

# Busca Tabu (TS) e Colônia de Formigas (ACO) para o Problema de Coloração de Grafos (GCP)

Guilherme Augusto Rocha de Figueiredo<sup>1</sup>

<sup>1</sup>Departamento de Informática – Universidade Federal de Viçosa (UFV)  
Viçosa – MG – Brazil

guilherme.figueiredo@ufv.br

**Abstract.** *This work presents the implementation of two metaheuristics, Tabu Search (TS) and Ant Colony Optimization (ACO), applied to a variant of the Graph Coloring Problem (GCP) in which the number of colors  $k$  is predefined ( $k$ -GCP). This formulation transforms the problem into a conflict minimization task. The metaheuristics were tested on benchmark instances from the literature, using values of  $k$  equal to and below the known chromatic number, in order to evaluate the progression of the error rate. The results showed similar performance between the methods, with Tabu Search standing out by achieving comparable solutions in a fraction of the time, highlighting its efficiency and relevance for this type of problem.*

**Resumo.** *Este trabalho apresenta a implementação de duas metaheurísticas, Busca Tabu (do inglês, Tabu Search - TS) e Colônia de Formigas (do inglês, Ant Colony Optimization - ACO), aplicadas a uma variante do Problema de Coloração de Grafos em que o número de cores  $k$  é previamente definido ( $k$ -GCP). Essa formulação transforma o problema em uma tarefa de minimização de conflitos. As metaheurísticas foram aplicadas a instâncias clássicas da literatura, considerando valores de  $k$  iguais e inferiores ao número cromático conhecido, com o objetivo de analisar a progressão da taxa de erros. Os resultados obtidos mostraram desempenhos similares entre as abordagens, com a Busca Tabu se destacando por alcançar soluções comparáveis em uma fração do tempo, o que reforça sua popularidade no contexto deste problema.*

## 1. Introdução

O Problema de Coloração de Grafos (GCP) é um problema clássico e amplamente estudado na teoria dos grafos. Sua definição formal envolve a atribuição de "cores" aos vértices de um grafo de forma que nenhum par de vértices adjacentes (isto é, conectados por uma aresta) receba a mesma cor. O objetivo tradicional do GCP é determinar o menor número de cores necessário para colorir um dado grafo, conhecido como seu número cromático ( $\chi(G)$ ) [Jensen and Toft 2011].

A variante abordada neste trabalho é o  $k$ -GCP, em que o número de cores  $k$  é previamente definido. Nessa versão, o foco não está em minimizar o número de cores utilizadas, mas sim em reduzir a quantidade de conflitos ao utilizar exatamente  $k$  cores. Conflitos, neste contexto, correspondem a arestas cujos vértices possuem a mesma cor. Essa mudança de foco transforma o problema em uma tarefa de otimização baseada em um  $k$  fixo, o que altera a métrica de sucesso e a função objetivo da solução.

O GCP é conhecido por sua alta complexidade computacional, sendo classificado como um problema NP-difícil (NP-hard). Essa característica justifica o uso de heurísticas e metaheurísticas para encontrar soluções de boa qualidade em tempo viável, já que abordagens exatas se tornam inviáveis para instâncias de grande porte. O problema possui diversas aplicações práticas, como a alocação de canais em redes sem fio para evitar interferências, o agendamento de tarefas (exames, reuniões) para prevenir sobreposições, e a alocação de registradores em compiladores com o objetivo de otimizar o uso de memória.

A escolha do  $k$ -GCP neste trabalho visa explorar um problema de minimização local sobre instâncias da literatura, com o número cromático já conhecido, além de analisar a progressão da taxa de erro ao tentar colorir um grafo com um número de cores inferior a  $\chi(G)$ . Duas metaheurísticas foram implementadas: uma baseada em busca local (Busca Tabu) e outra populacional (Colônia de Formigas), com o objetivo de encontrar boas soluções para essa variante do problema. O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta os trabalhos relacionados ao GCP; a Seção 3 descreve a metodologia e os algoritmos utilizados; a Seção 4 discute os resultados obtidos; e, por fim, a Seção 5 apresenta as conclusões do trabalho.

## 2. Trabalhos Relacionados

Diversas abordagens têm sido aplicadas ao Problema de Coloração de Grafos (GCP) ao longo das décadas, utilizando desde heurísticas construtivas até metaheurísticas sofisticadas. A seguir, apresentamos uma seleção representativa da literatura.

