

# Métodos Heurísticos e Exatos para Busca de um Subgrafo-Solução Ótimo de um Grafo x-y

Uéverton dos Santos Souza<sup>1,2</sup>, Rian G. S. Pinheiro<sup>1,3</sup>, Ivan C. Martins<sup>1</sup>

<sup>1</sup>Instituto de Computação – Universidade Federal Fluminense  
Niterói – RJ – Brasil

<sup>2</sup>Departamento de Informática – Centro Federal de Educação Tecnológica Celso Suckow da Fonseca  
Rio de Janeiro – RJ – Brasil

<sup>3</sup>Universidade Federal Rural de Pernambuco  
Garanhuns – PE – Brasil

{usouza, rgpinheiro, imartins}@ic.uff.br

## RESUMO

Grafos x-y são uma generalização de grafos E/OU, e consistem de grafos direcionados e acíclicos contendo um único vértice fonte. Nestes grafos, os arcos representam relações de dependência entre os vértices onde todo vértice  $v_i$  possui um rótulo  $x_i$ - $y_i$  indicando que  $v_i$  depende de  $x_i$  dentre seus  $y_i$  vizinhos de saídas. Dado um grafo x-y arco-ponderado  $G$  um subgrafo solução de  $G$  consiste de um subgrafo  $H$  contendo o vértice fonte e satisfazendo a seguinte restrição: para cada vértice  $v_i$  em  $H$ , exatamente  $x_i$  dentre seus  $y_i$  arcos de saídas também pertencem a  $H$ . Neste trabalho apresentamos um algoritmo aproximativo, um algoritmo genético de chaves aleatórias viciadas (biased random-key genetic algorithm – BRKGA) e uma formulação matemática para o problema de encontrar um subgrafo solução de custo mínimo em um grafo x-y. Os resultados mostram que o algoritmo aproximativo possui uma razão de aproximação menor ou igual a  $m$ , a formulação matemática resolve satisfatoriamente as instâncias propostas e o algoritmo genético produz soluções próximas à solução ótima.

**PALAVRAS CHAVE.** Grafos x-y. Meta-heurísticas. Formulação Matemática.

**Área principal** OC - Otimização Combinatória.

## ABSTRACT

X-y graphs are a generalization of and/or graphs, and consist of acyclic directed graphs which contains a single source vertex. In these graphs the edges represent dependency relationships between the vertices where every vertex  $v_i$  has a label  $x_i$ - $y_i$  indicating that  $v_i$  depends on  $x_i$  of its  $y_i$  out-neighbors. Given an weighted-edge x-y graph  $G$ , a subgraph solution of  $G$  is a subgraph  $H$  containing the source vertex and satisfying the following rule: for each vertex  $v_i$  in  $H$  exactly  $x_i$  of its  $y_i$  edges also belong to  $H$ . This paper presents a approximative algorithm, a biased random-key genetic algorithm - BRKGA, and a mathematical formulation for the problem of finding a solution subgraph of minimum cost in an x-y graph. The results show that the approximative algorithm has approximate ratio at most  $m$ , the mathematical formulation satisfactorily resolve instances proposals and genetic algorithm produces solutions close to the optimal solution.

**KEYWORDS.** x-y Graph. Metaheuristics. Mathematical Formulation.

**Main area** OC - Combinatorial Optimization.

## 1 Introdução

Um grafo E/Ou é um grafo direcionado (digrafo)  $G$ , tal que todo vértice  $v \in V(G)$  possui um rótulo  $f(v) \in \{E, Ou\}$ . Neste grafo, os arcos representam relações de dependência entre os vértices: vértices do tipo  $E$  dependem estritamente de todos os seus vizinhos de saída (dependência conjuntiva), enquanto vértices do tipo  $Ou$  dependem apenas de um dos seus vizinhos de saída (dependência optativa ou disjuntiva).

Grafos E/Ou vem sendo utilizados em inteligência artificial desde o início da década de 1970, para representação de decomposição de problemas (Nilsson, 1971; Hansen & Zilberstein, 2001). Desde então, ao longo do tempo, novas aplicações para estes grafos vem surgindo, como: versionamento de software em engenharia de software (Corandi & Westfechtel, 1998), geração de padrões de cortes em placas em pesquisa operacional (Morabito & Pureza, 2010; Arenales & Vianna, 2006), representação de hipergrafos em teoria dos grafos (Gallo et al., 1993) e representação de *game trees* em teoria dos jogos (Schaeffer et al., 2001), entre outras.

Um subgrafo solução  $H$  de um grafo E/OU deve conter o vértice fonte e obedecer as seguintes regras: se um e-vértice (resp. ou-vértice) é incluído em  $H$  então todas (resp. uma) de seus arcos de saídas devem também ser incluídas em  $H$ .

Grafos x-y por sua vez, são uma generalização de grafos E/OU onde: todo vértice  $v_i$  de um grafo x-y possui um rótulo  $x_i-y_i$  representando que o vértice, depende de  $x_i$  dentre seus  $y_i$  vizinhos de saída, ou seja, se um vértice  $v_i$  é incluído em uma solução  $H$  então exatamente  $x_i$  de seus  $y_i$  arcos de saída devem também ser incluídas em  $H$ . Um grafo E/Ou pode ser facilmente transformado em um grafo x-y, basta rotular cada ou-vértice  $v_i$  com o rótulo  $1-y_i$  e cada e-vértice com o rótulo  $y_i-y_i$ , onde  $y_i$  é o número de vizinhos de saída de um vértice  $v_i$ .

Um subgrafo solução  $H$  de um grafo x-y deve conter o vértice fonte e obedecer a seguinte regra: para cada vértice  $v_i$  em  $H$ , exatamente  $x_i$  dentre seus  $y_i$  arcos de saídas devem também estar contidas em  $H$ . A Figura 1 ilustra um grafo x-y  $G$  e a Figura 2 ilustra o subgrafo-solução de custo mínimo de  $G$ .

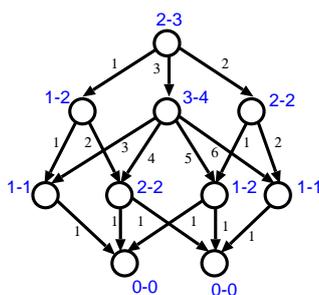


Figura 1: Um grafo x-y  $G$ .

