

## META-HEURÍSTICAS PARA O PROBLEMA BIOBJETIVO DA ÁRVORE GERADORA DE CUSTO E DIÂMETRO MÍNIMOS

**Ernando Gomes de Sousa<sup>1</sup>, Dario José Aloise<sup>2</sup>, Andréa Cynthia Santos<sup>3</sup>**

<sup>1</sup> Universidade Federal do Ceará, UFC  
Campus do Pici - Bloco 910 - CEP 60455-900 - Fortaleza – CE, Brasil.

<sup>2</sup> Universidade do Estado do Rio Grande do Norte, UERN  
Rua Almino Afonso, 478, CEP 59.610-210, Mossoró, RN, Brasil.

<sup>3</sup> ICD-LOSI, UMR CNRS 6281, Université de Technologie de Troyes  
12, rue Marie Curie, CS 42060, 10004, Troyes CEDEX, France.

{ernandogomes@lia.ufc.br, darioaloise@uern.br, andrea.duhamel@utt.fr}

### RESUMO

Este trabalho aborda o problema biobjetivo da Árvore Geradora de Custo e Diâmetro Mínimos (bi-AGCDM) por meio de duas meta-heurísticas. O bi-AGCDM é NP-Difícil e modela problemas em projeto de redes, onde todos os vértices devem comunicar-se entre si a um custo mínimo e com uma quantidade mínima de arestas. As meta-heurísticas utilizadas neste trabalho são a *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) e a *Multiobjective Discrete Particle Swarm Optimization* (MDPSO). Os experimentos são apresentados para três grupos de instâncias e o desempenho das meta-heurísticas é avaliado utilizando as métricas hipervolume, tempo e quantidade de soluções na melhor fronteira de Pareto produzida pelos métodos. Os resultados mostraram que o MDPSO fornece resultados competitivos em relação ao NSGA-II quando aplicados ao bi-AGCDM.

**PALAVRAS CHAVES:** Multiobjetivo, bi-AGCDM, Árvores geradoras.

**ÁREAS PRINCIPAIS:** MH - Metaheurísticas, OC - Otimização Combinatória.

### ABSTRACT

In this study, two multi-objective metaheuristics are presented for the bi-objective Minimum Diameter-Cost Spanning Tree problem (bi-MDCST). The bi-MDCST is a NP-Hard problem and models network design problems, where all nodes have to communicate with each other in a minimum cost and using a minimum number of hops (edges). The metaheuristics Non-dominated Sorting Genetic Algorithm II (NSGA-II) and Multi-objective Discrete Particle Swarm Optimization (MDPSO) have been developed. The experiments were performed with three groups of instances and the algorithms performance are evaluated in terms of hypervolume, time and number of solutions in the best Pareto front found so far. Results produced by the MDPSO are competitive whenever compared to the NSGA-II.

**KEYWORDS:** Multi-objective, bi-AGCDM, Spanning trees.

**MAIN AREA:** MH - Metaheuristics, CO - Combinatorial Optimization.

## 1. Introdução

Dado um grafo  $G=(V, E)$ , onde  $V$  representa o conjunto de vértices e  $E$  o conjunto de arestas, uma árvore geradora  $T$  de  $G$  é um subgrafo conexo e sem ciclo que contém todos os vértices de  $G$ . 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), sendo o diâmetro de uma árvore, a quantidade de arestas em seu maior caminho.

Para calcular a AGCM de um grafo existem diversos algoritmos polinomiais na literatura, tais como os clássicos algoritmos de Prim e Kruskal (Cormen, Leiserson e Rivest, 1990). Considerando  $m=|E|$  e  $n=|V|$ , a complexidade no pior caso dos algoritmos de Prim e Kruskal é  $O(m \log n)$  e  $O((m+n) \log n)$ , respectivamente. 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 (Salehi-Fathabadi e Ahrabian, 1995). Algoritmos eficientes para calcular a AGDM são apresentados em Bui, Butelle e Lavoult (2004).

Este trabalho aborda o problema Biobjetivo da Árvore Geradora de Custo e Diâmetro Mínimos (bi-AGCMD). O bi-AGCMD modela problemas em redes. No caso de redes de computadores, por exemplo, o custo da árvore visa garantir que as estações (vértices) possam comunicar-se entre si a um custo mínimo. O diâmetro mínimo da árvore é associado à qualidade de serviço de comunicação da rede, onde o interesse é diminuir o retardo de transmissão de dados para aumentar a confiabilidade da mesma.

