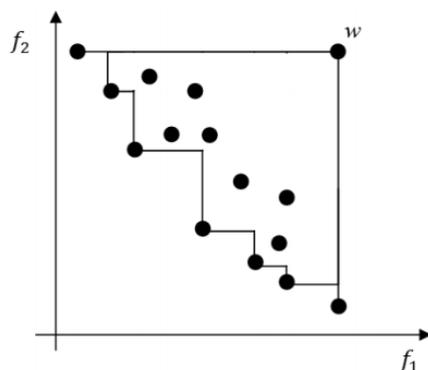


Figura 18 - Hipervolume. Fonte: Lima (2012).

### 5.2.1 CALIBRAÇÃO DOS ALGORITMOS

Os testes foram realizados com os grupos referenciados no início da seção 5.2. Três execuções diferentes foram realizadas variando a quantidade de iterações em 50, 90 e 130. A semente fornecida nas três execuções tem o valor a 7. Esta semente é a mesma que foi utilizada Lima (2012). Os resultados da calibração estão presentes na tabela 3 e 4. Na primeira coluna está descrito o nome das instâncias testadas e nas demais colunas os resultados das métricas e tempo para cada instância, de acordo com o número de iterações testadas.

Instância	50 Iterações				90 Iterações				130 Iterações			
	Q	S	H	Tempo	Q	S	H	Tempo	Q	S	H	Tempo
c_v20_a190_d8	13	0.121	0.771	0.83	13	0.121	0.771	1.34	13	0.121	0.772	1.97
c_v25_a300_d4	13	0.134	0.780	2.04	13	0.136	0.784	3.51	13	0.136	0.785	4.65
se_v40_a400_d5	16	0.098	0.788	11.32	16	0.098	0.791	19.41	16	0.115	0.807	27.92
se_v60_a600_d5	22	0.063	0.772	50.99	21	0.075	0.783	92.92	21	0.075	0.786	138.65
C_v10_0.3	2	0.0	0.0	0.0	2	0.0	0.0	0.01	2	0.0	0.0	0.0
P_v40_0.13	7	0.235	0.804	1.79	7	0.235	0.804	2.69	7	0.235	0.804	3.58

Tabela 3 - Calibrando o NSGA-II.

Instância	40 Iterações				90 Iterações				130 Iterações			
	Q	S	H	Tempo	Q	S	H	Tempo	Q	S	H	Tempo
c_v20_a190_d8	13	0.129	0.787	1.75	13	0.129	0.785	2.98	13	0.131	0.786	4.23
c_v25_a300_d4	13	0.141	0.790	3.38	13	0.123	0.785	5.53	13	0.130	0.783	7.88
se_v40_a400_d5	15	0.122	0.824	17.75	17	0.123	0.825	35.05	17	0.119	0.818	48.27
se_v60_a600_d5	25	0.204	0.960	97.09	23	0.212	0.960	175.57	25	0.204	0.960	255.51
C_v10_0.3	3	0.00	0.333	0.05	3	0.0	0.333	0.08	3	0.0	0.333	0.12
P_v40_0.13	10	0.426	0.643	15.14	10	0.400	0.714	28.19	10	0.400	0.714	40.16

Tabela 4 - Calibrando o PSO.

A medida que a quantidade de iterações aumenta, o hipervolume também cresce. Isso indica que a distância da fronteira de Pareto em relação ao ponto  $w$  está aumentando e as soluções estão convergindo para a fronteira de Pareto ótima. A métrica de espalhamento das soluções (*spacing*) é outra que não necessariamente diz que uma fronteira é melhor do que outra, já que ela indica apenas a uniformidade das soluções, independentemente da quantidade de elementos.

A execução da metaheurística NSGA-II com 130 iterações apresenta melhores resultados do que com outros grupos de iteração, sendo no PSO ela praticamente não existe em relação ao conjunto de 90 iterações. Na métrica Q praticamente os mesmos valores são encontrados em todas as iterações do PSO, exceto para *se\_v60\_a600\_d5*, e do NSGA-II. Na métrica *spacing*, algumas instâncias tem melhores resultados com 50 iterações, outras com 90 iterações e outras em 130 iterações. Neste trabalho, a coleta dos resultados será realizada com a quantidade de 90 iterações para o NSGA-II e PSO. Além do resultados das métricas, considerou-se que o tempo usando 90 iterações foi o melhor diante do resultados produzidos.