[Malaguti and Toth 2010] elaboraram um survey abrangente sobre variantes do GCP, abrangendo técnicas exatas, heurísticas e metaheurísticas, e reportando desempenho computacional em grandes instâncias (até milhares de vértices). Já [Musliu and Schwengerer 2013] vão além e investigaram a seleção automática de algoritmos para o GCP utilizando aprendizado de máquina: eles analisaram 78 características de instâncias e treinaram modelos para prever a metaheurística mais adequada para cada instância, demonstrando melhora em benchmarks clássicos. Ambos trabalhos evidenciam como o GCP já foi amplamente explorado na literatura.

No contexto das metaheurísticas que foram implementados neste trabalho, uma das primeiras implementações marcantes de Busca Tabu para o GCP foi proposta por [Hertz and Werra 1987]: eles desenvolveram o algoritmo *TABUCOL*, aplicando lista tabu e critério de aspiração em grafos com até 1.000 vértices, obtendo colorações quase ótimas e superando métodos como simulated annealing. Além disso, [Dowsland and Thompson 2008] propuseram uma heurística baseada em Colônia de Formigas, aprimorada com técnicas de construção e busca local tabu nas soluções geradas. Seus resultados em benchmarks mostraram desempenho competitivo com algoritmos de ponta da época.

A relevância desses estudos motiva a escolha deste trabalho: foi implementado versões modernas, mas conceitualmente alinhadas com *TABUCOL* e as propostas de ACO, e os resultados foram comparados de forma sistemática. Ademais, este trabalho se diferencia ao combinar: (i) comparação direta entre TS e ACO em ambiente controlado, (ii) análise da taxa de conflitos para variações de  $k$ , e (iii) uso de instâncias padronizadas, facilitando replicação e extensibilidade.

### 3. Metodologia

Esta seção descreve a metodologia adotada neste trabalho. Primeiramente, é apresentada a modelagem do problema e o processo de seleção das instâncias utilizadas na avaliação. Em seguida, são detalhadas as metaheurísticas implementadas, e por fim, são descritas as configurações experimentais adotadas.

#### 3.1. Modelagem do Problema

O problema abordado já possui uma representação computacional consolidada, o que facilita sua modelagem. Os grafos foram representados por meio de listas de adjacência. As soluções são codificadas como vetores de inteiros, em que cada posição representa um vértice e o valor correspondente indica a cor atribuída a esse vértice. A função objetivo é a contagem do número de conflitos, ou seja, o número de arestas cujos vértices terminais possuem a mesma cor. O conceito de vizinhança local utilizado considera soluções que diferem apenas na cor de um único vértice.

#### 3.2. Instâncias

Dado que o GCP é amplamente estudado na literatura, existem diversas instâncias públicas utilizadas como benchmark. Para a seleção das instâncias neste trabalho, foi utilizada como referência uma página da web <sup>1</sup>, que apresenta uma lista de instâncias acompanhada de metadados, incluindo uma ordenação por tempo médio de solução com base em hardwares e softwares de alto desempenho. Essa classificação permitiu filtrar apenas instâncias cuja resolução se dá na faixa de segundos ou minutos.

Para o download das instâncias, utilizou-se uma segunda página da web <sup>2</sup>, que disponibiliza as instâncias em um formato padronizado, facilitando a leitura e o processamento. Embora esta página não contenha todas as instâncias da fonte anterior, ela oferece 42 instâncias diversificadas, o que é suficiente para realizar uma avaliação significativa dos algoritmos. A Tabela 1 apresenta uma descrição detalhada dessas instâncias.

Como pode ser observado, a Tabela 1 contempla uma ampla variedade de instâncias, abrangendo grafos de pequeno porte ( $|V| < 250$ ), médio porte ( $350 < |V| < 500$ ) e grande porte ( $|V| > 550$ ). Há também variação no número cromático, o que possibilita uma análise abrangente sobre o desempenho das abordagens implementadas.

