

**VIETNAM NATIONAL UNIVERSITY HO CHI MINH CITY
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY
FACULTY OF COMPUTER SCIENCE AND ENGINEERING**



**REPORT
CAPSTONE PROJECT
SEMESTER 241 ACADEMIC YEAR 2023-2024**

<TITLE>

MAJOR: COMPUTER SCIENCE

COUNCIL: <NAME>

SUPERVISOR(s): ...

REVIEWER:

——o0o——

STUDENT 1: <NAME> - <ID>

STUDENT 2: <NAME> - <ID>

HO CHI MINH CITY, December 2024

Chapter 1

Introduction (Ân)

Overview, objectives and structure of the report.

1.1 Motivation

This work addresses the path planning problem on grid maps using Ant Colony Optimization (ACO). Classical ACO implementations for grid-based navigation can suffer from slow convergence, path jaggedness (unsuitable for continuous motion), and suboptimal exploitation/exploration balance. These limitations affect real-world use-cases such as mobile robot navigation, autonomous ground vehicles in structured environments, and automated guided vehicles (AGVs) in warehouses.

Prior approaches include standard ACO variants, heuristic-only planners (A^* , Dijkstra), and sampling-based planners (RRT, PRM). Strengths and weaknesses: - A^* / Dijkstra: exact on discrete grids but can produce unnatural paths and do not generalize well when using learned heuristics. - Sampling planners: good for continuous spaces but may be inefficient in dense discrete grids. - Baseline ACO: good at exploring multiple candidate paths but sensitive to parameter choices and produces discrete, jagged routes.

The team selected a set of complementary improvements (cone-shaped pheromone initialization, adaptive pheromone/heuristic factors, division of labor among ants, backtracking heuristics, B-spline smoothing, and coordinated parameter fine-tuning) because together they: 1. Introduce gentle directional guidance without overconstraining exploration. 2. Adapt search behavior during optimization to avoid premature convergence.

3. Allow role specialization to improve search efficiency. 4. Produce geometrically shorter, smooth, and executable trajectories suitable for real robotic platforms.

1.2 Goals

The primary goals of this project are:

1. Improve ACO convergence speed and robustness on grid benchmarks.
2. Reduce geometric path length (Euclidean) rather than node count.
3. Produce smooth, continuous paths suitable for motion controllers through post-processing.
4. Design complementary improvements that operate together without destructive interference.
5. Provide reproducible experiments and clear comparison to the baseline.

1.3 Scope

This report focuses on algorithmic improvements to ACO for static, known grid maps.

The work:

- Targets offline path planning and benchmark evaluation (no online dynamic re-planning).
- Uses occupancy-grid style maps provided in the project's maps/ folder.
- Evaluates improvements by numerical experiments (multiple runs) on benchmark maps.
- Does not address sensor noise, moving obstacles, or low-level control execution.

1.4 Thesis Structure

The report is organised as follows:

- Chapter 1: Introduction, motivation, goals and scope.
- Chapter 2: Theoretical Background, fundamentals of Ant Colony Optimization and relevant theory.
- Chapter 3: Overview and Approach, high-level description of the implemented ACO and the set of proposed improvements.
- Chapter 4: Detailed Improvements, dataset/benchmark description and per-improvement: idea, implementation, benefits, and experimental results.
- Chapter 5: Optional Team Improvements, additional ideas and extensions.
- Chapter 6: Conclusion, summary and future work.

Chapter 2

Overview and Approach (Phước)

2.1 ACO overview

This section provides a concise overview of Ant Colony Optimization as applied to grid path planning:

- Ant Colony Optimization basics: pheromone trails, heuristic information, probabilistic transition rules, pheromone evaporation and deposit.
- Heuristic design for grid navigation: distance heuristics, admissibility considerations when combining with ACO.
- Convergence issues: premature convergence, stagnation, and common mitigation strategies.
- Path quality metrics: node-count vs. Euclidean (geometric) length and runtime/statistical metrics for evaluation.

2.2 Overview of our approach

We present a modular set of improvements designed to be composable:

1. Cone pheromone initialization, adds directional bias toward goal.
2. Adaptive pheromone/heuristic factors dynamically adjust α and β over iterations.

