

Evolutionary Computation

Practical Assignment 1

1 Genetic Algorithm

In the first part we will look at experimental results of Genetic Algorithms (GAs) on four artificial functions in order to gain some insight in the convergence behavior of GAs. All 4 functions are defined over the binary domain and have a stringlength $\ell = 100$. The goal is to find the binary vector that maximizes the function value; in each case the optimal solution is the string of all ones.

1. Counting Ones Functions:

(a) Uniformly Scaled Counting Ones Function:

$$x_i \in \{0, 1\} : CO(x_1 \dots x_\ell) = \sum_{i=1}^{\ell} x_i$$

(b) Linearly Scaled Counting Ones Function:

$$x_i \in \{0, 1\} : SCO(x_1 \dots x_\ell) = \sum_{i=1}^{\ell} ix_i$$

2. Trap Functions:

$$TF(x_1 \dots x_\ell) = \sum_{j=0}^{\frac{\ell}{k}-1} B(x_{jk+1} \dots x_{j(k+k)})$$

with:

$$B(x_1 \dots x_k) = \begin{cases} k & \text{if } CO(x_1 \dots x_k) = k \\ k - d - \frac{k-d}{k-1} CO(x) & \text{if } CO(x_1 \dots x_k) < k \end{cases}$$

(a) Deceptive Trap Function: $k = 4$, $d = 1$

Number of Ones	4	3	2	1	0
Fitness Value	4	0	1	2	3

(b) Non-deceptive Trap Function: $k = 4$, $d = 2.5$

Number of Ones	4	3	2	1	0
Fitness Value	4	0	0.5	1.0	1.5

Both Trap functions consist of 25 concatenated subfunctions of length $k = 4$, each having an optimum at 1111 and at 0000. The overall function has therefore $2^{25} - 1$ local optima and 1 global optimum (= the string of all ones). The difference between the two trap functions is the difference d between the values of the two optima at each substring. As a result of this fitness difference the first trap function is so-called, fully deceptive, while the second is not. Deceptive functions are functions where the lower-order schema fitness averages that contain the local optimum 0000 have a higher value than the lower-order schema fitness averages that contain the global optimum 1111 in each subfunction.

(a) Deceptive Trap Function:

$$\begin{cases} F(111*) &= \frac{4+0}{2} = 2 \\ F(000*) &= \frac{3+2}{2} = 2.5 \end{cases}$$

$$\begin{cases} F(11**) &= \frac{4+0+1}{4} = 1.25 \\ F(00**) &= \frac{3+2+1}{4} = 2 \end{cases}$$

$$\begin{cases} F(1***) &= \frac{4+0+3+1+2}{8} = 1.125 \\ F(0***) &= \frac{3+3+2+3+1+0}{8} = 1.5 \end{cases}$$

(b) Non-deceptive Trap Function:

$$\begin{cases} F(111*) &= \frac{4+0}{2} = 2 \\ F(000*) &= \frac{1.5+1}{2} = 1.25 \end{cases}$$

$$\begin{cases} F(11**) &= \frac{4+0+0.5}{4} = 1.125 \\ F(00**) &= \frac{1.5+2(1)+0.5}{4} = 1 \end{cases}$$

$$\begin{cases} F(1***) &= \frac{4+0+3(0.5)+1}{8} = 0.813 \\ F(0***) &= \frac{1.5+3(1.0)+3(0.5)+0}{8} = 0.75 \end{cases}$$

The specific genetic algorithm (GA) we will use in the assignment is a so called incremental (or steady-state) GA: this means that we do not work with separate generations but that we generate a solution pair and immediately add one or both solutions to the current population if the fitness score of the solution is better than the lowest scoring solution in the current population:

1. Select and copy two parents from the population with tournament selection (tournament size s).

2. With probability p_c recombine the copies of the parents, with probability $1-p_c$ mutate both copies. When doing mutation first decide how many bits will be mutated: with probability $\frac{1}{2}$ flip 1 bit, with probability $\frac{1}{4}$ flip 2 bits, with probability $\frac{1}{8}$ flip 3 bits, ... until a mutation actually took place. This mutation probability distribution can easily be implemented by sequentially flipping a fair coin.
3. Let the best offspring solution compete with the current worst solution in the population: if the fitness of the new solution is better or equal then it replaces the worst solution in the population.

Run experiments with different parameter values:

- Compare different population sizes (only consider multiples of 10).
- Compare different crossover/mutation probabilities ($p_c = 1, \frac{1}{2}, 0$).
- Compare different tournament sizes ($s = 1$ and 2) (note that $s = 1$ means just a random selection).
- Compare two different crossover operators: uniform crossover versus 2-point crossover. For 2-point crossover, you should do the experiments with tightly linked and randomly linked Trap Functions. Tightly linked means that the 25 subfunctions are placed adjacent to each other on the binary string. With random linkage the 100 bits are placed at a random position on the binary string.

Running an experiment with a particular parameter setting requires executing 50 independent runs. The stopping criterium is met when the global optimum is reached or when the last V new solutions did not enter the population because their fitness is worse than the lowest ranked solution in the current population (choose $V = 10 \times \text{population size}$). Count the number of generated solutions T until the first occurrence of the global optimum ($T = \text{the first hitting time}$). A particular parameter setting is successful if 49 out of the 50 runs reach the global optimum.

Finally, you are asked to come up with a research question of your own choosing, set up the experiments and report the results. Examples of such topics are, use alternative genetic algorithms, increase the selection pressure for the UX and deceptive trap function case, compare with simulated annealing, use a hybrid GA (including a local search),

2 Deliverables

Write a report discussing your results and send it to d.thierens@uu.nl. The report should be in **PDF format !!!** The source code should be in SEPARATE, compressed archive file (program.tar.gz). **Do NOT include** your report.pdf file in this archive !

The report should at least contain the following topics:

1. Give a short description of the implemented program.
2. Tabulate and/or plot the experimental results.
3. Report the efficiency (in CPU time and in number of fitness function evaluations) and performance.
4. Plot the run length success distribution: the x-axis shows the number of fitness function evaluations (or solutions tested), and the Y-axis is the percentage of runs that are successful (this info is recorded by the 'first hitting time' measurements).
5. Discuss your findings.

3 Deadline

Sunday, **January 5**, 2014, 24:00 hrs.