#### 3.3. Busca Tabu (TS)

A primeira metaheurística implementada foi a Busca Tabu (Tabu Search - TS), uma técnica de busca local que expande o método clássico de *Hill Climbing* ao permitir a saída de ótimos locais por meio de uma lista tabu. Essa lista armazena movimentos recentes, impedindo que sejam revertidos por um número determinado de iterações, o que evita ciclos e reforça a exploração do espaço de soluções. A TS também utiliza um critério de aspiração, que permite realizar um movimento tabu caso ele produza uma solução melhor do que a melhor obtida até o momento.

Essa metaheurística é amplamente utilizada no contexto do GCP, tanto isoladamente quanto integrada a outras abordagens. Neste trabalho, foi adotada uma versão

---

<sup>1</sup><https://sites.google.com/site/graphcoloring/vertex-coloring>

<sup>2</sup><https://mat.tepper.cmu.edu/COLOR/instances.html>

**Table 1. Instâncias de Grafos**

<b>Nome</b>	<b>V</b>	<b>E</b>	$\chi(G)$	<b>Nome</b>	<b>V</b>	<b>E</b>	$\chi(G)$
anna	138	493	11	miles750	128	2113	31
david	87	406	11	multsol.i.1	197	3925	49
fpsol2.i.1	496	11654	65	multsol.i.2	188	3885	31
fpsol2.i.2	451	8691	30	multsol.i.3	184	3916	31
fpsol2.i.3	425	8688	30	multsol.i.4	185	3946	31
games120	120	638	9	multsol.i.5	186	3973	31
homer	561	1628	13	myciel3	11	20	4
huck	74	301	11	myciel4	23	71	5
inithx.i.1	864	18707	54	myciel5	47	236	6
inithx.i.2	645	13979	31	myciel6	95	755	7
inithx.i.3	621	13969	31	queen5_5	25	160	5
jean	80	254	10	queen6_6	36	290	7
le450_15a	450	8168	15	queen7_7	49	476	7
le450_25a	450	8260	25	queen5_5	96	1368	12
le450_25a	450	8263	25	queen5_5	64	728	9
le450_5c	450	9803	5	queen5_5	81	1056	10
le450_5d	450	9757	5	school1	385	19095	14
miles1000	128	3216	42	school1_nsh	352	14612	14
miles1500	128	5198	73	zeroin.i.1	211	4100	49
miles250	128	387	8	zeroin.i.2	211	3541	30
miles500	128	1170	20	zeroin.i.3	206	3540	30

clássica. Os hiperparâmetros utilizados foram: número de iterações tabu igual a 7 (como no algoritmo TABUCOL); lista tabu implementada como uma matriz Vértice x Cor; e critério de parada definido como  $2 \cdot |V|$  iterações sem melhora na solução global.

Experimentos alternativos foram realizados utilizando uma lista tabu simplificada (vetor de vértices) e definindo o número de iterações com base no tamanho do clique máximo do grafo. No entanto, testes estatísticos indicaram que não houve diferença significativa entre as variantes, e a versão inicial apresentou melhores médias e medianas nos resultados.

---

**Algorithm 1** Busca Tabu para o Problema de Coloração de Grafos

---

```

1: procedure TS-SOLVE
2:    $iteration \leftarrow 0$ 
3:    $bestSolution \leftarrow \text{EVALUATE}(\text{currentColoring})$ 
4:    $lastImprovement \leftarrow 0$ 
5:   while true do
6:     if  $iteration - lastImprovement > 2 \cdot |V|$  then
7:       break
8:     end if
9:      $neighbors \leftarrow \text{GENERATEALLCOLORCHANGES}$ 
10:     $\text{SORT}(neighbors \text{ by number of conflicts})$ 
11:    for all  $(u, c) \in neighbors$  do
12:      if  $\text{ISBETTER}(u, c)$  then
13:         $\text{APPLYMOVE}(u, c)$ 
14:         $\text{UPDATETABULIST}(u, c, iteration)$ 
15:         $bestSolution \leftarrow \text{UPDATEBEST}(iteration)$ 
16:        break
17:      else if not  $\text{ISTABU}(u, c, iteration)$  then
18:         $\text{APPLYMOVE}(u, c)$ 
19:         $\text{UPDATETABULIST}(u, c, iteration)$ 
20:        break
21:      end if
22:    end for
23:    if  $bestSolution = 0$  then
24:      break
25:    end if
26:     $iteration \leftarrow iteration + 1$ 
27:  end while
28:  return  $bestSolution$ 
29: end procedure

```

---

O pseudocódigo da Busca Tabu está apresentado no Algoritmo 1. Para a geração da solução inicial, foi utilizado um algoritmo guloso simples: partindo do vértice 1, percorre-se uma lista, e a cada passo, se o vértice ainda não foi processado, atribui-se a ele a cor que gera o menor número de conflitos locais naquele momento, e seus vértices adjacentes são adicionados à lista. O processo continua até que todos os vértices estejam coloridos.

