Ant Colony Optimization for Graph Coloring Problem

Population-based Metaheuristic

Mohammed Edris Mahdy (446910613)
Mohammed Ahmed Ewida (446910614)

Under the Supervision of:
Prof. Manar Hosny

King Saud University
College of Computer and Information Sciences
Department of Computer Science

December 7, 2025

Outline

Graph Coloring Problem (GCP)

Formal Definition

Given an undirected graph $G = (V, E)$, find a k-coloring function $C : V \rightarrow \{1, 2, \ldots, k\}$ such that:

- $\forall (u, v) \in E : C(u) \neq C(v)$ (adjacent vertices have different colors)
- Minimize $k = \chi(G)$ (chromatic number)

Objective Function

$$\text{Minimize } k \quad \text{subject to} \quad F(C) = \sum_{(u,v) \in E} \mathbb{I}(C(u) = C(v)) = 0$$

Challenge: NP-hard problem - exact methods infeasible for large graphs

Why Ant Colony Optimization?

ACO Key Advantages:

- **Constructive Approach:** Builds solutions from scratch (no initial solution dependency)
- **Collective Intelligence:** Multiple ants explore solution space in parallel
- **Adaptive Learning:** Pheromone-based memory guides search
- **Balanced Exploration-Exploitation:** Natural probabilistic mechanisms
- **Flexibility:** Adapts to different problem structures

Population-based Metaheuristic:

- Multiple solutions per iteration increase diversity
- Parallel execution leverages multi-core processors
- Collective experience accumulates over iterations

ACO Mechanism Overview

Key Components:

1. Pheromone Matrix $\tau[v][c]$
2. Heuristic Information $\eta[c]$
3. Probabilistic Selection
4. Pheromone Update

Probability Formula:

$$P(c) = \frac{[\tau[v][c]]^\alpha \cdot [\eta[c]]^\beta}{\sum_{c'}[\tau[v][c']]^\alpha \cdot [\eta[c']]^\beta}$$

Parameters:

- α: Pheromone importance
- β: Heuristic importance
- ρ: Evaporation rate
- m: Number of ants
- Q: Pheromone deposit

Parallel Execution:

- 82 ants per iteration
- Multi-threading

```
Initialize: pheromone matrix $\tau[v][c] \leftarrow \tau_0$
for iteration = 1 to max_iterations do
    solutions ← []
    for ant = 1 to m (parallel) do
        // Ant Solution Construction
        solution ← {}
        vertices ← shuffle(V)
        for each $v$ in vertices do
            // Sequential vertex coloring
            valid_colors ← {$c$ : no neighbor uses $c$}
            if valid_colors is empty then
                valid_colors ← {new color}
            end if
            $\eta[c] \leftarrow$ 2.0 if $c$ used, else 1.0 // Heuristic
            $P(c) \leftarrow \frac{[\tau[v][c]]^{\alpha} \cdot [\eta[c]]^{\beta}}{\sum_{c'} [\tau[v][c']]^{\alpha} \cdot [\eta[c']]^{\beta}}$ // Probability
            solution[$v$] ← select $c$ with probability $P(c)$
        end for
        solutions.append(solution)
    end for
    best_solution ← argmin(solutions) // Select iteration best
    Evaporate: $\tau[v][c] \leftarrow (1 - \rho) \cdot \tau[v][c]$ for all $v$, $c$
    Deposit: $\tau[v][c] \leftarrow \tau[v][c] + \frac{Q}{\text{num\_colors}}$ for ($v$, $c$) in best_solution
    if no improvement for patience × max_iterations then
        break // Early stopping
    end if
end for
return best solution found
```

Solution Representation

Dictionary mapping: Solution $= \{v_1 : c_1, v_2 : c_2, \ldots, v_n : c_n\}$

Constraint Handling: Preserving Strategy

- Only valid colors considered (no conflicts with neighbors)
- Dynamic pheromone matrix expansion when needed
- **Guarantees:** All solutions are feasible ($F(C) = 0$)

Population Generation

- New population generated at each iteration
- Random starting nodes for diversity
- Random visitation order

Diversification & Intensification

Intensification:

- Pheromone reinforcement on best solutions
- Heuristic preference for used colors ($\eta = 2.0$)
- High α increases exploitation
- High β increases intensification

Diversification:

- Pheromone evaporation $(1 - \rho)$
- Probabilistic (not greedy) selection
- Random node ordering
- Multiple parallel ants
- Low ρ promotes exploration

Balance: Adaptive mechanism between exploration and exploitation

Hyperparameter Tuning with Optuna

Optimization Setup

- **Framework:** Optuna TPE (Tree-structured Parzen Estimator)
- **Trials:** 40 independent trials
- **Parallel Workers:** 6 (n_jobs=6)
- **Duration:** 46 hours wall-clock (225 hours computational)
- **Tuning Dataset:** gc_500_9 (500 vertices, 90% density)
- **Objective:** Minimize total colors

