

## Os Novos Algoritmos da Classe *ColorAnt-RT*: Uma Classe de Algoritmos para a Resolução do PCG

**Anderson Faustino da Silva**

Departamento de Informática – Universidade Estadual de Maringá  
Avenida Colombo, 5790 – Bloco C56 – 87.020-900 – Maringá – PR – Brazil  
anderson@din.uem.br

**Mauro Henrique Mulati**

Departamento de Ciência da Computação – Universidade Estadual do Centro-Oeste  
Campus Cedeteg – Rua Camargo Varela de Sá, 03 – Vila Carli – 85.040-080 –  
Guarapuava – PR – Brazil  
mhmulati@gmail.com

**Carla Négrei Lintzmayer\***

Instituto de Computação – Universidade de Campinas  
Avenida Albert Einstein, 1251 – Cidade Universitária – 13.083-852 – Campinas – SP –  
Brazil  
carla0negri@gmail.com

### RESUMO

O problema de coloração de grafo é  $\mathcal{NP}$ -difícil e é utilizado em aplicações práticas, como escalonamento de tarefas e alocação de registradores. Para obter soluções para este problema em tempo aceitável foi desenvolvida a classe de algoritmos denominada *ColorAnt-RT*, cujos algoritmos são baseados no comportamento de formigas durante a busca por alimento em um ambiente. Este artigo apresenta os novos algoritmos desta classe, a saber: *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT*. Os resultados demonstram que a obtenção de bons resultados está relacionada as estratégias utilizadas na implementação de tais algoritmos.

**PALAVRAS CHAVE.** Otimização por Colônia de Formigas, Problema de Coloração de Grafo, *ColorAnt-RT*, Metaheurísticas.

### ABSTRACT

The graph coloring problem is  $\mathcal{NP}$ -hard and it is used in many practical applications, such as task scheduling and register allocation. To obtain solutions for this problem in acceptable time was proposed a class of algorithms that was named *ColorAnt-RT*, which algorithms are based on ant's behavior during the search for food in an environment. This paper presents the new *ColorAnt-RT* algorithms, namely: *ColorAnt<sub>4</sub>-RT* and *ColorAnt<sub>5</sub>-RT*. The results demonstrate that to obtain good results it necessary to take care of the used strategies.

**KEYWORDS.** Ant Colony Optimization, Graph Coloring Problem, *ColorAnt-RT*, Metaheuristics.

---

\*Bolsista do CNPq - Brasil

## 1. Introdução

O campo de pesquisa da inteligência coletiva é inspirado no comportamento social de enxames, os quais são compostos por indivíduos que cooperam e se organizam sem a necessidade de um controle central. Neste contexto uma metaheurística explorada é a Otimização por Colônia de Formigas Artificiais, ou *Ant Colony Optimization* (ACO), que se embasa no comportamento apresentado por formigas durante a busca por alimento em um ambiente (Dorigo e Stützle, 2004). Diversos trabalhos propõem o uso de algoritmos ACO para a resolução de problemas, tais como: organização de sites (Lin e Tseng, 2010), processamento de sequências de DNA (Kurniawan et al., 2008), escalonamento (Yin e Xiang, 2012), rotas de robôs (Sariff e Buniyamin, 2010) e coloração de grafo (Lintzmayer et al., 2011)(Lintzmayer et al., 2011)(Lintzmayer et al., 2011), entre outros.

Especificamente, obter uma solução para o problema de coloração de grafo (PCG) consiste basicamente em encontrar uma quantidade  $k$  de cores que possam ser atribuídas aos vértices de forma que não existam vértices adjacentes com a mesma cor. Trivialmente, se um grafo  $G$  possui  $n$  vértices, então basta escolher  $k = n$  cores, porém, o objetivo é encontrar o valor mínimo de  $k$  que respeite a restrição do problema, denominado número cromático do grafo e denotado por  $\chi(G)$ . Encontrar o número cromático de um grafo é um problema  $\mathcal{NP}$ -difícil (Karp, 1972). Assim, a menos que  $\mathcal{P} = \mathcal{NP}$ , não existem algoritmos exatos em tempo polinomial que possam resolver grandes instâncias (Cormen et al., 2009), sendo necessárias técnicas alternativas para obter soluções satisfatórias. O caráter  $\mathcal{NP}$ -difícil do PCG tem levado à realização de trabalhos que exploram metaheurísticas e algoritmos heurísticos (Plumettaz et al., 2010; Galinier e Hao, 1999; Johnson e Trick, 1996).

A investigação de uma solução para o PCG ocasionou o desenvolvimento da classe de algoritmos *ColorAnt-RT*. Em uma investigação inicial foi proposto o algoritmo *ColorAnt<sub>1</sub>-RT* no qual além de cada formiga da colônia ser utilizada para atualizar a trilha de feromônio, a melhor formiga da colônia no ciclo ( $s'$ ) e a melhor formiga até o momento ( $s^*$ ) também são utilizadas para este fim (Lintzmayer et al., 2011). No segundo algoritmo proposto, *ColorAnt<sub>2</sub>-RT*, apenas  $s'$  e  $s^*$  são utilizadas para atualizar a trilha de feromônio (Lintzmayer et al., 2011). E por fim no terceiro algoritmo proposto, *ColorAnt<sub>3</sub>-RT*,  $s'$  e  $s^*$  não atualizam a trilha de feromônio simultaneamente. Neste último, inicialmente  $s'$  atualiza mais frequentemente do que  $s^*$  e ocorre uma gradual mudança na frequência que é feita baseada no número máximo de ciclos do algoritmo, em cada intervalo de ciclos a quantidade de ciclos na qual  $s^*$  irá atualizar a trilha de feromônio (ao invés de  $s'$ ) é incrementada em uma unidade (Lintzmayer et al., 2011). Entre estes três algoritmos, *ColorAnt<sub>3</sub>-RT* é o melhor algoritmo da classe *ColorAnt-RT*.

Este presente artigo descreve os novos algoritmos da classe *ColorAnt-RT*, *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT*. O primeiro tenta reutilizar cores com o objetivo de reduzir a quantidade de conflitos, enquanto o segundo ajusta automaticamente o valor de  $\alpha$ ,  $\beta$  e  $\rho$ . Portanto, ambos são duas variações independentes de *ColorAnt<sub>3</sub>-RT* e não uma variação gradual da estratégia anterior.

A investigação realizada com estes novos algoritmos considerou 15 instâncias de grafos utilizados para avaliar algoritmos ACO. E os resultados demonstraram que a estratégia utilizada na implementação de *ColorAnt<sub>5</sub>-RT* proporciona a este um desempenho superior a *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>3</sub>-RT*.