### 3.4. Colônia de Formigas (ACO)

Como abordagem populacional, foi implementada a metaheurística Colônia de Formigas (Ant Colony Optimization - ACO), inspirada no comportamento coletivo de formigas em busca de alimento. No algoritmo, formigas constroem soluções iterativamente, influenciadas pela quantidade de feromônio depositada nos caminhos (pares vértice-cor) e por uma heurística que avalia a qualidade da escolha. Ao final de cada iteração, os feromônios são atualizados: ocorre evaporação e reforço nos caminhos que compõem as melhores soluções.

---

#### Algorithm 2 Colônia de Formigas para o Problema de Coloração de Grafos

---

```

1: procedure ACO-SOLVE
2:    $bestSolution \leftarrow |E| + 1$ 
3:   for  $it \leftarrow 1$  to  $maxIterations$  do
4:      $antSolutions \leftarrow []$  ▷ Lista de pares (valor, solução)
5:     for  $k \leftarrow 1$  to  $nAnts$  do
6:        $solution_k \leftarrow \text{RLFCONSTRUCTIONWITHPHEROMONES}$ 
7:        $value_k \leftarrow \text{EVALUATE}(solution_k)$ 
8:        $\text{APPEND}(antSolutions, (value_k, solution_k))$ 
9:       if  $value_k < bestSolution$  then
10:         $bestSolution \leftarrow value_k$ 
11:      end if
12:    end for
13:     $\text{UPDATEPHEROMONES}(antSolutions)$ 
14:    if  $bestSolution = 0$  then
15:      break
16:    end if
17:  end for
18:  return  $bestSolution$ 
19: end procedure

```

---

O pseudocódigo da ACO está apresentado no Algoritmo 2. Os hiperparâmetros adotados foram: 20 formigas, 10 iterações, taxa de evaporação de 10%,  $\alpha = 1.0$  e  $\beta = 2.0$ . Os parâmetros  $\alpha$  e  $\beta$  controlam, respectivamente, a influência do feromônio e da heurística no processo de decisão.

A construção da solução foi inspirada no algoritmo *Recursive Largest First* (RLF). A cada passo, a formiga seleciona o vértice não processado com maior grau (ou escolhe aleatoriamente entre empatados). A escolha da cor baseia-se em uma fórmula que combina feromônio e heurística local:

$$P_{i,j}^k = \frac{(\tau_{i,j})^\alpha (\eta_{i,j})^\beta}{\sum_{l \in N_i^k} (\tau_{i,l})^\alpha (\eta_{i,l})^\beta}$$

A heurística  $\eta_{i,j}$  é definida como  $\frac{1}{1 + \text{conflitos locais}}$ , de forma que cores que geram menos conflitos tenham maior chance de serem escolhidas. A seleção é feita por roleta, repetindo-se até que todas as formigas construam uma solução completa.

Após a construção, a matriz de feromônio é atualizada. Realiza-se a evaporação global e o reforço elitista baseado nas melhores soluções da iteração. Apenas a metade

superior das formigas (em termos de qualidade da solução) deposita feromônio extra. Além disso, uma busca local é aplicada à melhor solução da iteração antes da atualização. A melhor solução global também contribui com reforço ao final de cada iteração. A quantidade de feromônio depositado é proporcional à qualidade da solução, de acordo com a fórmula:

$$\frac{1}{1 + \text{número de conflitos da solução}}$$

Essa estratégia favorece soluções de melhor qualidade e acelera a convergência da busca.

### 3.5. Configuração Experimental

Com o objetivo de analisar não apenas a capacidade das metaheurísticas de minimizar conflitos usando o número cromático  $k$ , mas também sua performance em cenários mais restritivos, cada instância foi executada com quatro valores distintos de  $k$ :  $k$ ,  $k - 1$ ,  $k - 2$  e  $k - 3$ , onde  $k = \chi(G)$ . Para  $k$  é possível comparar os resultados com a literatura, pois soluções sem conflitos indicam que o ótimo global foi encontrado. Para os demais valores, não há ótimos conhecidos, mas ainda é possível comparar o desempenho relativo entre as abordagens e analisar a progressão da taxa de conflitos.

