Workshop GA2

The binary GA

Functional Programming and Intelligent Algorithms
Module: Genetic Algorithms

Spring 2017

Department of ICT and Natural Sciences
Norwegian University of Science and Technology

Robin T. Bye*

Revision date: 22 March 2017

*email: robin.t.bye@ntnu.no; website: www.robinbye.com; phone: 70 16 15 49; office: B315.

Contents

1 Workshop overview

1.1 Topics

Today's topics include:

- Components of the binary GA
- Solving 1D and 2D functional problems using a binary GA
- String-learning GA

1.2 Reading material

Compulsory reading to be studied *before* this workshop is Chapter 2 in Haupt & Haupt (2004) on binary GAs.

1.3 Specific learning outcomes

After completing this workshop, including self-study, reading and exercises, the students should be able to

- explain the components and algorithmic flow of the binary GA.
- demonstrate typical effects of changing parameters of the binary GA.
- implement and modify their own binary GAs to suit a variety of problems, including function optimisation and toy problems such as string learning.

1.4 Schedule

We begin at 8.15 with a recap of the last couple of weeks' activities and questions. Today's workshop will roughly follow the schedule below:

09.15 Overview and status update.

09.45 Lecture: The binary GA.

12.00 Workshop/tutorial rest of the day.

Description of a Basic GA

A chromosome c is an encoded candidate solution to the optimisation problem of optimising $f(c)$. The design parameters that we want to optimise must be translated (encoded) from their original domain to a format suitable for the GA, usually arrays of bits or real-valued numbers, often normalised to the interval $[0, 1]$. The bits or numbers are usually called genes. A chromosome is a list of genes, whilst a population is a list of chromosomes.

The objective function quantifies that quality of candidate solutions, that is, how well they fulfill the desired design criteria. The selection criterion determines how many chromosomes in a population survives from one iteration to the next. For examle, using the roulette wheel method, the cost (fitness) associated with each chromosome is evaluated and the chromosomes are given a weighted selection probability according to their cost, where a smaller cost (greater fitness) results in a greater probability.

A pre-determined fraction, of chromosomes (typically half the population) is then randomly picked, with low cost (high fitness) chromosomes having a greater chance of being picked and kept for survival and reproduction.

For mating, several crossover methods exists, where genes from two parent chromosomes are combined into one or several offspring, which are then put back into the population, replacing those chromosomes that were not selected for mating.

After mating, a fraction of the chromosomes will have one or several of their genes mutated. This means flipping (inverting) bits for binary chromosomes, or changing the values of these genes to random numbers within some allowable range.

Next, each of the chromosomes in the updated population is evaluated by the objective function and the population is sorted in descending order of performance (ability to minimise cost or maximise fitness).

The process repeats until the maximum number of iterations has been reached, or the solution (the best chromosome) has reached a satisfactory performance. Then the algorithm ends and returns the best chromosome, which is decoded back to its original domain. In our case, the decoded solution specifies the optimal values for selected design parameters of an offshore crane.

2 Exercises

2.1 Components of the binary GA

Exercise 2.1: Draw a diagram depicting the steps that constitute the algorithmic flow of a binary GA.

Exercise 2.2: Explain the following terms:

(a) N_{var}

(b) N_{gene}

(c) N_{bits}

(d) N_{pop}

(e) X_{rate}

(f) N_{keep}

Exercise 2.3: In terms of search and optimisation algorithms such as the GA, what is meant by the following terms:

(a) Elitism.

(b) Convergence rate.

(c) Exploration.

(d) Exploitation.

Exercise 2.4: Suppose you have a GA with some parameter settings that yield a particular level of exploration and a particular rate of convergence (exploitation). Explain what is likely to happen if you increase or decrease

(a) the population size.

(b) the mutation rate.

(c) the selection rate.

(You may have to come back to this question after having tried it for yourself in the following exercises).

2.2 String learning using a binary GA

Complete the tutorial on the binary GA for string learning on the course website.

3 Homework

- Complete all the exercises above.
- Read through (again!) the specific learning outcomes in Section 1.3 to check which outcomes you have not attained yet. Study today's material and prepare questions for tomorrow about learning outcomes you have missed.
- Prepare for the next workshop by reading about the continuous genetic algorithm (GA) in Chapter 3 of Haupt & Haupt (2004).

References

Haupt, R. L., & Haupt, S. E. (2004). *Practical Genetic Algorithms*. Wiley, 2nd ed.