Semester Project:

How does an Evolutionary Computation Work on the Test Function you chose?

Akira Imada

Abstract

One is given here a couple of test functions which are multi-dimensional and are commonly used to study how an Evolutionary Computation algorithm work. We can contrall the dimensionality and ruggedness of these functions by setting each of the parameters in the function to appropriate value. We pick up one of these functions and explore the function by, for example, drawing the graph of it's 2 or 3-dimensional version, observing locations of the global/local minima of the function. Then we learn a number of Evolutionary Computations including some hill-climbing techniqu, and apply some of them to the selected test function, with the goal of observing how the global optimum is serached for with avoiding to be got stucked to local minima.

1 Commonly Used Test Functions — to learn "How an EC works?"

1.1 Sphere Model F_1:

$$y = \sum_{i=1}^{n} x_i^2, \quad x_i \in [-5.12, 5.12].$$

1.2 Rastrigin's Function F_6:

$$y = nA + \sum_{i=1}^{n} (x_i^2 - A\cos(2\pi x_i)), \quad x_i \in [-5.12 - 5.12].$$

- Ruggedness might be controlled by modifying the value of A.
- A two dimensional example ($n = 1$): $y = 3 + x^2 - 3\cos(2\pi x)$.

1.3 Schwefel's Function F_7:

$$y = \sum_{i=1}^{n} (x_i \sin(|x_i|)), \quad x_i \in [-500, 500].$$

- A two dimensional example ($n = 1$): $y = x\sin(|x|)$.

1.4 Griewangk's Function F_8:

$$y = \sum_{i=1}^{n} x_i^2/4000 - \prod_{i=1}^{n} \cos(x_i/\sqrt{i}) + 1, \quad x_i \in [-600, 600].$$

· A two dimensional example: $y = x^2/4000 - \cos x + 1$.

1.5 Ackley's Function F_9:

$$y = -20 \sum_{i=1}^{n} \exp\left(-0.2\sqrt{x_i^2/n}\right) - \exp\left(\left(\sum_{i=1}^{n} \cos 2\pi x_i\right)/n\right) + 20 + e,$$

$$x_i \in [-30, 30].$$

· A two dimensional example: $y = -20 \exp\left(-0.2\sqrt{x^2}\right) - \exp\left(\cos 2\pi x\right) + 20 + e$.

1.6 Example Drowings of 4 function's Graph above

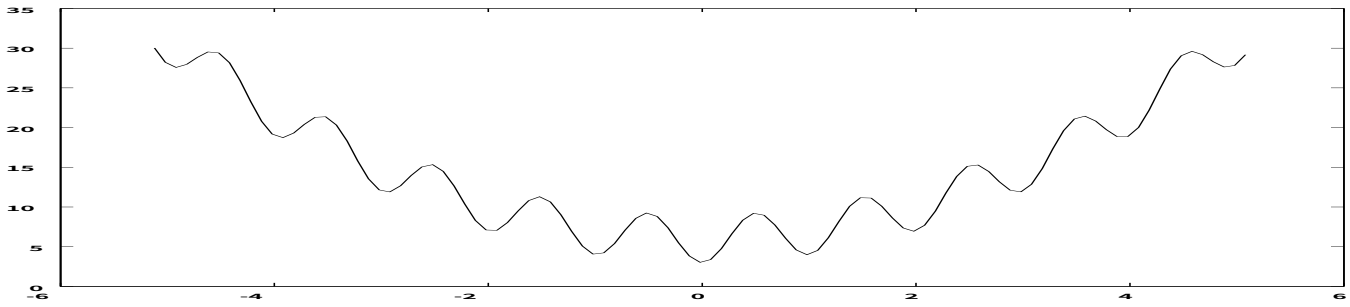


Figure 1: Rastrigin's Function F_6:

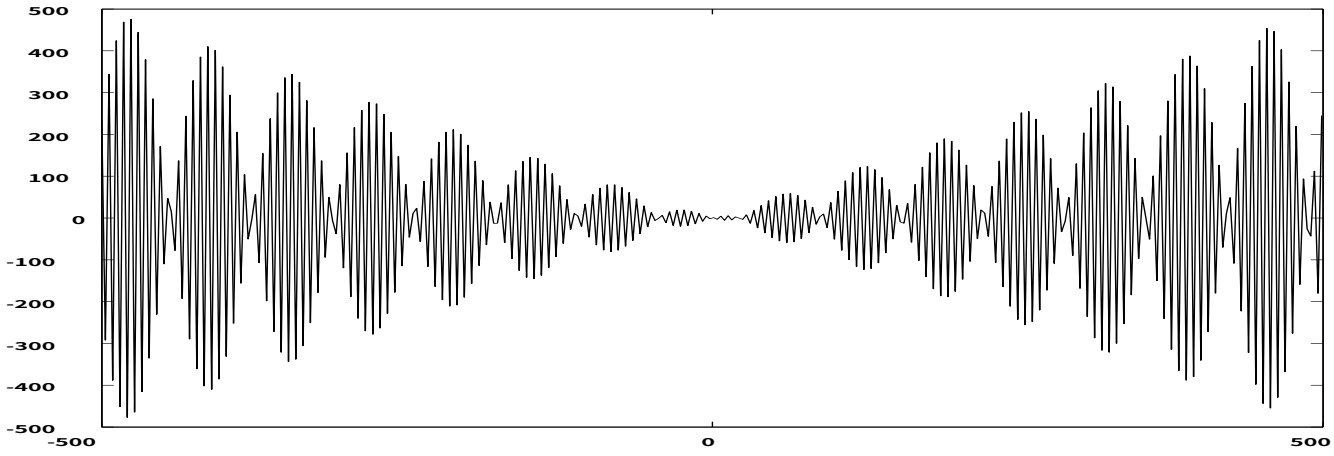


Figure 2: Schwefel's Function F_7:

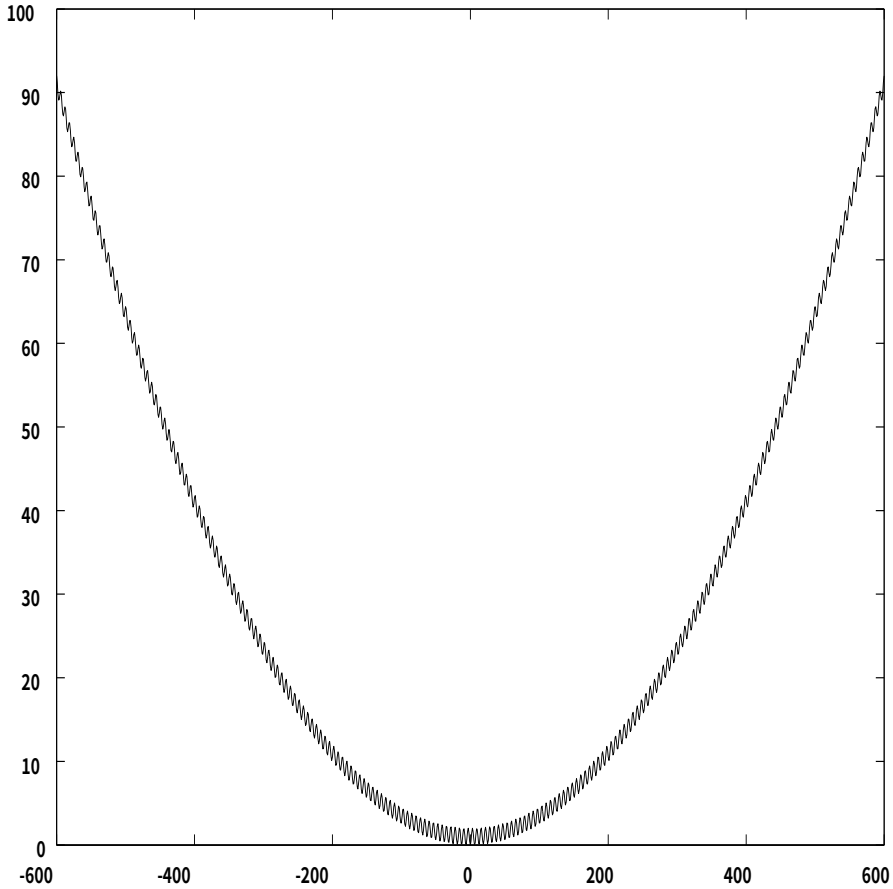


Figure 3: Griewangk's Function F_8:

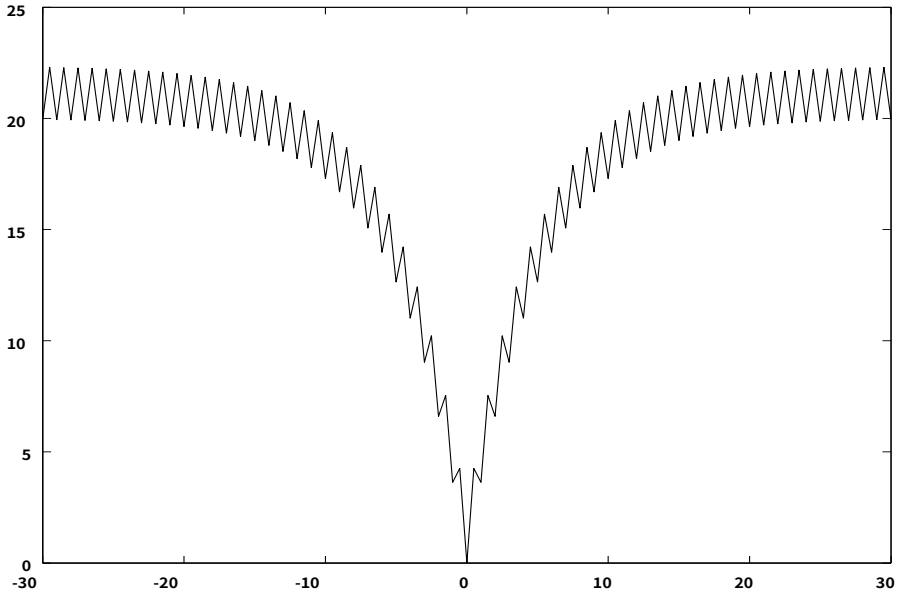
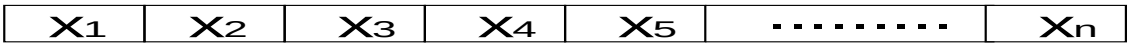


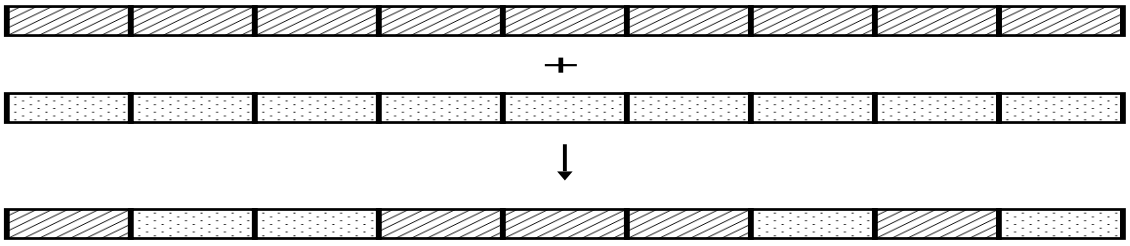
Figure 4: Ackley's Function F_9:

2 An Evolutionary Algorithm for the above testfunction

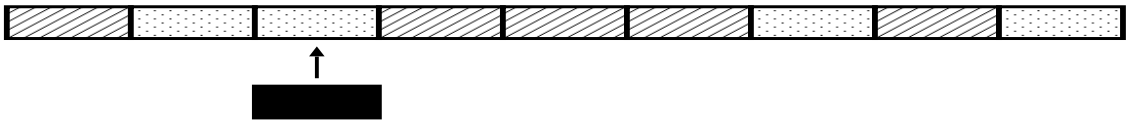
1. Create N chromosomes of the form:



2. Define a *fitness* evaluation.
   (= How good is each individual?)
3. Generate an initial *population* at random.
4. Reproduce a child chromosome by:
   - *Truncate Selection*
     * Select two parent chromosomes randomly from better half of the population
   - *Uniform Crossover*
     * Take genes either from two parents, gene by gene.