Tuning Results

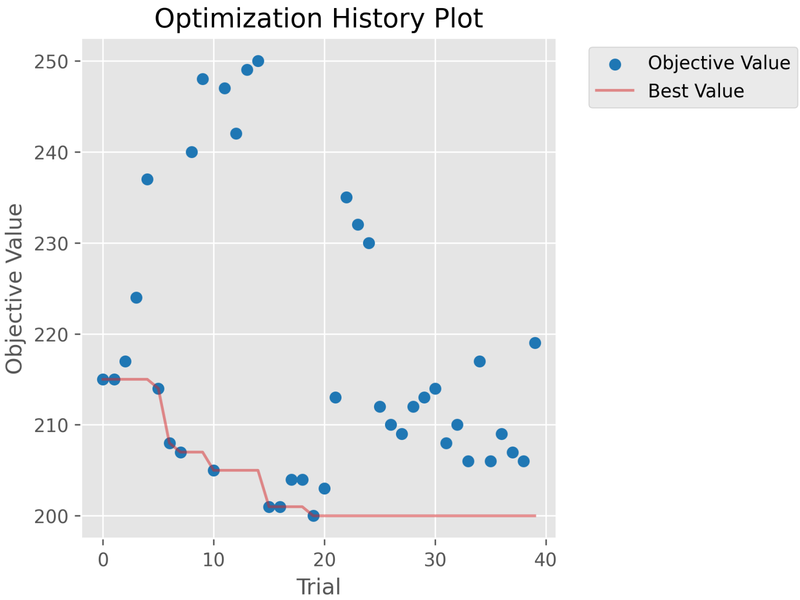
- Started at **215 colors**
- Best trial (Trial 19): **200 colors**
- **Improvement:** 15-color reduction (7%)

Optimal Parameters

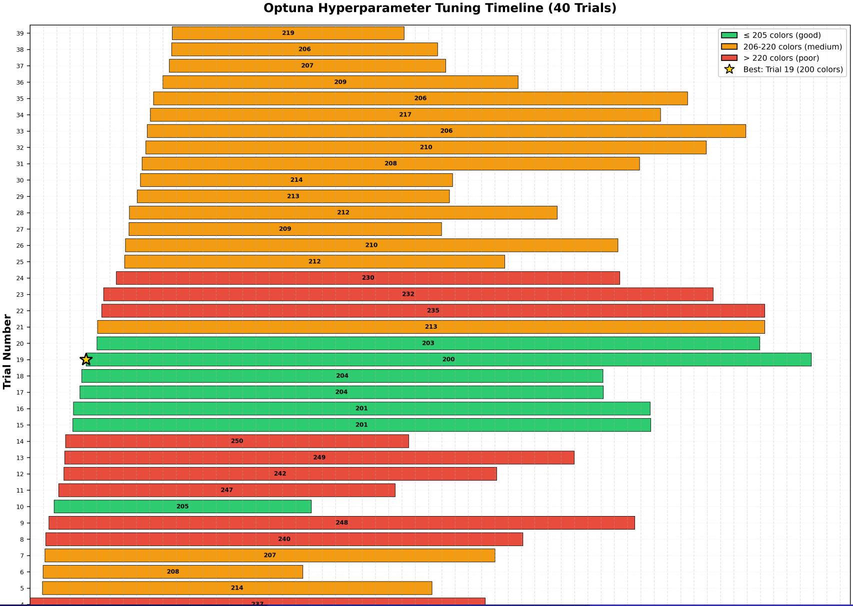
Parameter	Search Range	Optimal Value
Iterations (T)	[200, 500]	261
Alpha (α)	[0.5, 2.0]	1.536
Beta (β)	[1.0, 10.0]	**5.966**
Rho (ρ)	[0.01, 0.5]	**0.097**
Ant Count (m)	[20, 100]	82
Pheromone Deposit (Q)	[0.1, 5.0]	1.299
Patience Ratio	[0.3, 0.8]	0.577

Key Findings: High β (heuristic) and low ρ (evaporation)
$\rightarrow$ Intensification dominates over diversification