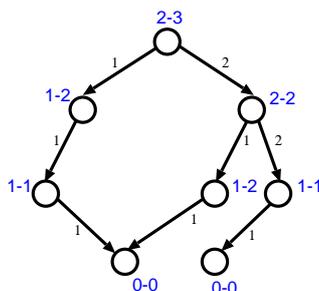


Figura 2: Subgrafo solução ótimo de custo mínimo do grafo x-y  $G$  apresentado na Figura 1.

Além das diversas aplicações dos grafos  $x$ - $y$ , estes possuem grande relevância em sistemas distribuídos pois modelam mecanismos clássicos para detecção de *deadlock* em sistemas distribuídos (Barbosa, 2002).

Em geral, a solução de um problema modelado por grafos  $x$ - $y$  pode ser representada por um subgrafo solução deste grafo, que satisfaz um determinado conjunto de restrições. O problema MIN-X-Y consiste basicamente em encontrar um subgrafo-solução de custo mínimo em grafos  $x$ - $y$ . A motivação de se estudar o problema MIN-X-Y vem do mesmo possuir várias e interessantes aplicações, além de não ser muito estudado na literatura especializada. As referências encontradas são poucas e, de uma forma geral, pouco recentes.

A seguir será apresentada a definição formal do problema MIN-X-Y.

---

MIN-X-Y

**Instância:** Um grafo  $x$ - $y$   $G = (V, E)$  com um vértice fonte  $s$ , onde cada arco possui um peso inteiro  $\tau_e > 0$ .

**Objetivo:** Determinar o peso mínimo do subdigrafo  $H = (V', E')$  de  $G$  que satisfaça as seguintes propriedades:

- $s \in V'$ ;
- para cada vértice  $v_i \in V'$ , exatamente  $x_i$  de seus  $y_i$  arcos de saída pertencem a  $E'$ .

---

Em 1974, Sahni (1974) mostrou através de uma redução do problema 3-SAT que é NP-difícil encontrar um subgrafo solução de um grafo E/OU e de um grafo  $x$ - $y$  (consequentemente). Em 2013, Souza et al. (2013, 2012) mostraram que determinar se existe em uma árvore  $x$ - $y$  uma subárvore solução de custo *exatamente*  $k$  é NP-completo, enquanto determinar a subárvore solução de custo mínimo de uma árvore  $x$ - $y$  pode ser feito em tempo polinomial. Além disso, é fácil ver que se todos os vértices  $v_i$  de um grafo  $x$ - $y$  possuem rótulo  $1-y_i$  (ou alternativamente rótulo  $y_i-y_i$ ) o problema MIN-X-Y pode ser solucionado em tempo polinomial. Sob o ponto de vista da complexidade parametrizada, Souza et al. (2013) mostraram que a versão parametrizada do problema MIN-X-Y (onde o custo da solução é o parâmetro) é W[1]-difícil.

Dada a dificuldade intrínseca de encontrar um subgrafo solução de custo mínimo em um grafo  $x$ - $y$ , neste trabalho são propostas uma heurística e uma meta-heurística para a busca de um subgrafo-solução ótimo de um grafo  $x$ - $y$ , além de uma formulação matemática para o problema.

O restante deste trabalho é dividido da seguinte forma: a Seção 2 apresenta um algoritmo aproximativo que encontra uma solução viável para o problema; na Seção 3 é proposto um algoritmo genético para o problema MIN-X-Y; a Seção 4 apresenta uma formulação matemática para o problema; na Seção 5 é feito um breve comentário sobre os experimentos computacionais e as ferramentas utilizadas e por fim, a Seção 6 apresenta as conclusões.

## 2 Algoritmo Aproximativo

Nesta seção será apresentado um algoritmo aproximativo para o problema MIN-X-Y. Dado uma ordenação topológica dos vértices de um grafo  $x$ - $y$   $G$ , o Algoritmo 1 consiste em selecionar a cada iteração o vértice que minimiza localmente o custo de seu subdigrafo-solução.

O conjunto  $S_{v_i}$  contém os arcos que formam o subgrafo solução enraizado em  $v_i$ . A função  $\tau(S)$  representa o somatório dos pesos dos arcos em  $S$ . Já o conjunto  $O_{v_i}$  representa os vizinhos de saída de  $v$ . No fim do Algoritmo 1,  $S_s$  contém os arcos que formam o subgrafo solução com o custo de  $\tau(S_s)$ .

---

**Algoritmo 1** Algoritmo Aproximativo

---

```
1: procedure HEURÍSTICA(Grafo  $G$ )
2:   Seja  $v_1, v_2, \dots, v_n$  uma ordenação topológica de  $G$ 
3:   for  $i = 1 \rightarrow n$  do
4:      $S_{v_i} \leftarrow \{\}$ 
5:   end for
6:   for  $i = n \rightarrow 1$  do
7:     if  $v_i$  não é sumidouro then
8:        $Aux \leftarrow O_{v_i}$ 
9:       for  $j = n \rightarrow x_i$  do
10:        Selecione o vértice  $w \in Aux$  que  $\min[\tau(S_w \setminus S_{v_i}) + \tau(v_i, w)]$ 
11:         $Aux \leftarrow Aux \setminus \{w\}$ 
12:         $S_{v_i} \leftarrow S_{v_i} \cup S_w \cup \{v_i, w\}$ 
13:      end for
14:    end if
15:  end for
16: end procedure
```

---

É fácil ver que o Algoritmo 1, de fato encontra uma solução viável para o problema MIN-X-Y pois cada vértice  $v_i$  em  $S_s$  satisfaz a restrição de possuir exatamente  $x_i$  arcos de saídas em  $S_s$ . Além disso, o Algoritmo 1 funciona de maneira *bottom-up* através de uma abordagem gulosa onde para cada vértice  $v_i$  pertencente a solução, será selecionado os  $x_i$  arcos de saída que minimizam localmente a função  $[\tau(S_w \setminus S_{v_i}) + \tau(v_i, w)]$ . Deste fato, segue que: sendo  $S^*$  o subgrafo solução ótimo do grafo  $G$  e  $S_s$  a solução viável obtida pelo Algoritmo 1 então

$$\tau(v_i, w) \leq \tau(S^*) \quad \forall (v_i, w) \in S_s.$$

Logo:  $\tau(S^*) \leq \tau(S_s) \leq \tau(S^*) \cdot m.$

Portanto o Algoritmo 1 possui uma fator de aproximação pelo menos  $m$  da solução ótima, mesmo para instâncias com arcos de pesos irrestritos.

