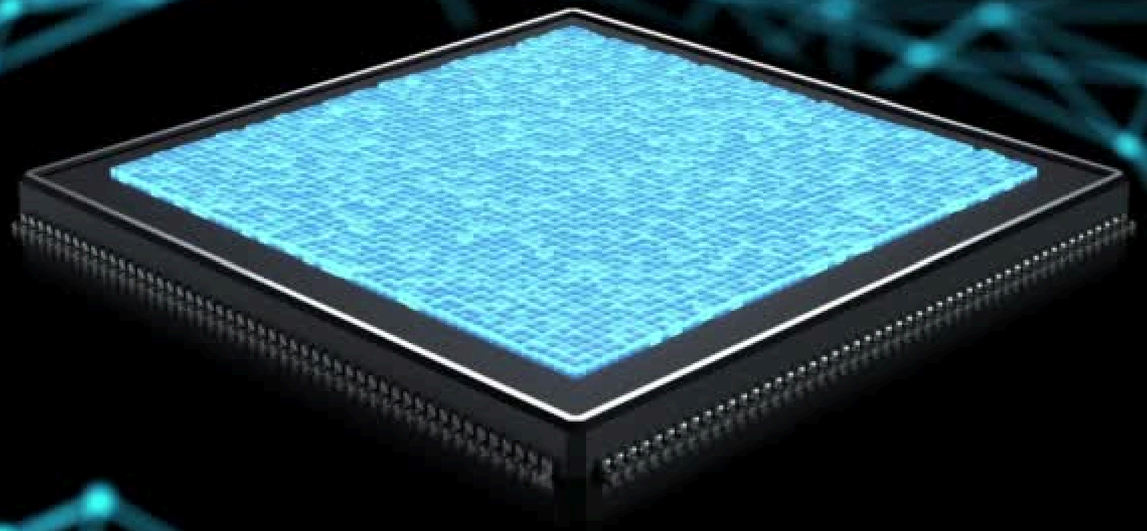
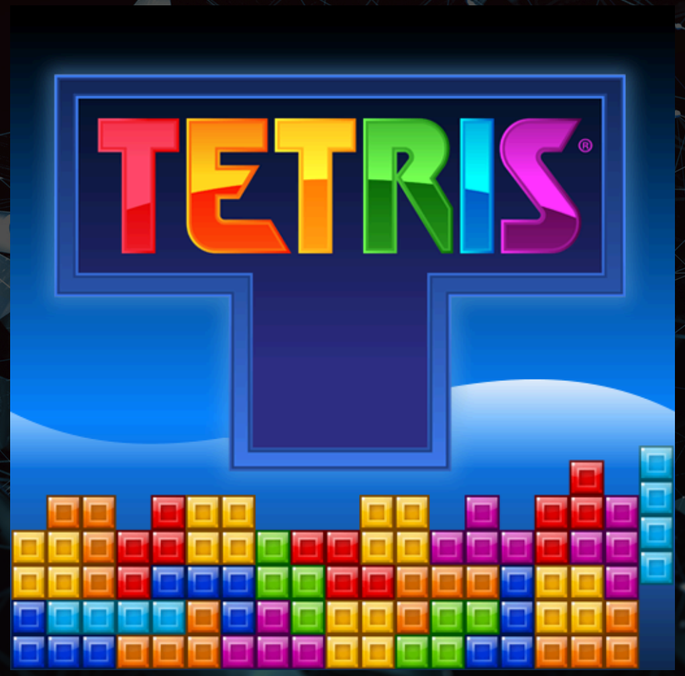
Computational Cognitive Science Project



Team Members

Moaz Gehad
Abdelrahman Ibrahim
Mahmoud Ehab Helmy
Rana Esmail Zekery
Menna Tullah Hany
Nadine Hazem

