

# Algoritmo de Colônia de Formigas para o Problema do Clique

Iuri Pedroso<sup>1</sup>, Yan Gabriel Reis<sup>2</sup>, Herich de Campos<sup>3</sup>

[57023140011@unicentro.edu.br](mailto:57023140011@unicentro.edu.br), [57023740006@unicentro.edu.br](mailto:57023740006@unicentro.edu.br),  
[57023740007@unicentro.edu.br](mailto:57023740007@unicentro.edu.br)

Curso de Ciência da Computação  
Universidade Estadual do Centro-Oeste (Unicentro)  
Guarapuava – PR – Brasil

**Abstract.** *The Maximum Clique Problem aims to find the largest subset of vertices in a graph where every pair is connected. Being NP-hard, exact solutions are costly for large graphs. This work applies the Ant Colony Optimization (ACO) algorithm, inspired by real ants' foraging behavior. Each ant incrementally builds a candidate clique guided by pheromones and heuristics. Pheromone reinforcement over iterations leads the colony toward better solutions. Experiments show that ACO efficiently identifies large cliques in random graphs, exploring the solution space cooperatively.*

**Resumo.** *O Problema do Clique Máximo busca o maior conjunto de vértices conectados entre si em um grafo. Por ser NP-difícil, soluções exatas são caras para grafos grandes. Este trabalho utiliza o Algoritmo de Colônia de Formigas (ACO), inspirado no comportamento cooperativo das formigas. Cada formiga constrói uma clique incrementalmente, guiada por feromônios e heurísticas. O reforço de feromônio ao longo das iterações conduz a colônia a soluções melhores. Os resultados mostram que o ACO identifica eficientemente grandes cliques, explorando o espaço de soluções de forma cooperativa.*

## 1. Introdução

O Problema do Clique Máximo consiste em encontrar o maior conjunto de vértices de um grafo em que todos os pares de vértices estão conectados. Esse problema é de natureza NP-difícil, tornando soluções exatas inviáveis para grafos grandes [1]. Por essa razão, optou-se pela utilização de metaheurísticas, que buscam boas soluções em tempo computacional aceitável ao custo de não garantir a melhor solução possível.

Neste trabalho, foi aplicado o Algoritmo de Colônia de Formigas (ACO), uma metaheurística bio-inspirada no comportamento de formigas reais que depositam feromônios para comunicar caminhos promissores [2]. No contexto do Problema do Clique, cada formiga constrói uma clique incrementalmente, guiada por feromônios e informações heurísticas sobre o grafo. Ao longo das iterações, o reforço de feromônio concentra a busca em regiões promissoras, permitindo à colônia encontrar grandes cliques. O ACO foi escolhido por sua capacidade de explorar de forma cooperativa o espaço de soluções, balanceando exploração e intensificação, sendo adequado para problemas combinatórios complexos como o Clique Máximo.

## **2. Fundamentação Teórica**

### **2.1 O Problema do Clique e Complexidade Computacional**

Segundo Cormen et al. [3], um clique em um grafo não orientado  $G=(V,E)$  é um subconjunto de vértices  $V' \subseteq V$  onde cada par é conectado por uma aresta, constituindo um subgrafo completo. O foco deste trabalho reside na abordagem de otimização, cujo objetivo é encontrar o clique de maior cardinalidade possível no grafo, conhecido como clique máximo.

Uma abordagem exata ingênuia para solucionar o problema de otimização exigiria verificar todos os subconjuntos de vértices possíveis para garantir que o maior agrupamento foi encontrado. O espaço de busca cresce exponencialmente com o número de vértices ( $2$  elevado a  $|V|$ ), tornando a verificação impraticável. Conforme a literatura aponta, se fosse possível resolver a otimização de forma eficiente, o problema de decisão, que é verificar a existência de uma clique de tamanho  $k$ , também seria resolvido, mas como este último é comprovadamente NP-difícil, a busca pelo clique máximo também possui essa complexidade para grafos grandes [1].