Segundo Ho *et al.* (1991), o bi-AGCMD é NP-difícil. Formalmente, esse problema é definido em um grafo  $G=(V,E)$ , onde um custo  $c_{ij} \geq 0$  é associado a cada aresta  $[i,j] \in E$ ,  $i < j$ . Sendo  $T$  uma árvore geradora de  $G$ , existe um único caminho  $p_{ij}$  em  $T$  ligando qualquer par de vértices  $i, j \in V$ . Neste caminho,  $d_{ij}$  corresponde ao número de arestas entre os vértices  $i, j \in p_{ij}$  e o diâmetro  $D$  de  $T$  é definido como  $D = \max\{d_{ij} : i, j \in V\}$ . O bi-AGCMD consiste em encontrar uma árvore geradora de custo e diâmetro mínimos. Apesar de ser fácil encontrar uma árvore geradora em um grafo, quando são adicionadas algumas restrições, tais como o diâmetro e custo, a busca dessa árvore passa a ser um problema NP-Difícil. Neste trabalho, abordamos o bi-AGCMD com as meta-heurísticas *Non-dominated Sorting Genetic Algorithm II* (NSGA-II) e a meta-heurística *Multiobjective Discrete Particle Swarm Optimization* (MDPSO).

Este trabalho está organizado da seguinte forma: na Seção 2 é realizada uma revisão bibliográfica do problema bi-AGCMD. Na Seção 3 o problema é formalmente definido e são apresentadas algumas definições relativas aos limites inferiores e superiores do espaço de busca para o bi-AGCMD. Nas Seções 4 e 5 são apresentados, respectivamente, o NSGA-II e o MDPSO para o bi-AGCMD. Na Seção 6 são exibidos os resultados obtidos com as meta-heurísticas utilizadas neste trabalho e na Seção 7 são apresentadas conclusões e trabalhos futuros.

## 2. Estado da Arte

O bi-AGCMD generaliza o problema da Árvore Geradora de Custo Mínimo com Restrição de Diâmetro (AGMRD). O AGMRD é bastante investigado na literatura e consiste em encontrar uma árvore geradora de custo mínimo, onde o diâmetro não excede um dado valor (inteiro positivo). Para árvores geradoras de custo mínimo com restrição de diâmetro iguais a 2 ou 3 existem algoritmos polinomiais para solucioná-las, mas se o diâmetro é maior do que 3, o problema é NP-difícil (Garey e Johnson, 1979). Existem diversos trabalhos na literatura que abordam o AGMRD: formulações são apresentadas por Gouveia e Magnanti (2003), Santos, Lucena e Ribeiro (2004), Gruber e Raidl (2005); algoritmos exatos por Noronha, Ribeiro e Santos (2010) e Gouveia, Simonetti e Uchoa (2011); heurísticas e meta-heurísticas foram propostas por Gruber e Raidl (2005), Requejo e Santos (2009), Lucena *et al.* (2010).

Alguns trabalhos na literatura tratam o problema biobjetivo da Árvore Geradora com duas funções de custos. Os autores Sourd e Spanjaard (2008) propõem um algoritmo *Branch-and-Bound* para este problema. Um algoritmo de enumeração em duas fases é apresentado em Ramos *et al.* (1998), Steiner and T. Radzik (2008). Os autores Arroyo, Vieira e Vianna (2008) utilizam o algoritmo *Greedy Randomized Adaptive Search Procedure* (GRASP). Um algoritmo evolucionário multiobjetivo (MOEA) é apresentado em Zhou e Gen (1999).

Para o bi-AGCDM, algoritmos aproximativos foram propostos em Marathe *et al.* (1998) e um MOEA é apresentado por Kumar, Singh e Chakrabarti (2005), Saha e Kumar (2011). Os autores Santos, Lima e Aloise (2014) desenvolveram uma formulação matemática multifluxo e compararam os resultados obtidos com as meta-heurísticas MOEA e NSGA-II. Procedimentos exatos baseados nos métodos  $\epsilon$ -constraint e *Parallel Partitioning Method* são apresentados em Sousa, Santos e Aloise (2013) e Sousa, Santos e Aloise (2014).

### 3. O Problema bi-AGCDM

Em problemas multiobjetivos (MOOP, do inglês *Multi-Objective Optimization Problem*) tem-se duas ou mais funções objetivos a serem otimizadas simultaneamente. A solução  $x$  de um tal problema em MOOP, sem perda de generalidade, é 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  $f: X \rightarrow Y$  atribui para  $x$  um vetor objetivo  $(f_1(x), f_2(x), f_3(x), \dots, f_l(x))$  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  $l$  objetivos, onde  $L$  é o conjunto de objetivos a serem otimizados e  $l \in L$  (Dhaenens, Lemestre e Talbi, 2010).

Funções objetivos de problemas MOOP conflitantes entre si, comumente não possuem uma solução única, mas um conjunto de soluções que apresentam um melhor compromisso entre os valores das funções objetivos, as quais são conhecidas como soluções não dominadas. O conjunto de soluções não dominadas por nenhuma outra no espaço de busca da solução recebe o nome de fronteira de Pareto ótima. Esse conjunto de soluções é definida por Dhaenens, Lemestre e Talbi (2010) como se segue:

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

$$\left\{ \begin{array}{l} \forall l \in L, f_l(y) \leq f_l(y'), \\ \exists l \in L, f_l(y) < f_l(y') \end{array} \right. \quad (1)$$

**Definição 2.** Uma solução é Pareto ótima se ela não é dominada por nenhuma outra solução no espaço de solução. Uma solução deste tipo é chamada suportada.