Part I: Genetic Algorithm Optimization for Tetris



01 INTRODUCTION TO TETRIS GAME

02 UNDERSTANDING GENETIC ALGORITHM TECHNIQUE

03 CONTRIBUTION FACTORS

04 GUI DEMONSTRATION

05 BEST RESULTS & TEST

06 CONCLUSIONS

THE TETRIS GAME OVERVIEW

INTERESTING FOR AI

Has a vast number of possible board states

Success depends on both immediate and future consequences of actions

Requires fast, real-time decision-making

TETRIS IS A CLASSIC PUZZLE VIDEO GAME WHERE PLAYERS MANIPULATE FALLING BLOCKS CALLED TETROMINOES TO FILL HORIZONTAL LINES ON A 10×20 GRID. EACH BLOCK IS MADE OF 4 CONNECTED SQUARES, AND CAN BE ROTATED OR MOVED LEFT/RIGHT BEFORE IT LANDS.
WHEN A LINE IS FULLY FILLED, IT DISAPPEARS AND THE PLAYER SCORES POINTS. THE GAME BECOMES FASTER OVER TIME, INCREASING THE CHALLENGE. THE OBJECTIVE IS TO SURVIVE AS LONG AS POSSIBLE WHILE MAXIMIZING THE SCORE.

Part II: Ant Colony Optimization for TSP

01 INTRODUCTION TO THE TSP PROBLEM

02 UNDERSTANDING ANT COLONY OPTIMIZATION

03 IMPLEMENTATION APPROACH

04 GUI DEMONSTRATION

05 ALGORITHM ANALYSIS RESULTS

06 CONCLUSIONS

THE TRAVELING SALESMAN PROBLEM

DEFINITION

Finding the shortest possible route that visits each city exactly once and returns to the origin

REAL-WORLD APPLICATIONS

Logistics, planning, manufacturing, DNA sequencing

MATHEMATICAL FORMULATION

Presentations are communication tools that can be used as lectures.

COMPLEXITY

NP-hard problem - no efficient exact algorithm for large instances

Traditional Solutions to TSP

Exact algorithms

Branch and bound, dynamic programming Guaranteed optimal solution but exponential time complexity

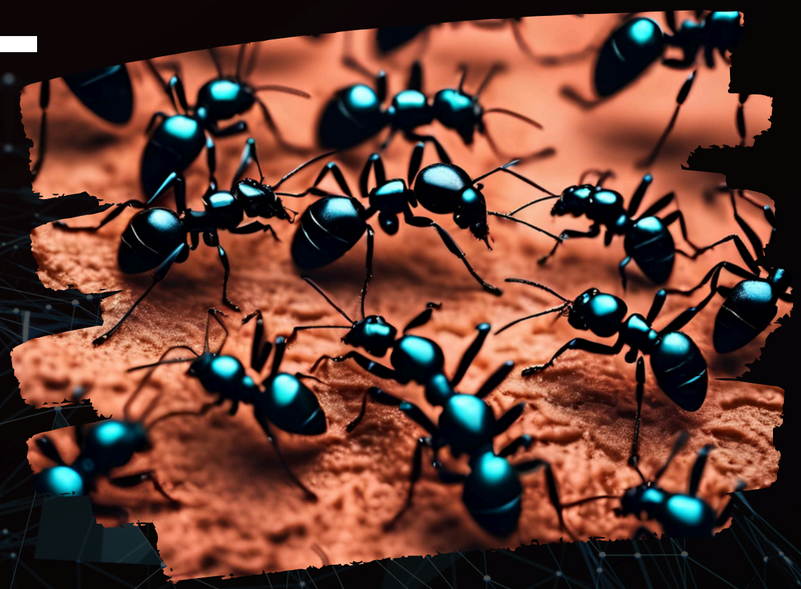
Heuristic approaches

Nearest neighbor, greedy algorithms Fast but often suboptimal

Metaheuristics

Genetic algorithms, simulated annealing Balance between solution quality and computational cost

Ant Colony Optimization – Inspiration



ACO ADVANTAGE

Nature-inspired, good balance of quality & speed

INSPIRATION

Ants lay pheromones to find shortest paths

ACO Algorithm Overview

How It Works

Virtual ants build solutions based on pheromone trails & distance

Probability Formula:

$$p_{ij}^{k}(t) = \frac{[\tau_{ij}(t)]^{\alpha} \cdot [\eta_{ij}]^{\beta}}{\sum_{l \epsilon J_i^k} [\tau_{il}(t)]^{\alpha} \cdot [\eta_{il}]^{\beta}}$$

Core Parameters

α: Pheromone influence
β: Heuristic (greediness)
ρ: Evaporation rate
Q: Pheromone quantity

