Laboratory 12 – Week 19

## Further Genetic Algorithms

## 12.1 Introduction

**Firstly, this worksheet is one of the worksheets from which your laboratory worksheets portfolio of work will be assessed.**

It is **VERY** important that you have completed Laboratory Sheet 11 “A Simple Genetic Algorithm Applied to the Scales Problem”. You should work on laboratory 11 and complete it before you start this one (sheet 12).

The laboratory involves the modification of Laboratory 11's simple Genetic Algorithm (GA) to solve a new problem. You will be running a number of experiments and plotting convergence graphs.

## 12.2 Preliminaries

Familiarise yourself with the lecture entitled “12.1 An Introduction to Genetic Algorithms”. Make sure you understand how Genetic Algorithms work.

## 12.2 The One-Max Problem

One of the first (and very simple) problems a Genetic Algorithm was applied to is called the **OneMax** problem. This is defined as follows:

If we have an *n* length binary string *S* (with individual binary digits *si*) then *OneMax*(*S*) is defined as follows:



That is, the number of ones in the binary string. The objective is to **maximise** this value.

## 12.3 Exercise 1: Modifying the GA

We are going to modify Laboratory 11's simple Genetic Algorithm (GA). Perform the following steps:

1. Copy and rename the CompareScalesChrome.java, ScalesChrome.java and the SimpleGeneticAlgorithms.java files. In the filename and within the files themselves (use the Eclipse Find/Replace menu option) change every instance of ScalesChrome to OneMaxChrome. Alternatively you can use the Eclipse Refactor tool.
2. Create a project Lab12 within Eclipse based around the Lab11 project.
3. The first task is to open up the CompareOneMaxChrome class and then change the signs of the two return statements(a 1 to a -1 and a -1 to a 1). This converts the problem from a minimisation problem to a maximisation one.
4. You should now only need to modify the OneMaxChrome.java file if you renamed everything correctly. Within this file you no longer need the weights field as the OneMax problem does not need any weights. Remove and modify the code so that there are no references to weights.
5. Remove the code within the GetFitness function and implement the OneMax fitness function.

## 12.4 Exercise 2: Experiments

Conduct a number of experiments to evaluate which version of the two Crossover operators performs best on the OneMax problem. Run each experiment at least ten times and vary the size of the problem you are solving.

Plot appropriate convergence graphs and record average performances.