

# Homework Module: **Programming the Basics of an Evolutionary Algorithm(EA)**

Sigve Sebastian Farstad

February 21, 2014

## **1 Introduction**

This report presents a solution to Homework Module: Programming the Basics of an Evolutionary Algorithm(EA), IT3708, spring 2014, NTNU. The assignment is to implement all of the basic components of an evolutionary algorithm, and use them to solve the One-Max problem.

## **2 The Evolutionary Algorithm Implementation**

The evolutionary algorithm solver presented in this solution is implemented as modular python library. A generic solver function solves arbitrary problems supplied by the user of the library.

### **2.1 Adult Selection**

Full generational replacement, over-production and generational mixing.

### **2.2 Parent Selection**

#### **2.2.1 Fitness-Proportionate Selection**

Fitness-proportionate selection is a selection scheme where an individual is selected at random, with the probability of being chosen equal to  $\frac{f(i)}{F}$ , where  $f(i)$  is the fitness of an individual, and  $F$  is the total fitness of the population.

#### **2.2.2 Sigma Scaling Selection**

Sigma-scaling selection is similar to fitness-proportionate scaling, but the probability of a given individual being chosen is scaled by the fitness variance of the population, as in equation 3 of [?].

### 2.2.3 Tournament Selection

Tournament selection is a selection scheme where a predefined  $k < |population|$  number of individuals are selected for participation in a tournament. Of these competing individuals, the one with the greatest fitness score gets selected. Tournament selection guarantees that the  $k - 1$  worst individuals in a population will not get selected.

### 2.2.4 Rank Selection

Rank selection is a selection scheme similar to fitness-proportionate selection, only the probability of selection is defined as in equation 1.

$$P_{selection} = 2 \times \frac{rank(i)}{|population| \times (|population| + 1)} \quad (1)$$

### 2.2.5 Random Selection

Random selection is a selection scheme that simply selects an individual at random. It is not a good selection scheme for an EA, but is included in the selection suite for benchmarking purposes.

## 2.3 Crossover

## 2.4 Mutation

# 3 The One-Max Problem

The One-Max problem is a simple problem popularly chosen as a demonstration problem for testing evolutionary algorithm implementations. It has no other real-world applications. The problem is: what is the binary string of length  $n$  that contains the largest number of Ones? The problem is trivial, but is good for basic evolutionary algorithm testing because of its transparency and simplicity.

## 3.1 Genotype and Phenotype Representation

A single genotype is represented as a binary vector of length  $n$ . Because of the simplicity of the program, the phenotype is identical.

## 3.2 Fitness Evaluation

The fitness function  $f_{ONE\text{MAX}}(g)$  of a genotype  $g$  is defined as equation 2, where  $g_i$  is the  $i^{\text{th}}$  bit of the genome  $g$ .

$$f_{ONE\text{MAX}}(g) = \sum_{i=1}^n g_i \quad (2)$$