Essa dificuldade computacional justifica o paralelo com o Problema do Caixeiro Viajante. Enquanto neste a dificuldade advém do número fatorial de rotas possíveis para visitar todas as cidades, no Problema do Clique a dificuldade reside na explosão combinatória de agrupamentos candidatos. Em ambos os cenários, a impossibilidade de varrer o espaço de busca completo em tempo polinomial motiva o uso de métodos heurísticos.

### **2.2 Aplicação da Otimização por Colônia de Formigas**

Dada a inviabilidade de soluções exatas, aplica-se a metaheurística de Otimização por Colônia de Formigas. Originalmente proposta para resolver o Problema do Caixeiro Viajante, onde o objetivo é encontrar o melhor caminho sequencial, a técnica é adaptada neste trabalho para o contexto de agrupamento de vértices.

Diferente do Problema do Caixeiro Viajante, onde as formigas artificiais depositam feromônio nas arestas para demarcar rotas de transição, na solução proposta para o Problema do Clique as formigas constroem soluções passo a passo selecionando vértices baseados em dois critérios principais: Conectividade: O vértice candidato deve obrigatoriamente estar conectado a todos os vértices já selecionados pela formiga na iteração atual; Informação Heurística e Feromônio: A escolha é guiada probabilisticamente pelo grau do nó, ou seja, a heurística local e pelo histórico de sucesso da colônia que neste caso se trata do feromônio acumulado nos nós que frequentemente pertencem a grandes cliques.

Essa abordagem cooperativa permite explorar o espaço de busca de forma muito mais eficiente do que a tentativa e erro da abordagem ingênuia descrita por Cormen,

balanceando a exploração de novas possibilidades com a intensificação nas melhores soluções encontradas.

### 3. Desenvolvimento

A implementação do algoritmo Colônia de Formigas (ACO), foi organizada para o problema do Clique Máximo em três módulos principais: grafo (graph\_clique.py), formigas (ant\_clique.py) e ACO (aco\_clique.py), onde cada componente possui um papel fundamental no processo e avaliação do problema. O código utilizado neste trabalho é de autoria própria, desenvolvido em Python, garantindo o controle completo sobre os métodos e parâmetros aplicados.

#### 3.1 Estrutura do Grafo

O grafo é representado por uma matriz de adjacência de dimensão  $N \times N$  onde  $N$ , é o número de nós. Cada elemento da matriz indica se existe uma aresta entre os vértices  $i$  e  $j$ :

- $a_{ij} = 1$ , se existe aresta entre  $i$  e  $j$
- $a_{ij} = 0$ , caso contrário

Cada nó também possui dois atributos principais:

- **Feromônio ( $\tau$ )**: inicialmente igual a 1 para todos os nós, representa a atração histórica das formigas para aquele nó.
- **Heurística ( $\eta$ )**: valor baseado no grau do nó, ou seja, na quantidade de conexões que ele possui. Nós com maior grau são considerados mais promissores para construir cliques grandes.

### 4. Metodologia Proposta: ACO para o Problema do Clique

#### 4.1. Mapeamento do Problema

A implementação do Algoritmo de Colônia de Formigas (ACO) para o Problema do Clique Máximo difere do modelo tradicional, como o utilizado no Problema do Caixeiro Viajante (TSP) [4], principalmente no mapeamento do feromônio. No TSP, o feromônio é depositado nas arestas (caminhos); neste trabalho, o feromônio ( $\tau$ ) é associado aos **nós** (vértices) do grafo  $G=(V,E)$ . O objetivo da formiga  $k$  é construir sequencialmente um conjunto de nós  $C_k \subseteq V$ , onde todo par de vértices em  $C_k$  possui uma aresta, maximizando a cardinalidade  $|C_k|$ .

A construção da clique começa em um nó aleatório. Em cada passo, a formiga  $k$  só pode escolher um nó  $j$  que seja adjacente a **todos** os nós já presentes em sua clique  $C_k$ .

