

## UM ALGORITMO EVOLUCIONÁRIO NSGA-II PARA RESOLVER O PROBLEMA BI-OBJETIVO DA ÁRVORE GERADORA DE CUSTO E DIÂMETRO MÍNIMOS

**Diego Rocha Lima<sup>1</sup>, Andréa Cynthia Santos<sup>2</sup>, Dario José Aloise<sup>1</sup>**

<sup>1</sup> Universidade do Estado do Rio Grande do Norte, UERN  
Rua Almino Afonso, 478, CEP 59.610-210, Mossoró, RN, Brasil.

<sup>2</sup> ICD-LOSI, UMR CNRS STMR, Université de Technologie de Troyes  
12, rue Marie Curie, BP 2060, 10010, Troyes CEDEX, France.  
{diegorocha@ufersa.edu.br, andrea.duhamel@utt.fr, aloisedj@gmail.com}

### RESUMO

Neste trabalho, o problema da Árvore Geradora de Custo e Diâmetro Mínimos (bi-AGCDM) é investigado. O bi-AGCDM é NP-difícil e consiste em encontrar árvores geradoras de custo e diâmetro mínimos. Problemas em redes e transportes podem ser modelados pelo bi-AGCDM. Uma formulação matemática bi-objetivo baseada em multifluxo é apresentada. Além disto, um algoritmo genético multiobjetivo NSGA-II é proposto. Experimentos computacionais foram realizados com 4 grupos de instâncias e os desempenhos avaliados através de três métricas clássicas: quantidade de indivíduos na fronteira de Pareto, espalhamento e hipervolume. Em nosso conhecimento, a formulação matemática e a metaheurística multiobjetivo são os primeiros métodos deste tipo propostos para o bi-AGCDM.

**PALAVRAS CHAVE.** Árvores geradoras, Multiobjetivo, NSGA-II, fronteira de Pareto.

**Áreas Principais:** Metaheurísticas multiobjetivo, Projeto de redes.

### ABSTRACT

In this work, the bi-objective Diameter Minimum Spanning Tree problem (bi-AGCDM) is investigated. The NP-hard bi-AGCDM consists in finding spanning trees with minimum total cost and minimum diameter. This problem can model several practical applications in networks and transportation. We propose a bi-objective mathematical formulation for the bi-AGCDM using multiflow and a multiobjective genetic algorithm NSGA-II. Four test sets are used in the computational experiments and the evaluation is measured by using the number of Pareto front solutions, the spacing, and the hypervolume. To the best of our knowledge, this is the first work to propose a multiflow mathematical formulation and a NSGA-II based metaheuristic for the bi-AGCDM.

**KEY WORDS.** Spanning trees, Multiobjective, NSGA-II, Pareto front.

**Main areas:** Multiobjective metaheuristics, Network design.

## 1 Introdução

Seja o grafo não orientado e conexo  $G = (V, E)$ , onde  $V$  é o conjunto de vértices e  $E$  é o conjunto de arestas, com custos  $c_{ij} \geq 0$  associado às arestas  $[i, j] \in E$ . O problema clássico da Árvore Geradora de custo Mínimo (AGM) consiste em encontrar uma árvore geradora de  $G$  com custo mínimo. Existem algoritmos eficientes para encontrar uma AGM em um grafo conexo como Prim e Kruskal, Cormen *et al.* (1990), que possuem respectivamente complexidade assintótica de  $O(|E| \log |E|)$  e  $O(|E| \log |V|)$ . O diâmetro de uma árvore é igual a quantidade de arestas em seu maior caminho e pode ser calculado através de uma adaptação de algoritmos de busca em largura. Neste trabalho, investigamos o problema bi-objetivo da Árvore Geradora de Custo e Diâmetro Mínimos (bi-AGCDM) que procura AGMs com custo e diâmetro mínimos.

O bi-AGCDM é NP-difícil, Ho *et al.* (1991), e modela diversos problemas de projetos em redes e transportes, Chen *et al.* (2008). Em projeto de redes, as restrições de diâmetro podem ser consideradas como restrições de qualidade de serviço, onde pequenos diâmetros reduzem o retardo e aumentam a confiabilidade das redes. Neste caso, os custos podem ser associados aos tempos ou aos custos de transmissão na rede. Outra aplicação interessante é encontrada na instalação de infraestrutura para trens bala, onde a minimizações do custo e do diâmetro melhoram os tempos de transporte entre as cidades e a qualidade de serviço.

O bi-AGCDM está relacionado com alguns problemas importantes, como o problema da Árvore Geradora de custo Mínimo com restrição de Diâmetro (AGM-D) e o problema da Árvore Geradora de Diâmetro Mínimo (AG-DM). O AGM-D consiste em encontrar uma árvore geradora de custo mínimo para um grafo dado, de modo que todos os caminhos da árvore tenham no máximo  $D$  arestas, onde  $D \geq 2$ . A diferença em relação ao bi-AGCDM está nas restrições de diâmetro sob os caminhos da AGM. Diversos trabalhos foram desenvolvidos para o AGM-D: Gouveia e Magnanti (2003); Santos *et al.* (2004) propuseram formulações matemáticas, Gouveia *et al.* (2011); Lucena *et al.* (2010); Gruber e Raidl (2005) trabalharam com algoritmos exatos, heurísticas e metaheurísticas. O AG-DM objetiva encontrar uma árvore geradora de diâmetro mínimo e difere do bi-AGCDM porque o custo total da árvore não é minimizado. Algoritmos eficientes e distribuídos para resolver este problema são apresentados em Bui *et al.* (2004); Hassin e Tamir (1995).

