Evolutionary Documentation

Release 0.1.0

Art Wild

January 05, 2016

Contents

Evolutionary.jl is a Julia package for evolutionary and genetic algorithms. It provides implementation of $(\mu/\rho \overset{+}{,} \lambda)$-Evolution Strategy, $(\mu/\mu_I, \lambda)$-Covariance Matrix Adaptation Evolution Strategy and Genetic Algorithms, as well as a rich set of mutation, recombination, crossover and selection functions.

Contents:

CHAPTER 1

Algorithms

1.1 Evolution Strategies

1.1.1 Evolution Strategies ($(\mu/\rho \overset{+}{,} \lambda)$-ES)

1.1.2 Covariance Matrix Adaptation Evolution Strategies ($(\mu/\mu_I, \lambda)$-CMA-ES)

1.2 Genetic Algorithms

CHAPTER 2

Operators

2.1 Mutations

2.2 Recombinations

2.3 Crossovers

2.4 Selections

Selection is a genetic operator used in GAs for selecting potentially useful solutions for recombination. The GAs are stochastic search methods using the concepts of Mendelian genetics and Darwinian evolution. According to Darwin's evolution theory the best ones should survive and create new offspring. There are many methods how to select the best individuals, for example roulette wheel selection, Boltzman selection, tournament selection, rank selection, steady state selection and some others.

2.4.1 Interface

All selection algorithms have following call interface:

`selection`(*fintess*, *N*)

Parameters

- **`fintess`** – The vector of population fitness values, a vector of `Float64` values of size `M`.
- **`N`** – The number of selected individuals.

Returns The vector of indexses of corresponding selected induviduals, a vector of `Int` values of size `N`. Values should be in range [1,M].

Note: Some of the selection algorithms imlemented as function closures, in order to provide additional parameters to the specified above selection interface.

2.4.2 Implementations

Roulette

roulette (*fintess*, *N*)

In roulette (fitness proportionate) selection, the fitness level is used to associate a probability of selection with each individual. If f_i is the fitness of individual i in the population, its probability of being selected is $p_i = \frac{f_i}{\sum_{j=1}^{M} f_j}$, where M is the number of individuals in the population.

Rank (linear)

ranklinear (*SP*)

Parameters **SP** – The selective presure value.

Returns Selection function, see *Interface*.

In rank-based fitness selection, the population is sorted according to the objective values. The fitness assigned to each individual depends only on its position in the individuals rank and not on the actual objective value *[BK85]*.

Consider M the number of individuals in the population, P the position of an individual in this population (least fit individual has $P = 1$, the fittest individual $P = M$) and SP the selective pressure. The fitness value for an individual is calculated as:

$Fitness(P) = 2 - SP + \frac{2(SP-1)(P-1)}{(M-1)}$

Linear ranking allows values of selective pressure in [1.0, 2.0].

Rank (uniform)

uniformranking (μ)

Parameters μ – Selection pool.

Returns Selection function, see *Interface*.

In uniform ranking, the best μ individuals are assigned a selection probability of $1/\mu$ while the rest are discarded *[SC95]*.

Stochastic universal sampling (SUS)

sus (*fintess*, *N*)

Stochastic universal sampling (SUS) provides zero bias and minimum spread *[BK87]*. The individuals are mapped to contiguous segments of a line, such that each individual's segment is equal in size to its fitness exactly as in roulette-wheel selection. Here equally spaced pointers are placed over the line as many as there are individuals to be selected.

Consider N the number of individuals to be selected, then the distance between the pointers are $1/N$ and the position of the first pointer is given by a randomly generated number in the range $[0, 1/N]$.

References

[SC95] Schwefel H.P., Evolution and Optimum Seeking, Wiley, New York, 1995.

[BK85] Baker J.E., Adaptive selection methods for genetic algorithms, In Proceedings of International Conference on Genetic Algorithms and Their Applications, pp. 100-111, 1985.

[BK87] Baker, J. E., Reducing Bias and Inefficiency in the Selection Algorithm. In [ICGA2], pp. 14-21, 1987.

R

S

U