Dado o uso de instâncias com diferentes tamanhos e densidades, foi necessário adotar uma métrica padronizada. Para isso, utilizou-se a **Taxa de Conflitos**, definida como a razão entre o número de conflitos e o número total de arestas do grafo. Essa métrica varia de 0% (nenhum conflito) até 100% (todos os vértices com a mesma cor), permitindo comparações justas entre instâncias de diferentes escalas.

As metaheurísticas foram implementadas em C++, enquanto a análise dos resultados e a geração de gráficos foram realizadas em Python. Todo o código-fonte encontra-se disponível publicamente no GitHub <sup>3</sup>.

## 4. Resultados

A Tabela 2 apresenta os resultados obtidos pelas metaheurísticas implementadas para o problema de coloração de grafos. Os melhores resultados entre os dois algoritmos estão destacados em negrito.

O primeiro aspecto notável ao analisar a tabela é a variabilidade entre os algoritmos, sem que um deles se destaque de forma consistente em todas as instâncias. Esse resultado é particularmente relevante, considerando as diferenças na complexidade computacional entre as abordagens. Enquanto a Busca Tabu, baseada em busca local, levou aproximadamente 15 minutos para executar todas as instâncias, a Colônia de Formigas, uma metaheurística populacional mais complexa, exigiu cerca de 20 horas para completar a execução. Um teste estatístico foi aplicado para avaliar diferenças entre os resultados, mas não apontou significância estatística, o que reforça o custo-benefício da Busca Tabu no contexto do GCP.

Com relação aos resultados para o valor  $k$  (número cromático conhecido), as metaheurísticas demonstraram bom desempenho: a TS encontrou o ótimo global em 45,24%

<sup>3</sup>[https://github.com/GuiFigueiredo/Trabalho\\_Final-INF\\_284](https://github.com/GuiFigueiredo/Trabalho_Final-INF_284)

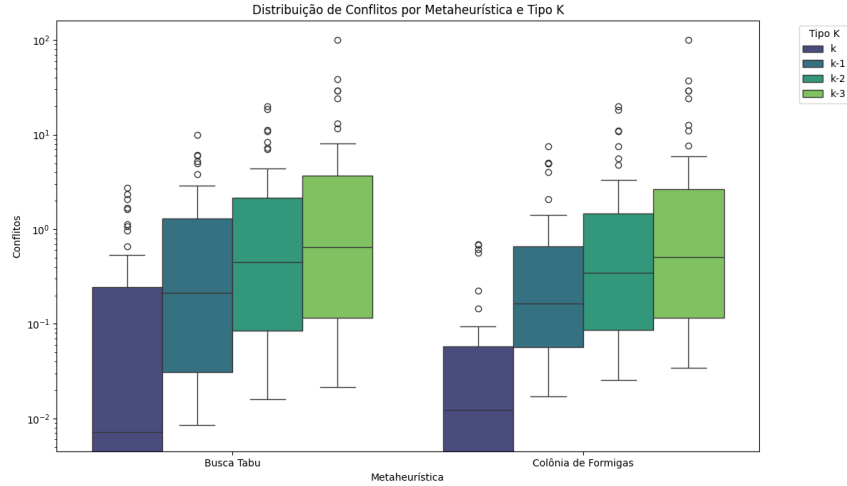
Table 2. Resultados dos Algoritmos TS e ACO