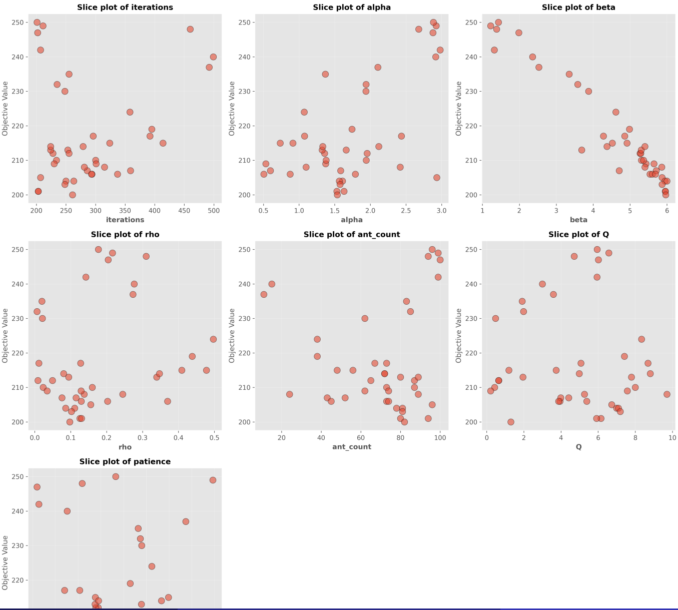
Optimization History



Optimization Timeline



Parameter Analysis



Parameter Importance

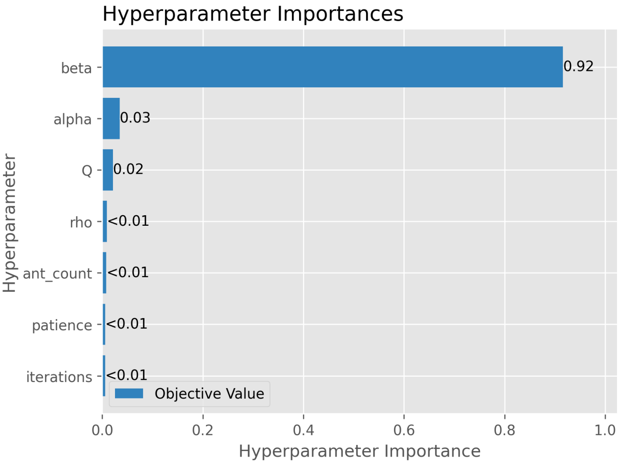


Figure: Iterations and Beta dominate performance; Rho has minimal impact

Best Trial Visualization

Benchmark Datasets

Instance	Vertices	Edges	Density	BKS
dsjc250.5	250	31,336	50.3%	28
dsjc500.9	500	224,874	90.0%	126
dsjc1000.5	1000	249,826	50.0%	85

Test Protocol:

- DIMACS benchmark instances
- 3 independent runs per instance
- Statistical robustness assessment

Three-Way Performance Comparison

Instance	Algorithm	Best	Avg.	Std.	Time (s)	Dev. (%)
dsjc250.5	Greedy	42	42.0	0.00	0.04	50.0
	Tabu Search	34	34.7	0.58	925.7	21.4
	ACO	**55**	**55.7**	**0.58**	**1134.4**	**96.4**
dsjc500.9	Greedy	174	174.0	0.00	0.24	38.1
	Tabu Search	164	166.0	2.00	6730.9	30.2
	ACO	**199**	**201.3**	**2.52**	**5853.5**	**57.9**
dsjc1000.5	Greedy	125	125.0	0.00	0.73	47.1
	Tabu Search	124	124.7	0.58	10026.2	45.9
	ACO	**226**	**227.0**	**1.00**	**13669.6**	**165.9**

Key Finding: ACO underperforms both Greedy and Tabu Search
Deviation: 58-167% vs BKS (TS: 21-46%, Greedy: 38-50%)

Best Colors Comparison

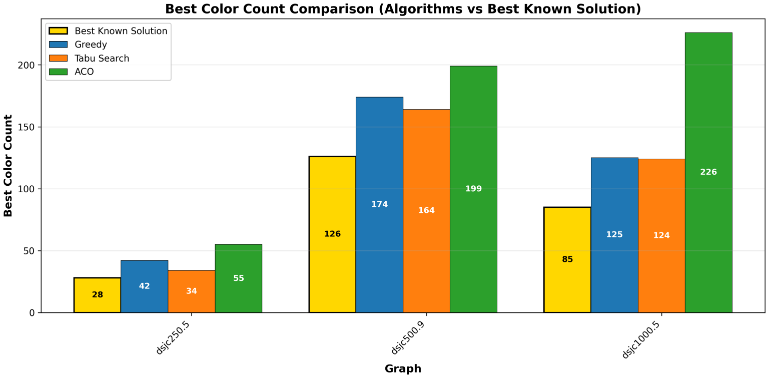


Figure: ACO produces 58-166% more colors than BKS

Average Colors Comparison

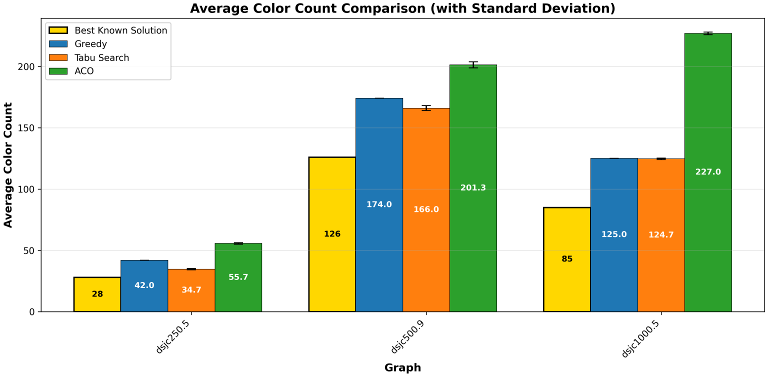


Figure: Performance gap widens with graph size: 61% (250v) → 82% (1000v)

