Statistical Mutation Operator for Generic Algorithms

Ghodrat Moghadampour, PhD
mg@puv.fi
Principal Lecturer
Vaasa University of Applied Sciences
Vaasa
Finland

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Outline

- Motivation
- Genetic Algorithms
- Parameters
- Mutation Operator
- Adaptation for Mutation Operators
- The statistical Mutation Operator
- Experimentation
- Conclusions

Motivation

- In the real world, there are numerous hard problems, which cannot be solved with conventional techniques within reasonable time, like optimization problems:

$$\frac{1}{4000}\sum_{i=1}^{n}(x_i - 100)^2 - \prod_{i=1}^{n}\cos\left(\frac{x_i - 100}{\sqrt{i}}\right) + 1$$

- Conventional techniques require rigid assumptions, like convexity, linearity, differentiability, explicitly defined objectives and so on.

Evolutionary Algorithms

- It is generally accepted that any evolutionary algorithm must have five basic components:
  1. a genetic representation of a number of solutions to the problem
  2. a way to create an initial population of solutions
  3. an evaluation function for rating solutions in terms of their “fitness”
  4. “genetic” operators that alter the genetic composition of offspring during reproduction
  5. values for the parameters, e.g. population size, probabilities of applying genetic operators

Genetic Algorithms

ii. Cross over the pair with crossover probability *Pc* at a randomly chosen point to form two offspring

iii. Mutate the two offspring at each locus with probability *Pm* and place the resulting individuals in the new population

4. Replace the current population with the new population

5. While the termination condition is false go to step 2.

Representation

-1.233011, 2.45612, 8.309812
14.840269,7.901482,-6.614903
10.710982,-42.002391,31.910283
...

♣ ♦ ♥ ♠ ⊗ • ⊗ ∅

∅ • ♥ ♠ ♣ ♦ • ♠

♠ ♠ ♠ ♣ ♣ ♦ ♦ ⊗
...

10110001
11111000
00011010
...

Evolutionary Process

♣ ♦ ♥ ♠ ⊗ ● ⊗ ∅

∅ ● ♥ ♠ ♣ ♦ ● ♠

♠ ♠ ♠ ♣ ♣ ♦ ♦ ⊗

...

♥ ♠ ♦ ♣ ⊕ ⊗ ∅ ●

Parameters in Evolutionary Algorithms

- Evolutionary algorithms are affected by more parameters than optimization methods typically

- A source of their robustness as well as a source of frustration in designing them.

Adaptation To The Evolutionary process

- Adaptation can be applied to problems as well as to evolutionary processes.
- Adapting evolutionary process refers to modifying some components of genetic algorithms to provide an appropriate form of the algorithm, which meets the nature of the given problem.
- These components could be any of representation, crossover, mutation and selection.

Adaptation To The Configuration of The Algorithm

- Adapting algorithm configuration suggests a way to tune the parameters of the changing configuration of genetic algorithms while solving the problem.
- Some of such parameters are:
  - population size and structure, like subpopulations
  - genome representation (floating point, binary, parse tree, matrix), precision and length
  - crossover type (arithmetic, -point, etc.), the number of crossover points and probability
  - mutation type (uniform, Gaussian, etc.), mutation variance and probability
  - selection type (tournament, proportional, etc.), tournament size.

Optimal Parameters

- The challenge is that optimal parameters of an EA are problem dependent and there is a large set of possible EA settings.
- The No-Free-Lunch theorem implies that no set of parameters for an EA is superior on all problems.
- Finding the right parameter values is a time-consuming task and it has been the subject of many researches.

Parameter Setting Methods

- The main criteria for classifying parameter setting methods are:

1) what is changed:
    - representation
    - evaluation function
    - variation operators (mutation and recombination)
    - selection
    - replacement
    - population

Parameter Setting Methods

2) How the change is made:

- deterministic (or fixed) parameter control (parameter tuning) in which the parameter-altering transformations takes no input variables related to the progress of search method
- adaptive (also called explicitly adaptive) parameter control in which there is some form of feedback from the search
- self-adaptive (implicitly adaptive) parameter control in which the parameters to be adapted are encoded into the chromosomes and undergo mutation and recombination

Mutation

- Mutation is a bit reversal event that occurs with small probabilities per bit.