### 3 BRKGA

O *Biased Random-Key Genetic Algorithm* (BRKGA) é uma meta-heurística eficiente para a resolução de problemas de otimização combinatória e contínua (Gonçalves & Resende, 2011). No BRKGA, os cromossomos são chamados de chaves aleatórias. Essas chaves consistem em vetores de valores reais no intervalo  $[0, 1)$ . O que difere entre cada problema é o módulo de decodificação das chaves aleatórias. A decodificação é um procedimento que mapeia um vetor de chaves aleatórias em uma solução do problema e calcula o custo desta solução (Gonçalves & Resende, 2011).

Para o Problema MIN-X-Y, a decodificação é feita da seguinte maneira: cada elemento  $c_i$  do vetor  $C$  codifica os arcos que saem do vértice  $v_i$ . O Algoritmo 2 é uma função de  $[0, 1)$  em  $\mathcal{C}_i$ , em que  $\mathcal{C}_i$  representa o conjunto das combinações de exatamente  $x_i$  arcos dentre os  $y_i$  que saem do vértice  $v_i$ . Em outras palavras, para cada gene o decodificador retorna um conjunto com  $x_i$  arcos.

A cada iteração do primeiro *for*, uma chave aleatória  $c_i$  é decodificada em um conjunto de arcos representado por  $E_{v_i}$ . As linhas 6 – 12, responsáveis pela decodificação, foram retiradas da função  $\text{KSUBSETREVDORUNRANK}(r, k, n)$  do livro de Kreher & Stinson (1999). Esta função tem como entrada os parâmetros  $r$ ,  $k$  e  $n$ , em que  $r = \ell - 1$  representa o  $\ell$ -ésimo conjunto de combinações;  $k$  indica quantas elementos o conjunto de saída deve conter e  $n$  o total de elementos.

---

**Algoritmo 2** Decodificador

---

```
1: procedure DECODIFICADOR(Vetor de Chaves_Aleatórias  $C$ )
2:   for  $c_i \in C$  do
3:      $q \leftarrow y_i$ 
4:      $k \leftarrow x_i$ 
5:      $r \leftarrow \lfloor c_i \binom{q}{k} \rfloor$ 
6:     for  $j \leftarrow k \rightarrow 1$  do
7:       while  $\binom{x}{j} > r$  do
8:          $q \leftarrow q - 1$ 
9:          $T_j \leftarrow q + 1$ 
10:         $r \leftarrow \binom{q+1}{j} - r - 1$ 
11:      end while
12:    end for
13:     $E_{v_i} \leftarrow T$ 
14:  end for
15:  return  $E$ 
16: end procedure
```

---

A Figura 3 mostra a saída da função para os parâmetros  $k = 3$ ,  $n = 5$  e diferentes valores de  $r$ .

$r = 0 \rightarrow 0 1 2$   
 $r = 1 \rightarrow 0 2 3$   
 $r = 2 \rightarrow 1 2 3$   
 $r = 3 \rightarrow 0 1 3$   
 $r = 4 \rightarrow 0 3 4$   
 $r = 5 \rightarrow 1 3 4$   
 $r = 6 \rightarrow 2 3 4$   
 $r = 7 \rightarrow 0 2 4$   
 $r = 8 \rightarrow 1 2 4$   
 $r = 9 \rightarrow 0 1 4$

Figura 3: Exemplo da função  $\text{KSUBSETREVDORUNRANK}(r, k = 3, n = 5)$

Note que, como o valor da chave aleatória  $c_i$  pertence ao intervalo  $[0, 1)$  e o valor do parâmetro  $r$  deve ser um inteiro entre 0 e  $\binom{y_i}{x_i} - 1$ ; o valor de  $r$  é calculado como  $\lfloor c_i \binom{y_i}{x_i} \rfloor$ . Dessa forma, cada um dos possíveis valores inteiros terá a mesma probabilidade ocorrer, assim, todas as possíveis combinações de exatamente  $x_i$  arcos terão a mesma probabilidade.

## 4 Formulação Matemática

Nesta seção será apresentada uma formulação matemática para o Problema MIN-X-Y. Os autores desconhecem formulações matemática para o problema. Neste caso, a seguinte formulação é proposta:

$$\min \sum_{ij \in E} \tau_{ij} a_{ij} \quad (1)$$

$$\text{s.a} \quad \sum_{j \in \delta(i)^+} a_{ij} = x_i p_i \quad \forall i \in V \quad (2)$$

$$\sum_{i \in \delta(j)^-} a_{ij} \leq y_j p_j \quad \forall j \in V \quad (3)$$

$$p_s = 1 \quad (4)$$

$$a_{ij} \in \mathbb{B} \quad \forall i, j \in V \quad (5)$$

$$p_i \in \mathbb{B} \quad \forall i \in V. \quad (6)$$

Nesta formulação:

- $\tau_{ij}$  é o custo dos arco  $ij$ ;
- $a_{ij}$  é uma variável binária que vale 1 se o arco  $ij$  pertence a solução e 0 caso contrário;
- $p_i$  é uma variável binária que vale 1 se o vértice  $i$  pertence a solução e 0 caso contrário;
- $x_i$  é o rótulo que indica quantos arcos devem sair do vértice  $i$  na solução;
- $y_i$  é o rótulo que indica quantos arcos saem do vértice  $i$  no grafo de entrada;
- $\delta(j)^-$  (resp.  $\delta(j)^+$ ) é o conjunto de arcos que entram (resp. saem) no vértice  $j$ .

A Função Objetivo do problema (1) calcula o custo dos arcos pertencentes à solução. As Restrições (2) obrigam que todo vértice  $i$  pertencente a solução deve ter  $x_i$  arcos de saída. As Restrições (3) que ativam todo vértice que possui um arco de entrada. O vértice fonte é incluído na solução pela Restrição (4). Por fim, (5) e (6) são restrições de integralidade.

## 5 Experimentos Computacionais

Nesta seção descreveremos as ferramentas utilizadas para a realização de testes computacionais, de forma a avaliar o método exato e as heurísticas descritas neste artigo. Entraremos em detalhes sobre a seleção dos parâmetros dos métodos e a geração das instâncias.