O bi-AGCDM também é referenciado na literatura como o problema de árvores geradoras de custo mínimo multiobjetivo (multicritério). Ho *et al.* (1991) apresentam um estudo teórico para provar que certos problemas de árvores geradoras, como o bi-AGCDM, são NP-difíceis. Marathe *et al.* (1998) também realizaram um estudo teórico e propuseram algoritmos aproximativos para os problemas de bi-AGCDM e AGM com minimização de grau (quantidade de arestas incidentes a cada nó de  $G$ ). Resultados computacionais não são apresentados. Em nosso conhecimento, os únicos trabalhos dedicados ao desenvolvimento de métodos multiobjetivo para o bi-AGCDM foram propostos por Saha e Kumar (2011). Um algoritmo genético é proposto e utiliza uma codagem dos cromossomos do tipo conjunto de arestas. Os resultados da fronteira de Pareto produzidos são comparados com heurísticas construtivas simples, tais como One Time Tree, Deo e Abdalla (2000), Randomized Greedy Heuristic, Raidl e Julstrom (2003b) e Refinamento Iterativo, Deo e Abdalla (2000). O problema é que essas heurísticas produzem em geral soluções distantes do ótimo.

Neste trabalho, nós propomos uma formulação matemática multiobjetivo para o bi-AGCDM inspirada no modelo multifluxo proposto por Gouveia e Magnanti (2003), e também desenvolvemos um algoritmo genético multiobjetivo *fast Nondominated Sorting Genetic Algorithm* (NSGA-II) para o problema, Deb *et al.* (2002). O objetivo é encontrar a superfície de compromisso (fronteira de Pareto) para otimização dos dois objetivos. A estratégia multiobjetivo proposta foi testada com quatro grupo de instâncias. Três são provenientes de trabalhos importantes do AGM-D: Lucena *et al.* (2010); Raidl e Julstrom (2003b); Gouveia e Magnanti (2003); e o último grupo foi proposto por Santos *et al.* (2012). As seções 2 e 3 são respectivamente dedicadas a uma formulação bi-objetivo e um algoritmo do tipo NSGA-II para o bi-AGCDM. Resultados são apresentados na

seção 4 e conclusões são realizadas na seção 5.

## 2 Modelo multifluxo

O bi-AGCDM é definido em um grafo conexo e não orientado  $G = (V, E)$ , como mencionado anteriormente. Em uma AGM,  $\mathcal{T}$  de  $G$ , existe um caminho único  $\mathcal{P}_{ij}$ ,  $i, j \in V$ , conectando cada par de nós de  $\mathcal{T}$ . Seja  $d_{ij}$  a quantidade de arestas em  $\mathcal{P}_{ij}$ . O diâmetro  $D = \max\{d_{ij} : i, j \in V\}$ . A formulação bi-objetivo para o bi-AGCDM apresentada de (1) a (11) utiliza multifluxo para controlar a conectividade da solução, seguindo a idéia de modelagem multifluxo para o AGM-D proposta em Gouveia e Magnanti (2003). As variáveis de decisão  $x_{ij}$ ,  $[i, j] \in E$  são usadas para definir  $\mathcal{T}$ . Se a aresta  $[i, j]$  pertence à solução, então  $x_{ij} = 1$ , senão  $x_{ij} = 0$ . As variáveis orientadas de fluxo  $y_{ij}^{pq}$  especificam se o caminho de  $p \in V$  até  $q \in V$ , com  $i \neq q$  e  $j \neq p$ , passa pela aresta  $[i, j]$ . Neste caso,  $y_{ij}^{pq} = 1$ , senão  $y_{ij}^{pq} = 0$ . As variáveis  $d_{pq}$  determinam a quantidade de arestas no caminho de  $p \in V$  até  $q \in V$ .

$$z_1(x) = \min \sum_{[i,j] \in E} c_{ij} \cdot x_{ij} \quad (1)$$

$$z_2 = \min D \quad \text{st.} \quad (2)$$

$$(3)$$

$$\sum_{[i,j] \in E} x_{ij} = |V| - 1 \quad (4)$$

$$\sum_{j:[i,j] \in E} y_{ij}^{pq} - \sum_{j:[j,i] \in E} y_{ji}^{pq} = \begin{cases} 1, & \text{if } i = p \\ 0, & \text{if } i \neq p \text{ and } i \neq q \\ -1, & \text{if } i = q \end{cases} \quad \forall i \in V, \forall p, q \in V \quad (5)$$