## 5.2.2 AVALIAÇÃO DO NSGA-II

O NSGA-II proposto neste trabalho é referenciado por NSGA-II (A) e o proposto por Lima, Santos e Aloise (2012) é referenciado por NSGA-II (B). Para gerar um novo indivíduo na população, o NSGA-II (B) utiliza dois indivíduos do conjunto elite, enquanto que o NSGA-II (A) utiliza um indivíduo do conjunto elite

e outra do conjunto não elite. No NSGA-II (B) foi utilizado somente a busca local 1-opt, sendo que o NSGA-II (A) utiliza todas as buscas locais presentes na seção 4.2. Uma comparação precisa entre as duas propostas analisando somente os resultados das 3 métricas utilizadas neste trabalho não é possível de ser feita. Em cada instância a quantidade de elementos na fronteira de Pareto é diferente para cada algoritmo e isto é decisivo no resultado das métricas *spacing* e hipervolume. Nos resultados colhidos percebe-se que enquanto uma proposta tem um melhor para o *spacing*, a mesma tem um resultado pior para seu hipervolume, e vice versa.

Alguns resultados colhidos com as metaheurísticas podem ser comparados aos resultados ótimos encontrados pelo procedimento exato proposto nesta dissertação. Na Tabela 6 são apresentados os valores das métricas de algumas instâncias para o procedimento exato, NSGA-II (A) e NSGA-II (B), considerando 14 instâncias do grupo Santos, Lucena e Ribeiro (2004). Nesta tabela, para a métrica Q, o procedimento exato e o NSGA-II (A) apresentam resultados idênticos, exceto para 1 instância, das 14 apresentadas na tabela, enquanto que o NSGA-II (B) não se iguala a nenhum resultado do procedimento exato. O mesmo padrão de resultado se repete para a métrica de hipervolume. Considerando a métrica *spacing*, o NSGA-II (A) apresenta resultados mais próximos ao procedimento exato para 11 instâncias e o NSGA-II (B) para 3 instâncias.

Na tabela 7 são abordadas 12 instâncias do grupo Santos, Lima e Aloise (2013). Para a métrica Q, o NSGA-II (A) apresenta os mesmos resultados do procedimento exato para 5 instâncias, enquanto que o NSGA-II (B) apresenta o mesmo resultado somente para 1 instância. Considerando a métrica de hipervolume, o NSGA-II (A) apresenta resultados mais próximos ao do procedimento exato para 9 instâncias. Considerando a métrica *spacing*, o NSGA-II (A) apresenta resultados mais próximo ao procedimento exato para 7 instâncias, o NSGA-II (B) para 4 instâncias e para 1 instância tem-se o mesmo resultado para ambas metaheurística.

Com os resultados apresentados nas tabelas 6 e 7, pressupõe-se que o NSGA-II (A) tem um melhor desempenho em relação ao NSGA-II (B) e por isso o mesmo será utilizado na comparação com o PSO.

As tabelas 5 e 8 apresentam a comparação entre o NSGA-II (A) e o NSGA-II (B) para outras instâncias.

Instâncias Santos, Lucena e Ribeiro (2004)								
Instância	NSGA-II (A)				NSGA-II (B)			
	Q	S	H	Tempo	Q	S	H	Tempo
c_v25_a300_d4	13	0.136	0.784	3.38	12	0.169	0.793	25
c_v25_a300_d5	13	0.143	0.806	3.20	12	0.192	0.810	25
c_v25_a300_d6	11	0.123	0.747	3.23	13	0.175	0.808	25
c_v25_a300_d8	15	0.116	0.817	3.38	13	0.174	0.835	25
c_v25_a300_d9	10	0.161	0.741	3.11	9	0.190	0.733	25
s_v40_a100_d5	15	0.164	0.819	23.26	12	0.095	0.794	30
s_v40_a100_d6	11	0.163	0.759	20.70	9	0.148	0.758	30
s_v60_a150_d5	15	0.147	0.811	133.19	13	0.143	0.733	40

Tabela 5 – Comparação de resultados entre NSGA-II (A) e NSGA-II (B) para Santos, Lucena e Ribeiro (2004).

