Assignment 1 Natural Computing

l.nies s4136748, g.zuidhof s4160703

February 2015

1

Search problems where there is no possible distance-to-the-goal heuristic. Representation is very hard for these problems. For these problems there is no indication how close it is to an answer, and as such it's very hard to come up with a fitness. There is no one answer is better than the other: there is only one correct answer. An example binary search, for this an algorithm exists, but EAs will have a hard time.

2

2.1 Representation

The genome consists of 6 real numbers, which represent the scalars a, b, c, d, e, f in the formula.

2.2 Initialisation

Generate 5 random, real numbers $[r_a, r_b, r_c, r_d, r_e, r_f]$ between the minimum and the maximum of the y coordinates of the data points. The genome then becomes $f = r_f, e = \sqrt[max(x)]{r_e}, d = \sqrt[2]{r_d}, c = \sqrt[3]{r_c}, b = \sqrt[4]{r_b}, a = \sqrt[5]{r_a}$

2.3 Parent selection

Roulette wheel selection with simulated annealing.

2.4 Recombination

Linear interpolation between the two values, with a random value $t \in [0, 1)$ for the first offspring, and $1 - t$ for the other.

With the linear interpolation method in pseudocode (from Wikipedia):

```
float lerp(float v0, float v1, float t) {
  return (1-t)*v0 + t*v1;
}
```

2.5 Mutation

Generate 5 random, real numbers $[r_a, r_b, r_c, r_d, r_e, r_f]$ between 0 and 1, and one additional random real number s between -1 and 1, that is not 0. For every number in the genome there a 1/6 probability that it mutates:
$f := f + sign(s) * r_f$
$e := e + sign(s) * \sqrt[max(x)]{r_e}$
$d := d + sign(s) * \sqrt[2]{r_d}$
$c := c + sign(s) * \sqrt[3]{r_c}$
$b := b + sign(s) * \sqrt[4]{r_b}$
$a := a + sign(s) * \sqrt[5]{r_a}$

2.6 Replacement

No elitism, stirct generational.

2.7 Termination Condition

Terminate 10000 generations or after sufficient fit is found.

2.8 Fitness

Each individual has its own polynomial function y, given by: $y = ax^5 + bx^4 + cx^3 + dx^2 + e^x + f$

where a, b, c, d, e, f are taken from the genome. The fitness is the distance of every data point to the polynomial curve y vertically.

3

$P(No\ mutation) = \prod_{x \in L} 1/L$

4

4.1 With $f(x) = x^2$

$f(1) = 1^2 = 1$
$f(2) = 2^2 = 4$
$f(3) = 3^2 = 9$
roulettesize = f(1) + f(2) + f(3) = 14
P(*select 1*) = f(1) / roulettesize = 1/14 = 0.07
P(*select 1*) = f(2) / roulettesize = 4/14 = 0.29
P(*select 3*) = f(3) / roulettesize = 9/14 = 0.64

4.2 With $f1(x) = f(x) + 10$

$f1(1) = f(1) + 10 = 11$
$f1(2) = f(2) + 10 = 14$
$f1(3) = f(3) + 10 = 19$
roulettesize = f(1) + f(2) + f(3) = 44
P(*select 1*) = f1(1) / roulettesize = 11/44 = 0.25
P(*select 1*) = f1(2) / roulettesize = 14/44 = 0.32

4.3 Selection Pressure

The selection pressure of the second fitness function $f1$ is lower, the influence of the fitness of an individual on the selection probability is smaller.