#### 3.2. Limite do espaço de busca da solução do bi-AGCDM

Definir o limite do espaço de busca da solução para o problema bi-AGCDM significa identificar um subespaço que possui todas as árvores geradoras da fronteira de Pareto ótima para uma determinada instância. Isto é importante porque analisar todas as possíveis árvores geradoras de um grafo pode ser computacionalmente muito caro em termos de tempo de cálculo. Em grafos completos, por exemplo, a quantidade de árvores geradoras é definida por  $n^{n-2}$ , segundo o Teorema de Caley, (Norman, Lloyd e Wilson, 1986).

Os limites para o espaço de busca do bi-AGCDM em um grafo  $G$  são estabelecidos por meio da AGCM e da AGDM. Sabe-se que na fronteira de Pareto ótima não haverá uma árvore geradora de custo menor do que AGCM, pois o seu custo é o mínimo. Qualquer outra solução de

custo idêntico ao da AGCM e diâmetro maior será considerada uma solução dominada, de acordo com a Definição 1. Da mesma forma, considerando o diâmetro, sabe-se que na fronteira de Pareto ótima não haverá uma solução de diâmetro menor do que a AGDM. Qualquer outra solução de mesmo diâmetro e custo maior do que a AGDM será considerada dominada, de acordo com o Definição 1.

#### 4. Meta-heurística NSGA-II aplicada ao bi-AGCDM

O NSGA-II é uma meta-heurística baseada em algoritmos genéticos e destinada a resolver problemas multiobjetivos. A ideia principal do NSGA-II é gerar novos indivíduos (soluções) para a iteração seguinte a partir de uma seleção de indivíduos provenientes da iteração atual. Os indivíduos são classificados em elite e não elite por meio de dois operadores: *ranking* e *crowding distance*. O *ranking* é aplicado na população de cada iteração do NSGA-II para que seus indivíduos sejam classificados em fronteiras. Dado um indivíduo  $z$  qualquer, a medida *ranking*  $r_j$  guarda o índice de qual fronteira o indivíduo  $z$  pertence. Dessa forma, todos os indivíduos que possuem  $r_j = 0$  pertencem a fronteira  $F_1$ , os indivíduos com  $r_j = 1$  pertencem a fronteira  $F_2$  e assim por diante. Na primeira fronteira estão todas as soluções não-dominadas encontradas pelo algoritmo. O *crowding distance* ( $w_i$ ) define a distância de um indivíduo em relação aos seus vizinhos mais imediatos na fronteira. Assim sendo, quanto maior for o valor do *crowding distance*, mais distante este indivíduo está de seus vizinhos, indicando a necessidade de gerar mais soluções nesta região. Detalhes relacionados ao *ranking* e *crowding distance* são encontrados em (Deb *et al.*, 2002).

Nas iterações do NSGA-II, após o cálculo do *ranking* e *crowding distance*, a população é ordenada utilizando o operador *crowded* definido da seguinte forma: dados dois indivíduos  $i$  e  $j$ ,  $i$  é melhor do que  $j$  se  $(r_i < r_j)$  ou  $((r_i = r_j) e (w_i < w_j))$ . A população da iteração seguinte é construída com a primeira metade dos indivíduos da iteração atual (soluções elite) e os outros indivíduos são gerados por meio da aplicação de operadores genéticos entre indivíduos selecionados aleatoriamente do conjunto de soluções elite.

No NSGA-II aplicado ao bi-AGCDM, um indivíduo representa uma árvore geradora, com custo e diâmetro específicos. A primeira etapa desta meta-heurística consiste em determinar os limites do espaço de busca da solução do problema, como mencionado na Seção 3.2. Uma vez definidos os limites do espaço de busca, indivíduos são gerados considerando todos os possíveis diâmetros entre a AGCM e a AGDM. O tamanho de uma população do NSGA-II é  $2n$ . A quantidade de indivíduos na população com um diâmetro específico é determinada pela variável  $e$  definida na equação (2), onde  $2n$  representa o tamanho da população,  $s$  representa o diâmetro da AGCM e  $i$  o diâmetro da AGDM, sendo  $s > i > 0$ . Os indivíduos com diâmetro 2, 3 e o diâmetro da AGCM são utilizados para determinar os limites do espaço de busca, mas não participam das operações de evolução na população do NSGA-II.

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

Os indivíduos da população inicial são construídos por meio do algoritmo de Prim aleatorizado. Na versão original do algoritmo de Prim, em cada iteração, o nó selecionado para fazer parte da solução é aquele que se conecta com o menor custo à solução parcial. Na versão aleatorizada, um sorteio é realizado para escolher qual nó entrará na solução parcial. De acordo com um diâmetro específico  $D'$  determinado a priori, inicialmente constrói-se um caminho com este diâmetro  $D'$ . Em seguida, os outros nós serão inseridos na solução parcial sem violar  $D'$  por meio da aresta de menor custo possível.