Instâncias Santos, Lucena e Ribeiro (2004)												
Instância	PROCEDIMENTO EXATO				NSGA-II (A)				NSGA-II (B)			
	Q	S	H	Tempos	Q	S	H	Tempo	Q	S	H	Tempo
c_v10_a45_d4	8	0.211	0.730	0,70	8	0.211	0.730	0.17	7	0.122	0.623	10
c_v10_a45_d5	5	0.202	0.517	0,31	5	0.202	0.517	0.07	4	0.289	0.545	10
c_v10_a45_d6	5	0.206	0.619	0,26	5	0.262	0.615	0.16	4	0.283	0.548	10
c_v10_a45_d7	8	0.219	0.746	0,72	8	0.218	0.745	0.17	7	0.140	0.642	10
c_v10_a45_d8	6	0.257	0.697	0,30	6	0.257	0.697	0.16	5	0.168	0.562	10
c_v10_a45_d10	7	0.231	0.724	0,55	7	0.231	0.724	0.14	6	0.241	0.662	10
c_v15_a105_d4	11	0.167	0.809	25,16	11	0.157	0.803	0.56	10	0.176	0.782	15
c_v15_a105_d8	7	0.180	0.662	23,81	7	0.171	0.653	0.51	6	0.236	0.650	15
s_v20_a50_d4	9	0.143	0.725	11,47	9	0.137	0.718	1.14	8	0.181	0.747	20
s_v20_a50_d5	9	0.191	0.722	14,15	9	0.179	0.696	1.08	8	0.242	0.752	20
s_v20_a50_d6	10	0.170	0.751	36,35	9	0.153	0.743	1.86	7	0.155	0.651	20
s_v20_a50_d7	9	0.201	0.753	22,86	9	0.160	0.722	1.23	7	0.287	0.770	20
s_v20_a50_d8	10	0.183	0.768	49,60	10	0.179	0.759	1.72	7	0.200	0.662	20
s_v40_a100_d4	18	0.146	0.839	46690,4	17	0.131	0.808	31.56	17	0.156	0.850	30

Tabela 6 - Comparação de resultados entre o procedimento exato, NSGA-II (A) e o NSGA-II (B) para Santos, Lucena e Ribeiro (2004).

Instâncias Santos, Lima e Aloise (2013)												
Instância	PROCEDIMENTO EXATO				NSGA-II (A)				NSGA-II (B)			
	Q	S	H	Tempo	Q	S	H	Tempo	Q	S	H	Tempo
c_v10_0.3	2	0.0	0.0	0.02	2	0.0	0.0	0.01	2	0.0	0.0	5
c_v10_0.4	3	0.0	0.417	0.1	3	0.0	0.417	0.71	2	0.0	0.0	5
c_v15_0.3	7	0.175	0.673	1.87	5	0.205	0.646	1.97	3	0.290	0.638	10
c_v15_0.4	6	0.259	0.684	1.81	6	0.302	0.764	2.22	5	0.229	0.588	10
c_v20_0.2	6	0.264	0.669	1.96	5	0.192	0.515	1.10	4	0.135	0.405	15
c_v20_0.3	5	0.321	0.653	22.78	4	0.427	0.519	0.78	3	0.469	0.484	15
c_v20_0.4	7	0.205	0.702	16.94	7	0.219	0.689	4.48	6	0.174	0.611	15
c_v25_0.2	8	0.202	0.727	57.94	6	0.274	0.532	5.74	6	0.139	0.593	20
c_v25_0.3	5	0.163	0.664	20.28	6	0.151	0.492	4.65	4	0.314	0.696	20
c_v25_0.4	8	0.144	0.711	59.55	8	0.233	0.739	8.05	7	0.188	0.691	20
p_v40_0.09	6	0.191	0.721	418.89	3	0.0	0.069	0.17	4	0.283	0.582	25
p_v40_0.13	13	0.175	0.838	534.58	7	0.235	0.804	5.96	9	0.157	0.766	25

Tabela 7 - Comparação de resultados entre o procedimento exato, NSGA-II (A) e o NSGA-II (B) para Santos, Lima e Aloise (2013).

Instâncias Gouveia e Magnanti (2003)									
Instância	NSGA-II (A)				NSGA-II (B)				
	Q	S	H	Tempo	Q	S	H	Tempo	
se_v40_a400_d5	16	0.098	0.791	19.95	13	0.126	0.810	60	
se_v60_a600_d5	21	0.075	0.783	96.06	19	0.149	0.876	100	
sr_v40_a400_d5	9	0.159	0.732	16.69	8	0.156	0.649	60	
sr_v60_a600_d5	12	0.088	0.699	89.27	13	0.198	0.788	100	

Tabela 8 – Comparação de resultados entre o NSGA-II (A) e o NSGA-II (B) para Gouveia e Magnanti (2003).

### 5.2.3 COMPARANDO O NSGA-II E O PSO

As tabelas abaixo comparam o PSO com NSGA-II integrado as estratégias sugeridas neste trabalho. Considerando as 38 instâncias tratadas neste trabalho, em 12 instâncias o PSO encontrou mais elementos na fronteira de Pareto e somente em 1 caso o NSGA-II encontrou mais elementos.

Considerando a métrica de hipervolume, das 24 instâncias em que as duas metaheurísticas apresentam valores diferente, somente em 3 instâncias o NSGA-II apresenta um melhor resultado.