$$y_{ij}^{pq} + y_{ji}^{pq} \leq x_{ij} \quad \forall [i, j] \in E, \forall p, q \in V \quad (6)$$

$$\sum_{[i,j] \in E} (y_{ij}^{pq} + y_{ji}^{pq}) \leq d_{pq} \quad \forall p, q \in V \quad (7)$$

$$D \geq d_{pq} \quad \forall p, q \in V \quad (8)$$

$$y_{ij}^{pq} \in \{0, 1\} \quad \forall [i, j] \in E, \forall p, q \in V, i \neq q, j \neq p \quad (9)$$

$$x_{ij} \in \{0, 1\} \quad \forall [i, j] \in E \quad (10)$$

$$d_{pq} \geq 1 \quad \forall p, q \in V \quad (11)$$

As duas funções objetivo para minimização do custo total e do diâmetro são apresentadas nas equações (1) e (2). A Equação (4) garante que a árvore geradora tenha no máximo  $|V| - 1$  arestas. As clássicas restrições de conservação de fluxo são definidas em (5). As inequações (6) especificam que nenhum fluxo passa através da aresta  $[i, j]$  se  $[i, j]$  não é escolhida como parte da solução, i.e. se  $x_{ij} = 0$ . As restrições (7) calculam a quantidade de arestas no caminho entre  $p$  e  $q$ . A minimização do diâmetro é resultado das restrições (2) e (8). O domínio das variáveis é definido de (9) a (11).

## 3 Algoritmo evolucionário multiobjetivo NSGA-II

O *fast Nondominated Sorting Genetic Algorithm* (NSGA-II) foi proposto por Deb *et al.* (2002) e é atualmente um dos métodos mais eficazes para resolução de problemas multiobjetivo, Zhou *et al.* (2011). Esta estratégia possui complexidade assintótica de pior caso  $O(MN^2)$ , onde  $M$  é o número de objetivos e  $N$  é o tamanho da população. A idéia é classificar as fronteiras utilizando dois procedimentos: o primeiro ordena as soluções não-dominadas (chamada de *ranking*)

e o segundo, estima a densidade de soluções no entorno de um indivíduo (chamado de *crowding distance*).

As fronteiras são camadas no espaço de soluções que limitam uma região de dominância. Na prática, diversos objetivos  $f_m(x)$ ,  $m = 1, \dots, M$ , onde  $x \in \mathbb{R}^n$  (vetores que definem soluções do sistema), podem ser considerados, ocupando diversas dimensões no espaço. Sem perda de generalidade, pode-se considerar que todos os objetivos são de minimização  $\min[f_1(x), f_2(x), \dots, f_m(x)]$ . A Figura 1 exemplifica fronteiras  $F_1, F_2, \dots, F_j$  no espaço de soluções para dois objetivos de minimização. Na prática, as fronteiras podem ser descontínuas e formar camadas irregulares. A camada mais importante é a que forma o envelope do espaço de soluções dos diversos objetivos, chamada de fronteira de Pareto. No exemplo da Figura 1 corresponde à fronteira  $F_1$ . As soluções Pareto-ótimas são aquelas que não se pode melhorar em nenhum dos objetivos e que formam o espaço de soluções não-dominadas.

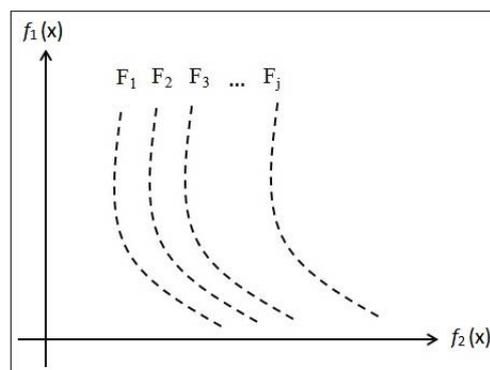


Figura 1: Camadas ilustrando fronteiras.

Uma solução  $x_i$  domina  $x_j$  se as seguintes condições são satisfeitas: (a)  $f_m(x_i) \leq f_m(x_j)$ , para todo  $m = 1, \dots, M$ , ou seja, não piora nenhum dos objetivos e (b)  $f_m(x_i) < f_m(x_j)$  para algum  $m = 1, \dots, M$ , ou seja, é extritamente melhor em pelo menos um objetivo.

O procedimento geral do NSGA-II é ilustrado na Figura 2, onde  $t$  refere-se a iteração corrente. Seja uma população  $P_t$  de tamanho  $N$  formada por indivíduos gerados aleatoriamente e ordenados usando o *ranking*. Os outros  $N$  indivíduos da população inicial  $Q_t$  são obtidos a partir de operadores genéticos, tais como cruzamento e mutação, utilizando indivíduos escolhidos aleatoriamente do conjunto  $P_t$ . A população completa  $R_t$  é dada por  $R_t = P_t \cup Q_t$ , onde  $|R_t| = 2N$ .