Instância	k		k-1		k-2		k-3	
	TS	ACO	TS	ACO	TS	ACO	TS	ACO
anna	0.2028	<b>0.0</b>	0.4057	<b>0.2028</b>	0.6085	<b>0.4057</b>	0.8114	<b>0.6085</b>
david	<b>0.0</b>	<b>0.0</b>	<b>0.2463</b>	<b>0.2463</b>	<b>0.7389</b>	<b>0.7389</b>	1.2315	<b>0.9852</b>
fpsol2.i.1	<b>0.0</b>	0.0086	<b>0.0086</b>	0.0172	<b>0.0172</b>	0.0257	<b>0.0257</b>	0.0343
fpsol2.i.2	<b>0.0115</b>	0.0345	<b>0.023</b>	0.0575	<b>0.046</b>	0.0805	<b>0.069</b>	0.1036
fpsol2.i.3	<b>0.0115</b>	0.0576	<b>0.023</b>	0.0460	<b>0.046</b>	0.0576	<b>0.0691</b>	0.0921
games120	<b>0.0</b>	<b>0.0</b>	<b>0.3135</b>	<b>0.3135</b>	1.4107	<b>1.2539</b>	2.8213	<b>2.5078</b>
homer	<b>0.0</b>	<b>0.0</b>	0.1229	<b>0.0614</b>	0.2457	<b>0.1843</b>	0.4914	<b>0.3686</b>
huck	<b>0.0</b>	<b>0.0</b>	<b>0.3322</b>	<b>0.3322</b>	<b>0.9967</b>	<b>0.9967</b>	2.3256	<b>1.9934</b>
inithx.i.1	<b>0.0053</b>	0.0160	<b>0.0107</b>	0.0214	<b>0.016</b>	0.0267	<b>0.0214</b>	0.0535
inithx.i.2	<b>0.0072</b>	0.0644	<b>0.0143</b>	0.0715	<b>0.0215</b>	0.0644	<b>0.0429</b>	0.0715
inithx.i.3	<b>0.0072</b>	0.0716	<b>0.0143</b>	0.0573	<b>0.0215</b>	0.0931	<b>0.043</b>	0.0787
jean	<b>0.0</b>	<b>0.0</b>	<b>0.3937</b>	<b>0.3937</b>	<b>0.7874</b>	<b>0.7874</b>	1.9685	<b>1.5748</b>
le450_15a	<b>0.5387</b>	0.6121	<b>0.9305</b>	0.9549	<b>1.3467</b>	<b>1.3467</b>	<b>1.7997</b>	<b>1.7997</b>
le450_25a	0.0484	<b>0.0363</b>	<b>0.1695</b>	<b>0.1695</b>	<b>0.2906</b>	<b>0.2906</b>	0.4600	<b>0.3995</b>
le450_25b	0.0605	<b>0.0242</b>	0.1815	<b>0.1573</b>	<b>0.2662</b>	0.2783	<b>0.3994</b>	0.4115
le450_5c	1.0813	<b>0.2244</b>	5.9574	<b>5.0597</b>	<b>10.7926</b>	<b>10.7926</b>	29.2768	<b>29.2053</b>
le450_5d	1.6398	<b>0.041</b>	6.0674	<b>4.94</b>	11.1817	<b>10.9665</b>	<b>29.22</b>	<b>29.22</b>
miles1000	0.0933	<b>0.0622</b>	0.1244	<b>0.0622</b>	0.1555	<b>0.0933</b>	0.1866	<b>0.1244</b>
miles1500	0.0192	<b>0.0</b>	0.0385	<b>0.0192</b>	0.0577	<b>0.0385</b>	0.0770	<b>0.0577</b>
miles250	0.2584	<b>0.0</b>	<b>0.5168</b>	<b>0.5168</b>	1.8088	<b>1.0336</b>	2.8424	<b>2.584</b>
miles500	0.1709	<b>0.0</b>	0.2564	<b>0.0855</b>	0.2564	<b>0.1709</b>	<b>0.3419</b>	<b>0.3419</b>