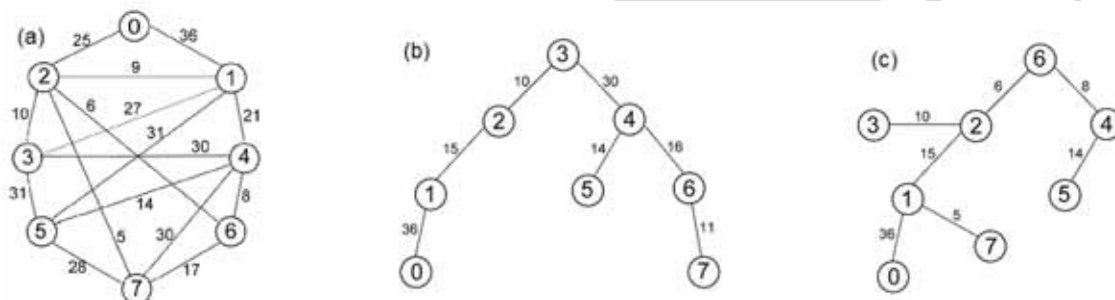
Uma vez que a população inicial é gerada, aplica-se os operadores *ranking* e *crowding distance* para classificar os indivíduos da população em um conjunto elite e em um conjunto não elite, ambos os conjuntos de tamanho  $n$ . Em seguida, os indivíduos do conjunto elite são



mantidos para a próxima iteração. Os outros indivíduos que irão completar a população na iteração seguinte são gerados por meio de um cruzamento entre um indivíduo da população elite e outro da população não elite. Embora o NSGA-II padrão recomende o cruzamento entre dois indivíduos da população elite, em teste realizados com o bi-AGCDM percebeu-se que o cruzamento entre indivíduos elite e não elite geram melhores resultados. Estes cruzamentos foram realizados de acordo com o sugerido por Carrano *et al.* (2007). Duas soluções são unidas em um grafo suporte e um novo indivíduo é gerado por meio do algoritmo de Prim aleatorizado e outro por meio do Prim original. Aquele que tiver o menor custo será inserido na população.

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

Considere como nó raiz de uma árvore aquele que encontra-se a no máximo  $D/2$  arestas de distância de todos os outros nós da árvore, quando  $D$  é par (resp. no máximo  $(D-1)/2$  arestas de todos os outros nós da árvore quando  $D$  é ímpar). Na segunda busca local, uma permutação de nós é realizada com o nó(s) raiz(es) da árvore. O movimento é aceito se a nova árvore gerada com a permutação é conexa e tem um custo inferior ao da árvore original. Esta busca local tem um impacto interessante no problema bi-AGCDM porque ela pode gerar facilmente soluções com diâmetros pequenos, tais como  $D=4$  ou  $D=5$ . Soluções com diâmetro e custo pequenos são as mais difíceis de serem encontradas. Quando o diâmetro da nova árvore é reduzido em relação ao da árvore inicial, o movimento também pode ser aceito mesmo com o custo da nova árvore aumentando. Neste caso, ela entrará na população substituindo alguma árvore de mesmo diâmetro e com o custo maior. Além disso, quando o diâmetro da árvore é alterado pode ocorrer uma modificação do nó raiz. A figura 1 ilustra um exemplo desta situação. Na árvore inicial, Figura 1 (b), o nó 3 é a raiz da árvore e o diâmetro da mesma é  $D=6$ . Supondo que uma permutação será realizada entre os nós 3 e 6 da árvore na Figura 1 (b), o árvore resultante, Figura 1 (c), tem  $D=5$  e nós raízes iguais a 2 e 6.



**Figura 1 - Busca local de troca do nó raiz.**

A terceira busca local é baseada no grafo complementar de uma árvore. Seja  $T$  a árvore geradora em que esta busca local será aplicada. Um grafo complementar é construído contendo todas as arestas  $[i,j] \in G \setminus T$ . Em seguida, duas árvores geradoras são criadas, sendo que uma utiliza o algoritmo de Prim, a qual chamamos de  $T'$ , e a outra árvore utiliza versão aleatorizada do algoritmo Prim, a qual chamamos de  $T''$ . Se alguma das árvores  $T'$  e  $T''$  tiverem o mesmo diâmetro e custo menor do que  $T$ , esta árvore será inserida na população no lugar de  $T$ . Se alguma das árvores  $T'$  ou  $T''$  tiver diâmetro diferente de  $T$ , tenta-se encontrar na população outra árvore com mesmo diâmetro e custo maior para seja substituída.

## 5. Meta-heurística MDPSO para o bi-AGCDM

O MDPSO é uma versão do *Particle Swarm Optimization* (PSO) para solucionar problemas multiobjetivos. O MDPSO já foi aplicado, entre outros trabalhos, para otimização multiobjetivo de registro de imagens (Senthilnath *et al.* 2011), clusterização multiobjetivo (Gong *et al.* 2014) e ao problema de roteamento de veículos com janelas de tempo (Gong *et al.* 2012).