3. Division of labor and role assignment (explorers vs exploiters) with smooth transitions.
4. Backtracking heuristics □ allow ants to recover from dead-ends efficiently.
5. Smooth-line (B-spline) post-processing convert discrete waypoint lists into continuous trajectories.
6. Finetune coordinated parameters so all features work together.

A high-level algorithmic workflow is given, and detailed implementation notes are provided in Chapter 5.

Chapter 3

Detailed Improvements (Chung)

Dataset / benchmark description and detailed description for each improvement.

3.1 Dataset / Benchmark maps (Ân)

Describe the benchmark maps used for evaluation:

- Source: project maps in the repository (e.g. `maps/map1.txt`, `maps/map2.txt`, `maps/map3.txt`).
- Map format: ASCII grid with markers: S (start), F (goal), E (empty/free), O (obstacle).
- Evaluation protocol: run each configuration multiple times (e.g. 20 runs), record min/mean/std of path length (Euclidean), runtime, and success rate.
- Preprocessing: conversion to occupancy map and node graph (diagonal connectivity allowed when free).

3.2 Cone Pheromone (Ân)

3.2.1 Idea

Initialize pheromone distribution with a cone-shaped bias pointed toward the goal. This provides gentle directional guidance without forcing a fixed path.

3.2.2 Implementation

Compute initial pheromone for each node/edge using a combination of a normalized cone term and an inverse-distance term:

$$\tau_0 = \tau_{\text{base}} + c \cdot \frac{|x - y|}{L} + \frac{1}{d}$$

where c is a small coefficient, L is map scale and d is Euclidean distance to goal.

3.2.3 Benefits

Faster convergence, lower variance across runs, maintains exploration while guiding ants toward promising regions.

3.2.4 Experimental results

Insert per-map quantitative results here: min/mean/std path length and runtime compared to baseline.

3.3 Adaptive Pheromone / Heuristic Factors (Nhur)

3.3.1 Idea

Dynamically adapt α and β during iterations to encourage exploration early and exploitation later.

3.3.2 Implementation

Use a smooth schedule (e.g. quadratic/integral schedule):

$$\alpha'(n) = \alpha + \xi \left(\frac{n}{N} \right)^2 / 2, \quad \beta'(n) = \beta + \xi \left(\frac{n}{N} \right)^2 / 2$$

Tune ξ to control adaptation strength.

3.3.3 Benefits

Reduces premature convergence and balances exploration/exploitation.

3.3.4 Experimental results

Include comparison tables/plots showing effect of adaptive scheduling on path quality and variance.

3.4 Division of Labor (Author: Thương)

3.4.1 Idea

Assign dynamic roles to ants (e.g. explorers vs exploiters) with a smooth transition based on iteration progress and convergence metrics.

3.4.2 Implementation

Define a sigmoid-based mixing coefficient to determine role proportions. Configure different (α, β) scalings per role.

3.4.3 Benefits

Improves runtime efficiency and can speed convergence by leveraging specialized search behaviours.

3.4.4 Experimental results

Report trade-offs: faster runtime vs. potential increase in variance; include runtimes and path statistics.

3.5 Backtracking (Phước)

3.5.1 Idea

Allow ants to perform controlled backtracking when encountering dead-ends or low-quality branches instead of restarting immediately.

3.5.2 Implementation

Implement a lightweight backtrack policy that revisits recent nodes with probability based on local pheromone/heuristic cues and a maximum backtrack depth.

3.5.3 Benefits

Reduces wasted exploration steps, increases probability of discovering valid routes in constrained regions.

3.5.4 Experimental results

Include measurements showing success rate improvements on maps with narrow corridors or many obstacles.

3.6 Smooth Line (B-spline) (Thương)

3.6.1 Idea

Post-process discrete grid paths using cubic B-spline interpolation to create continuous, smooth trajectories suitable for controllers.

3.6.2 Implementation

Convert grid waypoint sequence to control points, apply cubic B-spline with chosen smoothing factor and sample densely to form the final trajectory.

3.6.3 Benefits

Removes sharp corners, yields C2-continuous curves, increases point density for motion planning.

3.6.4 Experimental results

Provide before/after figures and quantitative change in curvature/point density.