miles750	<b>0.0947</b>	<b>0.0947</b>	0.1420	<b>0.0947</b>	0.1893	<b>0.142</b>	0.2366	<b>0.1893</b>
multsol.i.1	<b>0.0</b>	<b>0.0</b>	<b>0.0255</b>	<b>0.0255</b>	<b>0.051</b>	<b>0.051</b>	<b>0.0764</b>	<b>0.0764</b>
multsol.i.2	<b>0.0</b>	0.0257	<b>0.0515</b>	<b>0.0515</b>	<b>0.103</b>	<b>0.103</b>	<b>0.1544</b>	<b>0.1544</b>
multsol.i.3	<b>0.0</b>	<b>0.0</b>	<b>0.0511</b>	<b>0.0511</b>	<b>0.1021</b>	<b>0.1021</b>	<b>0.1532</b>	<b>0.1532</b>
multsol.i.4	<b>0.0</b>	<b>0.0</b>	<b>0.0507</b>	<b>0.0507</b>	<b>0.1014</b>	<b>0.1014</b>	<b>0.1521</b>	<b>0.1521</b>
multsol.i.5	<b>0.0</b>	<b>0.0</b>	<b>0.0252</b>	<b>0.0252</b>	<b>0.0755</b>	<b>0.0755</b>	<b>0.1258</b>	<b>0.1258</b>
myciel3	<b>0.0</b>	<b>0.0</b>	<b>5.0</b>	<b>5.0</b>	<b>20.0</b>	<b>20.0</b>	<b>100.0</b>	<b>100.0</b>
myciel4	<b>0.0</b>	<b>0.0</b>	<b>1.4085</b>	<b>1.4085</b>	7.0423	<b>5.6338</b>	<b>23.9437</b>	<b>23.9437</b>
myciel5	<b>0.0</b>	<b>0.0</b>	<b>0.4237</b>	<b>0.4237</b>	2.1186	<b>1.6949</b>	8.0508	<b>7.6271</b>
myciel6	<b>0.0</b>	<b>0.0</b>	<b>0.1325</b>	<b>0.1325</b>	<b>0.6623</b>	<b>0.6623</b>	<b>2.5166</b>	2.6490
queen5_5	<b>0.0</b>	<b>0.0</b>	10.0000	<b>7.5</b>	18.7500	<b>18.125</b>	38.7500	<b>37.5</b>
queen6_6	2.0690	<b>0.6897</b>	3.7931	<b>2.069</b>	7.2414	<b>4.8276</b>	11.7241	<b>11.0345</b>
queen7_7	2.7311	<b>0.0</b>	5.2521	<b>3.9916</b>	8.4034	<b>7.563</b>	13.2353	<b>12.605</b>
queen8_12	0.6579	<b>0.1462</b>	0.9503	<b>0.6579</b>	1.7544	<b>1.462</b>	2.7047	<b>2.3392</b>
queen8_8	0.9615	<b>0.6868</b>	1.5110	<b>1.3736</b>	4.3956	<b>3.2967</b>	6.5934	<b>5.9066</b>
queen9_9	1.1364	<b>0.5682</b>	1.9886	<b>0.7576</b>	3.4091	<b>2.8409</b>	5.1136	<b>4.6402</b>
school1	1.6758	<b>0.0</b>	2.8751	<b>0.6651</b>	2.1524	<b>1.4402</b>	3.9330	<b>2.2362</b>
school1_nsh	2.3542	<b>0.0274</b>	2.4364	<b>0.5543</b>	3.0728	<b>1.2866</b>	3.0181	<b>1.9915</b>
zero.in.i.1	<b>0.0</b>	0.0244	<b>0.0244</b>	0.0488	<b>0.0488</b>	0.0732	<b>0.0732</b>	<b>0.0732</b>
zero.in.i.2	<b>0.0</b>	0.0565	<b>0.0282</b>	0.0565	0.0847	<b>0.0565</b>	0.1130	<b>0.0847</b>
zero.in.i.3	<b>0.0</b>	0.0282	<b>0.0282</b>	0.0565	<b>0.0847</b>	<b>0.0847</b>	<b>0.113</b>	<b>0.113</b>