No MDPSO aplicado ao bi-AGCDM, uma solução (árvore geradora) é chamada de partícula, a qual é representada em um vetor, e um conjunto de partículas é chamado de enxame (conjunto de soluções). Considerando uma partícula  $i$  em específico, o  $pbest$  desta partícula é um ótimo local, *i.e.* uma árvore geradora de mesmo diâmetro que  $i$  e com menor custo até a iteração corrente do algoritmo. Já o  $gbest$  é a melhor partícula  $i$  dentre todas as partículas encontradas para um diâmetro específico. No contexto da otimização multiobjetivo, o  $gbest$  seria uma solução não dominada para um diâmetro alvo. Para cada partícula  $i$  do enxame é atribuída uma velocidade a cada iteração, onde a velocidade de uma partícula  $i$  é um conjunto de arestas sorteadas aleatoriamente do grafo original, de seu  $pbest$  e de seu  $gbest$ . Estas arestas serão utilizadas para modificar a partícula  $i$ , com o objetivo de melhorá-la. A mudança da partícula  $i$  (árvore geradora) é comumente conhecida como mudança de posição dentro do espaço de busca. A quantidade de arestas que serão utilizadas para modificar  $i$  é dada pela Equação (3), adaptada para o bi-AGCDM e proveniente do trabalho (Eberhart e Shi, 1998); onde, as partes da Equação (3) - (a), (b) e (c) determinam, respectivamente, a quantidade de arestas provenientes do grafo, do  $pbest$  e do  $gbest$ . Na Equação (3),  $U_i^k$  é a quantidade de arestas do conjunto de arestas da velocidade atual da partícula  $i$  na iteração  $k$ . A variável  $W^k$  é um valor, inicialmente igual a 1, que determina a aceleração da partícula  $i$  na iteração  $k$ . A cada iteração, o valor de  $W^k$  é reduzido até chegar ao valor 0 (zero). Valores maiores de  $W^k$  são empregados para realizar uma diversificação no espaço de busca, já valores menores intensificam a busca local.  $C_1 \in \mathbb{R}^+$  e  $C_2 \in \mathbb{R}^+$  são constantes (parâmetros a serem calibrados nos experimentos) e  $Z_1, Z_2 \in [0, 1]$  são variáveis cujo valores são sorteados em cada iteração do algoritmo.

$$U_i^k = \underbrace{W^k |U_i^{k-1}|}_{(a)} + \underbrace{C_1 Z_1 (|pbest_i^k - X_{il}|)}_{(b)} + \underbrace{C_2 Z_2 (|gbest_i^k - X_{il}|)}_{(c)} \quad (3)$$

No bi-AGCDM, as partículas são tratadas como um ponto em um espaço  $d$ -dimensional associadas a um custo e diâmetro. O princípio da meta-heurística MDPSO consiste, em cada iteração, modificar as soluções (partículas) utilizando movimentos definidos pela velocidade. Ao longo das iterações do algoritmo, as partículas mudam de posição no espaço de busca. No MDPSO desenvolvido para o bi-AGCDM, o enxame de partículas tem tamanho igual a  $2n$ . Os limites do espaço de busca da solução são definidos de acordo com a Seção 3.2. No início da execução do algoritmo cada partícula é gerada utilizando o algoritmo de Prim aleatorizado. A quantidade máxima de partículas a serem relacionadas com um diâmetro presente nos limites do espaço de busca é determinada pela Equação (2). Durante a definição de uma velocidade que será aplicada a uma partícula, para que arestas com custos muito alto não participem dessa velocidade, a mesma é criada a partir de uma lista de arestas restritas. O tamanho desta lista é igual a metade das arestas do grafo, se o mesmo for completo, e as arestas que a compõe são aquelas de menores custo. Caso o grafo seja esparso, este é transformado em um grafo completo através da inserção de arestas de custo infinito e o procedimento acima se aplica.

Uma característica do MDPSO clássico é sempre memorizar a velocidade que foi aplicada a uma partícula na iteração  $k$ , de modo a utilizá-la na iteração  $k + 1$ . No entanto, considerando a definição de velocidade utilizada no bi-AGCDM, se a velocidade a ser aplicada

em uma partícula for idêntica, ou quase idêntica, em todas as iterações da meta-heurística, a mudança de posição das partículas será mínima. Para evitar isto, no MDPSO aplicado ao bi-AGCDM, em cada iteração do algoritmo uma nova velocidade (escolha das arestas) é calculada para cada partícula. Outra característica particular do MDPSO aplicado ao problema bi-AGCDM é a utilização de mais de um *gbest*, um associado a cada diâmetro dentro dos limites do espaço de busca, como mencionado anteriormente. Desta forma, ao calcular a velocidade da partícula *i*, o *gbest* utilizado será aquele relacionado ao diâmetro da partícula *i*.