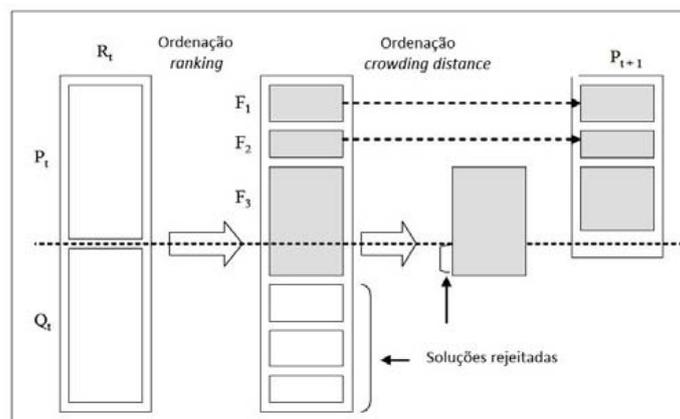


Figura 2: Procedimento geral do NSGA-II extraído de Deb *et al.* (2002).

Os procedimentos *ranking* e *crowding distance* são aplicados à população  $R_t$  para

classificar as fronteiras  $\{F_1, F_2, F_3, \dots, F_j\}$  utilizando o operador de *crowded*, detalhado abaixo. Os primeiros  $N$  elementos sobrevivem e compõem  $P_{(t+1)}$ . Operadores genéticos são aplicados, como mencionados acima, para gerar  $Q_{(t+1)}$  e o procedimento é repetido até satisfazer um critério de parada.

Dado um indivíduo  $j$ , a medida *ranking*  $r_j$  corresponde a quantidade de soluções que dominam  $j$  na iteração  $t$ . Logo, todos os indivíduos que possuem  $r_j = 0$  são armazenados em  $F_1$ ; os indivíduos com  $r_j = 1$  são armazenados em  $F_2$  e assim por diante. No final de uma iteração do NSGA-II, todos os indivíduos da população encontram-se classificados em uma fronteira. O objetivo do *crowding distance* é de definir a distância dos pontos na vizinhança do indivíduo  $j$ , na fronteira considerada. Assim sendo, quanto maior for o valor do *crowding distance*, mais distantes estão os pontos, indicando a necessidade de gerar mais soluções nesta região. Seja o vetor de *crowding distance*  $d$  de tamanho  $M$ . O cálculo do *crowding distance* é dado na Equação (12), onde  $suc(j)$  e  $pred(j)$  correspondem respectivamente aos valores do sucessor e predecessor do indivíduo  $j$  no vetor  $d$ .  $f_m^{max}$  e  $f_m^{min}$  são os valores máximos e mínimos da função objetivo  $m$ . Detalhes do algoritmo (inicialização do vetor  $d$ , tratamento do primeiro e último elemento de  $d$ , etc) são encontrados em Deb *et al.* (2002).

$$d_j = \sum_{m=1}^M \left( \frac{f_m^{suc(j)} - f_m^{pred(j)}}{f_m^{max} - f_m^{min}} \right) \quad (12)$$

Nas iterações do NSGA-II, 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 do *crowding distance*, a população é reordenada utilizando o operador *crowded* definido da seguinte forma: dados dois indivíduos  $i$  e  $j$ ,  $i$  domina  $j$  se  $(r_i < r_j)$  ou  $((r_i = r_j) \text{ e } (d_i > d_j))$ . A nova população é construída com os  $N$  primeiros indivíduos selecionados após o uso do operador *crowded*, formando assim o conjunto  $P_{(t+1)}$ . Os outros  $N$  indivíduos, pertencentes ao conjunto  $Q_{(t+1)}$ , são resultantes da aplicação de operados genéticos a indivíduos selecionados aleatoriamente no conjunto  $P_{(t+1)}$ .

### 3.1 NSGA-II aplicado ao bi-AGCDM

As funções objetivo para o bi-AGCDM minimizam o custo e o diâmetro, ver Equações (1) e (2). No NSGA-II desenvolvido para o bi-AGCDM, uma solução (indivíduo), para o bi-AGCDM é caracterizada por uma estrutura de árvore geradora, o valor total do custo da árvore e seu correspondente diâmetro. Existem diversas maneiras de representar (codificar) uma árvore em algoritmos genéticos. Alguns trabalhos avaliam a eficiência das diferentes representações em algoritmos genéticos, Raidl e Julstrom (2003a); Carrano *et al.* (2007). Neste trabalho, um indivíduo (solução) é representada através de uma lista de predecessores que refere-se ao nó predecessor na árvore. A quantidade de indivíduos geradas é  $N = |V|$ .

Na primeira população, duas soluções particulares são geradas e correspondem a melhor solução em termos de custo (AGM) e diâmetro (AG-DM). Para obter a AGM de  $G$ , basta calculá-la utilizando um algoritmo clássico como o de Prim. A AG-DM de  $G$  é realizada utilizando uma adaptação do algoritmo de busca em largura. Inicialmente, o diâmetro de  $G$  é determinado e um dos caminhos que determina o diâmetro é utilizado para construir uma AGM com o respectivo diâmetro. Os nós não pertencentes ao caminho são conectados ao primeiro nó encontrado no caminho que não aumenta o diâmetro. As  $N - 2$  soluções da população inicial que formam o conjunto  $P_t$  são geradas utilizando um algoritmo aleatorizado de Prim. Na versão original de Prim, a cada iteração, o nó selecionado é aquele que se conecta com o menor custo a um outro já pertence à solução. Na versão aleatorizada, um sorteio é feito para selecionar um dos possíveis nós que se conectam a algum outro da solução. A conexão não é necessariamente de menor custo.