O restante deste artigo está organizado como segue. A Seção 2 apresenta alguns

trabalhos relacionados. A Seção 3 descreve o algoritmo *ColorAnt<sub>3</sub>-RT*. A Seção 4 descreve as modificações que foram realizadas em *ColorAnt<sub>3</sub>-RT* ocasionando o surgimento de *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT*. A Seção 5 apresenta os resultados obtidos pelos novos algoritmos, juntamente com uma discussão. E por fim, a Seção 6 apresenta as considerações finais.

## 2. Trabalhos Relacionados

No primeiro trabalho sobre coloração de grafos com colônia de formigas, o *ANTCOL* (Costa e Hertz, 1997), cada formiga tenta colorir o grafo com o menor valor de  $k$  possível, utilizando os métodos construtivos *RLF* (*Recursive Large First*) (Brélaz, 1979) e *Dsatur* (Leighton, 1979). A diferença entre este e os algoritmos da classe *ColorAnt-RT* está no uso da probabilidade, que envolve o feromônio e a informação heurística: no *ANTCOL* é utilizada para escolher um novo vértice a ser colorido, e em *ColorAnt-RT* é utilizada para escolher a cor que irá colorir um vértice.

Uma abordagem diferente trabalha com cada formiga se movendo em uma iteração com certa probabilidade, para um vértice adjacente que tenha o maior número de arestas conflitantes (Comellas e Ozón, 1998). Nesse vértice, a formiga substitui, também com certa probabilidade, a cor atual por uma nova que minimize os conflitos. O algoritmo utiliza a experiência dos eventos passados, porém não mantém uma matriz ou lista para tal. Essa abordagem não se assemelha com o que é realizado em *ColorAnt-RT*.

Em outro algoritmo para o  $k$ -PCG cada formiga é um procedimento iterativo que tenta minimizar o número de conflitos (Shawe-Taylor e Zerovnik, 2001). A trilha de feromônio é modelada com base em um grafo  $G'$ , inicialmente igual a  $G$ , ao qual vão sendo adicionadas arestas caso muitas formigas atribuíam diferentes cores a nós não-adjacentes. Também não é semelhante aos algoritmos *ColorAnt-RT*.

Um algoritmo mais recente, *ALS-COL* (*Ant Local Search*) (Plumettaz et al., 2010), é o único que compete com os melhores algoritmos de coloração de grafo. Nele, cada formiga é uma busca local derivada da busca tabu para o  $k$ -PCG. Este algoritmo é bem diferente de *ColorAnt-RT*.

## 3. O Algoritmo *ColorAnt<sub>3</sub>-RT*

*ColorAnt<sub>3</sub>-RT* é um algoritmo ACO para coloração de grafos que usa *React-Tabucol*, uma busca local baseada em Busca Tabu. Cada formiga em *ColorAnt<sub>3</sub>-RT* é um método construtivo chamado  $k$ -ANTCOL, que por sua vez é baseado no método ANTCOL (Costa e Hertz, 1997) modificado para colorir um grafo com  $k$  cores.

No  $k$ -ANTCOL, apresentado no Algoritmo 1, a cada passo de construção, deve-se escolher um vértice  $v$  (ainda não colorido) com o maior grau de saturação  $gsat(v)$ <sup>1</sup> e deve-se escolher uma cor  $c$  com probabilidade  $p$  para atribuir a  $v$ .

A probabilidade  $p$  é apresentada na Equação 1 e é calculada com base na trilha de feromônio  $\tau$ , apresentada na Equação 2, e na informação heurística  $\eta$ , apresentada na Equação 3.

<sup>1</sup>Grau de saturação é o número de cores diferentes que já foram atribuídas aos nós adjacentes de um vértice.

---

**Algoritmo 1** Pseudocódigo do  $k$ -ANTCOL.

---

$\kappa$ -ANTCOL( $G = (V, E), k$ ) //  $G$ : grafo;  $V$ : vértices;  $E$ : arestas  
1  $NC = V$ ; // vértices ainda não coloridos  
2  $cor_i = 0 \quad \forall i \in V$ ; // vetor  $cor$  mantém um mapeamento vértice-cor  
3  $ncoloridos = 0$ ;  
4 **while**  $ncoloridos < |V|$  **do**  
5  $v = \arg \max\{gsat(v') \mid v' \in NC\}$ ;  
6 escolher uma cor  $c \in \{1..k\}$  com probabilidade  $p(s, v, c)$  dada pela Equação 1;  
7  $cor_v = c$ ;  
8  $NC = NC \setminus \{v\}$ ;  
9  $ncoloridos++$ ;  
10 **return**  $cor$ ;

---

$$p(s, v, c) = \frac{\tau(s, v, c)^\alpha \cdot \eta(s, v, c)^\beta}{\sum_{i \in \{1..k\}} \tau(s, v, i)^\alpha \cdot \eta(s, v, i)^\beta} \quad (1)$$

onde  $\alpha$  e  $\beta$  são parâmetros do algoritmo que controlam a influência dos valores associados a eles na equação.

$$\tau(s, v, c) = \begin{cases} 1 & \text{se } C_c(s) = \{\} \\ \frac{\sum_{u \in C_c(s)} F_{uv}}{|C_c(s)|} & \text{caso contrário} \end{cases} \quad (2)$$

$$\eta(s, v, c) = \begin{cases} 1 & \text{se } N_{C_c(s)}(v) = \{\} \\ \frac{1}{|N_{C_c(s)}(v)|} & \text{caso contrário} \end{cases} \quad (3)$$

onde  $F_{uv}$  é a trilha de feromônio entre os vértices  $u$  e  $v$ ,  $C_c(s)$  é o conjunto de vértices já coloridos com a cor  $c$  na solução  $s$ , e  $N_{C_c(s)}(v)$  são os vértices  $x \in C_c(s)$  adjacentes a  $v$  na solução  $s$ .

A trilha de feromônio é armazenada na matriz  $F_{|V| \times |V|}$  e inicializada com 1 nas arestas de vértices não-adjacentes e 0 nas arestas de vértices adjacentes. Sua atualização envolve persistir a trilha atual por um fator  $\rho$  ( $1 - \rho$  é a taxa de evaporação), conforme a Equação 4, e reforçá-la por meio da experiência obtida nas soluções geradas, cuja forma geral é mostrada na Equação 5.

$$F_{uv} = \rho F_{uv} \quad \forall u, v \in V \quad (4)$$

$$F_{uv} = F_{uv} + \frac{1}{f(s)} \quad \forall u, v \in C_c(s) \mid (u, v) \notin E, c = 1..k \quad (5)$$

onde  $s$  é uma solução,  $C_c(s)$  é o conjunto de vértices coloridos com a cor  $c$  na solução  $s$  e  $f$  é a função objetivo e retorna o número de arestas conflitantes da solução.