### 5.1 Geração de Instâncias

Como grafos x-y ainda são pouco explorados, não existem instâncias criadas na literatura. Dessa forma propõe-se a criação de instâncias — grafos aleatórios — utilizando uma adaptação do modelo  $\mathbb{G}(n, p)$ , também conhecido como modelo binomial (Gilbert, 1959), específica para digrafos acíclicos. Neste modelo, cada arco do grafo com  $n$  vértices terá probabilidade  $p$  de existir. Note que, este modelo permite a geração de instâncias sem arcos em seu vértice raiz. Nos testes computacionais, estes casos foram descartados.

Para escolher o rótulo de um vértice  $i$ , basta escolher de um evento de uma variável aleatória discreta que segue uma distribuição uniforme  $\mathcal{U}(0, |\delta(i)^+|)$ . Com relação aos pesos dos arcos, foram atribuídos pesos gerados uniformemente entre 1 e  $2n$ .

---

**Algoritmo 3** Adaptação do  $\mathbb{G}(n, p)$  de Gilbert para digrafos acíclicos

---

```
1: procedure GNP(int  $n$ , real  $p$ )
2:   Crie a matrix  $M_{n \times n}$ 
3:   for  $i = 0 \rightarrow n$  do
4:     for  $j = i \rightarrow n$  do
5:       if  $p > \text{Random}(0,1)$  then
6:          $M[i][j] \leftarrow 1$ 
7:       else
8:          $M[i][j] \leftarrow 0$ 
9:       end if
10:    end for
11:  end for
12:  return  $M$ 
13: end procedure
```

---

## 5.2 Ferramentas

O método exato baseada na formulação matemática apresentada na Seção 4 foi desenvolvido utilizando a ferramenta CPLEX 12.5, que é um dos pacotes de *software* de otimização linear mista mais utilizados na literatura. O algoritmo genético (BRKGA) foi desenvolvido utilizando o *framework* brkgaAPI desenvolvido por Toso & Resende (2011). Todos os métodos, exatos e heurísticos, foram desenvolvidos em C++ (gcc-4.9.0) e executados numa máquina Intel Core i7-4500U com 1,80 GHz e 8 Gb de memória RAM no sistema operacional Arch Linux 3.14.1

## 5.3 Seleção de Parâmetros

Para a realização dos testes do BRKGA, iniciamos com a fixação e seleção dos parâmetros. Na Tabela 1 relacionamos os parâmetros fixados de forma proporcional ao número de vértices  $n$  de cada instância. Na primeira coluna temos o nome, em seguida o valor ao qual será fixado e por último qual sua função no algoritmo.

Tabela 1: Parâmetros fixados pelo tamanho da instância

| Parâmetro | Valor         | Significado  |
|-----------|---------------|--|
| MAX_GEN   | $n \times 10$ | Número máximo de gerações                          |
| SIZE_P    | $n \times 10$ | Tamanho da população                               |
| X_INTVL   | $n$           | Troca as melhores soluções a cada X_INTVL gerações |

Na Tabela 2, temos uma relação dos parâmetros utilizados no algoritmo e seus possíveis valores. Esta segue o mesmo esquema que a Tabela 1, no entanto, para cada parâmetro especificado, os valores foram variados sobre um conjunto, especificado na coluna Valores. Uma bateria de testes foi realizada para decidir qual valor deve ser utilizado para cada parâmetro.

A bateria de teste consiste em executar o algoritmo utilizando todas as 576 combinações de parâmetros apresentadas na Tabela 2 para cada uma das 120 instâncias sem pesos consideradas neste trabalho. As 10 combinações de parâmetros que obtiveram a melhor taxa de sucesso (% de instâncias em que se alcançou uma solução ótima), são apresentados na Tabela 3. Para decidirmos qual a combinação de parâmetros melhor descreve o problema, além da taxa de sucesso, pontuamos cada combinação de acordo com a soma do número de vezes que cada parâmetro aparece nesta lista. Por fim, selecionamos aquela que obteve a maior taxa de sucesso, e a pontuação estabelecida como um critério de desempate.

Tabela 2: Conjunto de parâmetros testados

| Parâmetro | Valores              | Significado  |
|-----------|----------------------|--|
| $p_e$     | {10%, 15%, 20%, 25%} | Fração da população considerada elite                                  |
| $p_m$     | {2%, 5%, 10%, 15%}   | Fração da população a sofrer mutação                                   |
| $\rho$    | {50%, 60%, 70%}      | probabilidade de uma solução filha herdar o alelo de uma solução elite |
| $K$       | {1, 2, 3}            | Número de populações independentes                                     |
| $X$       | {2, 3, 4, 5}         | Troca as $X$ melhores soluções a cada geração                          |

Tabela 3: 10 melhores combinações de parâmetros testados

|    | combinação                                | succ % | pontuação                |
|----|---|--------|--------------------------|
| 1  | $p_e=10\%, p_m=10\%, \rho=50\%, K=3, X=3$ | 77,50  | $7 + 5 + 4 + 7 + 4 = 27$ |
| 2  | $p_e=10\%, p_m=15\%, \rho=60\%, K=3, X=4$ | 77,50  | $7 + 5 + 4 + 7 + 1 = 24$ |
| 3  | $p_e=25\%, p_m=15\%, \rho=60\%, K=3, X=3$ | 77,50  | $1 + 5 + 4 + 7 + 4 = 21$ |
| 4  | $p_e=10\%, p_m=10\%, \rho=50\%, K=3, X=5$ | 75,83  | $7 + 5 + 4 + 7 + 3 = 26$ |
| 5  | $p_e=10\%, p_m=15\%, \rho=60\%, K=3, X=5$ | 75,83  | $7 + 5 + 4 + 7 + 3 = 26$ |
| 6  | $p_e=10\%, p_m=15\%, \rho=60\%, K=3, X=2$ | 75,00  | $7 + 5 + 4 + 7 + 2 = 25$ |
| 7  | $p_e=10\%, p_m=15\%, \rho=70\%, K=3, X=3$ | 75,00  | $7 + 5 + 2 + 7 + 4 = 25$ |
| 8  | $p_e=10\%, p_m=10\%, \rho=70\%, K=2, X=2$ | 74,17  | $7 + 5 + 2 + 3 + 2 = 19$ |
| 9  | $p_e=15\%, p_m=10\%, \rho=50\%, K=2, X=3$ | 74,17  | $2 + 5 + 4 + 3 + 4 = 18$ |
| 10 | $p_e=15\%, p_m=10\%, \rho=50\%, K=2, X=5$ | 74,17  | $2 + 5 + 4 + 3 + 3 = 17$ |

