

SET10107 Computational Intelligence  
Practical 02  
Introduction to Evolutionary Algorithms

**Date: Jan/2018**

<b>Duration / Location</b>	2 Hour Practical Session in the JKCC
<b>Level</b>	10
<b>Aims</b>	Introduction to Evolutionary Algorithms
<b>Learning Outcomes</b>	The session partially covers the requirements for Learning Outcomes 1,2,3 and 4
<b>Equipment required</b>	Windows PC

<b>Description</b>	<p>During this session, you will be completing exercises that introduce you to evolutionary algorithms and methods for evaluating their effectiveness as techniques for solving computationally intractable problems.</p> <p>You should complete this practical during the scheduled 2 hour session in the JKCC</p>
--------------------	---

## Tutorial Week 2: Experimenting with a Grouping Evolutionary Algorithm

### Aim:

You are provided with some software that allows you to experiment with using an Evolutionary Algorithm for solving grouping problems. The aim of the tutorial is to investigate whether it is possible to get an EA to solve this type of problem, and to get a feel for how the various parameters of the algorithm affect its performance.

### Materials (on Moodle)

- Download the zip file GroupingGA.zip from Moodle and save onto the **D drive** – unzip it.
- The folder contains an executable **GroupingGA.exe** and a set of input files called \*.epp.
- Start the executable by double-clicking the exe file to start the program

### Running the program

(Read the accompanying application notes if you want to understand how the algorithm works)

- Click “Solve Problem”
- Select “Load Problem File” – select exercise1.epp from the folder you saved it in
- Change the number of groups to 9 in the orange problem details box

### To run the algorithm, there are two steps you must always follow:

- Press initialise to set up a random population
- Press go to run the algorithm

### Things to notice:

- The fitness of the best solution found in the initial population is displayed in the box at the bottom left
- The chromosomes in the current population can be viewed in the box at the top left – it defaults to showing the best chromosome in the population. Use the arrows underneath to cycle through the population to see the different chromosomes.
- When the algorithm finishes running, the best solution found is displayed in the grey box on the bottom right – remember you are trying to minimise this value.

## Exercise 1 – Population Diversity

### Part I

EAs are population based algorithms – the first step is always to initialise a random population of solutions. Each member of the population is a potential (though not necessarily good!) solution to the problem

Select settings for the various parameters as given below and press Initialise: notice the fitness of the best solution that was found in the initial population.

- Population Size: 100 Selection: Tournament, tournament size 2
- Mutation Rate: 0.001 Crossover method: uniform Iterations: 1000
- Using the forward and backward arrows, cycle through the population and view each member of the population in the display box on the left:

*Notice the variation between solutions, in terms of their layout and quality. Some solutions may have equal fitness, but different assignments of blocks to groups.*

- Press GO - the algorithm will run for 1000 iterations (an iteration = selection of parents, crossover and mutation).
- *What is the fitness of the best solution found ? .....*
- Continue pressing GO and fill in the table below:

<i>Iterations</i>	<i>Best Fitness</i>
1000	9
2000	6
3000	7
4000	5
5000	7
6000	7
7000	5
8000	6
9000	6
10000	5

- *Did the algorithm find the optimum solution (fitness 0 ) YES/NO*
- *Did the algorithm continue to improve as the iterations increased ? YES/NO*
- *If the answer is NO, why do you think this happened? (Hint: use the display box on the left to examine each solution in the population again)*
- Now re-initialise the population, and set the number of iterations to 10,000. Run the algorithm 10 times and record the best result found here:

<i>Attempt</i>	<i>Best Fitness</i>
1	7
2	5
3	5
4	5
5	5
6	3
7	3
8	3
9	3
10	3

- *Is the algorithm reliable? YES/NO*
- *Do you ever find the optimum? YES/NO*

## Part II

You have probably found that you can't get a good answer using the settings above. Systematically experiment with some of the parameters of the algorithm to see if you can improve it. Some hints are given below:

- Population Size: Try increasing it (up to around 1000)
- Selection Method: Try using tournament selection, start with a tournament of size 2. What happens if you increase the tournament size (try increasing from 2 in steps of 1)
- Mutation Rate: The mutation rate has a big impact on the performance of this algorithm. Try a very small mutation rate (0.005 or even less!) Try increasing it from this value and see what happens to the performance.
- Crossover: Does the choice of crossover have much effect?
- If you find a set of parameters which find the optimum, how reliable are they? If you initialise and run the algorithm several times with these parameters, do you always find the optimum?

*If you cannot obtain the optimum answer on this problem, consider why this might be so.*

## Part III

When you have found the best answer you can for the file "exercise1.epp", then experiment with the file "exercise2.epp". Can you find the optimum answer? If so, what combination of parameters did you use?

## Part IV - Selection Pressure

### Aim:

To understand the effect of selection pressure on the performance of an EA, and how it can be controlled using (a) tournament selection and (b) population size

### a) Investigating the effect of population size

**Objective:** to understand the effect of increasing population size whilst maintaining a fixed tournament size in an evolutionary algorithm

### Method:

- Load the problem **input1n.epp**
- Set the selection = tournament, tournament size=2, mutation rate = 0.005, crossover = 2point, iterations = 100000.
- Set the population size to 10, and run the algorithm 10 times – you can do this manually by pressing re-initialise the population before you press Run each time

- OR .....you can set the number of experiments to 10 – the program will ask you for a filename. The output will be saved to this file – the number on each line of the output file is the fitness of the best solution found in each run
- Record the best solution found at the end of each run in an Excel Spreadsheet (or paste the values directly from the output file)
- Repeat the above step for population sizes of 25, 50, 75, 100, keeping all other parameter values fixed.
- Calculate the average and standard deviation of the best result at each population
- Plot a graph of population size against average fitness
- Comment on the trends observed in the graph and complete the following statement

***Increasing the population size whilst maintaining a fixed tournament size\_\_\_\_\_ the selection pressure in the algorithm.***

#### **b) Investigating the effect of tournament size for a fixed population size**

**Objective:** to understand the effect of increasing tournament size, given a fixed population size

#### **Method:**

- Set the selection = tournament, population size = 100, mutation rate = 0.005, crossover = 2point, iterations = 100000.
- Set the tournament size to 2, and run the algorithm 10 times. Record the best solution found each time in an Excel spreadsheet.
- Repeat the above step for tournament sizes of 10, 50, 75, 90, keeping all other parameter values fixed.
- Calculate the average and standard deviation of the best result at each population
- Plot a graph of population size against average fitness
- Comment on the trends observed in the graph and complete the following statement

***Increasing the tournament size whilst maintaining a fixed population size\_\_\_\_\_ the selection pressure in the algorithm.***

In order to confirm your conclusions, repeat Parts IV a and IV b using the other test problems given. Confirm that the same effect is observed using each problem.

## Conclusions

Consider:

- From your results, which factor appears to affect performance more strongly – tournament size or population size?
- What effect does using a high selection pressure have on the population as the algorithm runs? Why does this matter? (examine the population using the display box on the left to give you some insight into this)

## Reading

Read chapter 2 “*What Is an Evolutionary Algorithm?*” from [1], which is available on Moodle.

[1] Agoston E. Eiben and J. E. Smith. 20015. *Introduction to Evolutionary Computing*. SpringerVerlag.