Em relação ao tratamento da trilha de feromônio, *ColorAnt<sub>3</sub>-RT* utiliza a melhor formiga da iteração após a aplicação da busca local ( $s'$ ) e a melhor formiga encontrada durante toda a execução até o momento ( $s^*$ ), mas não simultaneamente. Inicialmente  $s'$  reforça a trilha mais frequentemente do que  $s^*$  e uma troca gradual nesta frequência é feita com base na quantidade máxima de ciclos do algoritmo.

Desde as versões iniciais (Lintzmayer et al., 2011)(Lintzmayer et al., 2011)(Lintzmayer et al., 2011), *ColorAnt<sub>3</sub>-RT* passou por algumas alterações em relação às regras de atualização do feromônio, permitindo melhores valores de  $k$  e principalmente diminuindo a quantidade de conflitos das soluções encontradas. *ColorAnt<sub>3</sub>-RT* é descrito no Algoritmo 2.

A busca *React-Tabucol* utilizada por *ColorAnt<sub>3</sub>-RT* funciona sobre um espaço de soluções  $S$  onde cada solução é formada por  $k$  classes de cores e todos os vértices estão coloridos. Um movimento é a troca da cor de um único vértice que ocorre entre soluções vizinhas e quando ocorre, seu inverso é armazenado em uma *lista tabu*, sendo proibido de ser realizado nas próximas  $t$  gerações. Partindo de uma solução inicial  $s_0 \in S$ , a busca gera uma sequência  $s_1, s_2, \dots$  de soluções em  $S$ , com  $s_{i+1}$  sendo vizinha de  $s_i$  a partir de um movimento não-tabu. Entre elas deve-se escolher a solução com o menor número de conflitos.

---

### Algoritmo 2 Pseudocódigo do *ColorAnt<sub>3</sub>-RT*.

---

COLORANT<sub>3</sub>-RT( $G = (V, E), k$ ) //  $G$ : grafo;  $V$ : vértices;  $E$ : arestas

```

1   $F_{uv} = 1 \forall (u, v) \notin E$ ;
2   $F_{uv} = 0 \forall (u, v) \in E$ ;
3   $f^* = \infty$ ; // melhor valor da função objetivo na execução (número de conflitos)
4  while  $ciclo < max\_ciclos$  and  $f^* \neq 0$ 
    and  $tempo < max\_tempo$  and  $converg < 4 \cdot \sqrt{max\_ciclos}$  do
5       $f' = \infty$ ; // melhor valor da função objetivo no ciclo
6      for  $a = 1$  to  $nformigas$  do
7           $s = K\text{-ANTCOL}(G, k)$ ;
8          REACT_TABUCOL( $s$ );
9          if  $f(s) < f'$  then
10              $s' = s$ ;
11              $f' = f(s')$ ;
12         if  $f' < f^*$  then
13              $s^* = s'$ ;
14              $f^* = f(s^*)$ ;
15              $converg = 0$ ;
16          $F_{uv} = \rho F_{uv} \forall u, v \in V$ ;
17         if  $ciclo \bmod \sqrt{max\_ciclos} == 0$  then
18              $fero\_cont = ciclo \div \sqrt{max\_ciclos}$ ;
19              $s = s'$ ;
20         if  $fero\_cont > 0$  then
21              $s = s^*$ ;
22          $F_{uv} = F_{uv} + \frac{1}{f(s)} \forall u, v \in C_c(s) \mid (u, v) \notin V, c = 1..k$ ;
23          $fero\_cont = fero\_cont - 1$ ;
24          $ciclo = ciclo + 1$ ;
25          $converg = converg + 1$ ;

```

---

## 4. Os Novos Algoritmos da Classe *ColorAnt-RT*

### 4.1. O Algoritmo *ColorAnt<sub>4</sub>-RT*

*ColorAnt<sub>4</sub>-RT* tem como objetivo tentar economizar cores. As três versões anteriores utilizam a mesma regra para o cálculo da trilha  $\tau$  a ser utilizada no cálculo da probabilidade na escolha da cor a ser utilizada, conforme apresentado nas Equações 1 e 2. Nesta última, de cálculo do  $\tau$ , caso a classe de cores  $C_c$  esteja vazia, ou seja, nenhum vértice do grafo foi colorido com a cor  $c$ , essa cor ganha uma chance maior de ser escolhida para colorir  $v$ . Desta forma, em *ColorAnt<sub>4</sub>-RT* houve uma modificação na forma de calcular o  $\tau$ , conforme apresentado na Equação 6.

Para o novo cálculo de  $\tau$  é necessário que seja verificado um terceiro caso, no qual nenhum vizinho de  $v$  foi colorido com  $c$ , apesar dessa cor já ter sido utilizada por

algum outro vértice (o caso em que  $C_c(s) \neq \{\} \wedge N_{C_c(s)}(v) = \{\}$ ). Neste caso, também é interessante a preferência pela escolha da cor  $c$ , já que ela não geraria nenhum conflito caso fosse atribuída ao vértice em questão, além disto não ocasionar a utilização de uma cor que ainda não havia sido utilizada anteriormente, justificando a criação da nova fórmula.

$$\tau(s, v, c) = \begin{cases} x & \text{se } C_c(s) \neq \{\} \wedge N_{C_c(s)}(v) = \{\} \\ y & \text{se } C_c(s) = \{\} \\ \frac{\sum_{u \in C_c(s)} F_{uv}}{|C_c(s)|} & \text{caso contrário} \end{cases} \quad (6)$$

onde  $(x; y)$  são valores a serem avaliados.

É importante observar que é desejável que  $x > y$ , caso no qual uma cor já utilizada seria escolhida com maior preferência. Desta forma, os experimentos foram conduzidos com  $x = 2$  e  $y = 1$ .

#### 4.2. O Algoritmo *ColorAnt<sub>5</sub>-RT*

Diferentemente de *ColorAnt<sub>4</sub>-RT*, *ColorAnt<sub>5</sub>-RT* tem como objetivo tentar tornar os parâmetros  $\alpha$ ,  $\beta$  e  $\rho$  *auto-adaptáveis* ao problema, considerando a variação da função objetivo, utilizando uma idéia similar à apresentada em (Mulati e Constantino, 2011).

Para a implementação desta nova versão houve a inserção de um novo parâmetro,  $\gamma$ , chamado *fator de ajuste*, que será relacionado aos dois parâmetros da seguinte forma:

$$\begin{aligned} \alpha &= \alpha_{base} \cdot \gamma \\ \beta &= \beta_{base} \cdot (1 - \gamma) \\ \rho &= \rho_{base} \cdot \gamma \end{aligned} \quad (7)$$

onde  $\alpha_{base}$ ,  $\beta_{base}$  e  $\rho_{base}$  são sempre os mesmos (são os valores iniciais passados ao algoritmo) e  $\alpha$ ,  $\beta$  e  $\rho$  são os parâmetros que serão utilizados no cálculo da probabilidade.

