ENG335
Computational Intelligence

Assignment 3 - Genetic Algorithms

Tyler Robards - 651790, Baley Eccles - 652137

Contents

1 Introduction

This assignment applies a Genetic Algorithm (GA) to optimise the placement of an Emergency Response Unit (ERU) with a 7×7 km city grid. Each grid sector represent the annual rate of emergency incidents, λ. The goal is to minimise the average weighted distance between the ERU and all city sectors, therefore reducing the average response time to emergencies.
Two cases are analysed:

1) Flat City - No obstacles in the city grid, the ERU can be placed anywhere.

2) River City - A river divides the city at $x = 5$ km with a single bridge at $(5.0, 5.5)$ km forcing detours across the bridge.

The Genetic Algorithm is developed in MATLAB to locate the global optimum ERU position, and is visualised through a progress plot. Simple testing of GA parameters and result visitation is implemented through an interactive GUI built with the MATLAB App Designer.

2 Domain Problem

This optimisation problem seeks to minimise total travel distance to a single ERU location.
The specification defines the objective function as the weighted total emergency response distance,

$$f(x_{eru}, y_{eru}) = \sum_{n=1}^{49} \lambda_n \sqrt{(x_n - x_{eru})^2 + (y_n - y_{eru})^2}$$

where,

- λ_n is the number of emergencies per year in sector n,
- (x_n, y_n) are the coordinates of the center of sector n, and
- (x_{eru}, y_{eru}) are the coordinates of the ERU location

The goal is to minimise the this sum, $f(x, y)$, since a smaller value means a shorter response time. However, most genetic algorithms are designed for maximisation problems. To use the GA for this task, the reciprocal of the objective function is used as the fitness function,

$$f_{GA}(x, y) = \frac{1}{f(x, y) + \epsilon}$$

where ϵ is a small constant to avoid division by zero.
This allows the algorithm to maximise the fitness while minimising the original objective.
#TODO write about the city layouts

Table 1: Flat City

3	4	1	2	1	3	8
2	1	3	1	3	9	7
5	1	2	1	4	9	8
4	2	1	2	5	9	9
8	9	6	3	2	5	9
9	8	5	2	1	7	9
8	9	6	1	1	8	9

2.1 The Travelling Salesman Problem

The travelling salesman problem (TSP) asks the following question "Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?" The TSP is a similar optimisation problem to that in this assignment, both aim to minimise the total travel distance, however this task has a single ERU location rather than a closed route.

3 Genetic Algorithm Description

#TODO

3.1 Fitness Evaluation

3.2 Parameter Selection

3.3 Algorithm Testing

To test the genetic algorithm the GUI shown in figure 1 was developed. This allows the user to easily change the GA parameters and display the results.
#TODO More testing on the GA and write about it

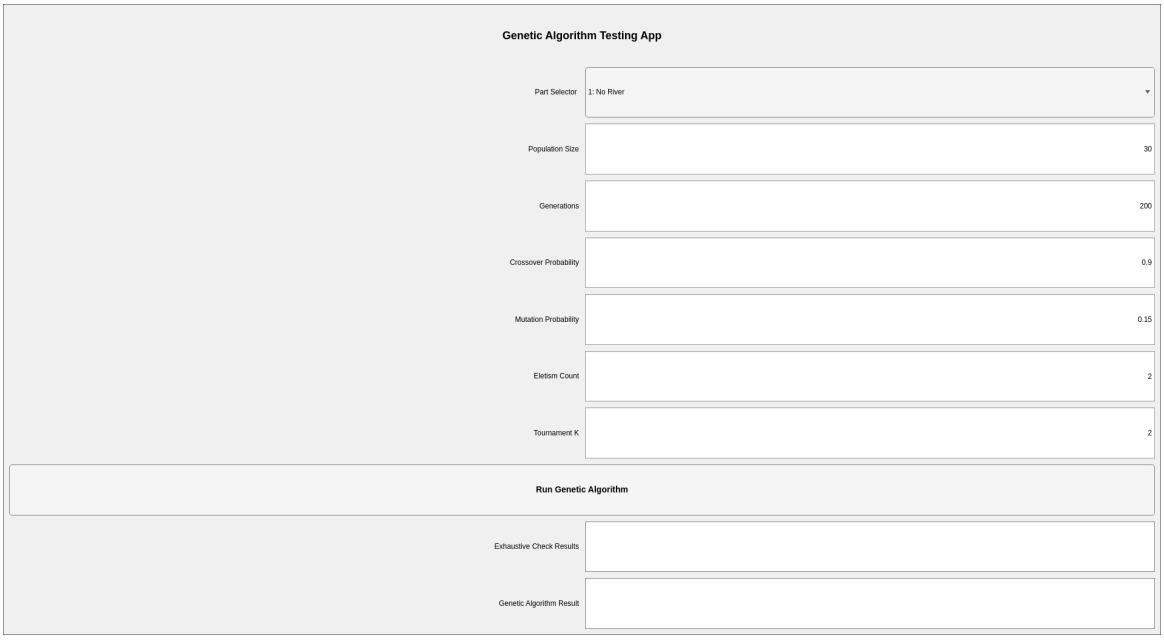

Figure 1: GUI

3.3.1 GUI User Guide

Features:

- Drop down selection to choose if the river is enabled
- Editable fields for population, number of generations, Crossover and Mutation probabilities, elitism count and tournament size.

Steps:

1) Extract `Simulation Files.zip` and open `Main.m`

2) Run `Main.m`. This will open the GUI

3) Enter desired GA parameters

4) Click `Run GA` to execute

5) Results will be displayed in the bottom text areas. The plot progess plot will be displayed, showing the result from both the Genetic Algorithm (red circle) and Exhaustive Check (star)

4 Conclusions

#TODO

End of Assignment 3