3.7 Finetuning Parameters (\hat{A}_n)

3.7.1 Idea

Coordinate parameter choices so multiple improvements work together without interference.

3.7.2 Implementation

Provide a parameter table and tuning strategy:

- Ant count, iterations, evaporation factor, pheromone constant.
- Specific coefficients for cone term, adaptive schedule ξ , role scalings, smoothing factor.

3.7.3 Benefits

Improved combined performance (balanced quality, run-time, and consistency).

3.7.4 Experimental results

Report results of combined configuration (“Mix All”) vs single-improvement runs.

Chapter 4

Additional Team Improvements (Optional) (Như - dự bị)

4.1 Optional Team Improvements

Use this section to document any additional experiments or features the team wants to explore (e.g. multi-objective fitness, dynamic map handling, GPU acceleration, or learned heuristics).

Chapter 5

Conclusion (Ân)

5.1 Conclusion

This work improves Ant Colony Optimization (ACO) for grid-based path planning by introducing a set of complementary algorithmic and post-processing enhancements. The project focused on: (1) cone-shaped pheromone initialization to provide gentle directional guidance; (2) adaptive pheromone/heuristic scheduling to balance exploration and exploitation over iterations; (3) division of labor to specialise ant roles for speed and robustness; (4) controlled backtracking to recover from dead-ends; (5) B-spline smoothing to produce continuous executable trajectories; and (6) coordinated parameter fine-tuning so all features work together.

Key findings

- All proposed configurations maintain the optimal minimum path found on benchmark maps while improving secondary metrics (consistency, mean path quality, runtime) compared to the baseline.
- The cone-pheromone initialization yields the best mean path length and the lowest variance across runs, making it the best single improvement for reproducible quality.
- Division of labor provides the largest runtime improvement at the cost of higher variance, making it appropriate for time-critical scenarios.

- The combined configuration (“Mix All”) achieves the best balance between quality, consistency and runtime, demonstrating synergy among the improvements.
- B-spline smoothing converts discrete grid paths into smooth, high-density trajectories suitable for real-world motion controllers.

Limitations

- Experiments were performed on static, known occupancy-grid maps; dynamic or partially-observed environments were not evaluated.
- Parameter sensitivity remains: although coordinated tuning reduces conflicts, further automated tuning (e.g. Bayesian optimization) may improve robustness.
- Low-level control and execution (actuator models, vehicle dynamics) are out of scope and require integration and validation on target platforms.

Practical recommendations

- For quality-critical deployments (robotics, AGVs): enable cone-pheromone and Euclidean-distance fitness, apply B-spline smoothing.
- For time-critical applications: enable division of labor; accept slightly higher variance for faster results.
- For general use: enable the combined configuration (Mix All) with the tuned parameter set provided in the implementation notes and appendices.

Future work

- Extend evaluation to dynamic maps and moving obstacles with online re-planning.
- Integrate learning-based heuristics or learned initial pheromone fields to further reduce tuning effort.
- Validate the full pipeline on a physical robot or high-fidelity simulator to measure real-world execution performance.
- Add automated parameter search and multi-objective optimization (trade-off between runtime, smoothness and path length).

In summary, the presented enhancements make ACO a more practical and robust option for grid-based path planning. The modular design allows selective activation of improvements depending on application constraints (quality vs. speed), and the smoothing stage produces trajectories ready for downstream control.

5.2 References

Bibliography

- [1] Richard H. Bartels, John C. Beatty, and Brian A. Barsky. *An Introduction to Splines for Use in Computer Graphics and Geometric Modeling*. Morgan Kaufmann, 1987.
- [2] Marco Dorigo and Thomas Stützle. *Ant Colony Optimization*. MIT Press, 2004.
- [3] P. E. Hart, N. J. Nilsson, and B. Raphael. “A Formal Basis for the Heuristic Determination of Minimum Cost Paths”. In: *IEEE Transactions on Systems Science and Cybernetics* 4.2 (1968), pp. 100–107.
- [4] Thomas Stützle and Holger H. Hoos. “A Max–Min Ant System for Combinatorial Optimization”. In: *Future Generation Computer Systems* 16.8 (2000), pp. 889–914.