O fator de ajuste, por sua vez, também deverá ser ajustado, pois é ele que deve reagir à possível convergência da colônia:

- ao final de  $X$  ciclos sem melhora no valor da função objetivo de  $s^*$ ,  $\gamma$  deve diminuir em um fator  $\omega$  (ou seja,  $\gamma = (1 - \omega) \cdot \gamma$ ), causando então uma diminuição no valor de  $\alpha$  e  $\rho$  e um aumento no valor de  $\beta$ , permitindo assim uma maior exploração guiada pela informação heurística (diversificação);
- ao final de  $X$  ciclos com melhora ou mantendo o valor da função objetivo de  $s^*$ ,  $\gamma$  deve aumentar em um fator  $\omega$  (ou seja,  $\gamma = (1 + \omega) \cdot \gamma$ ), causando então um aumento no valor de  $\alpha$  e  $\rho$  e uma diminuição no valor de  $\beta$ , permitindo assim uma maior exploração guiada pela trilha de feromônio (intensificação).

Assim como em *ColorAnt<sub>4</sub>-RT* é desejável que certos valores sejam utilizados na inicialização de *ColorAnt<sub>5</sub>-RT*, a saber:  $\alpha_{base} = 10$ ,  $\beta_{base} = 10$  e  $\rho_{base} = 1$ ,  $\gamma = 0,5$ ,  $\omega = 0,2$  e  $iteracoes = 5$ . Estes são os valores utilizados nos experimentos realizados com *ColorAnt<sub>5</sub>-RT*.

### 5. Avaliação Experimental

*ColorAnt<sub>3</sub>-RT*, *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT* foram implementados na linguagem C, compilados com GCC 4.4.3 utilizando o nível de otimização O3. Todos foram executados em um computador Intel Xeon E5504 de 2.00 GHz (2 processadores com 4 núcleos

**Tabela 1. Características das instâncias dos grafos.**

| Grafo         | $(\chi/k^*)$ | Vértices | Arestas | Densidade |
|---------------|--------------|----------|---------|-----------|
| dsjc250.5     | (?/28)       | 250      | 15668   | 0,50      |
| dsjc250.9     | (?/72)       | 250      | 27897   | 0,90      |
| dsjc500.5     | (?/48)       | 500      | 62624   | 0,50      |
| dsjc500.9     | (?/126)      | 500      | 224874  | 0,90      |
| dsjr500.1     | (12/12)      | 500      | 3555    | 0,03      |
| dsjr500.1c    | (?/85)       | 500      | 121275  | 0,97      |
| dsjr500.5     | (122/122)    | 500      | 58862   | 0,47      |
| flat300_20_0  | (20/20)      | 300      | 21375   | 0,48      |
| flat300_26_0  | (26/26)      | 300      | 21633   | 0,48      |
| flat300_28_0  | (28/28)      | 300      | 21695   | 0,48      |
| flat1000_50_0 | (50/50)      | 1000     | 245000  | 0,49      |
| le450_15c     | (15/15)      | 450      | 16680   | 0,17      |
| le450_15d     | (15/15)      | 450      | 16750   | 0,17      |
| le450_25c     | (25/25)      | 450      | 17343   | 0,17      |
| le450_25d     | (25/25)      | 450      | 17425   | 0,17      |

cada), 24GB de memória RAM e sistema operacional Ubuntu 10.04.3 LTS com Kernel 2.6.32-37-server.

## 5.1. Metodologia

Os experimentos foram realizados em 15 grafos<sup>2</sup> do Desafio DIMACS (Culberson e Luo, 1995; Johnson e Trick, 1996), a saber:

- dsjc250.5, dsjc250.9, dsjc500.5, dsjc500.9: grafos aleatórios padrão dsjcn.d que possuem n vértices e d probabilidade de quaisquer dois vértices formarem uma aresta;
- dsjr500.1, dsjr500.1c e dsjr500.5: grafos aleatórios geométricos dsjrn.d gerados escolhendo-se aleatoriamente e uniformemente n pontos em um quadrado e configurando arestas entre pares de vértices com distância menor que d. Uma letra ‘c’ ao final do nome indica que o grafo é complemento do grafo geométrico correspondente;
- flat300\_20\_0, flat300\_26\_0, flat300\_28\_0 e flat1000\_50\_0: grafos flatn\_χ\_0 gerados pelo particionamento dos n vértices em χ classes (que têm quase o mesmo tamanho) e pela seleção de arestas entre vértices de diferentes classes (assim, esses grafos possuem n vértices e número cromático χ conhecido);
- le450\_15c, le450\_15d, le450\_25c e le450\_25d: grafos le450\_χl que possuem sempre 450 vértices e número cromático χ conhecido.

A Tabela 1 apresenta mais detalhes de cada instância utilizada na avaliação. Nesta tabela a segunda coluna apresenta o par  $\chi/k^*$  (com ‘?’ caso  $\chi$  não seja conhecido), onde  $k^*$  é o valor da melhor solução encontrada até o momento. A terceira, a quarta e a quinta colunas apresentam respectivamente: quantidade de vértices, quantidade de arestas e densidade.

Os algoritmos *ColorAnt<sub>3</sub>-RT* e *ColorAnt<sub>4</sub>-RT* utilizam a mesma estratégia de calibragem de parâmetros, que foi realizada utilizando o Algoritmo 3, onde o parâmetro  $x$  pode assumir os valores 3 ou 4. Se houver empate em configurações para retorno, o desempate é feito pelo tempo de execução apresentado pelas configurações. Portanto, neste caso é retornada a configuração com o menor tempo de execução. Cada configuração foi avaliada em 3 tentativas para o valor conhecido de  $k^*$ , indicando que o valor retornado é a medida entre 3 execuções.

<sup>2</sup>Disponível em <http://mat.gsia.cmu.edu/COLOR/instances.html>, acessado em janeiro de 2013.