Os indivíduos que fazem parte do conjunto  $Q_t$  são gerados utilizando o operador de cruzamento proposto por Carrano *et al.* (2007). O cruzamento consiste em construir uma nova árvore a partir de dois indivíduos selecionados aleatoriamente no conjunto elite  $P_t$ . Entretanto, a probabilidade de gerar árvores inviáveis é muito alta, em particular em grafos esparsos. Carrano *et al.* (2007) propôs realizar a união das duas árvores que consiste a construir um grafo de suporte contendo todas as arestas das duas árvores. Em seguida, calcular o novo indivíduo a partir do grafo de suporte. Por exemplo, dado duas soluções, como ilustrado na Figura 3-(a) e 3-(b), a união que forma o grafo de suporte é dado na Figura 3-(c). A nova árvore é obtida utilizando a versão aleatorizada do algoritmo de Prim mencionada acima.

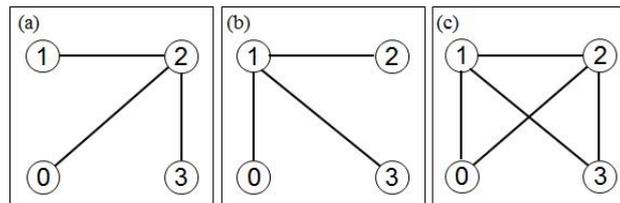


Figura 3: Exemplo de união de grafos.

O *ranking* e o *crowding distance* são calculados para a população inicial e classificados utilizando o operador *crowded*. A construção da população é realizada como mencionado anteriormente na Seção 3.

#### 4 Resultados preliminares

O NSGA-II foi desenvolvido utilizando a linguagem de programação C ANSI com o compilador DevCpp versão 4.9.9.2. Os testes foram realizados em uma máquina Intel core i3 com 2.1GHz e 4Gb de memória RAM. Quatro grupos de instâncias são utilizados nos experimentos, onde três  $L$ ,  $R$ ,  $G$  são respectivamente provenientes dos trabalhos de Lucena *et al.* (2010); Raidl e Julstrom (2003a); Gouveia e Magnanti (2003). O novo grupo de instâncias  $S$  contém grafos esparsos provenientes de Santos *et al.* (2012), cuja conectividade do grafo é respectivamente garantida pela construção de um ciclo Hamiltoniano. As outras arestas são selecionadas aleatoriamente e adicionadas de acordo com a densidade do grafo requerida  $\{0.2, 0.3, 0.4\}$ .

Três métricas de avaliação são empregadas nos testes e são calculadas a partir das soluções obtidas na última iteração do NSGA-II. A métrica  $Q$  indica a quantidade de indivíduos da fronteira  $F_1$ . A segunda métrica *Spacing*, referida neste trabalho por  $S$ , foi proposta por Veldhuizen e Lamont (2000) e é baseada no espalhamento das soluções. Para obter  $S$ , a média das distâncias Euclidianas entre todos os pontos vizinhos em  $F_1$  é calculada. Quanto menor o valor de  $S$ , melhor é  $S$  porque indica que a distribuição em  $F_1$  está mais uniformizada. Seja o ponto  $w$  definido como sendo a pior solução conhecida no espaço de busca. Para o bi-AGCDM pode-se considerar  $w$  como sendo a solução que tem o diâmetro da AGM e o custo da AG-DM. O hipervolume  $H$  fornece a área coberta pelas soluções não dominadas em relação ao ponto  $w$ , Zitzler *et al.* (2003). Quanto maior o hipervolume, melhor será a solução porque indica que a fronteira  $F_1$  está mais afastada de  $w$ .

Testes preliminares foram realizados com os quatro grupos de instâncias. O objetivo inicial é verificar a convergência das soluções produzidas pelo NSGA-II. Três execuções diferentes foram realizadas variando a quantidade de iterações em 100, 300 e 500. A semente fornecida para geração de números aleatórios nas três execuções são idênticas e iguais a 7.

Os resultados preliminares são apresentados na Tabela 1. Cada linha corresponde a uma instância e as colunas “Id”, “Nome”, “ $|V|$ ” e “ $|E|$ ” indicam respectivamente o identificador do grupo de instâncias, o nome da instância, a quantidade de vértices e a quantidade de arestas para cada instância. O custo ótimo e o diâmetro da AGM são exibidos respectivamente nas colunas  $C^*$  e

Tabela 1: Resultados preliminares do NSGA-II para o bi-AGCDM.