## 5.4 Avaliação

Nessa seção são apresentados os resultados obtidos pelos métodos utilizados nesse artigo: método exato, algoritmo aproximativo e BRKGA. O método exato é definido pela formulação matemática definida na Seção 4 implementado na ferramenta CPLEX. Já o algoritmo aproximativo, foi implementado de acordo com o definido no Algoritmo 1. Temos por fim, o BRKGA definido no Algoritmo 2, e seguindo o critério definido e as avaliações realizadas na Seção 5.3, trabalhamos os parâmetros:  $p_e = 10\%$ ,  $p_m = 10\%$ ,  $\rho = 50\%$ ,  $K = 3$  e  $X = 3$ .

Para os testes, foram gerados aleatoriamente vários grupos de instâncias de acordo com o método apresentado no Algoritmo 3. Cada grupo de instâncias é definido por 3 fatores: número  $n \in \{10; 20; 50\}$  de vértices, presença ou não de pesos nos arcos, e probabilidade  $p \in \{0,3; 0,4; \dots; 1,0\}$  de existência de arcos. Cada resultado foi separado em 4 tabelas. As 3 primeiras apresentam os resultados obtidos pelas instâncias sem peso nos arcos, cada uma com um valor de  $n$  e composta por 5 instâncias para cada valor de  $p$ , totalizando 120 instâncias desse tipo. Por último, a Tabela 7 apresenta as instâncias com peso nos arcos, composta por uma instância para cada valor de  $p$  e  $n$ , totalizando 24 instâncias desse tipo.

Cada tabela apresenta o valor ótimo e o tempo obtido pelo método exato; o melhor resultado obtido, o *gap* desse resultado em relação ao ótimo e o tempo de execução do método aproximativo e do BRKGA. Para o método aproximativo, o tempo de execução ficou sempre abaixo de 0,01 s em todas as tabelas, e portanto foi omitida a coluna referente ao tempo de execução gasto pelo método.

A Tabela 4 exhibe os resultados dos métodos nas instâncias sem peso com  $n = 10$ . Pode-se observar que o método exato é muito rápido, com um tempo médio de execução em torno de 5 ms, praticamente desprezível assim como o algoritmo aproximativo. No entanto, o algoritmo aproximativo obteve o valor ótimo em 40 % das instâncias com um *gap* médio de 34,79 %, muito abaixo do intervalo de aproximação teórico. Já o BRKGA também apresenta um tempo baixo, porém muito maior que os demais, em média de 180 ms, apesar de obter o ótimo em 100 % das vezes.

Tabela 4: Avaliação do Exato, Heurística e BRKGA com  $n = 10$  sem pesos

| $p$  | Exato |           | Aproximativo |        | BRKGA |           |       |
|------|-------|-----------|--------------|--------|-------|-----------|-------|
|      | ótimo | tempo (s) | valor        | gap %  | valor | tempo (s) | gap % |
| 0,30 | 03    | 0,00      | 03           | 00     | 03    | 00,10     | 00,00 |
|      | 05    | 0,00      | 07           | 40     | 05    | 0,12      | 00,00 |
|      | 10    | 0,00      | 10           | 00     | 10    | 0,15      | 00,00 |
|      | 02    | 0,00      | 02           | 00     | 02    | 0,11      | 00,00 |
|      | 08    | 0,00      | 08           | 00     | 08    | 0,10      | 00,00 |
| 0,40 | 13    | 0,00      | 19           | 46,15  | 13    | 00,20     | 00,00 |
|      | 04    | 0,00      | 04           | 00     | 04    | 0,14      | 00,00 |
|      | 13    | 0,00      | 13           | 00     | 13    | 0,15      | 00,00 |
|      | 03    | 0,01      | 03           | 00     | 03    | 0,14      | 00,00 |
|      | 16    | 0,01      | 17           | 06,25  | 16    | 0,21      | 00,00 |
| 0,50 | 10    | 0,00      | 10           | 00     | 10    | 00,14     | 00,00 |
|      | 06    | 0,01      | 06           | 00     | 06    | 0,15      | 00,00 |
|      | 04    | 0,01      | 09           | 125    | 04    | 0,17      | 00,00 |
|      | 05    | 0,01      | 06           | 20     | 05    | 0,14      | 00,00 |
| 14   | 0,00  | 14        | 00           | 14     | 0,22  | 00,00     |       |
| 0,60 | 09    | 0,01      | 10           | 11,11  | 09    | 00,22     | 00,00 |
|      | 03    | 0,01      | 07           | 133,33 | 03    | 0,12      | 00,00 |
|      | 08    | 0,00      | 13           | 62,50  | 08    | 0,16      | 00,00 |
|      | 06    | 0,01      | 09           | 50     | 06    | 0,16      | 00,00 |
|      | 14    | 0,00      | 14           | 00     | 14    | 0,19      | 00,00 |
| 0,70 | 02    | 0,00      | 02           | 00     | 02    | 00,15     | 00,00 |
|      | 14    | 0,01      | 16           | 14,29  | 14    | 0,19      | 00,00 |
|      | 04    | 0,01      | 07           | 75     | 04    | 0,17      | 00,00 |
|      | 14    | 0,00      | 22           | 57,14  | 14    | 0,17      | 00,00 |
|      | 21    | 0,00      | 25           | 19,05  | 21    | 0,21      | 00,00 |
| 0,80 | 02    | 0,01      | 02           | 00     | 02    | 00,19     | 00,00 |
|      | 23    | 0,01      | 26           | 13,04  | 23    | 0,36      | 00,00 |
|      | 09    | 0,01      | 14           | 55,56  | 09    | 0,21      | 00,00 |
|      | 12    | 0,01      | 19           | 58,33  | 12    | 0,20      | 00,00 |
|      | 26    | 0,00      | 26           | 00     | 26    | 0,29      | 00,00 |
| 0,90 | 10    | 0,01      | 25           | 150    | 10    | 00,17     | 00,00 |
|      | 03    | 0,01      | 06           | 100    | 03    | 0,15      | 00,00 |
|      | 14    | 0,00      | 14           | 00     | 14    | 0,41      | 00,00 |
|      | 16    | 0,01      | 19           | 18,75  | 16    | 0,20      | 00,00 |
|      | 05    | 0,01      | 10           | 100    | 05    | 0,15      | 00,00 |
| 1,00 | 07    | 0,00      | 07           | 00     | 07    | 0,17      | 00,00 |
|      | 20    | 0,01      | 25           | 25     | 20    | 0,22      | 00,00 |
|      | 18    | 0,01      | 26           | 44,44  | 18    | 0,23      | 00,00 |
|      | 06    | 0,01      | 08           | 33,33  | 06    | 0,28      | 00,00 |
|      | 03    | 0,01      | 07           | 133,33 | 03    | 0,13      | 00,00 |
| -    | -     | 0,005     | 40,00 %      | 34,79  | 100 % | 0,18      | 00,00 |