---

**Algoritmo 3** Calibrar *ColorAnt<sub>3</sub>-RT* ou *ColorAnt<sub>4</sub>-RT*.

---

CALIBRAR( $x, G = (V, E), k^*$ ) //  $G$ : grafo;  $V$ : vértices;  $E$ : arestas

```

1 for  $x = 1$  to 9 do
2    $\alpha = x$ ;
3    $\beta = 10 - x$ ;
4    $\rho = \alpha/10$ ;
5   for ( $nfor = 10$ ;  $nfor \leq 1000$ ;  $nfor * = 10$ ) do
6     for ( $bl\_ciclos = 10$ ;  $bl\_ciclos \leq 1000000$ ;  $bl\_ciclos * = 10$ ) do
7       for tentativa = 1 to 3 do
8         COLORANT $_x$ -RT( $G, k^*, nfor, \alpha, \beta, \rho, bl\_ciclos$ );
9   Retornar a configuração com a menor quantidade de conflitos

```

---

**Tabela 2.** Parâmetros de *ColorAnt<sub>3</sub>-RT*, *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT*.

| Grafo         | <i>ColorAnt<sub>3</sub>-RT</i> |          |         |        |              | <i>ColorAnt<sub>4</sub>-RT</i> |          |         |        |              | <i>ColorAnt<sub>5</sub>-RT</i> |              |
|---------------|--------------------------------|----------|---------|--------|--------------|--------------------------------|----------|---------|--------|--------------|--------------------------------|--------------|
|               | $nfor$                         | $\alpha$ | $\beta$ | $\rho$ | $bl\_ciclos$ | $nfor$                         | $\alpha$ | $\beta$ | $\rho$ | $bl\_ciclos$ | $nfor$                         | $bl\_ciclos$ |
| dsjc250.5     | 10                             | 3        | 7       | 0,3    | 100000       | 10                             | 6        | 4       | 0,6    | 1000000      | 10                             | 1000000      |
| dsjc250.9     | 10                             | 6        | 4       | 0,6    | 10000        | 100                            | 2        | 8       | 0,2    | 100000       | 10                             | 100000       |
| dsjc500.5     | 10                             | 2        | 8       | 0,2    | 1000000      | 1000                           | 8        | 2       | 0,8    | 1000000      | 10                             | 1000000      |
| dsjc500.9     | 1000                           | 7        | 3       | 0,7    | 1000000      | 1000                           | 3        | 7       | 0,3    | 1000000      | 1000                           | 1000000      |
| dsjr500.1     | 10                             | 1        | 9       | 0,1    | 100          | 10                             | 7        | 3       | 0,7    | 1000000      | 10                             | 1000         |
| dsjr500.1c    | 10                             | 5        | 5       | 0,5    | 100000       | 10                             | 1        | 9       | 0,1    | 100000       | 100                            | 10000        |
| dsjr500.5     | 100                            | 1        | 9       | 0,1    | 1000         | 10                             | 8        | 2       | 0,8    | 100000       | 1000                           | 100          |
| flat300_20_0  | 10                             | 1        | 9       | 0,1    | 1000         | 10                             | 8        | 2       | 0,8    | 10000        | 10                             | 10000        |
| flat300_26_0  | 10                             | 2        | 8       | 0,2    | 1000000      | 10                             | 8        | 2       | 0,8    | 1000000      | 10                             | 1000000      |
| flat300_28_0  | 10                             | 9        | 1       | 0,9    | 1000000      | 10                             | 1        | 9       | 0,1    | 1000000      | 10                             | 1000000      |
| flat1000_50_0 | 1000                           | 1        | 9       | 0,1    | 1000000      | 100                            | 2        | 8       | 0,2    | 1000000      | 1000                           | 1000000      |
| le450_15c     | 10                             | 8        | 2       | 0,8    | 1000000      | 10                             | 8        | 2       | 0,8    | 1000000      | 1000                           | 100000       |
| le450_15d     | 10                             | 1        | 9       | 0,1    | 10000        | 10                             | 6        | 4       | 0,6    | 1000000      | 100                            | 10000        |
| le450_25c     | 1000                           | 8        | 2       | 0,8    | 10000        | 10                             | 4        | 6       | 0,4    | 1000000      | 1000                           | 1000000      |
| le450_25d     | 1000                           | 5        | 5       | 0,5    | 1000000      | 10                             | 8        | 2       | 0,8    | 1000000      | 100                            | 100000       |

*ColorAnt<sub>5</sub>-RT* usa uma estratégia diferente, pois não é necessário realizar calibragem para os parâmetros  $\alpha$ ,  $\beta$  e  $\rho$ , pois os mesmos são adaptativos, i.e., o algoritmo os altera no decorrer de sua execução. Sua calibragem é feita em procedimento semelhante ao apresentado no Algoritmo 3, porém sem o parâmetro  $x$ , considerando-se apenas o trecho das Linhas 5 a 9 e com substituição da Linha 8 por “COLORANT<sub>5</sub>-RT( $G, k^*, nfor, bl\_ciclos$ );”.

A Tabela 2 apresenta os parâmetros selecionados pelo processo de calibragem, separados por algoritmo.

Os experimentos com cada instância foram realizadas iniciando a quantidade de cores com o menor valor dentre  $\chi/k^*$ , de modo que a execução é repetida até que se tenha 10 sucessos<sup>3</sup> em 10 tentativas. Se nenhum sucesso for obtido, a execução não é apresentada na Tabela 3, a qual contém os resultados. Além disto os algoritmos possuem um tempo máximo de duração de 1 hora e quantidade máximo de 841 ciclos.

## 5.2. Resultados e Discussão

Na Tabela 3 são apresentados os resultados das execuções dos algoritmos *ColorAnt<sub>3</sub>-RT*, *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT*, utilizando as instâncias contidas na Tabela 1 com os parâmetros mostrados na Tabela 2. Os resultados na Tabela 3 contém os grafos com informações de suas colorações ótimas ou melhores conhecidas, seguidas do valor  $k$  utilizado nas tentativas de cada linha. Na sequência, há os resultados por algoritmo, onde cada um deles apresentam os dados: S/10, que indica a quantidade de sucessos ocorrida a

<sup>3</sup> Sucesso ocorre quando algoritmo encontra coloração com a quantidade de cores  $k$  proposta na tentativa.

cada 10 tentativas (que compõem uma execução); média de tempo, em segundos, das tentativas em uma execução (M.T(s)); média de ciclos, das tentativas de uma execução, para alcançar a primeira melhor solução (M.CPM); média total de ciclos executadas por cada tentativa de uma execução (M.TC); e média de conflitos das tentativas de uma execução (M.Cfs).

As instâncias estão agrupadas de acordo com suas classes, a saber: **dsjc**, **dsjr**, **flat** e **le450**. Há média dos valores M.T(s) e M.Cfs das instâncias de cada classe.