das instâncias ( $\frac{19}{42}$ ) e a ACO em 47,62% ( $\frac{20}{42}$ ). Esses números indicam que ambas as abordagens foram capazes de resolver aproximadamente metade das instâncias sem conflitos. Para os demais valores de  $k$  ( $k-1$ ,  $k-2$  e  $k-3$ ), não é possível saber o valor ótimo global, mas a comparação relativa entre as heurísticas permanece válida. A Figura 1 ilustra os resultados em formato de boxplot.

**Figure 1. Boxplot com a distribuição dos resultados (escala logarítmica).**



A Figura 1 utiliza escala logarítmica devido à concentração da maioria dos valores abaixo de 1%. A necessidade desta escala já mostra como os algoritmos em geral encontraram bons resultados. Além disso, o gráfico evidencia a proximidade entre os algoritmos, com a Busca Tabu apresentando maior variabilidade e ligeiramente piores médias. Em relação ao impacto da redução no número de cores disponíveis, observa-se que a diferença entre  $k$  e  $k-1$  resulta em poucos conflitos adicionais. Isso sugere que a última cor utilizada no número cromático é empregada de maneira pontual, em regiões específicas do grafo com alta densidade. À medida que o número de cores diminui, a taxa de conflitos aumenta progressivamente.

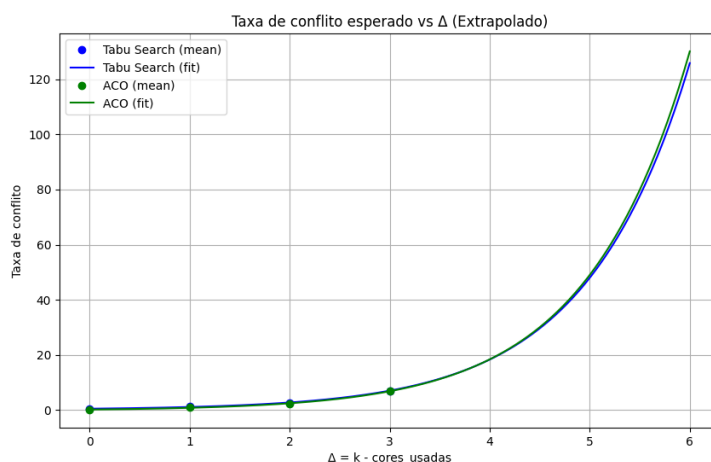
Foi ajustada uma curva de progressão da taxa média de conflitos utilizando a função `curve_fit` da biblioteca **Scipy**, com o objetivo de extrapolar o comportamento esperado. O resultado é mostrado na Figura 2.

Embora a extrapolação seja teórica e simplificada, pois 100% de conflitos ocorre apenas com  $k = 1$  e a progressão de conflitos é particular a cada grafo, a curva reforça a ideia de que as primeiras cores no número cromático possuem relevância localizada, enquanto a necessidade de múltiplas cores cresce de forma exponencial para evitar conflitos.

## 5. Conclusão

Este trabalho implementou duas metaheurísticas, Busca Tabu (TS) e Colônia de Formigas (ACO), para o problema  $k$ -GCP, uma versão do Problema de Coloração de Grafos em que o número de cores é pré-definido. Ambas as abordagens apresentaram desempenho semelhante, encontrando o ótimo global em quase metade das instâncias avaliadas, além de produzir bons resultados de forma geral. A Busca Tabu se destacou por sua simplicidade e eficiência, obtendo soluções comparáveis às da ACO em uma fração do tempo.

**Figure 2. Curva extrapolada da progressão da taxa de conflitos.**



Esse resultado reforça sua popularidade como estratégia eficaz e de baixo custo computacional para o GCP.

A análise dos resultados também evidenciou que a diferença entre utilizar  $k$  e  $k - 1$  cores resulta em um pequeno aumento na taxa de conflitos, o que sugere que a última cor é utilizada apenas em regiões críticas do grafo. À medida que se reduz ainda mais o número de cores, a taxa de erro cresce exponencialmente, indicando a crescente dificuldade em encontrar soluções viáveis.

De forma geral, este trabalho proporcionou um exercício completo de implementação e comparação entre metaheurísticas de natureza distinta (local e populacional) aplicadas a um problema clássico da literatura. Como contribuição adicional, foi realizada uma análise da progressão da taxa de conflitos em função da redução do número de cores, oferecendo uma perspectiva mais ampla sobre a dificuldade incremental do problema.

Como trabalho futuro, propõe-se a investigação de abordagens híbridas, combinando metaheurísticas para potencializar a qualidade das soluções e reduzir o tempo de execução, além de explorar variantes do problema com restrições adicionais.

## References

- Dowsland, K. A. and Thompson, J. M. (2008). An improved ant colony optimisation heuristic for graph colouring. *Discrete Applied Mathematics*, 156(3):313–324.
- Hertz, A. and Werra, D. d. (1987). Using tabu search techniques for graph coloring. *Computing*, 39(4):345–351.
- Jensen, T. R. and Toft, B. (2011). *Graph coloring problems*. John Wiley & Sons.
- Malaguti, E. and Toth, P. (2010). A survey on vertex coloring problems. *International transactions in operational research*, 17(1):1–34.
- Musliu, N. and Schwengerer, M. (2013). Algorithm selection for the graph coloring problem. In *International conference on learning and intelligent optimization*, pages 389–403. Springer.