Metaheuristic Algorithms for Spatial Optimization in GIS using Python

Yasin Mohammadi

March 3, 2025

Course Overview

- Optimization and heuristics
- Classification of metaheuristic algorithms
- Single vs. multi-objective optimization
- Applications in GIS

GA Fundamentals

- Evolution and natural selection
- Selection, Crossover, Mutation
- Encoding spatial data
- GIS applications (clustering, land-use optimization)

GA Implementation

- Implementing GA in Python

ACO Fundamentals

- Pheromone update and evaporation
- ACO for shortest path optimization
- Route optimization in GIS

PSO Fundamentals

- Swarm intelligence and movement
- PSO for spatial clustering and classification

Nature-Inspired Algorithms

- Bee Algorithm for resource allocation
- Firefly Algorithm, Cuckoo Search, Bat Algorithm

Multi-Objective Optimization

- Pareto Front and NSGA-II
- Trade-offs in GIS (cost vs. environmental impact)

Applications in GIS

- Land-use planning with GA
- Traffic flow optimization with ACO
- Environmental monitoring with PSO

Course Project

- Choose a real-world GIS problem
- Apply a metaheuristic algorithm
- Example projects: wildfire evacuation, transportation network optimization

Introduction

Genetic Algorithm (GA) is a search heuristic inspired by natural selection. It is used to solve optimization problems.

Key Steps:

- Initialize population
- Evaluate fitness
- Select parents
- Perform crossover and mutation
- Repeat until convergence

Example 1: Maximizing $f(x) = x^2$

Problem: Find x (0 to 15) that maximizes $f(x) = x^2$.
Solution Representation: 4-bit binary string.
Steps:

1. Initialize population (e.g., 1010, 0111, 1100, 0011).
2. Evaluate fitness (e.g., $10^2 = 100$, $7^2 = 49$, etc.).
3. Select parents using roulette wheel selection.
4. Perform crossover (e.g., $1100 + 1010 \to 1110$ and 1000).
5. Repeat until convergence (optimal solution: $x = 15$).

Example 1: Fitness Evaluation

Individual	Binary	x	Fitness (x^2)
1	1010	10	100
2	0111	7	49
3	1100	12	144
4	0011	3	9

Total Fitness: $100 + 49 + 144 + 9 = 302$

Example 1: Crossover

Parents:

- Parent 1: 1100 (12)
- Parent 2: 1010 (10)

Crossover Point: 2

Offspring:

- Offspring 1: 11—00 → 1110 (14)
- Offspring 2: 10—10 → 1000 (8)

Example 2: Knapsack Problem

Problem: Select items with maximum value without exceeding 15 kg capacity.
Items:

- Item 1: 2 kg, $10
- Item 2: 4 kg, $20
- Item 3: 6 kg, $30
- Item 4: 8 kg, $40

Solution Representation: 4-bit binary string ($1 =$ include, $0 =$ exclude).

Example 2: Fitness Evaluation

Individual	Binary	Items	Weight	Fitness
1	1010	1, 3	8 kg	$40
2	0101	2, 4	12 kg	$60
3	1100	1, 2	6 kg	$30
4	0011	3, 4	14 kg	$70

Total Fitness: $40 + 60 + 30 + 70 = 200$

Example 2: Crossover

Parents:

- Parent 1: 0011 (Items 3, 4)
- Parent 2: 0101 (Items 2, 4)

Crossover Point: 2

Offspring:

- Offspring 1: 00—11 $\rightarrow$ 0001 (Items 1, 4)
- Offspring 2: 01—01 $\rightarrow$ 0111 (Items 2, 3, 4)

Conclusion

Key Takeaways:

- GA is a powerful optimization tool inspired by natural selection.
- It works well for problems with large search spaces.
- Examples demonstrated: Maximizing $f(x) = x^2$ and solving the knapsack problem.

Problem Definition

Knapsack Capacity: 15 kg
Items:

Item	Weight (kg)	Value ($)
1	2	10
2	4	20
3	6	30
4	8	40

Objective: Select a subset of items with maximum total value without exceeding the 15 kg capacity.

Solution Representation

Representation: 4-bit binary string ($1 =$ include, $0 =$ exclude).
Example:

- 1010: Include Item 1 and Item 3.
- 0101: Include Item 2 and Item 4.

Step 1: Initialize Population

Population Size: 4
Initial Population:

Individual	Binary	Items Selected
1	1010	1, 3
2	0101	2, 4
3	1100	1, 2
4	0011	3, 4

Step 2: Evaluate Fitness

Fitness Function:

- If total weight le 15 kg: Fitness = Total value.
- If total weight gt 15 kg: Fitness = 0 (invalid solution).

Fitness Calculation:

Individual	Binary	Items	Weight (kg)	Fitness ($)
1	1010	1, 3	8	40
2	0101	2, 4	12	60
3	1100	1, 2	6	30
4	0011	3, 4	14	70

Total Fitness: $40 + 60 + 30 + 70 = 200$

Step 3: Selection

Method: Roulette Wheel Selection

Selection Probabilities:

- Individual 1: $40/200 = 20\%$
- Individual 2: $60/200 = 30\%$
- Individual 3: $30/200 = 15\%$
- Individual 4: $70/200 = 35\%$

Selected Individuals:

- Individual 4 (70)
- Individual 2 (60)
- Individual 4 (70)
- Individual 1 (40)

Step 4: Crossover

Method: Single-Point Crossover
Crossover Point: 2
Parents:

- Parent 1: 0011 (Items 3, 4)
- Parent 2: 0101 (Items 2, 4)

Offspring:

- Offspring 1: 00—11 $\rightarrow$ 0001 (Items 1, 4)
- Offspring 2: 01—01 $\rightarrow$ 0111 (Items 2, 3, 4)

Step 5: Mutation

Mutation Probability: 1% per bit
Example: No mutation occurs in this generation.
New Population:

Individual	Binary	Items Selected
1	0001	1, 4
2	0111	2, 3, 4
3	0010	3
4	1011	1, 3, 4

Step 6: Evaluate New Population

Fitness Calculation:

Individual	Binary	Items	Weight (kg)	Fitness ($)
1	0001	1, 4	10	50
2	0111	2, 3, 4	18	0
3	0010	3	6	30
4	1011	1, 3, 4	16	0

Total Fitness: $50 + 0 + 30 + 0 = 80$

Step 7: Repeat Until Convergence

Stopping Condition: Maximum generations or optimal solution found.
Optimal Solution: 0101 (Items 2 and 4) with total value = \$60 and total weight = 12 kg.
Conclusion: The population evolves to the optimal solution over generations.

Summary

Steps in Genetic Algorithm:

1. Initialize population.
2. Evaluate fitness.
3. Select parents.
4. Perform crossover and mutation.
5. Repeat until convergence.

Example 2 Result: The GA successfully found the optimal subset of items (Items 2 and 4) with a total value of $60.

Thank You!