Embora o NSGA-II apresente um melhor resultado para algumas instâncias em relação a métrica de *spacing*, como por exemplo, instâncias

$c_{v15\_a105\_d4}$  (figura 19) e  $sr_{v40\_a400\_d5}$  (figura 20), observa-se no geral que o PSO é melhor. A instância  $c_{v25\_0.4}$  (figura 21) também apresenta um melhor resultado para o PSO, essa com um melhor valor para *spacing* e hipervolume.

Sendo assim, observando as tabelas 9, 10 e 11, juntamente com as figuras 19, 20 e 21, o PSO possui uma melhor performance em relação ao NSGA-II.

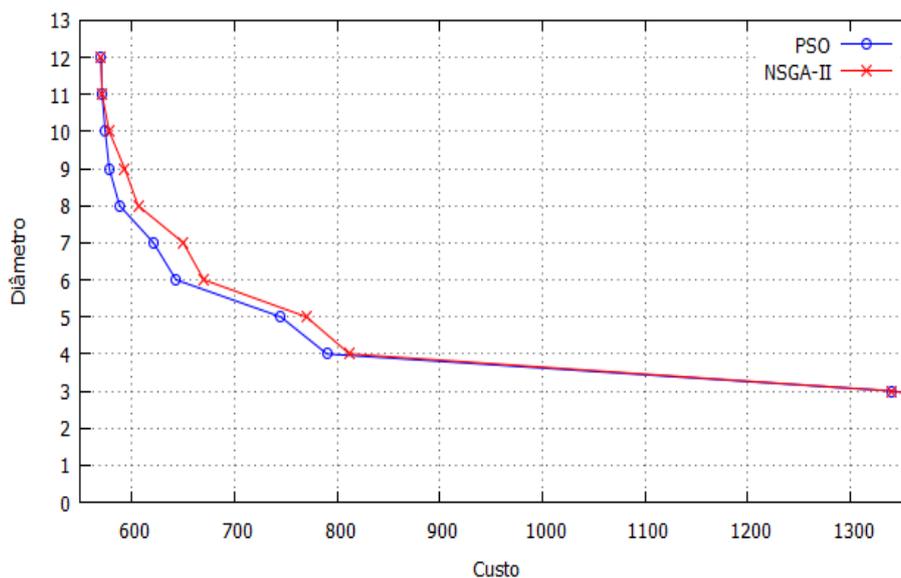


Figura 19 - Comparação entre o PSO e NSGA-II para a instância  $s_{v40\_a100\_d6}$

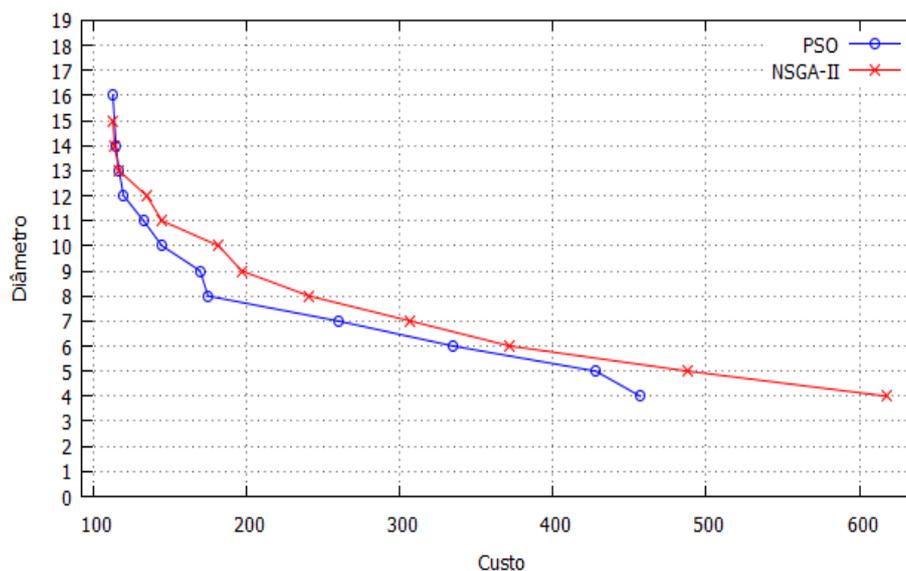


Figura 20 - Comparação entre o PSO e o NSGA-II para a instância  $sr_{v60\_a600\_d5}$