5

For an EA using roulette wheel selection with a bit string length of 25, the mean and standard deviation of the completion time are 0.85 and 0.05 respectively. The optimum doesn't seem to be found, so the average amount of generations is 100. The plot of one run of the EA is shown in figure 1

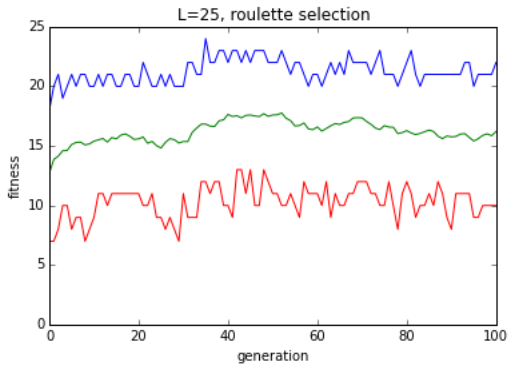

Figure 1: The best, mean and worst case of an EA using roulette wheel selection with a bitstring of length 25

6

For an EA using roulette wheel selection with a bit string length of 75, the mean and standard deviation of the completion time are 1.02 and 0.09 respectively. The optimum doesn't seem to be found, so the average amount of generations is 100. The plot of one run of the EA is shown in figure 2

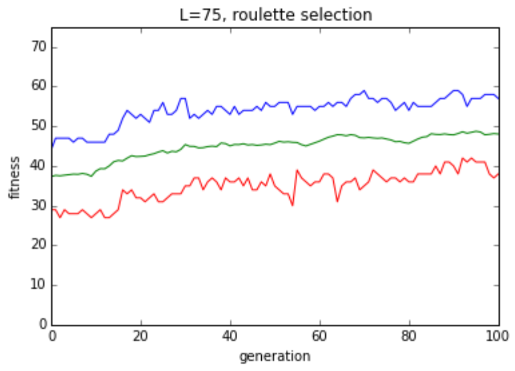

Figure 2: The best, mean and worst case of an EA using roulette wheel selection with a bitstring of length 75

7

For an EA using tournament selection with k = 2 and with a bit string length of 25, the mean and standard deviation of the completion time are 0.15 and 0.04 respectively. The mean and standard deviation of the amount of generations needed until convergence is 16.20 and 3.79 respectively. The plot of one run of the EA is shown in figure 3

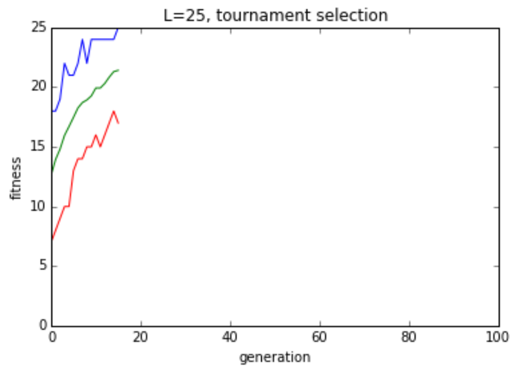

Figure 3: The best, mean and worst case of an EA using tournament selection with a bitstring of length 25

For an EA using tournament selection with k = 2 and with a bit string length of 75, the mean and standard deviation of the completion time are 0.71 and 0.17 respectively. The mean and standard deviation of the amount of generations needed until convergence is 67.90 and 15.55 respectively. The plot of one run of the EA is shown in figure 4

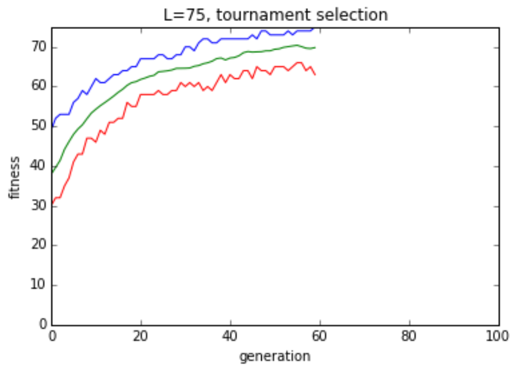

Figure 4: The best, mean and worst case of an EA using tournament selection with a bitstring of length 75

The big difference is that the roullette wheel selecetion didn't seem to convere to the optimal solution, but the tournament selection did.