Execution Time Comparison

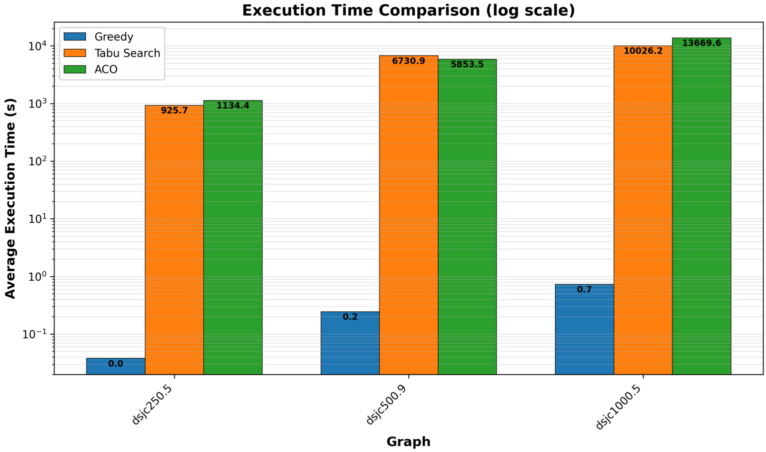


Figure: ACO time comparable to TS despite 82 parallel ants

Algorithm Comparison Summary

Greedy (Baseline)

Strength: Extremely fast (¡ 1 second)
Weakness: Moderate quality (38-50% deviation)
Use Case: Rapid prototyping, time-critical applications

Tabu Search (Assignment 2)

Strength: Best solution quality (21-46% deviation)
Weakness: High computational cost (926-10,026 seconds)
Use Case: Quality-critical applications with time budget

ACO (This Work)

Strength: Consistent, no initial dependency
Weakness: Poorest quality (58-167% deviation), comparable time to TS
Current Status: Not recommended in present form

Why ACO Underperforms

1. **Constructive Nature:**
   - Builds from scratch vs TS starts with 40 colors and improves to 35
   - Must discover good assignments purely through pheromone learning
2. **Local Construction:**
   - Ants make locally optimal decisions without global view
   - TS has global view of all conflicts
3. **Pheromone Sparsity:**
   - Large color sets (50-200 colors) dilute pheromone signals
   - TS's tabu list prevents specific moves without dilution

Scalability Issue: Performance degradation increases with graph size (96% deviation on 250v $\rightarrow$ 167% on 1000v)

Key Insights

Quality-Time Trade-off

- **Greedy:** Fast but moderate quality
- **Tabu Search:** Excellent quality, high cost
- **ACO:** Poor quality despite comparable cost to TS

Paradigm Suitability

- **Population-based $\neq$ automatically better**
- Improvement-based (TS) benefits from feasible starting solutions
- Constructive methods (ACO) must learn from scratch
- Problem structure determines paradigm suitability

Tuning Insights

High β and low ρ essential $\rightarrow$ **Intensification ¿ Diversification**
Questions whether ACO's pheromone learning is well-suited for GCP

Contributions

1. **Implementation:**
   - Constructive ACO with dynamic pheromone matrix
   - 82 parallel ants using multi-threading
2. **Comprehensive Tuning:**
   - Optuna framework: 40 trials, 6 parallel workers, 46 hours
   - 15-color improvement (215 $\rightarrow$ 200 colors)
3. **Empirical Comparison:**
   - Three-way analysis: Greedy vs TS vs ACO
   - DIMACS benchmarks (250-1000 vertices)
   - Reveals constructive vs improvement-based trade-offs
4. **Insights:**
   - Population-based methods not inherently superior
   - For GCP: heuristics and search time ¿ pheromone dynamics
   - No universally best algorithm - problem-dependent

Future Work

Potential Enhancements:

- Improved heuristics (degree-based, conflict-prediction)
- Hybrid approaches (TS-quality initialization for pheromones)
- Local search post-processing
- Alternative pheromone models (edge-color instead of node-color)
- Adaptive parameter control

Research Directions:

- Investigation of why pheromone learning fails for GCP
- Comparison with other population-based methods (GA, PSO)
- Problem structure analysis for metaheuristic selection

Thank You!

Questions?