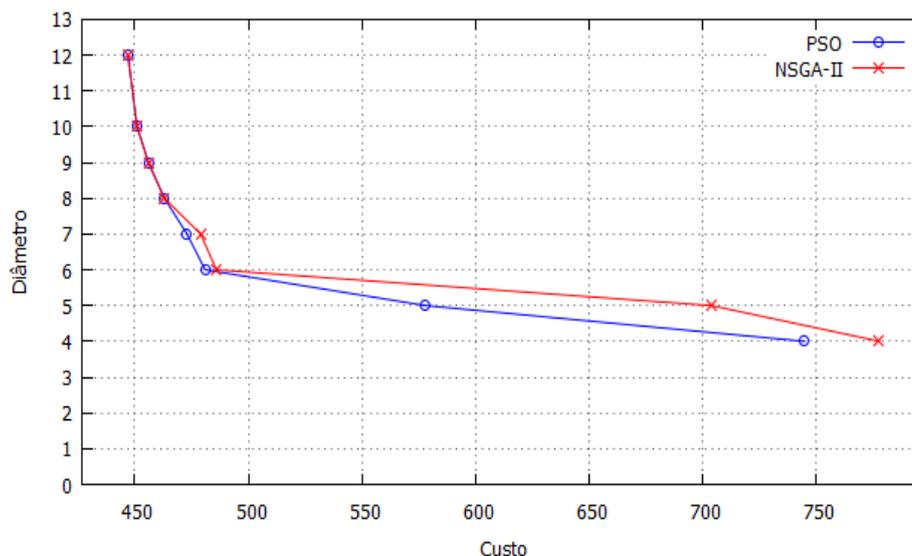


Figura 21 - Comparação entre o PSO e o NSGA-II para a instância c\_v25\_0.4

Instâncias Santos, Lucena e Ribeiro (2004)								
Instância	NSGA-II				PSO			
	Q	S	H	Tempo	Q	S	H	Tempo
c_v10_a45_d4	8	0.211	0.730	0.17	8	0.211	0.730	0.26
c_v10_a45_d5	5	0.202	0.517	0.07	5	0.202	0.517	0.26
c_v10_a45_d6	5	0.262	0.615	0.16	5	0.206	0.619	0.24
c_v10_a45_d7	8	0.218	0.745	0.17	8	0.219	0.746	0.27
c_v10_a45_d8	6	0.257	0.697	0.16	6	0.257	0.697	0.22
c_v10_a45_d10	7	0.231	0.724	0.14	7	0.231	0.724	0.26
c_v15_a105_d4	11	0.157	0.803	0.56	11	0.164	0.807	0.94
c_v15_a105_d8	7	0.171	0.653	0.51	7	0.171	0.657	0.86
c_v25_a300_d4	13	0.136	0.784	3.38	13	0.123	0.785	5.42
c_v25_a300_d5	13	0.143	0.806	3.20	13	0.148	0.819	6.09
c_v25_a300_d6	11	0.123	0.747	3.23	10	0.132	0.723	7.74
c_v25_a300_d8	15	0.116	0.817	3.38	15	0.129	0.830	6.98
c_v25_a300_d9	10	0.161	0.741	3.11	10	0.165	0.750	4.03
s_v20_a50_d4	9	0.137	0.718	1.14	9	0.141	0.722	2.55
s_v20_a50_d5	9	0.179	0.696	1.08	9	0.187	0.713	2.72
s_v20_a50_d6	9	0.153	0.743	1.86	10	0.170	0.751	2.76
s_v20_a50_d7	9	0.160	0.722	1.23	9	0.198	0.750	3.24
s_v20_a50_d8	10	0.179	0.759	1.72	10	0.184	0.766	2.95
s_v40_a100_d4	17	0.131	0.808	31.56	18	0.139	0.825	48.92
s_v40_a100_d5	15	0.164	0.819	23.26	15	0.166	0.828	41.12
s_v40_a100_d6	11	0.163	0.759	20.70	11	0.170	0.773	46.14
s_v60_a150_d5	15	0.147	0.811	133.19	16	0.151	0.819	202.88

Tabela 9 – Comparação de resultados entre NSGA-II e o PSO para Santos, Lima e Aloise (2012).

Instâncias Santos, Lima e Aloise (2013)								
Instância	NSGA-II				PSO			
	Q	S	H	Tempo	Q	S	H	Tempo
c_v10_0.3	2	0.0	0.0	0.01	2	0.0	0.0	0.42
c_v10_0.4	3	0.0	0.417	0.71	3	0.0	0.417	1.42
c_v15_0.3	5	0.205	0.646	1.97	7	0.176	0.560	4.81
c_v15_0.4	6	0.302	0.764	2.22	6	0.234	0.666	5.03
c_v20_0.2	5	0.192	0.515	1.10	5	0.212	0.667	7.47
c_v20_0.3	4	0.427	0.519	0.78	4	0.408	0.589	5.47
c_v20_0.4	7	0.219	0.689	4.48	7	0.264	0.750	9.57
c_v25_0.2	6	0.274	0.532	5.74	7	0.123	0.601	15.98
c_v25_0.3	6	0.151	0.492	4.65	5	0.186	0.629	13.51
c_v25_0.4	8	0.233	0.739	8.05	8	0.209	0.781	21.00
p_v40_0.09	3	0.0	0.069	0.17	3	0.0	0.376	7.36
p_v40_0.13	7	0.235	0.804	5.96	9	0.201	0.806	33.21

Tabela 10 - Comparação de resultados entre o NSGA-II e o PSO para Santos, Lima e Aloise (2013).

