

Project 1.A:
Programming the Basics of an
Evolutionary Algorithm(EA)

Mikael Brevik

February 14, 2012

Contents

1	Introduction	2
1.1	About the project	2
1.2	EA Overview	2
1.3	The One-Max Problem	2
2	Deliverables	3
2.1	Description of the EA code	4
2.2	Justification of the code's modularity and reusability	5
2.3	Analysis of the performance	6
2.4	Selection mechanism with the best results	7
2.5	Target bit string as a random vector	8

Chapter 1

Introduction

1.1 About the project

The purpose with this project is to gain experience with the basic mechanisms underlying EAs by programming them.

The entire project consists of two parts; A) Implementing basic modular EA , and B) using that algorithm to solve a Colonel Blotto problem.

This document will focus on part A, implementing the basic evolutionary algorithm and testing that algorithm by solving a One Max problem

1.2 EA Overview

The evolutionary algorithm we're implementing in this assignment includes the following aspects:

1. A population for representing individuals with genotypes.
2. A development phase, for converting genotypes to phenotypes.
3. A fitness test for calculating fitness of all phenotypes.
4. Selection protocols for finding adults from the individuals
5. Selection mechanism for finding parents from the adults.
6. Reproduction for creating children from parents. Taking parts from each of the parent's genos.
7. Mutation for altering the childrens genotype.

The EA will loop through these items many generations. There must also exist a plotting functionality for analyzing and visualizing the results.

1.3 The One-Max Problem

For testing out the EA, the One-Max problem will be used. The goal of the One-Max problem is to find a bit string consisting only of 1's. The initial genotype will be a random n-bit string. With using the number of 1's as fitness, throughout the generations the bit strings should evolve to consist only of 1's.

The programmed EA should be able to solve a 40-bit One-Max problem.

Chapter 2

Deliverables

- 2.1 Description of the EA code
- 2.2 Justification of the code's modularity and reusability
- 2.3 Analysis of the performance
- 2.4 Selection mechanism with the best results
- 2.5 Target bit string as a random vector