**Tabela 3. Resultados obtidos por ColorAnt<sub>3</sub>-RT, ColorAnt<sub>4</sub>-RT e ColorAnt<sub>5</sub>-RT.**

| Grafo<br>( $\chi/k^*$ )  | k   | ColorAnt <sub>3</sub> -RT |                  |           |          |            |          | ColorAnt <sub>4</sub> -RT |           |          |            |          | ColorAnt <sub>5</sub> -RT |           |          |            |  |
|--------------------------|-----|---------------------------|------------------|-----------|----------|------------|----------|---------------------------|-----------|----------|------------|----------|---------------------------|-----------|----------|------------|--|
|                          |     | S/<br>10                  | M.<br>T(s)       | M.<br>CPM | M.<br>TC | M.<br>Cfs  | S/<br>10 | M.<br>T(s)                | M.<br>CPM | M.<br>TC | M.<br>Cfs  | S/<br>10 | M.<br>T(s)                | M.<br>CPM | M.<br>TC | M.<br>Cfs  |  |
| dsjc250.5<br>(?/28)      | 28  | 10                        | 60,434           | 21,1      | 21,1     | 0          | 10       | 38,115                    | 0,5       | 1,5      | 0          | 10       | 42,028                    | 1,6       | 1,6      | 0          |  |
| dsjc250.9<br>(?/72)      | 72  | 3                         | 59,973           | 40,3      | 123,6    | 1,7        | 8        | 1828,748                  | 22,1      | 34,4     | 0,4        | 8        | 233,475                   | 29,5      | 53,3     | 0,4        |  |
|                          | 73  | 10                        | 4,115            | 8,2       | 8,2      | 0          | 10       | 43,870                    | 1,0       | 1,0      | 0          | 10       | 7,950                     | 1,9       | 1,9      | 0          |  |
| dsjc500.5<br>(?/48)      | 49  | 9                         | 1453,398         | 20,3      | 23,9     | 0,4        | 5        | 6507,481                  | 1,0       | 1,0      | 1,6        | 8        | 1785,314                  | 21,0      | 27,8     | 0,7        |  |
|                          | 50  | 10                        | 46,434           | 1,0       | 1,0      | 0          | 10       | 4721,790                  | 1,0       | 1,0      | 0          | 10       | 56,159                    | 1,1       | 1,1      | 0          |  |
| dsjc500.9<br>(?/126)     | 128 | 6                         | 14816,141        | 1,0       | 1,0      | 1,0        | 4        | 15007,497                 | 1,0       | 1,0      | 1,6        | 8        | 9107,522                  | 1,0       | 1,0      | 0,3        |  |
|                          | 129 | 10                        | 13776,089        | 1,0       | 1,0      | 0          | 10       | 13985,468                 | 1,0       | 1,0      | 0          | 10       | 898,295                   | 1,0       | 1,0      | 0          |  |
| <b>dsjc</b>              |     |                           | <b>4097,487</b>  |           |          | <b>0,8</b> |          | <b>5845,460</b>           |           |          | <b>0,9</b> |          | <b>2792,085</b>           |           |          | <b>0,4</b> |  |
| dsjr500.1<br>(12/12)     | 12  | 10                        | 0,040            | 1,0       | 1,0      | 0          | 10       | 0,106                     | 1,0       | 1,0      | 0          | 10       | 0,050                     | 1,0       | 1,0      | 0          |  |
| dsjr500.1c<br>(?/85)     | 85  | 10                        | 46,892           | 9,2       | 9,2      | 0          | 7        | 513,787                   | 38,4      | 75,1     | 0,5        | 8        | 246,976                   | 15,1      | 38,9     | 0,2        |  |
|                          | 86  | 10                        | 54,798           | 7,8       | 8,8      | 0          | 10       | 12,230                    | 1,6       | 1,6      | 0          | 10       | 1,6                       | 1,6       | 0        |            |  |
| dsjr500.5<br>(?/122)     | 124 | 1                         | 251,050          | 11,2      | 118,3    | 1,5        | -        | -                         | -         | -        | -          | 1        | 2414,744                  | 15,2      | 119,7    | 1,0        |  |
|                          | 125 | 6                         | 113,277          | 6,2       | 53,8     | 0,5        | -        | -                         | -         | -        | -          | 5        | 1355,327                  | 7,2       | 66,7     | 0,6        |  |
|                          | 126 | 10                        | 5,285            | 2,3       | 2,3      | 0          | -        | -                         | -         | -        | -          | 8        | 581,893                   | 4,9       | 28,7     | 0,2        |  |
|                          | 127 | -                         | -                | -         | -        | -          | -        | -                         | -         | -        | -          | 9        | 291,838                   | 1,9       | 13,8     | 0,1        |  |
|                          | 128 | -                         | -                | -         | -        | -          | -        | -                         | -         | -        | -          | 10       | 37,561                    | 1,7       | 1,7      | 0          |  |
|                          | 129 | 9                         | 1232,139         | 6,6       | 8,1      | 0,3        | 10       | 178,034                   | 0,4       | 1,4      | 0          | -        | -                         | -         | -        | -          |  |
|                          | 130 | 10                        | 178,034          | 0,4       | 1,4      | 0          | -        | -                         | -         | -        | -          | -        | -                         | -         | -        | -          |  |
| <b>dsjr</b>              |     |                           | <b>99,327</b>    |           |          | <b>0,5</b> |          | <b>582,011</b>            |           |          | <b>0,3</b> |          | <b>887,257</b>            |           |          | <b>0,4</b> |  |
| flat300_20_0<br>(20/20)  | 20  | 10                        | 0,960            | 4,7       | 4,7      | 0          | 10       | 1,164                     | 1,0       | 1,0      | 0          | 10       | 0,157                     | 1,0       | 1,0      | 0          |  |
| flat300_26_0<br>(26/26)  | 26  | 10                        | 228,692          | 2,3       | 2,3      | 0          | 10       | 252,129                   | 1,4       | 2,4      | 0          | 10       | 282,905                   | 3,2       | 3,2      | 0          |  |
| flat300_28_0<br>(28/28)  | 30  | 1                         | 3422,375         | 32,1      | 91,3     | 4,8        | -        | -                         | -         | -        | -          | 1        | 3289,616                  | 27,3      | 86,5     | 4,8        |  |
|                          | 31  | 10                        | 131,219          | 4,5       | 4,5      | 0          | 10       | 114,556                   | 2,7       | 3,7      | 0          | 10       | 94,431                    | 3,5       | 3,5      | 0          |  |
| flat1000_50_0<br>(50/50) | 50  | 10                        | 148116,972       | 1         | 1        | 0          | 5        | 14773,218                 | 1,0       | 1,0      | 15,2       | 10       | 34555,589                 | 1,0       | 1,0      | 0          |  |
|                          | 51  | 9                         | 15598,448        | 1,0       | 1,0      | 2,8        | 9        | 14932,006                 | 1,0       | 1,0      | 3,4        | -        | -                         | -         | -        |            |  |
|                          | 52  | 9                         | 14932,006        | 1,0       | 1,0      | 3,4        | 10       | 15021,714                 | 1,0       | 1,0      | 0          | -        | -                         | -         | -        |            |  |
|                          | 53  | 10                        | 15021,714        | 1,0       | 1,0      | 0          | -        | -                         | -         | -        | -          | -        | -                         | -         | -        |            |  |
| <b>flat</b>              |     |                           | <b>37942,250</b> |           |          | <b>1,2</b> |          | <b>3785,267</b>           |           |          | <b>3,8</b> |          | <b>9532,067</b>           |           |          | <b>1,2</b> |  |
| le450_15c<br>(15/15)     | 15  | 4                         | 889,641          | 4,3       | 75,7     | 1,5        | 10       | 759,969                   | 14,0      | 15,0     | 0          | 9        | 2733,762                  | 2,6       | 2,6      | 0,3        |  |
|                          | 16  | 10                        | 34,194           | 1,0       | 1,0      | 0          | 10       | 36,639                    | 1,0       | 1,0      | 0          | 10       | 36,639                    | 1,0       | 1,0      | 0          |  |
| le450_15d<br>(15/15)     | 15  | 8                         | 23,835           | 32,7      | 56,5     | 1,0        | 7        | 1963,835                  | 15,4      | 28,9     | 0,8        | 10       | 168,384                   | 20,9      | 20,9     | 0          |  |
|                          | 16  | 10                        | 9,501            | 7,8       | 7,8      | 0          | 10       | 39,968                    | 1,0       | 1,0      | 0          | -        | -                         | -         | -        |            |  |
| le450_25c<br>(25/25)     | 26  | 3                         | 3162,489         | 43,6      | 74,5     | 3,3        | 10       | 29,774                    | 0,2       | 1,2      | 0          | 10       | 6,569                     | 1,0       | 1,0      | 0          |  |
|                          | 27  | 10                        | 39,824           | 1,0       | 1,0      | 0          | -        | -                         | -         | -        | -          | -        | -                         | -         | -        |            |  |
| le450_25d<br>(25/25)     | 26  | 10                        | 2301,327         | 1,0       | 1,0      | 0          | 10       | 28,266                    | 0,2       | 1,2      | 0          | 10       | 914,917                   | 26,2      | 26,2     | 0          |  |
| <b>le450</b>             |     |                           | <b>1594,323</b>  |           |          | <b>1,5</b> |          | <b>695,461</b>            |           |          | <b>0,2</b> |          | <b>955,908</b>            |           |          | <b>0,1</b> |  |