Instâncias Gouveia e Magnanti (2003)								
Instância	NSGA-II				PSO			
	Q	S	H	Tempo	Q	S	H	Tempo
se_v40_a400_d5	16	0.098	0.791	19.95	17	0.123	0.825	35.34
se_v60_a600_d5	21	0.075	0.783	96.06	22	0.096	0.828	175.73
sr_v40_a400_d5	9	0.159	0.732	16.69	9	0.159	0.736	21.66
sr_v60_a600_d5	12	0.088	0.699	89.27	12	0.099	0.706	125.15

Tabela 11 – Comparação de resultados entre NSGA-II e o PSO para Gouveia e Magnanti (2003).

## 6 CONCLUSÕES E TRABALHOS FUTUROS

Neste trabalho, um procedimento exato para calcular fronteira de Pareto ótima para o bi-AGCDM é proposto. Embora ele não trabalhe em um tempo viável com grandes instâncias, como por exemplo, grafo completos com 40 nós, as fronteiras de Pareto ótimas geradas com este procedimento para algumas instâncias da literatura constituem uma contribuição importante. Em particular, esses resultados servem de base para análises da qualidade dos

resultados heurísticos para o bi-AGCDM. O resultado da fronteira Pareto ótima para diversas instâncias foram publicados no endereço <http://di.uern.br/dario/bi-mdcst-problem/>.

O procedimento exato desenvolvido pode ser melhorado em alguns aspectos. Por exemplo, pode-se desenvolver ou integrar outros métodos para a resolução dos subproblemas em cada partição do espaço de busca, tais como *Branch-and-Cut*, geração de colunas e algoritmos de enumeração.

Outra contribuição deste trabalho é a integração de novas estratégias ao NSGA-II na abordagem do problema bi-AGCDM, afim de que ele melhore seu desempenho, conforme foi verificado nos testes, quando comparado a outro trabalho que também utilizou o NSGA-II para tratar o problema bi-AGCDM.

Além disso, a metaheurística PSO é proposta como uma alternativa ao NSGA-II na abordagem do problema bi-AGCDM. A metaheurística PSO, no geral, apresentou melhores resultados.

Na população do NSGA-II existem grupos de indivíduos com um diâmetro específico. Para tanto, como trabalho futuro sugere-se aplicar o procedimento de *ranking* e *crowding distance* em cada grupo de maneira separada, pois cada grupo possui suas particularidades e acredita-se que isso melhorará os resultados do NSGA-II.

## 7 REFERÊNCIAS BIBLIOGRÁFICAS

- Achuthan, N. R.; L.Caccetta; Caccetta, P. A.; Geelen, J. F.; **Computational methods for the diameter restricted minimum weight spanning tree problem.** *Australasian Journal of Combinatorics*, v. 10, p. 51–71, 1994.
- Arroyo, J. E. C.; Vieira, P. S.; Vianna, D. S. **A GRASP algorithm for the multi-criteria minimum spanning tree problem.** *Annals of Operations Research*, v. 159, p. 125–133, 2008.
- Aloise, D. J. **bi-MDCST Problem.** Disponível em: <http://di.uern.br/dario/bi-mdcst-problem/>. Acessado em 01 de fevereiro de 2014.
- Binh, H. T. T.; Mckay, R. I.; Hoai, N. X. **New Heuristic and Hybrid Genetic Algorithm for Solving the Bounded Diameter Minimum Spanning Tree Problem.** *Genetic and Evolutionary Computation Conference*. Montreal, Quebec, Canadá, 2009.
- Bui, M.; Butelle, F.; Lavault, C. **A distributed algorithm for constructing the minimum spanning tree problem.** *Journal of Parallel and Distributed Computing*, v. 64, p. 571 – 577, 2004.
- Carrano, E. G.; Fonseca, C. M.; Takahashi, R. H. C.; Pimenta, L. C. A. E.; Neto, O. M. **A preliminary comparison of tree encoding schemes for evolutionary algorithms.** p. 1969 –1974, Montreal, Canadá, 2007.
- Cormen, T.; Leiserson, C.; Rivest, R. **Introduction to algorithms.** McGraw-Hill, New York, EUA, 1990.
- Deb, K.; Pratap, K.; Sameer, A. A.; Meyarivan, T. **A Fast and Elitist Multiobjective Genetic Algorithm: NSGA-II.** *IEEE Transactions on Evolutionary Computation*, v. 6. ISSN 2, 2002.
- Desrochers, M.; Laporte, G. **Improvements and extensions to the Miller-Tucker-Zemlin subtour elimination constraints.** *Operations Research Letters*, v. 10, p. 27–36, 1991.
- Dhaenens, C.; Lemestre, J.; Talbi, E.G. **K-PPM: A New Exact Method To Solve Multi-Objective Combinatorial Optimization Problems.** *European Journal of Operational Research*. p. 45-53, 2010.
- Freitas, F. G.; Maia, C. L B.; Campos, G. A. L.; Souza, J. T. **Otimização em Teste de Software com Aplicação de Metaheurísticas.** *Revista de Sistemas de Informação da FSMA*. n. 5. p. 3-13, 2010.