## 6. Resultados das meta-heurísticas

Três grupos de instâncias provenientes dos trabalhos de Gouveia e Magnanti (2003), Santos, Lucena e Ribero (2004), e Santos, Lima e Aloise (2014) foram utilizados nos experimentos. Para medir a qualidade e eficiência dos algoritmos desenvolvidos, as seguintes métricas foram utilizadas: Q, H e tempo de execução, onde Q indica a quantidade de elementos encontrados na fronteira de Pareto e a métrica H (Zitzler *et al.*, 2003) representa o Hipervolume. O Hipervolume é calculado em função da área coberta pelas soluções presentes na fronteira de Pareto em relação a um ponto  $\theta$  específico. O ponto  $\theta$  é a pior solução para a instância que está sendo tratada. Neste trabalho, definimos  $\theta$  como sendo uma solução formada pelos piores valores em termos de custo e diâmetro (custo da AGDM e diâmetro da AGCM). Quanto maior o hipervolume, melhor a fronteira de Pareto encontrada.

Devido os valores de diâmetro e custo serem em escalas diferentes, todos os valores foram normalizados entre 0 (zero) e 1. Experimentos foram realizados para calibrar o número de iterações das meta-heurísticas (50, 90 e 130 iterações). De acordo com as métricas descritas acima, os resultados obtidos com 90 iterações apresentaram o melhor compromisso em termos de tempo de cálculo e qualidade das soluções. No MDPSO aplicado ao bi-AGCDM, os melhores valores encontrados para  $C_1$  e  $C_2$  foram  $C_1=2,5$  e  $C_2=2,5$  após realização de testes preliminares de calibração.

As Tabelas 1, 2 e 3 mostram os resultados das métricas Q, H e tempo de cálculo para um total de 36 instâncias provenientes, respectivamente, dos trabalhos de Santos, Lucena e Ribeiro (2004), Santos, Lima e Aloise (2014) e Gouveia e Magnanti (2003). Das 36 instâncias tratadas neste trabalho, considerando a métrica Q, os dois algoritmos encontraram valores idênticos para Q em 27 instâncias, sendo que MDPSO encontrou mais elementos na fronteira de Pareto para 7 instâncias, enquanto que o NSGA-II encontrou melhores valores em Q para 2. Em relação a métrica H, em 28 instâncias o MDPSO encontrou melhores resultados e o NSGA-II encontrou 3 instâncias melhores resultados. Os resultados indicam que o MDPSO encontrou bons resultados para as instâncias testadas, mas com um tempo computacional um pouco mais elevado que o NSGA-II.

Instância	NSGA-II			MDPSO		
	Q	H	Tempo	Q	H	Tempo
c_v10_a45_d4	8	0.730	0.17	8	0.730	0.26
c_v10_a45_d5	5	0.517	0.07	5	0.517	0.26
c_v10_a45_d6	5	0.615	0.16	5	<b>0.619</b>	0.24
c_v10_a45_d7	8	0.745	0.17	8	<b>0.746</b>	0.27
c_v10_a45_d8	6	0.697	0.16	6	0.697	0.22
c_v10_a45_d10	7	0.724	0.14	7	0.724	0.26
c_v15_a105_d4	11	0.803	0.56	11	<b>0.807</b>	0.94
c_v15_a105_d8	7	0.653	0.51	7	<b>0.657</b>	0.86

c_v25_a300_d4	13	0.784	3.38	13	<b>0.785</b>	5.42
c_v25_a300_d5	13	0.806	3.20	13	<b>0.819</b>	6.09
c_v25_a300_d6	<b>11</b>	<b>0.747</b>	3.23	10	0.723	7.74
c_v25_a300_d8	15	0.817	3.38	15	<b>0.830</b>	6.98
c_v25_a300_d9	10	0.741	3.11	10	<b>0.750</b>	4.03
s_v20_a50_d4	9	0.718	1.14	9	<b>0.722</b>	2.55
s_v20_a50_d5	9	0.696	1.08	9	<b>0.713</b>	2.72
s_v20_a50_d6	9	0.743	1.86	<b>10</b>	<b>0.751</b>	2.76
s_v20_a50_d7	9	0.722	1.23	9	<b>0.750</b>	3.24
s_v20_a50_d8	10	0.759	1.72	10	<b>0.766</b>	2.95
s_v40_a100_d4	17	0.808	31.56	<b>18</b>	<b>0.825</b>	48.92
s_v40_a100_d5	15	0.819	23.26	15	<b>0.828</b>	41.12
s_v40_a100_d6	11	0.759	20.70	11	<b>0.773</b>	46.14
s_v60_a150_d5	15	0.811	133.19	<b>16</b>	<b>0.819</b>	202.88

**Tabela 1 – Comparação de resultados entre NSGA-II e o PSO para Santos, Lima e Aloise (2012).**

Instância	NSGA-II			MDPSO		
	Q	H	Tempo	Q	H	Tempo
c_v15_0.3	5	<b>0.646</b>	1.97	7	0.560	4.81
c_v15_0.4	6	<b>0.764</b>	2.22	6	0.666	5.03
c_v20_0.2	5	0.515	1.10	5	<b>0.667</b>	7.47
c_v20_0.3	4	0.519	0.78	4	<b>0.589</b>	5.47
c_v20_0.4	7	0.689	4.48	7	<b>0.750</b>	9.57
c_v25_0.2	6	0.532	5.74	<b>7</b>	<b>0.601</b>	15.98
c_v25_0.3	<b>6</b>	0.492	4.65	5	<b>0.629</b>	13.51
c_v25_0.4	8	0.739	8.05	8	<b>0.781</b>	21.00
p_v40_0.09	3	0.069	0.17	3	<b>0.376</b>	7.36
p_v40_0.13	7	0.804	5.96	<b>9</b>	0.806	33.21