Os resultados de ColorAnt<sub>3</sub>-RT apresentados na Tabela 3 denotam que ele foi capaz de encontrar valores de ( $\chi/k^*$ ) para 9 das 15 instâncias utilizadas, sendo que em 6 dessas instâncias o algoritmo atingiu 10 sucessos em 10 tentativas. Nota-se que ColorAnt<sub>3</sub>-RT obteve valor 1,2 como média de M.Cfs da classe de problemas **flat**, que é menor que o valor de ColorAnt<sub>4</sub>-RT e igual ao de ColorAnt<sub>5</sub>-RT, porém o tempo computacional é 4 vezes maior que o deste último. Um grande destaque é o tempo computacional de 99,327s da classe **dsjr**, contra tempos nas casas de 500 e 800 segundos para os outros algoritmos.

*ColorAnt<sub>3</sub>-RT* não venceu nenhuma média de M.Cfs isoladamente. A ordem de melhores resultados em qualidade de solução é **dsjr**, **dsjc**, **flat** e **le450**. Em termos de tempo computacional a ordem é **dsjr**, **le450**, **dsjc** e **flat**. O grupo **flat** apresenta altíssimo tempo computacional causado pelo processamento da instância *flat1000\_50\_0*, que, no entanto, tem solução ótima encontrada. O tempo da instância é 148116,972s, muito mais do que o limite de 7200s imposto ao algoritmo. Observando a M.TC, a conclusão é de que a grande parte desse tempo computacional foi gasto dentro da busca local, onde não é feita verificação do critério de parada que limita o tempo do algoritmo.

O algoritmo *ColorAnt<sub>3</sub>-RT* foi proposto no trabalho Lintzmayer et al. (2011), que apresentou experimentos utilizando instâncias de grafos da mesma base utilizada pelo presente trabalho. O processo de calibragem apresentado em Lintzmayer et al. (2011) é menos abrangente. Das instâncias utilizadas em ambos os trabalhos, nove são coincidentes, e, considerando a qualidade de solução, o presente trabalho foi cinco vezes melhor, empatou em uma e foi pior em uma. Este é um bom indício da melhoria causada pelo melhor processo de calibragem.

O *ColorAnt<sub>4</sub>-RT* teve melhor tempo computacional em **le450** que os outros dois algoritmos, obtendo também uma boa qualidade de solução com média de M.Cfs com valor 0,2. Porém, teve um desempenho bastante ruim nas classes **flat** e **dsjr**. Em **flat**, teve a pior qualidade de solução com valor de 3,8 como média de M.Cfs, além do fato de não ter encontrado coloração com  $k = 30$  para a instância *flat300\_28\_0*. Em **dsjr**, o valor de 0,3 para média de M.Cfs não pode ser equiparado aos dos outros algoritmos, pois *ColorAnt<sub>4</sub>-RT* não encontrou valores de  $k$  tão bons quanto os outros para a instância **dsjr500.5** (na verdade, os valores de *ColorAnt<sub>4</sub>-RT* foram bastante piores). Em **dsjc** tempo computacional e qualidade de solução foram ambos piores que os dos outros algoritmos.

Desse modo, a tentativa de se realizar o aproveitamento de cores proposta por *ColorAnt<sub>4</sub>-RT* não surtiu um efeito geral positivo, de acordo com os resultados apresentados.

Em termos de qualidade de solução, o *ColorAnt<sub>5</sub>-RT* vence ou empata com os outros algoritmos em todas as classes de problemas. Vence em **dsjc**, **dsjr** e **le450**; empatando com *ColorAnt<sub>4</sub>-RT* em **flat**, porém com um tempo computacional muito melhor. O valor de **dsjr** de *ColorAnt<sub>4</sub>-RT* é 0,3, contra 0,4 de *ColorAnt<sub>5</sub>-RT*, porém, *ColorAnt<sub>5</sub>-RT* encontra coloração com  $k = 124$ , enquanto que a variação 4 do algoritmo encontra coloração com  $k = 129$ . *ColorAnt<sub>5</sub>-RT* tem o melhor tempo computacional que os outros algoritmos apenas em **dsjc**.

A características de auto-adaptação de parâmetros surtiu um efeito positivo na qualidade de solução, mas investigações no sentido de diminuir seu tempo de execução são válidas. Um outro ponto bastante importante em *ColorAnt<sub>5</sub>-RT* é a calibragem: como o algoritmo realiza auto-adaptação de  $\alpha$ ,  $\beta$  e  $\rho$  alterando seus valores em tempo de execução guiado pela execução do algoritmo, não é necessário que esse valores sejam calibrados, diminuindo em muito o tempo necessário para este processo.