- Eberhart, R.C.; Shi, Y. **Evolving artificial neural networks**. In: *Proceedings of the International Conference on Neural Networks and Brain*, p. PL5–PL13, 1998.
- Garey, M.; Johnson, D. **Computers and intractability: A guide to the theory of NP-completeness**. W. H. Freeman, New York, EUA, 1979.
- Gouveia, L.; Magnanti, T. L. **Network flow models for designing diameter-constrained minimum-spanning and steiner trees**. *Networks*, v. 41, p. 159–173, 2003.
- Gouveia, L.; Simonetti, L.; Uchoa, E. **Modeling hop-constrained and diameter-constrained minimum spanning tree problems as steiner tree problems over layered graphs**. *Mathematical Programming*, v. 128, p. 123–148, 2011.
- Gruber, M. e Raidl, G. R. **Variable neighborhood search for the bounded diameter minimum spanning tree problem**. Hansen, P., Mladenovic, N., Pérez, J. A. M., Batista, B M. e MorenoVega, J. M. (Eds.), *Proceedings of the 18th Mini Euro Conference on Variable Neighborhood Search*, p. 1–11, 2005.
- Handler, G. **Minimax location of a facility in an undirected graph**. *Transportation Science*, v. 7, p. 287–293, 1978.
- Haimes, Y.; Ladson, L.; **On a bicriterion formulation of the problems of integrated system identification and system optimization**. *IEEE Transactions on Systems, Man and Cybernetics*, v. SMC-1, n. 3, p. 296–297, 1971.
- Ho, J. M.; Lee, D. T.; Chang, C.-H.; Wong, K. **Minimum diameter spanning trees and related problems**. *SIAM Journal on Computing*, v. 20, n. 5, p. 987–997, 1991.
- Kennedy, J.; Eberhart, R.C. **“Particle Swarm Optimization”**, *Proceedings of the International Conference on Neural Networks*, Perth, Australia, 1995.
- Kropiwiiec, D. D. **MO405**. Disponível em: <http://www.ic.unicamp.br/~meidanis/courses/mo405/2002s2/atas/2002-08-26.html>. Acessado em 24 de janeiro de 2014.
- Lemestre, J.; Dhaenens, C.; Talbi, E.G. **Parallel Partitioning Method (PPM): A New Exact Method To Solve Bi-Objective Problems**. P. 2450 – 2462, 2007.

- Lucena, A.; Ribeiro, C. C.; Santos, A. C. **A hybrid heuristic for the diameter constrained minimum spanning tree problem.** *Journal of Global Optimization*, v. 46, p. 363 - 381, 2010.
- Lima, Diego Rocha. **Modelos e metaheurísticas multi-objetivos para o problema da árvore geradora de custo e diâmetro mínimos.** Dissertação de Mestrado. Universidade do Estado do Rio Grande do Norte. Mossoró – RN. 2012.
- Lima, D. R.; Santos, A. C.; Aloise, D. J. **Um Algoritmo Evolucionário Nsga-ii Para Resolver o Problema Bi-Objetivo Da Árvore Geradora De Custo e Diâmetro Mínimos.** *Congreso Latino-Iberoamericano Investigación Operativa (CLAIO – SBPO)*. Rio de Janeiro – RJ, Brasil, 2012.
- Marathe, M. V.; Ravi, R.; Sundaram, R.; Ravi, S.; Rosenkrantz, D. J.; HuntIII, H. B. **Bicriteria network design problems.** *Journal of Algorithms*, v. 28, n. 1, p. 142–171. 1998.
- Miller, C.; Tucker, A.; Zemlin, R. **Integer programming formulations and traveling salesman problems.** *Journal of the ACM*, v. 7, p. 326 – 329. 1960.
- Neumann, F.; Wegener, I. **Minimum spanning trees made easier via multi-objective optimization.** *Natural Computing*, 5:305–319, 2006.
- Nghia, N. D.; Binh, H. T. T. **New Recombination Operator in Genetic Algorithm For Solving the Bounded Diameter Minimum Spanning Tree Problem.** *International Conference on Research, Innovation and Vision for the Future*. p. 108 – 113, 2007.
- Oliveira, M. C. S.; Silva, T. L.; Aloise, D. J. **Otimização por nuvem de partículas: diferença entre Aplicações a problemas contínuos e discretos.** XXXVI SBPO. São João Del-Rei – MG. 2004.
- Patvardhan, C.; Prakash, V. P. **Novel Deterministic Heuristics for Building Minimum Spanning Trees with Constrained Diameter.** *International Conference on Pattern Recognition and Machine Intelligence*, v. 5909, p. 68 - 73, 2009.
- Raidl, G. R.; Julstrom, B. A. **Edge sets: an effective evolutionary coding of spanning trees.** *IEEE Transactions on Evolutionary Computation*, v. 7, n. 3, p. 225 – 239, 2003.