Já na Tabela 5, exibe os resultados das instâncias sem peso e  $n = 20$ . De forma semelhante à Tabela 4, o tempo do método exato se manteve muito baixo, em torno de 13ms. No entanto, dessa vez o algoritmo aproximativo obteve o valor ótimo em apenas 7,5% das instâncias com um *gap* médio de 89,27%. O BRKGA manteve o tempo computacional mais alto, em torno de 1,62s em média, e alcançando o valor ótimo em 100% das instâncias.

Diferentemente das outras tabelas, na Tabela 6 há uma mudança no padrão dos resultados. Os testes realizados pelo método exato em algumas instâncias apresentou um tempo de processamento considerável, o que fez a média do tempo de execução subir para 63,75 s, superior que a do BRKGA de 48,05 s, principalmente nas instâncias com maior densidade de arcos (com  $p > 0,7$ ). No entanto, a qualidade dos resultados obtidos pelos métodos heurísticos sofreram uma piora, alcançando o valor ótimo em 0% e 40% das instâncias e gaps em torno de 101,78% e 11,82%, relativos respectivamente ao algoritmo aproximativo e o BRKGA.

Pelos resultados apresentados na Tabela 6, pode-se observar uma tendência de piora no tempo de execução do método exato em contrapartida a uma piora na qualidade da solução dos métodos heurísticos à medida que a instância torna-se maior ou mais difícil. Com o intuito de investigar o comportamento dos métodos apresentados em grafos com pesos, foi executada uma bateria de testes seguindo os mesmo moldes descritos para grafos sem pesos. A Tabela 7 descreve os resultados desse experimento, utilizando uma instância para cada combinação de  $n$  e  $p$ .

Pelo apresentado na Tabela 7, pode-se observar a mesma tendência apresentada nas instâncias sem pesos: à medida que o valor de  $n$  aumenta, o tempo de execução do método exato passou a se tornar maior que o método heurístico e uma piora na qualidade das soluções obtidas pelos métodos heurísticos.



Tabela 5: Avaliação do Exato, Heurística e BRKGA com  $n = 20$  sem pesos

| p    | Exato |           | Aproximativo |        | BRKGA    |           |       |
|------|-------|-----------|--------------|--------|----------|-----------|-------|
|      | ótimo | tempo (s) | valor        | gap %  | valor    | tempo (s) | gap % |
| 0,30 | 27    | 0,00      | 32           | 18,52  | 27       | 1,22      | 00,00 |
|      | 10    | 0,00      | 10           | 00     | 10       | 0,95      | 00,00 |
|      | 03    | 0,00      | 09           | 200    | 03       | 1,04      | 00,00 |
|      | 05    | 0,01      | 12           | 140    | 05       | 1,04      | 00,00 |
|      | 06    | 0,01      | 11           | 83,33  | 06       | 1,20      | 00,00 |
| 0,40 | 15    | 0,01      | 16           | 06,67  | 15       | 1,26      | 00,00 |
|      | 05    | 0,00      | 21           | 320    | 05       | 1,13      | 00,00 |
|      | 03    | 0,00      | 15           | 400    | 03       | 0,99      | 00,00 |
|      | 04    | 0,01      | 04           | 00     | 04       | 0,96      | 00,00 |
|      | 16    | 0,01      | 33           | 106,25 | 16       | 1,29      | 00,00 |
| 0,50 | 09    | 0,01      | 34           | 277,78 | 09       | 1,26      | 00,00 |
|      | 35    | 0,01      | 51           | 45,72  | 35       | 1,62      | 00,00 |
|      | 13    | 0,01      | 15           | 15,38  | 13       | 1,60      | 00,00 |
|      | 06    | 0,01      | 12           | 100    | 06       | 1,20      | 00,00 |
|      | 12    | 0,02      | 32           | 166,67 | 12       | 1,35      | 00,00 |
| 0,60 | 43    | 0,01      | 44           | 2,33   | 43       | 1,63      | 00,00 |
|      | 05    | 0,01      | 12           | 140    | 05       | 1,36      | 00,00 |
|      | 03    | 0,01      | 03           | 00     | 03       | 1,26      | 00,00 |
|      | 03    | 0,01      | 09           | 200    | 03       | 1,13      | 00,00 |
|      | 12    | 0,01      | 15           | 25     | 12       | 1,35      | 00,00 |
| 0,70 | 16    | 0,02      | 43           | 168,75 | 16       | 1,63      | 00,00 |
|      | 10    | 0,01      | 19           | 90     | 10       | 1,36      | 00,00 |
|      | 03    | 0,01      | 06           | 100    | 03       | 1,31      | 00,00 |
|      | 06    | 0,01      | 18           | 200    | 06       | 1,41      | 00,00 |
|      | 48    | 0,01      | 49           | 02,08  | 48       | 1,73      | 00,00 |
| 0,80 | 65    | 0,01      | 73           | 12,31  | 65       | 2,28      | 00,00 |
|      | 48    | 0,02      | 51           | 06,25  | 48       | 2,44      | 00,00 |
|      | 13    | 0,02      | 25           | 92,31  | 13       | 2,02      | 00,00 |
|      | 02    | 0,00      | 02           | 00     | 02       | 1,50      | 00,00 |
|      | 52    | 0,01      | 66           | 26,92  | 52       | 2,34      | 00,00 |
| 0,90 | 28    | 0,04      | 52           | 85,71  | 28       | 2,14      | 00,00 |
|      | 08    | 0,02      | 23           | 187,5  | 08       | 1,98      | 00,00 |
|      | 43    | 0,04      | 53           | 23,26  | 43       | 2,48      | 00,00 |
|      | 86    | 0,01      | 100          | 16,28  | 86       | 2,81      | 00,00 |
|      | 20    | 0,02      | 27           | 35     | 20       | 1,85      | 00,00 |
| 1,00 | 95    | 0,01      | 104          | 9,47   | 95       | 2,66      | 00,00 |
|      | 65    | 0,05      | 70           | 7,69   | 65       | 2,57      | 00,00 |
|      | 14    | 0,02      | 19           | 35,71  | 14       | 1,96      | 00,00 |
|      | 03    | 0,01      | 05           | 66,67  | 03       | 1,64      | 00,00 |
|      | 14    | 0,01      | 22           | 57,14  | 14       | 1,80      | 00,00 |
| -    | -     | 0,013     | 07,50 %      | 89,27  | 100,00 % | 1,62      | 00,00 |