Key Classes

GRAPH

Generates distances, computes path cost

ANT

Builds path using probabilities

COLONY

Manages ants, updates pheromones, tracks best path

Partial vs Full Path Construction

1.STRATEGY COMPARISON:

- **PARTIAL PATH**

  Higher exploration, potentially better quality

- **FULL PATH**

  Simpler implementation, faster execution

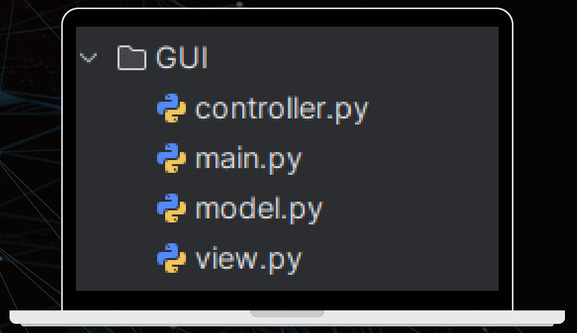
2.PERFORMANCE TRADEOFFS:

- SOLUTION QUALITY VS COMPUTATIONAL EFFICIENCY
- EARLY CONVERGENCE VS THOROUGH SEARCH

SYSTEM ARCHITECTURE

01 PATTERN:

Model-View-Controller (MVC)