- Requejo, C.; Santos, E. **Greedy heuristics for the diameter-constrained minimum spanning tree problem.** *Journal of Mathematical Sciences*, v. 161, p. 930–943, 2009.
- Reyes-Sierra, M.; Coelho, C. A. C. **Multi-Objective Particle Swarm Optimizers: A Survey of the State-of-the-Art.** *International Journal of Computational Intelligence Research*. Vol.2, No.3, p. 287–308, 2006.
- Saha, S.; Kumar, R. **Bounded-diameter MST instances with hybridization of multi-objective EA.** *International Journal of Computer Applications*, v. 18, p. 17–25, 2011.
- Salehi-Fathabadi, H.; Ahrabian, H. **A new Algorithm for minimum spanning tree using depth-first-search in a unidirected graph.** *International Journal of Computer Mathematics*. Volume 57, Issue 3-4, 1995.
- Santos, A. C. **Modelos e Algoritmos para o Problema da Árvore Geradora de Custo Mínimo com Restrição de Diâmetro.** Tese de Doutorado, PUC-Rio, Rio de Janeiro, Brasil. 2004.
- Santos, A. C.; Lima, D. R.; Aloise, D. J. **Modeling the bi-objective diameter minimum spanning tree problem.** Proceedings of the Global Optimization Workshop 2012, p. 125-128, Natal-RN, Brazil, 2012.
- Santos, A. C.; Lucena, A.; Ribeiro, C. C. **Solving diameter constrained minimum spanning tree problem in dense graphs.** *Lecture Notes in Computer Science*, v. 3059, p. 458–467, 2004.
- Santos, A. C.; Lima, D. R.; Aloise, D. J. **Modeling and solving the bi-objective minimum diameter-cost spanning tree problem.** *Journal of Global Optimization*, v. 57, p. 4, 2013.
- Sourd, F.; Spanjaard, O. **A multiobjective branch-and-bound framework: Application to the biobjective spanning tree problem.** *INFORMS Journal on Computing*, v. 20, n. 3, p. 472–484, 2008.
- Sousa, E. G.; Santos, A. C.; Aloise, D. J. **Um procedimento para gerar fronteiras de pareto ótimas para o problema biobjetivo da árvore geradora de custo e diâmetro mínimos.** XLV Simpósio Brasileiro de Pesquisa Operacional. Natal – RN. Brasil. 2013.
- Steiner, S.; Radzik, T. **Computing all efficient solutions of the biobjective minimum spanning tree problem.** *Computers & Operations Research*, v. 35, n. 1, p. 198–211, 2008.

- Noronha, T. F.; Ribeiro, C. C.; Andrea, C. S. **Solving diameter-constrained minimum spanning tree problems by constraint programming.** *International Transactions in Operational Research*, v. 17, n. 5, p. 653–665. 2010.
- Ulungu, E.L.; Teghem, J. **The two-phase method: An efficient procedure to solve bi-objective combinatorial optimization problems,** *Foundations of Computing and Decision Sciences*, p. 149–165. 1995.
- Veldhuizen, D. A. V.; Lamont, G. B. **On measuring multiobjective evolutionary algorithm performance.** Proceedings of the Congress on Evolutionary Computation, p. 204 – 211, California, USA, 2000.
- Zhou, G.; Gen, M. **Genetic algorithm approach on multi-criteria minimum spanning tree problem.** *European Journal of Operational Research*, v. 114, p. 141–152. 1999.
- Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C; Da Fonseca.; V. **Performance assessment of multiobjective optimizers: an analysis and review.** *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117 – 132, 2003.