#### **4.2. Transição de Estado e Heurística Dinâmica**

A escolha do próximo nó  $j$  a ser adicionado à clique  $C_k$  é regida pela probabilidade  $P_{ijk}$ , que balanceia a atratividade do nó  $j$  baseada no feromônio ( $\tau_j$ ) e na informação heurística ( $\eta_j$ ). A fórmula de transição de estado é definida por:

$$P_{ij}^k = \frac{(\tau_j)^\alpha \cdot (\eta_j)^\beta}{\sum_{l \in J_i^k} (\tau_l)^\alpha \cdot (\eta_l)^\beta}$$

**Fórmula 1. Probabilidade de Transição**

Onde:

- $\tau_j$  é a quantidade de feromônio no nó  $j$ .
- $\eta_j$  é o valor da heurística local do nó  $j$ .
- $J_i^k$  é o conjunto de nós candidatos elegíveis, ou seja, todos os nós fora de  $C_k$  que são adjacentes a todos os nós em  $C_k$ .
- $\alpha$  e  $\beta$  são os parâmetros que controlam a influência do feromônio e da heurística, respectivamente.

Neste trabalho, foi empregada uma Heurística Dinâmica para  $\eta_j$ . Em vez de usar o grau estático do nó (total de conexões),  $\eta_j$  representa o número de conexões que o nó candidato  $j$  possui com os outros nós candidatos em  $J_k$ . Essa abordagem prioriza nós com maior potencial de expansão da clique no passo atual, aumentando a eficácia da busca local.

#### **4.3. Estratégia de Atualização de Feromônio (Elite Ant System)**

Após todas as  $N$  formigas completarem a construção de suas cliques, o feromônio é atualizado no grafo. Foi adotada a estratégia do Elite Ant System, onde apenas a formiga que encontrou a melhor clique na iteração deposita feromônio. Isso intensifica a busca na região da melhor solução encontrada. A atualização ocorre em duas fases:

##### **4.3.1 Evaporação**

O feromônio de todos os nós é reduzido, simulando a volatilidade do rastro, onde  $\rho$  é a taxa de evaporação (self.decay):

$$\tau_j \leftarrow (1 - \rho) \cdot \tau_j$$

### Fórmula 2. Evaporação do Feromônio

#### 4.3.2 Deposição de Elite

A formiga que encontrou a clique de maior tamanho (Celite) na iteração deposita o feromônio:

$$\tau_j \leftarrow \tau_j + \Delta\tau_j^{elite}$$

### Fórmula 3. Depósito de Elite

Onde  $\Delta\tau_j^{elite}$  é o feromônio depositado no nó j, calculado como:

$$\Delta\tau_j^{elite} = \text{deposit\_strength} \cdot |C_{elite}|$$

### Fórmula 4. Feromônio Depositado

O depósito é realizado apenas nos nós  $j \in \text{Celite}$ . O prêmio é o tamanho da clique, o que recompensa diretamente os nós que fizeram parte das melhores soluções.

## 5. Resultados e Discussão

Esta seção apresenta os resultados obtidos pela execução do Algoritmo de Colônia de Formigas (ACO) em diferentes configurações de grafos. O objetivo é demonstrar a eficácia da abordagem e analisar a curva de convergência da Estratégia de Elite (Elite Ant System) e da Heurística Dinâmica implementadas.

### 5.1 Configuração dos Experimentos

Os experimentos foram realizados utilizando o algoritmo ACO desenvolvido em Python, com a seguinte configuração de parâmetros fixos:

- Influência do Feromônio ( $\alpha$ ): 2.0
- Influência da Heurística ( $\beta$ ): 1.0
- Taxa de Evaporação ( $\rho$ ): 0.5
- Força de Depósito (deposit\_strength): 1.0
- Número de Formigas (Nants): 10
- Número de Iterações: 100

Foram testados três cenários de grafos aleatórios para avaliar a performance do ACO em diferentes complexidades:

Cenário	Nº de Nós	Densidade da Aresta	Complexidade (Esperada)
Pequeno/Denso	20	0.8	Bom para ACO
Médio/Denso	50	0.7	Desafio Intermediário
Grande/Esparsa	100	0.4	Desafio com menos candidatos

## 5.2 Análise de Convergência



**Figura 1 - Convergência do tamanho da clique por iteração**

A Figura 1 demonstra a evolução do tamanho da melhor clique encontrada ao longo das 100 iterações para o Cenário 2 (50 nós, densidade 0.7). Observa-se que a solução converge rapidamente nas primeiras iterações (intensificação), característica da Estratégia de Elite, que foca o depósito de feromônio nos nós que pertenceram à melhor solução local. A estabilização posterior indica que o algoritmo encontrou uma região ótima do espaço de busca.

