BLM22332E, BLM20364E Heuristic Optimization Algorithms

Final Exam as Term Project: Optimization with Genetic Algorithms

Due Date: ……..
Submission Format: *ID_Name_Surname.zip (Report (PDF) + Source Code (ZIP) + Presentation)*

1. Project Overview

In this term project, you will implement a Genetic Algorithm (GA) to solve one of three classic optimization problems: Graph Coloring, Knapsack, or Traveling Salesman Problem (TSP). Working in groups of 2-3 members, you will develop, optimize, and analyze your solution for a specific problem instance.

2. Group Formation

- Each project group must have exactly 2-3 members (no exceptions)
- Groups must register via the provided Google Sheet
- Problem instances will be randomly distributed evenly (approximately 33% of groups per problem type)

3. Problem : ONE of the following optimization problems:

I. Graph Coloring Problem

Problem Definition: Assign colors to vertices of a graph such that no adjacent vertices share the same color.
- Objective: Minimize the number of colors used
- Problem Instances: DIMACS benchmark instances (select from provided list)

II. Knapsack Problem

- Problem Definition: Select items with given weights and values to maximize total value while respecting capacity constraints.
- Objective: Maximize total value while keeping total weight under capacity
- Problem Instances: OR-Library instances (select from provided list)

III. Traveling Salesman Problem (TSP)

- Problem Definition: Find the shortest possible route visiting each city exactly once and returning to the origin.
- Objective: Minimize total distance
- Problem Instances: TSPLIB instances (select from provided list)

4. Project Requirements

a. Problem Formulation

- Define chromosome representation appropriate for your chosen problem
- Implement the objective function
- Understand the structure and constraints of your selected problem instance

b. Genetic Algorithm Implementation

- Implement a complete GA with:
  - Population initialization
  - Selection mechanism (tournament, roulette wheel, etc.)
  - Crossover operator appropriate for your problem
  - Mutation operator appropriate for your problem
  - Replacement strategy

c. Performance Analysis

- Evaluate your algorithm's performance:
  - Convergence analysis

- Solution quality compared to known optimal/best solutions
- Computational efficiency
- Conduct parameter sensitivity analysis (population size, crossover rate, mutation rate)

Advanced Features
Implement at least TWO of the following advanced features:

d. Parameter Tuning/Optimization
- Systematically optimize GA parameters
- Implement adaptive parameter control

e. Advanced Selection or Elitism
- Implement specialized selection mechanisms
- Design a custom elitism strategy

f. Enhanced Genetic Operators
- Design problem-specific crossover operators
- Implement repair mechanisms for infeasible solutions
- Implement local search/memetic components

g. Hybridization
- Combine GA with another technique (PSO, Simulated Annealing, etc.)
- Implement a coevolutionary approach

h. Visualization Tools
- Create visual representations of solutions
- Develop an interactive tool for tracking GA progress

5. Deliverables

Submission Format: *ID_Name_Surname.zip (Report (PDF) + Source Code (ZIP) + Presentation)*

1. **Project Report (5-10 pages)**
   - Problem formulation and mathematical definition
   - Algorithm design and implementation details
   - Experimental setup and parameter settings
   - Results and analysis
   - Discussion of challenges and solutions
   - Contributions of each group member
2. **Source Code**
   - Well-documented implementation
   - Include README with instructions for running your code
   - Include the problem instance file used
3. **Presentation (10 minutes)**
   - Overview of your approach
   - Key innovations/points
   - Results visualization
   - Demo
   - Lessons learned

6. Evaluation Criteria

Component	Weight	Description
Problem formulation	15%	Appropriate representation and objective definition
GA implementation	35%	Correctness and efficiency of implementation
Performance analysis	15%	Thoroughness of testing and analysis
Advanced features	20%	Implementation of additional techniques
Report and presentation	15%	Quality of writing, visuals, and presentation

7. Academic Integrity

All work must be original to your group. While you may discuss general concepts with other groups, your implementation, code, and report must be your own work. All references must be properly cited. Any form of plagiarism will result in a failing grade for all group members.

If you have any questions about the project requirements, please post them on the course discussion forum over LMS or contact the instructor directly.

Good luck!
Cumali Türkmenoğlu