Tabela 6: Avaliação do Exato, Heurística e BRKGA com  $n = 50$  sem pesos

| p    | Exato |           | Aproximativo |        | BRKGA   |           |       |
|------|-------|-----------|--------------|--------|---------|-----------|-------|
|      | ótimo | tempo (s) | valor        | gap %  | valor   | tempo (s) | gap % |
| 0,30 | 02    | 0,01      | 11           | 450    | 02      | 18,76     | 00    |
|      | 20    | 0,05      | 98           | 390    | 20      | 23,87     | 00    |
|      | 18    | 0,18      | 26           | 44,44  | 18      | 20,47     | 00    |
|      | 07    | 0,03      | 18           | 157,14 | 07      | 20,79     | 00    |
|      | 07    | 0,04      | 12           | 71,43  | 07      | 18,25     | 00    |
| 0,40 | 74    | 0,05      | 149          | 101,35 | 74      | 29,00     | 00    |
|      | 96    | 0,48      | 153          | 59,37  | 101     | 34,04     | 05,21 |
|      | 05    | 0,03      | 14           | 180    | 05      | 23,91     | 00    |
|      | 39    | 0,06      | 91           | 133,33 | 39      | 28,94     | 00    |
|      | 43    | 0,04      | 114          | 165,12 | 43      | 27,24     | 00    |
| 0,50 | 04    | 0,01      | 04           | 00     | 04      | 25,67     | 00    |
|      | 03    | 0,02      | 06           | 100    | 03      | 29,27     | 00    |
|      | 19    | 0,17      | 45           | 136,84 | 19      | 30,52     | 00    |
|      | 116   | 0,96      | 149          | 28,45  | 133     | 44,24     | 14,66 |
|      | 19    | 0,19      | 49           | 157,89 | 19      | 31,71     | 00    |
| 0,60 | 240   | 0,12      | 357          | 48,75  | 253     | 57,02     | 05,42 |
|      | 21    | 0,67      | 40           | 90,48  | 22      | 37,36     | 04,76 |
|      | 159   | 0,12      | 268          | 68,55  | 187     | 52,53     | 17,61 |
|      | 69    | 3,04      | 130          | 88,41  | 83      | 45,38     | 20,29 |
|      | 80    | 0,74      | 142          | 77,50  | 82      | 45,48     | 02,50 |
| 0,70 | 67    | 3,08      | 147          | 119,40 | 91      | 49,97     | 35,82 |
|      | 109   | 9,56      | 226          | 107,33 | 132     | 56,24     | 21,10 |
|      | 163   | 86,49     | 225          | 38,04  | 187     | 59,51     | 14,72 |
|      | 384   | 0,05      | 472          | 22,92  | 402     | 66,89     | 04,69 |
|      | 16    | 0,18      | 34           | 112,50 | 17      | 39,05     | 06,25 |
| 0,80 | 227   | 163,84    | 371          | 63,44  | 335     | 79,69     | 47,58 |
|      | 354   | 0,03      | 391          | 10,45  | 362     | 67,17     | 02,26 |
|      | 170   | 4,07      | 298          | 75,29  | 190     | 62,55     | 11,76 |
|      | 200   | 20,32     | 343          | 71,50  | 205     | 64,11     | 02,50 |
|      | 32    | 2,13      | 61           | 90,62  | 60      | 55,65     | 87,50 |
| 0,90 | 11    | 0,11      | 22           | 100    | 11      | 47,47     | 00,00 |
|      | 17    | 0,16      | 54           | 217,65 | 21      | 55,50     | 23,53 |
|      | 13    | 0,28      | 32           | 146,15 | 13      | 49,18     | 00    |
|      | 167   | 1972,72   | 222          | 32,93  | 217     | 70,95     | 29,94 |
|      | 248   | 16,56     | 454          | 83,06  | 328     | 81,97     | 32,26 |
| 1,00 | 02    | 0,01      | 02           | 0      | 02      | 46,40     | 00    |
|      | 401   | 131,80    | 477          | 18,95  | 437     | 92,46     | 08,98 |
|      | 338   | 58,74     | 428          | 26,63  | 425     | 93,20     | 25,74 |
|      | 369   | 17,21     | 559          | 51,49  | 420     | 86,91     | 13,82 |
|      | 77    | 58,55     | 103          | 33,77  | 103     | 83,25     | 33,77 |
| -    | -     | 63,75     | 00,00 %      | 101,78 | 40,00 % | 48,05     | 11,82 |