### **5.3 Comparativo de Performance**

A Tabela 1 resume o desempenho do algoritmo ACO em termos de qualidade da solução (tamanho da clique) e custo computacional (tempo de execução).

Cenário	V	Densidade	Melhor Clique Encontrada	Tempo de execução
1	20	0.8	10	0.44 seg
2	50	0.7	12	0.96 seg
3	100	0.4	8	0.69 seg

### **5.4 Discussão da Heurística**

A utilização da Heurística Dinâmica (Seção 4.2), que prioriza nós candidatos com alto potencial de conexões dentro do conjunto de movimentos possíveis, se mostrou crucial para a exploração eficiente. Em grafos densos (Cenários 1 e 2), essa heurística evitou que as formigas ficassem presas prematuramente, resultando em cliques de maior cardinalidade quando comparado à heurística estática (grau do nó).

## **6. Conclusão**

Este trabalho teve como objetivo aplicar o Algoritmo de Colônia de Formigas (ACO) para solucionar o Problema do Clique Máximo, um problema de natureza NP-difícil que exige o uso de metaheurísticas para obtenção de soluções viáveis em tempo computacional aceitável.

A metodologia proposta utilizou uma abordagem adaptada, mapeando o feromônio para os nós do grafo e implementando duas estratégias para otimizar o processo de busca: uma Heurística Dinâmica que guia a formiga para nós com maior potencial de expansão na clique atual, e o Elite Ant System, que concentra o depósito de feromônio apenas na melhor formiga da iteração.

Os resultados experimentais demonstraram a eficácia e a rápida convergência da abordagem. Em particular, a análise da curva de convergência para o cenário de 50 nós e densidade 0.7 revelou que a solução ótima de 12 nós foi encontrada nas iterações iniciais, o que valida a capacidade do *Elite Ant System* de intensificar rapidamente a busca. Apesar do aumento na complexidade do grafo, o custo computacional permaneceu baixo (menor que 1 segundo para grafos de até 50 nós), confirmando a adequação do ACO como uma solução eficiente para o Problema do Clique Máximo.

Como trabalho futuro, sugere-se a realização de testes com grafos do mundo real (como redes sociais ou biológicas) e a implementação de uma etapa de Busca Local (como o 2-opt ou Busca Tabu) após o término da construção da clique pela formiga, visando refinar a solução e potencialmente alcançar cliques de maior cardinalidade.

## 7. Referências

- [1] SINGH, K. K.; GOVINDA, L. A simple and efficient heuristic algorithm for maximum clique problem. In: IEEE 8TH INTERNATIONAL CONFERENCE ON INTELLIGENT SYSTEMS AND CONTROL (ISCO), Coimbatore, India, 2014. p. 269-273. DOI: 10.1109/ISCO.2014.7103958.
- [2] DORIGO, M.; DI CARO, G. Ant colony optimization: a new meta-heuristic. In: PROCEEDINGS OF THE 1999 CONGRESS ON EVOLUTIONARY COMPUTATION-CEC99 (CAT. NO. 99TH8406). IEEE, 1999. p. 1470-1477.
- [3] CORMEN, T. H.; LEISERSON, C. E.; RIVEST, R. L.; STEIN, C. *Algoritmos: Teoria e Prática*. 4. ed. Rio de Janeiro: LTC, 2024.
- [4] DORIGO, M.; STÜTZLE, T. *Ant Colony Optimization*. Cambridge, MA: The MIT Press, 2004.