Id	Instâncias			AGM		AG-DM		100 iterações				300 iterações				500 iterações			
	Nome	V	E	C*	D	C	D*	Q	S	H	t(s)	Q	S	H	t(s)	Q	S	H	t(s)
G	se_v40_a400	40	400	450	19	1206	4	14	113,62	21,94	2,21	14	91,91	22,02	6,71	14	144,04	22,10	11,40
	se_v60_a600	60	600	638	33	1933	5	17	262,11	25,90	6,78	20	408,61	30,54	25,08	21	310,47	32,35	40,10
	sr_v40_a400	40	400	151	13	766	4	9	222,18	13,44	1,81	9	149,84	14,09	5,66	9	168,07	14,17	9,14
S	Hc_v25_e120	25	120	447	12	528	4	8	73,84	11,35	0,57	8	61,86	11,37	1,52	8	69,36	11,37	2,55
	Hc_v35_e47	35	47	1604	20	1677	10	7	18,63	7,02	1,87	7	20,73	7,87	6,16	7	20,04	7,87	9,95
	Hc_v40_e78	40	78	1313	19	1520	8	9	77,61	11,83	2,39	9	64,67	13,21	7,29	9	66,93	13,22	12,74
L	c_v15_a105	15	105	277	13	681	2	11	106,32	15,45	0,13	11	89,05	15,54	0,28	11	83,59	15,54	0,43
	c_v20_a190	20	190	271	15	629	2	13	101,98	19,59	0,27	13	99,12	19,62	0,67	13	105,63	19,71	1,24
	c_v25_a300	25	300	366	15	941	2	12	190,62	16,80	0,54	12	199,82	16,81	1,48	12	191,45	16,81	2,36
R	c_v70_d7_1	70	2.415	5,56	36	25,54	2	17	5,51	25,00	17,47	20	6,06	41,00	37,21	22	4,55	45,00	53,96
	c_v100_d10_1	100	4.950	6,61	45	37,47	2	14	14,20	27,00	22,26	28	11,65	55,00	89,89	30	11,60	59,00	169,30
	c_v250_d15_1	250	31.125	10,61	65	94,26	2	15	22,14	29,00	348,92	19	24,61	37,00	1180,34	19	24,12	37,00	2143,89

$D$ . Além disto, o diâmetro ótimo  $D^*$  e o custo  $C$  da AG-DM também são fornecidos. Em seguida, para cada execução com 100, 300 e 500 iterações, os valores obtidos com as métricas  $Q$ ,  $S$ ,  $H$  e os tempos de execução em segundos “t(s)” são dados.

Quando a quantidade de iterações aumenta, os resultados indicam que o hipervolume cresce devido a melhoria da fronteira  $F_1$ . Conseqüentemente, aumenta a distância de  $F_1$  do ponto  $w$ . A medida que a quantidade de iterações aumenta, a métrica  $Q$  permanece com valores similares ou senão aumenta, indicando que novos pontos foram descobertos em  $F_1$ .

A Figura 4 ilustra a evolução da fronteira  $F_1$  no final das três execuções para a instância  $R : c\_v100\_d10\_1$ . Para as outras instâncias um comportamento similar é obtido. Observa-se que com 100 iterações a qualidade de  $F_1$  é pior do que com 300 ou 500 iterações, o que é um comportamento esperado. Além disto, a passagem de 300 a 500 iterações permitiu de melhorar um pouco a fronteira  $F_1$ , em particular, de uniformizá-la. Mas, o ganho em termos de qualidade da fronteira  $F_1$  é gradativamente reduzido com o aumento do número de iterações.

As Figuras 5 e 6 fornecem toda a população considerando três iterações específicas do NSGA-II (a primeira, a do meio e a última). As instâncias utilizadas são  $G : se\_v60\_a600$ ,  $S : Hc\_v25\_e120$ ,  $L : c\_v25\_a300$ , e  $R : c\_v100\_d10\_1$ . Essas instâncias foram escolhidas porque representam um comportamento similar ao que ocorreu no seu respectivo grupo de instâncias. A população gerada na primeira iteração encontra-se concentrada em uma região do espaço de pouca qualidade. Isto ocorre porque  $N$  árvores são geradas aleatoriamente e as outras  $N$  são obtidas a partir de cruzamentos das árvores aleatórias. A convergência é realizada rapidamente. Na metade das iterações, já existem diversas soluções que encontram-se em  $F_1$  no final das iterações.

Em termos de tempos de processamento, o algoritmo é eficiente. Por exemplo, executa 500 iterações com as instâncias de 100 nós em grafos completos em menos de 1 minuto e as instâncias de grafos completos com 250 em aproximadamente 15 minutos.