## 6. Conclusões e Trabalhos Futuros

Os algoritmos *ColorAnt-RT* tem demonstrado serem boas opções para solucionar o PCG, sendo capazes de encontrar boas soluções para diversas instâncias de grafos.

Embora *ColorAnt<sub>3</sub>-RT* tenha demonstrado em trabalhos anteriores possuir pon-

tencial, novas pesquisas ocasionaram o desenvolvimento de dois novos algoritmos: *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT*. Desta forma, o objetivo deste artigo foi descrever as modificações realizadas em *ColorAnt<sub>3</sub>-RT* que ocasionaram o surgimento destes novos algoritmos.

Como foi demonstrado nos experimentos apresentados, as modificações realizadas em *ColorAnt<sub>3</sub>-RT* viabilizaram o aumento de desempenho dos algoritmos *ColorAnt-RT*. *ColorAnt<sub>4</sub>-RT* e *ColorAnt<sub>5</sub>-RT* foram capazes de encontrar as melhores soluções conhecidas para diversas instâncias, enquanto que para outras a distância para as melhores soluções foi significativamente curta. Isto indica que tais algoritmos estão próximos dos melhores algoritmos propostos na literatura. Contudo, *ColorAnt<sub>5</sub>-RT* se mostrou melhor que os seus antecessores em termos da qualidade do resultado como também do tempo de execução. Além disto, *ColorAnt<sub>5</sub>-RT* possui o atrativo de não necessitar de uma prévia calibragem de  $\alpha$ ,  $\beta$  e  $\rho$ .

Uma questão que deve ser abordada em novos trabalhos é a necessidade de verificar a condição de parada durante a execução de um ciclo e não apenas ao seu final. Desta forma, será possível a finalização dos algoritmos de acordo com o parâmetro de tempo máximo de execução. Outro ponto a ser abordado será a implementação de outras heurísticas para avaliá-las (e compará-las) sobre as mesmas condições. Além disto, um outro passo será tornar a quantidade de formigas e de ciclos da busca local adaptáveis como  $\alpha$ ,  $\beta$  e  $\rho$  em *ColorAnt<sub>5</sub>-RT*.

## Agradecimentos

Os autores agradecem o apoio da Fundação Araucária.

## Referências

- Brélaz, D.** (1979). New Methods to Color the Vertices of a Graph. *Communications of the ACM*, 22(4):251–256.
- Comellas, F. e Ozón, J.** (1998). An Ant Algorithm for the Graph Colouring Problem. In *International Workshop on Ant Colony Optimization*, pages 151–158.
- Cormen, T. H., Leiserson, C. E., Rivest, R. L., e Stein, C.** (2009). *Introduction to Algorithms*. MIT Press, Cambridge, Massachusetts, 3rd edition.
- Costa, D. e Hertz, A.** (1997). Ants Can Colour Graphs. *The Journal of the Operational Research Society*, 48(3):295–305.
- Culberson, J. C. e Luo, F.** (1995). Exploring the k-colorable Landscape with Iterated Greedy. In *Dimacs Series in Discrete Mathematics and Theoretical Computer Science*, pages 245–284. American Mathematical Society.
- Dorigo, M. e Stützle, T.** (2004). *Ant Colony Optimization*. MIT Press.
- Galinier, P. e Hao, J.-K.** (1999). Hybrid Evolutionary Algorithms for Graph Coloring. *Journal of Combinatorial Optimization*, 3(4):379–397.
- Johnson, D. S. e Trick, M. A.** (1996). *Cliques, Coloring, and Satisfiability: Second DIMACS Implementation Challenge*. DIMACS series in discrete mathematics and theoretical computer science. American Mathematical Society, Providence, RI, EUA.
- Karp, R. M.** (1972). Reducibility among combinatorial problems. In Miller, R. e Thatcher, J., editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, New York, NY, EUA.

- Kurniawan, T. B., Khalid, N. K., Ibrahim, Z., Khalid, M., e Middendorf, M.** (2008). An ant colony system for dna sequence design based on thermodynamics. In *Proceedings of the International Conference on Advances in Computer Science and Technology*, pages 144–149, Anaheim, CA, USA. ACTA Press.
- Leighton, F. T.** (1979). A Graph Coloring Algorithm for Large Scheduling Problems. *Journal of Research of the National Bureau of Standards*, 84(6):489–506.
- Lin, C.-C. e Tseng, L.-C.** (2010). Website reorganization using an ant colony system. *Expert Systems with Applications*, 37(12):7598–7605.
- Lintzmayer, C. N., Mulati, M. H., e da Silva, A. F.** (2011a). Algoritmo Heurístico Baseado em Colônia de Formigas Artificiais ColorAnt2 com Busca Local Aplicado ao Problema de Coloração de Grafo. In *X Congresso Brasileiro de Inteligência Computacional*.
- Lintzmayer, C. N., Mulati, M. H., e da Silva, A. F.** (2011b). RT-ColorAnt: Um Algoritmo Heurístico Baseado em Colônia de Formigas Artificiais com Busca Local para Colorir Grafos. In *Anais do Simpósio Brasileiro de Pesquisa Operacional*.
- Lintzmayer, C. N., Mulati, M. H., e da Silva, A. F.** (2011c). Toward Better Performance of ColorAnt ACO Algorithm. In *XXX International Conference of the Chilean Computer Science Society*.
- Mulati, M. e Constantino, A.** (2011). Ant-line: A line-oriented aco algorithm for the set covering problem. In *Computer Science Society (SCCC), 2011 30th International Conference of the Chilean*, pages 265–274.
- Plumettaz, M., Schindl, D., e Zufferey, N.** (2010). Ant Local Search and Its Efficient Adaptation to Graph Colouring. *Journal of the Operational Research Society*, 61(5):819–826.
- Sariff, N. B. e Buniyamin, N.** (2010). Ant colony system for robot path planning in global static environment. In *Proceedings of the International Conference on System Science and Simulation in Engineering*, pages 192–197, Stevens Point, Wisconsin, USA. World Scientific and Engineering Academy and Society.
- Shawe-Taylor, J. e Zerovnik, J.** (2001). Ants and Graph Coloring. In *International Conference on Artificial Neural Nets and Genetic Algorithms, ICANNGA'01*, pages 276–279, Berlin, Heidelberg. Springer.
- Yin, J. e Xiang, W.** (2012). Ant colony algorithm for surgery scheduling problem. In *Proceedings of the Third International Conference on Advances in Swarm Intelligence - Volume Part I*, pages 198–205, Berlin, Heidelberg. Springer-Verlag.