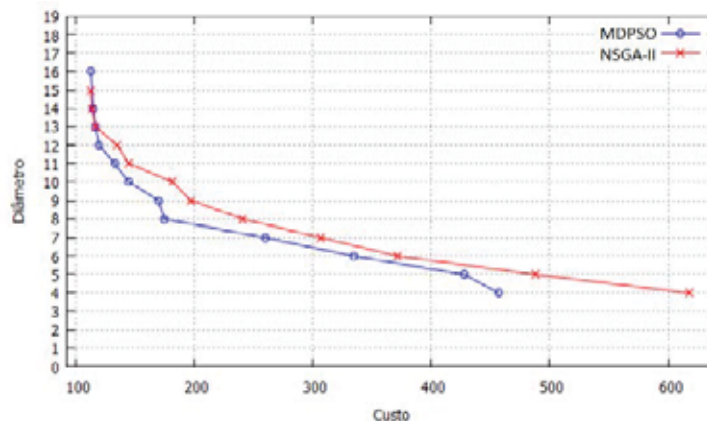
**Tabela 2 - Comparação de resultados entre o NSGA-II e o PSO para Santos, Lima e Aloise (2013).**

Instância	NSGA-II			MDPSO		
	Q	H	Tempo	Q	H	Tempo
se_v40_a400_d5	16	0.791	19.95	<b>17</b>	<b>0.825</b>	35.34
se_v60_a600_d5	21	0.783	96.06	<b>22</b>	<b>0.828</b>	175.73
sr_v40_a400_d5	9	0.732	16.69	9	<b>0.736</b>	21.66
sr_v60_a600_d5	12	0.699	89.27	12	<b>0.706</b>	125.15

**Tabela 3 – Comparação de resultados entre NSGA-II e o PSO para Gouveia e Magnanti (2003).**

Na Figura 5 é exibida uma comparação entre as duas meta-heurísticas utilizadas neste trabalho, considerando a instância *sr\_v60\_a600\_d5*. Nela, podemos observar o melhor desempenho do MDPSO em relação ao NSGA-II, quando aplicados ao bi-AGCDM.





**Figura 2 - Comparação entre o MDPSO e o NSGA-II para a instância *sr\_v60\_a600\_d5*.**

## 7. Conclusões

O problema bi-AGCDM é NP-difícil e por isso é inviável tratar algumas instâncias desse problema por meio de métodos exatos, visto a enorme quantidade de tempo que será necessário para solucioná-las. Meta-heurísticas, apesar de não garantir encontrar a solução ótima de um problema, na maioria dos casos encontram soluções de boa qualidade para problemas altamente complexos em tempos computacionais relativamente rápidos.

Os resultados obtidos com o MDPSO indicam que o mesmo tem um desempenho interessante para o bi-AGCDM quando comparado com o NSGA-II. A meta-heurística NSGA-II também já foi aplicada ao bi-AGCDM no trabalho de Santos, Lima e Aloise (2014). Contudo, o presente artigo apresenta melhores resultados com NSGA-II do que o desenvolvido no trabalho de Santos, Lima e Aloise (2014). A provável explicação é a integração das buscas locais implementadas nesse trabalho.

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 influenciar os resultados do NSGA-II.

## 8. Referências

- Arroyo, J. E. C.; Vieira, P. S.; Vianna, D. S.** (2008), A GRASP algorithm for the multi-criteria minimum spanning tree problem. *Annals of Operations Research*, v. 159, p. 125 - 133.
- Biggs, Norman L.; Lloyd, E. Keith; Wilson, Robin J.** (1986), Graph theory: 1736-1936. Oxford, Clarendon.
- Bui, M.; Butelle, F.; Lavault, C.** (2004), A distributed algorithm for constructing the minimum spanning tree problem. *Journal of Parallel and Distributed Computing*, v. 64, p. 571-577.
- Carrano, E. G.; Fonseca, C. M.; Takahashi, R. H. C.; Pimenta, L. C. A. E.; Neto, O. M.** (2007), A preliminary comparison of tree encoding schemes for evolutionary algorithms. p. 1969 -1974, Montreal, Canada.
- Cormen, T.; Leiserson, C.; Rivest, R.** (1990), Introduction to algorithms. McGraw-Hill, New York, EUA.
- Deb, K.; Pratap, K.; Sameer, A. A.; Meyarivan, T.** (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6. ISSN 2.

**Dhaenens, C.; Lemestre, J.; Talbi, E.G.** (2010), K-PPM: A new exact method to solve multi-objective combinatorial optimization problems. *European Journal of Operational Research*. p. 45-53.

**Eberhart, R.C.; Shi, Y.** (1998), Evolving artificial neural networks. In: *Proceedings Of The International Conference On Neural Networks and Brain*, p. PL5-PL13.