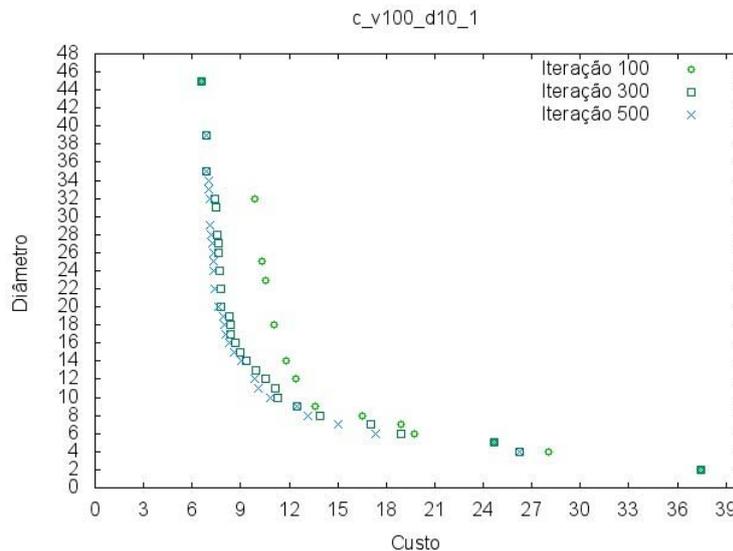


Figura 4: Evolução de  $F_1$  para a instância  $R : c\_v100\_d10\_1$ .

## 5 Conclusões e perspectivas

Neste trabalho, uma formulação multifluxo inspirada nos modelos propostos por Gouveia e Magnanti (2003) é apresentada. Esta formulação fornece uma formalização matemática para o bi-AGCDM e poderá guiar a construção de algoritmos exatos para o problema, objetivando conhecer completamente a fronteira de Pareto para instâncias de pequeno e médio porte.

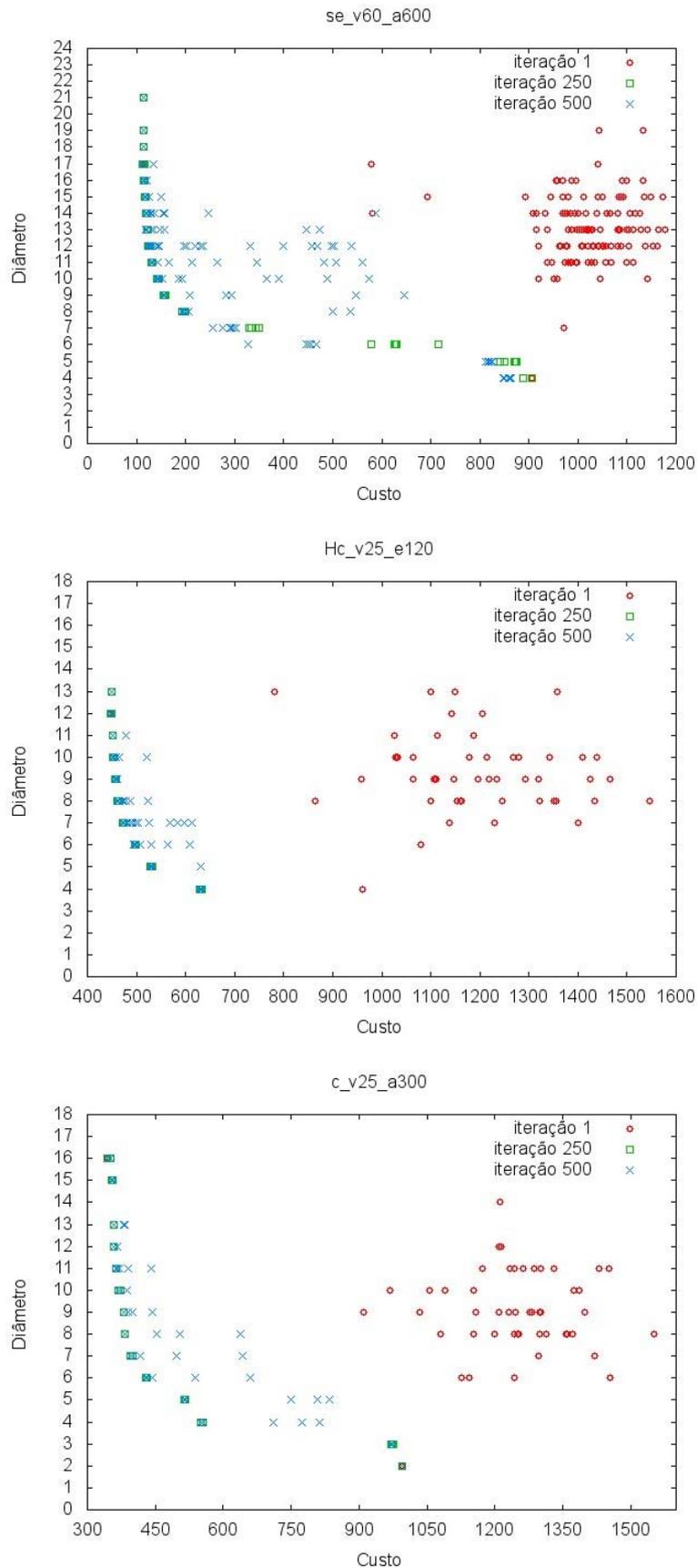


Figura 5: Evolução da população em diferentes iterações.