   - *Mutation*
     * Pick up a gene with a probability P_m and replace it with other random number. (A commonly used mutation rate is one over the length of chromosome.)



5. Repeat 4 until N chromosomes of the next generation are reproduced.
6. Repeat 3-5 until a stopping criteria is fulfilled
7. Thus better solutions will be expected from *generation* to *generation.*

2.1 Selection & Crossover — A little more in detail

2.1.1 Selection

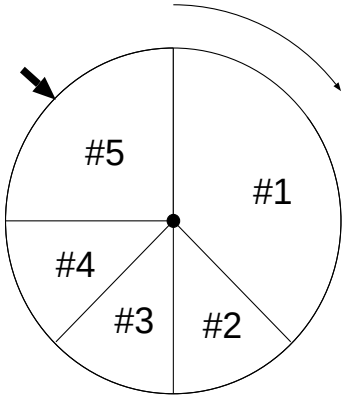
⋆ Truncation Selection

· Select parents from the best some-percentage of the population.

⋆ Roulette Wheel Selection (Fitness Proportionate Selection)

· Select so that the probability to be selected is proportional to the fitness value.

	fitness
individual #1	0.375
individual #2	0.125
individual #3	0.125
individual #4	0.125
individual #5	0.250

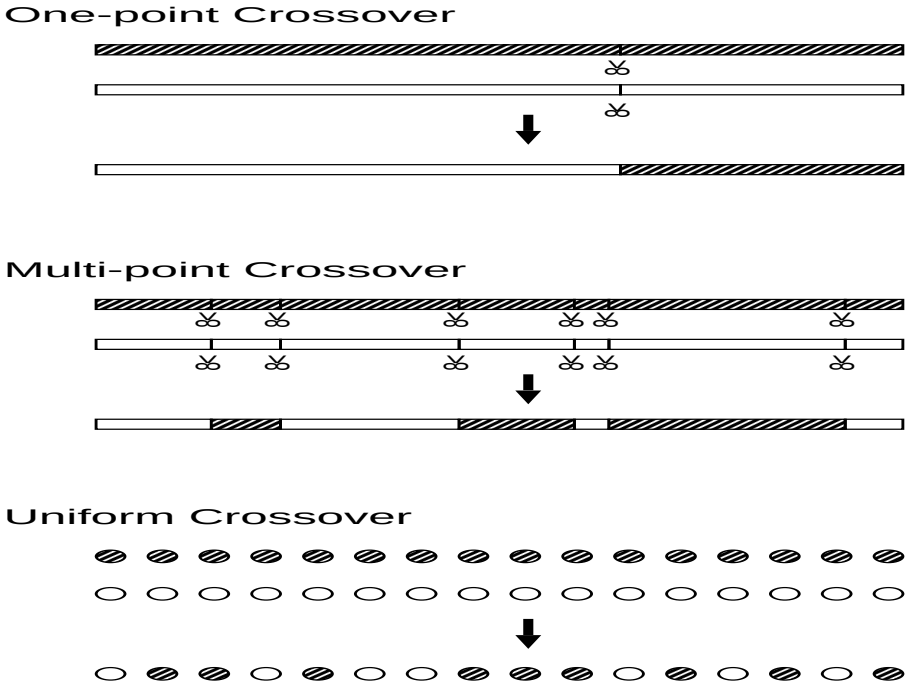


⋆ Tournament Selection

· Assume we have the original μ parents and their μ children. The fitness value of each of the 2μ individuals are compared to those of q individuals which are chosen randomly from the whole 2μ points at every time of the comparison Then the 2μ points are ranked according to the number of wins, and the best μ points survive (q-tournament selection).

2.1.2 Crossover

- ⋆ One-point Crossover.
- ⋆ Multi-point Crossover.
- ⋆ Uniform Crossover.



Excersise 1 *Choose one out of four functions above, and explore the function. That is, draw surface of its 3-dimensional version, check the location of the global minimum, learn distribution of local minima, etc...*

Excersise 2 *Apply Evolutionary Computation to locate the global minimum. Try various parameters and selection/crossover system, for example, try with* $N = 100, p_m = 0.05$*, tornament selection and one-point crossover, etc.*