**Garey, M. e Johnson, D.** (1979), *Computers and Intractability: A Guide to the theory of NP-Completeness*. W. H. Freeman, New York.

**Gong, M.; Cai, Q.; Chen, X.; Ma, L.** (2014), Complex network clustering by multiobjective discrete particle swarm optimization based on decomposition. *IEEE Transactions on Evolutionary Computation*, v. 18 (1), p. 82-97.

**Gong, Y. J.; Zhang, J.; Liu, O.; Huang, R. Z.; Chung, H. S. H.; Y. Shi, H.** (2012), Optimizing the vehicle routing problem with time windows: a discrete particle swarm optimization approach. *IEEE Transactions on Systems Man and Cybernetics-Part C Applications Reviews*, v. 42, p. 254.

**Gouveia, L.; Magnanti, T. L.** (2003), Network flow models for designing diameter-constrained minimum-spanning and Steiner trees. *Networks*, v. 41, p. 159-173.

**Gouveia, L., Simonetti, L. e Uchoa, E.** (2011), Modeling hop-constrained and diameter constrained minimum spanning tree problems as steiner tree problems over layered graphs. *Mathematical Programming*, v. 128, p. 123-148.

**Gruber, M. e Raidl, G. R.** (2005), Variable Neighborhood Search For The Bounded Diameter Minimum Spanning Tree Problem. Hansen, P., Mladenovic, N., Perez, 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.

**Ho, J. M.; Lee, D. T.; Chang, C.-H.; Wong, K.** (1991), Minimum Diameter Spanning Trees And Related Problems. *SIAM Journal on Computing*, v. 20, n. 5, p. 987-997.

**Lucena, A.; Ribeiro, C. C.; Santos, A. C.** (2010), A hybrid heuristic for the diameter constrained minimum spanning tree problem. *Journal of Global Optimization*, v. 46, p. 363 - 381.

**Marathe, M. V.; Ravi, R.; Sundaram, R.; Ravi, S.; Rosenkrantz, D. J.; Hunt III, H. B.** (1998), Bicriteria network design problems. *Journal of Algorithms*, v. 28, n. 1, p. 142-171.

**Noronha, T. F.; Ribeiro, C. C.; Andrea, C. S.** (2010), Solving diameter-constrained minimum spanning tree problems by constraint programming. *International Transactions in Operational Research*, v. 17, n. 5, p. 653-665.

**Requejo, C.; Santos, E.** (2009), Greedy heuristics for the diameter-constrained minimum spanning tree problem. *Journal of Mathematical Sciences*, v. 161, p. 930-943.

**Salehi-Fathabadi, H.; Ahrabian, H.** (1995), A new algorithm for minimum spanning tree using depth-first-search in a undirected graph. *International Journal of Computer Mathematics*. Volume 57, Issue 3-4.

**Saha, S.; Kumar, R.** (2011), Bounded-diameter MST instances with hybridization of multi-objective EA. *International Journal of Computer Applications*, v. 18, p. 17-25.

**Santos, A. C.; Lucena, A.; Ribeiro, C. C.** (2004), Solving diameter constrained minimum spanning tree problem in dense graphs. *Lecture Notes in Computer Science*, v. 3059, p. 458-467.

**Santos, A. C.; Lima, D. R.; Aloise, D. J.** (2014), Modeling and solving the biobjective minimum diameter-cost spanning tree problem. *Journal of Global Optimization*, v. 57, p. 4.

**Senthilnath, J.; Omkar, S. N.; Mani, V.; Karthikeyan, T.** (2011), Multi-objective optimization of satellite image registration using discrete particle swarm optimisation. India Conference (INDICON), Annual IEEE. p.1 - 5, 2011.

**Sourd, F.; Spanjaard, O.** (2008), A multiobjective branch-and-bound framework: application to the biobjective spanning tree problem. *INFORMS Journal on Computing*, v. 20, n. 3, p. 472-484.

**Sousa, E. G.; Santos, A. C.; Aloise, D. J.** (2013), Um procedimento para gerar fronteiras de Pareto ótimas para o problema biobjetivo da árvore geradora de custo e Diâmetro mínimos. XLV Simposio Brasileiro de Pesquisa Operacional. Natal - RN. Brasil. p. 483-493.

**Sousa, E. G.; Santos, A. C.; Aloise, D. J.** (2014), An exact method for solving the bi-objective minimum diameter-cost spanning tree problem. *RAIRO Operations Research*. v. 49 p. 143-160.

**Steiner, S.; Radzik, T.** (2008), Computing all efficient solutions of the biobjective minimum spanning tree problem. *Computers & Operations Research*, v. 35, n. 1, p. 198-211.

**Zhou, G.; Gen, M.** (1999), Genetic algorithm approach on multi-criteria minimum spanning tree problem. *European Journal of Operational Research*, v. 114, p. 141-152.

**Zitzler, E.; Thiele, L.; Laumanns, M.; Fonseca, C; Da Fonseca, V.** (2003), Performance assessment of multiobjective optimizers: an analysis and review. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 2, p. 117 - 132.