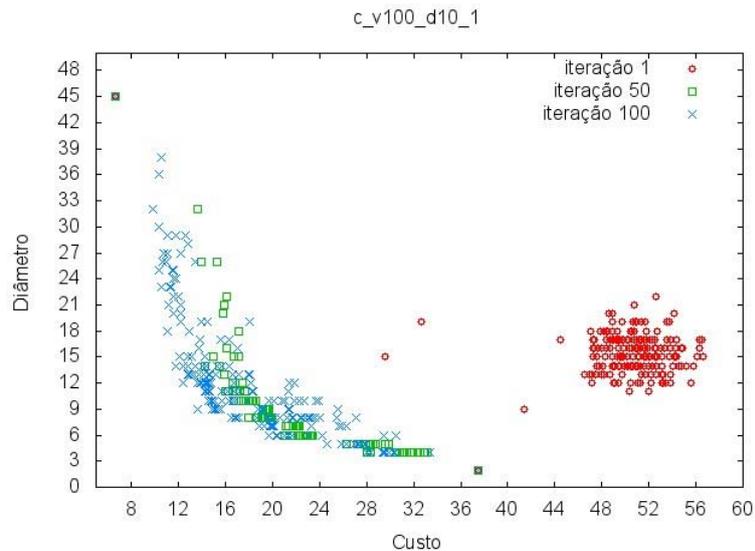


Figura 6: Evolução da população em diferentes iterações para instância  $R : c_v100_d10_1$ .

Nós propomos a utilização de um método de alto desempenho para problemas multiobjetivo. Em nosso conhecimento, o NSGA-II proposto neste trabalho é a primeira estratégia deste tipo empregada na literatura para o bi-AGCDM. O algoritmo proposto converge rapidamente ao longo das iterações autorizadas.

Estamos atualmente trabalhando na definição de outros operadores genéticos a serem integrados na construção da população e no uso de outras métricas de avaliação. Um outro ponto de estudo é a codificação de árvores possíveis de serem utilizadas em algoritmos genéticos. O objetivo é melhorar os tempos de execução nas instâncias de grande porte. Além disso, existem diversas perspectivas de trabalhos futuros, como o desenvolvimento de outros métodos multiobjetivo heurísticos e exatos. Em seguida, uma comparação entre as diferentes estratégias para o bi-AGCDM será realizada.

### Referências

- Bui, M., Butelle, F. e Lavault, C.** (2004), A distributed algorithm for constructing the minimum spanning tree problem. *Journal of Parallel and Distributed Computing Archive*, v. 64, p. 571–577.
- 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.
- Chen, H., Campbell, A. M. e Thomas, B. W.** (2008), Network design for time-constrained delivery. *Naval Research Logistics*, v. 55, p. 493–515.
- Cormen, T., Leiserson, C. e Rivest, R.** *Introduction to algorithms*. McGraw-Hill, New York, 1990.
- Deb, K., Pratap, A., Agarwal, S. e Meyarivan, T.** (2002), A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, v. 6, n. 2, p. 182 –197.
- Deo, N. e Abdalla, A.** (2000), Computing a diameter-constrained minimum spanning tree in parallel. *Lecture Notes in Computer Science*, v. 1767, p. 17–31.

- Gouveia, L. e 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.** 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, Tenerife, Espanha, 2005.
- Hassin, R. e Tamir, A.** (1995), On the minimum diameter spanning tree problem. *Information Processing Letters*, v. 53, n. 2, p. 109–111.
- Ho, J.-M., Lee, D. T., Chang, C.-H. e 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. e 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. e HuntIII, H. B.** (1998), Bicriteria network design problems. *Journal of Algorithms*, v. 28, n. 1, p. 142–171.
- Raidl, G. R. e Julstrom, B. A.** (2003a), Edge sets: an effective evolutionary coding of spanning trees. *IEEE Transactions on Evolutionary Computation*, v. 7, n. 3, p. 225 – 239.
- Raidl, G. R. e Julstrom, B. A.** Greedy heuristics and an evolutionary algorithm for the bounded-diameter minimum spanning tree problem. *Proceedings of the 18th ACM Symposium on Applied Computing*, p. 747–752, Melbourne, USA, 2003b.
- Saha, S. e 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., Lima, D. R. e Aloise, D. J.** Modeling the bi-objective diameter minimum spanning tree problem. *Proceedings of the Global Optimization Workshop (GOW)*, p. 125–128, Natal, Brasil, 2012.
- Santos, A. C., Lucena, A. e Ribeiro, C. C.** (2004), Solving diameter constrained minimum spanning tree problem in dense graphs. *Lecture Notes in Computer Science*, v. 3059, p. 458–467.
- Veldhuizen, D. A. V. e Lamont, G. B.** On measuring multiobjective evolutionary algorithm performance. *Proceedings of the Congress on Evolutionary Computation*, p. 204 –211, California, USA, 2000.
- Zhou, A., Qu, B.-Y., Li, H., Zhao, S.-Z., Suganthan, P. N. e Zhang, Q.** (2011), Multiobjective evolutionary algorithms: A survey of the state of the art. *Swarm and Evolutionary Computation*, v. 1, n. 1, p. 32 – 49.
- Zitzler, E., Thiele, L., Laumanns, M., Fonseca, C. e 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.