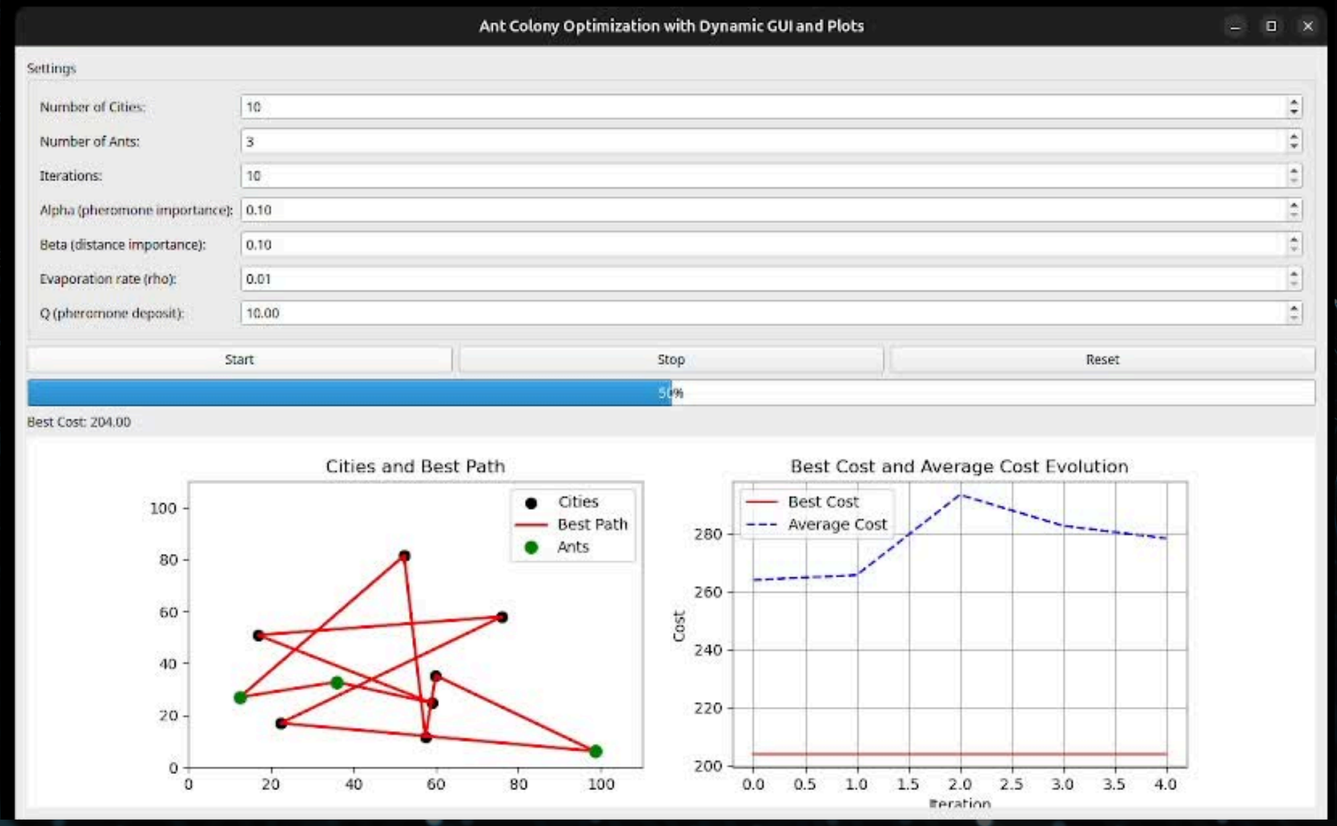


02 COMPONENTS

- Model: ACO algorithm (Graph, Ant, Colony)
- View: PyQt6 GUI
- Controller: Connects logic and interface

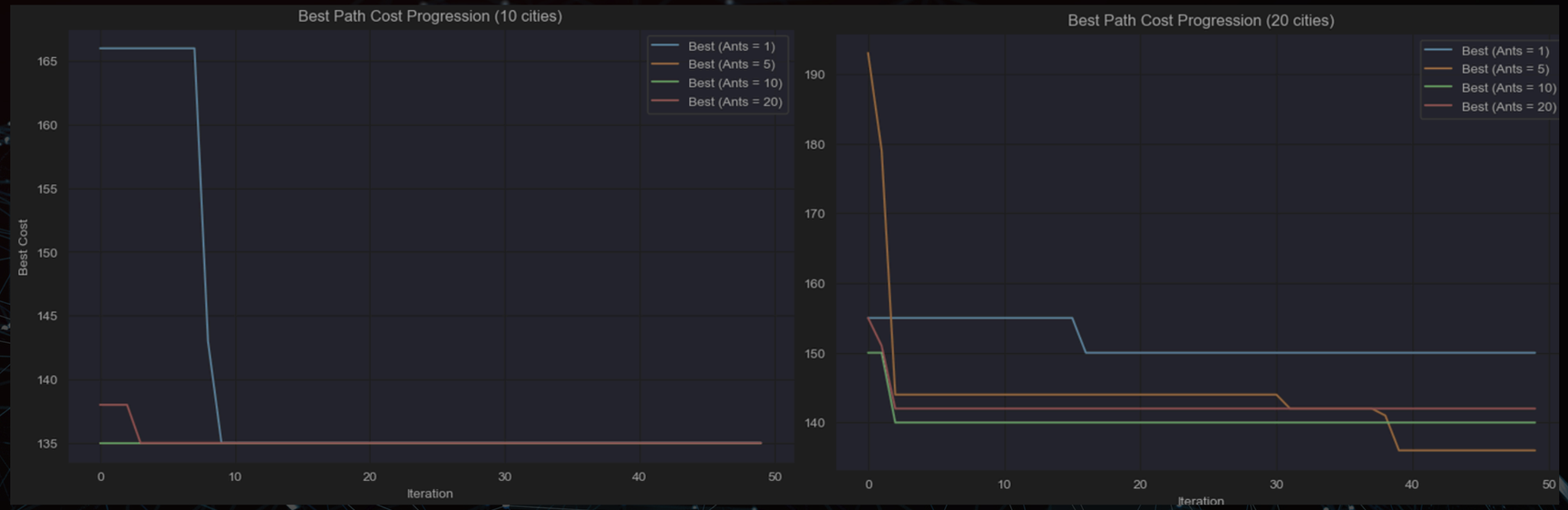
GUI Features

- **Interactive Controls:** Set α, β, ρ, ants, cities
- **Visualizations:**
  - City map & current best path
  - Real-time path updates
  - Cost progression graph



Results

FINDINGS



- **COLONY SIZE:**

  LARGER COLONIES = BETTER RESULTS, BUT DIMINISHING RETURNS

- **PROBLEM SIZE:**

  MORE CITIES = SLOWER CONVERGENCE, HIGHER VARIANCE

CONCLUSIONS

- VISUAL INTERFACE AIDS COMPREHENSION
- ACO PERFORMS WELL ON MEDIUM-SIZED TSP INSTANCES
- EFFECTIVE BALANCE OF PARAMETERS CRITICAL FOR PERFORMANCE

Q&A

Repository link: Optimization-Algorithms-Tetris-TSP

Scan me