Tabela 7: Avaliação do Exato, Heurística e BRKGA em grafos com pesos

| n  | p    | Exato |          | Aproximativo |        | BRKGA   |          |       |
|----|------|-------|----------|--------------|--------|---------|----------|-------|
|    |      | opt   | time (s) | best         | gap %  | best    | time (s) | gap % |
| 10 | 0,30 | 29    | 0,01     | 29           | 0      | 29      | 0,10     | 0,00  |
|    | 0,40 | 85    | 0,00     | 85           | 0      | 85      | 0,11     | 0,00  |
|    | 0,50 | 83    | 0,01     | 116          | 39,76  | 83      | 0,15     | 0,00  |
|    | 0,60 | 14    | 0,00     | 14           | 0      | 14      | 0,14     | 0,00  |
|    | 0,70 | 185   | 0,00     | 185          | 0      | 185     | 0,20     | 0,00  |
|    | 0,80 | 65    | 0,01     | 107          | 64,62  | 65      | 0,15     | 0,00  |
|    | 0,90 | 353   | 0,00     | 353          | 0      | 353     | 0,25     | 0,00  |
|    | 1,00 | 22    | 0,01     | 22           | 0      | 22      | 0,15     | 0,00  |
| 20 | 0,30 | 237   | 0,00     | 237          | 0      | 237     | 0,98     | 0,00  |
|    | 0,40 | 19    | 0,00     | 59           | 210,53 | 19      | 0,94     | 0,00  |
|    | 0,50 | 442   | 0,01     | 1051         | 137,78 | 442     | 1,48     | 0,00  |
|    | 0,60 | 33    | 0,06     | 79           | 139,39 | 33      | 1,02     | 0,00  |
|    | 0,70 | 18    | 0,01     | 30           | 66,67  | 18      | 1,06     | 0,00  |
|    | 0,80 | 161   | 0,02     | 526          | 226,71 | 161     | 1,42     | 0,00  |
|    | 0,90 | 623   | 0,01     | 1527         | 145,10 | 633     | 2,19     | 1,61  |
|    | 1,00 | 1177  | 0,03     | 1623         | 37,89  | 1202    | 2,75     | 7,61  |
| 50 | 0,30 | 689   | 10,10    | 3334         | 383,89 | 689     | 20,60    | 0,00  |
|    | 0,40 | 6664  | 5,70     | 7865         | 18,02  | 6753    | 36,89    | 1,34  |
|    | 0,50 | 637   | 84,62    | 2417         | 279,43 | 637     | 29,71    | 0,00  |
|    | 0,60 | 349   | 26,05    | 1204         | 244,99 | 349     | 33,87    | 0,00  |
|    | 0,70 | 88    | 14,58    | 176          | 100,00 | 88      | 33,17    | 0,00  |
|    | 0,80 | 3343  | 188,90   | 9926         | 192,92 | 3431    | 64,04    | 2,63  |
|    | 0,90 | 4675  | 2344,26  | 7672         | 64,10  | 5195    | 76,63    | 11,12 |
|    | 1,00 | 13050 | 553,93   | 20924        | 60,34  | 13276   | 74,50    | 1,73  |
|    |      | -     | 134,51   | 29,17 %      | 100,51 | 75,00 % | 15,94    | 1,08  |

No geral, os resultados apresentados indicam um bom desempenho do método exato para instâncias com até 50 vértices, com pouca ou nenhuma variação de desempenho entre as diferentes densidades do grafo de entrada. Enquanto que com os métodos heurísticos, existe uma clara relação entre densidade e a dificuldade da instância. Além disso, algo menos óbvio foi a constatação de que o BRKGA obteve um desempenho melhor quando aplicado a grafos com peso nos arcos.

## 6 Conclusão

Neste trabalho, foi abordado o Problema MIN-X-Y. Este problema possibilita a modelagem de diversos problemas reais. Desta forma, foram propostos uma heurística construtiva que gera uma solução pelo menos  $m$ -aproximativa, um algoritmo genético (BRKGA) e uma formulação matemática. Além disso, os algoritmos desenvolvidos foram comparados com a formulação e os resultados foram próximos do método exato.

Como trabalhos futuros, sugere-se o estudo de buscas locais e algoritmos híbridos para esta versão do problema, além da derivação novas desigualdades válidas para a formulação proposta.

## Referências

- Arenales, M. N. & Vianna, A. C. G.** (2006), ‘O problema de corte de placas defeituosas’, *Pesquisa Operacional* **26**(2), 185–202.
- Barbosa, V. C.** (2002), The combinatorics of resource sharing, in M. Fialles & F. Gomes, eds, ‘Models for Parallel and Distributed Computation: Theory, Algorithmic Techniques and Applications’, Kluwer Academic Publishers, The Netherlands, pp. 27–50.
- Corandi, R. & Westfachtel, B.** (1998), ‘Version models for software configuration management’, *ACM Computing Surveys* **30**(2), 332–282.
- Gallo, G., Longo, G., Nguyen, S. & Pallottino, S.** (1993), ‘Directed hypergraphs and applications’, *Discrete Applied Mathematics* **42**, 177–201.
- Gilbert, E.** (1959), ‘Random graphs’, *Annals of Mathematical Statistics* **30**, 1141–1144.
- Gonçalves, J. F. & Resende, M. G. C.** (2011), ‘Biased random-key genetic algorithms for combinatorial optimization’, *Journal of Heuristics* **17**(5), 487–525.



- Hansen, E. & Zilberstein, S.** (2001), 'Lao\*: A heuristic search algorithm that finds solutions with loops', *Artificial Intelligence* **129**, 35–62.
- Kreher, D. L. & Stinson, D. R.** (1999), *Combinatorial algorithms : generation, enumeration, and search*, Boca Raton, Fla. ; London : CRC Press.
- Morabito, R. & Pureza, V.** (2010), 'A heuristic approach based on dynamic programming and and/or-graph search for the constrained two-dimensional guillotine cutting problem', *Annals of Operations Research* **179**, 297–315.
- Nilsson, N. J.** (1971), Problem-reduction representations, in R. F. Dojny & M. Eakins, eds, 'Problem Solving Methods in Artificial Intelligence', McGraw-Hill, United States of America, pp. 80–112.
- Sahni, S.** (1974), 'Computationally related problems', *Siam Journal on Computing* **3**(4), 262–279.
- Schaeffer, J., Plaat, A. & Junghanns, A.** (2001), 'Unifying single-agent and two-player search', *Information Sciences* **135**(3-4), 151–175.
- Souza, U. S., Protti, F. & Dantas da Silva, M.** (2012), 'Complexidade parametrizada para problemas em grafos e/ou', *Pesquisa Operacional para o Desenvolvimento*.
- Souza, U. S., Protti, F. & Dantas da Silva, M.** (2013), 'Revisiting the complexity of and/or graph solution', *Journal of Computer and System Sciences* **79**, 1156–1163.
- Toso, R. F. & Resende, M. G.** (2011), A C++ application programming interface for biased random-key genetic algorithms, Technical report, AT&T Labs.