Binary Mutation:

1 0 0 1 1 1 0 0 0 0 0 0 0 1 1 ⇒ 10.95

$-50 \le x \le 50$

0 0 0 1 1 1 0 0 0 0 0 0 0 1 1 0 ⇒ -39.05

Real Value Mutation:

6.290351 ⇒ 6.290102

“Optimal” Parameter Values

- Efforts to tune the mutation probability have resulted to different values and hence leaving practitioners in ambiguity.
- As results of tuning “optimal” mutation rate, the best rate found to be Pm=0.001 (De Jong 1975), Pm=0.01(Grefenstette 1986), 0.005≤Pm≤0.01 (Schaffer et al. 1989) and Pm=1/L (Mühlenbein 1992), where L is the length of the bit string (Michalewicz et al. 2004).

Improving Mutation Operators

- Problem with the classical mutation operators: they are dependent on many parameters and setting the parameters is already a challenging task while the main objective is to assist the algorithm in avoiding stagnation and finding problem solution
- Solution: invent operators, which are independent of parameters and are capable of exploring search space comprehensively

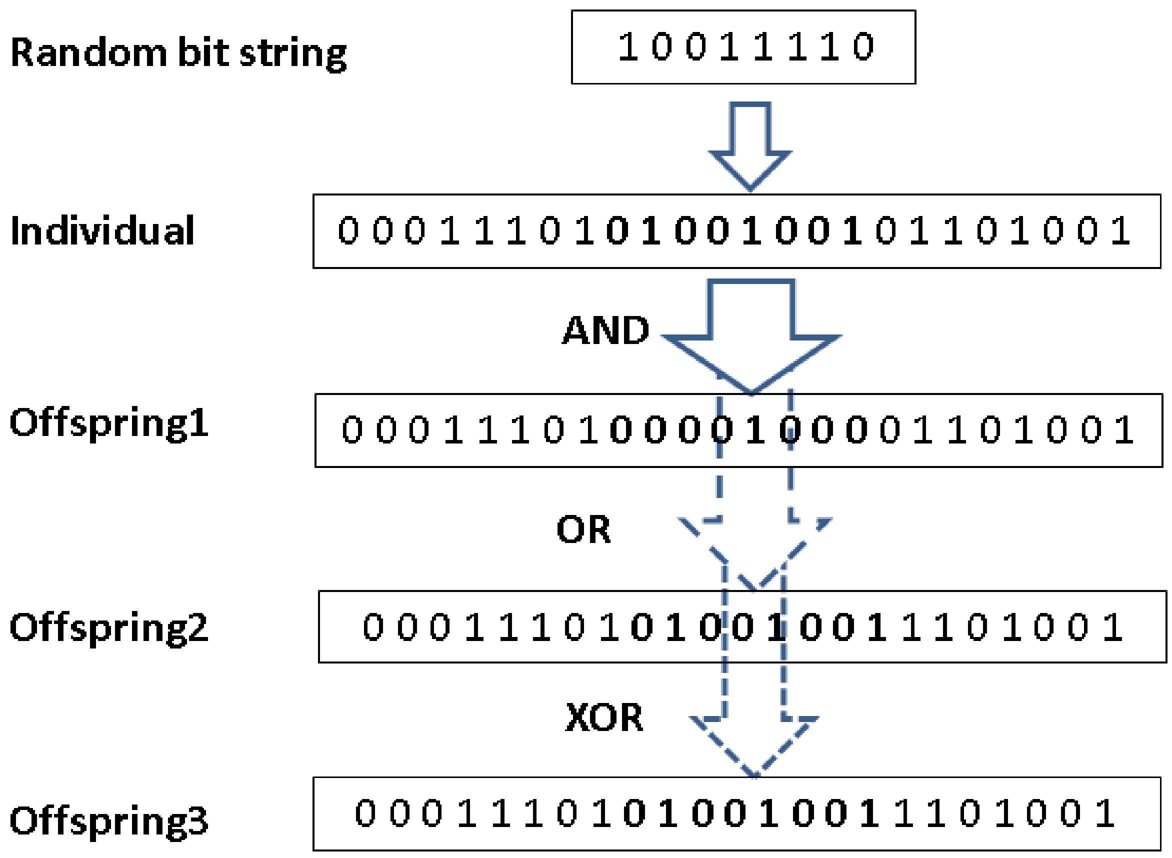
Polymorphic Random Building Block Operator

- The polymorphic random building block operator involves only one individual.
- It does not require any pre-defined parameter value and it automatically takes into account the length (number of bits) of the individual at hand.
- In practice, the polymorphic random building block operator selects a section (s_1) of random length (l) from the binary presentation of the individual at hand.

Polymorphic Random Building Block Operator

- In the next step the operator produces randomly a binary string (s_2) of the same size (l) and then applies AND, OR and XOR bitwise operators between s_1 and s_2 in turn in order to produce three new offspring.
- In the next step these newly generated offspring go through selection procedure one by one to be either selected or discarded

Polymorphic Random Building Block Operator



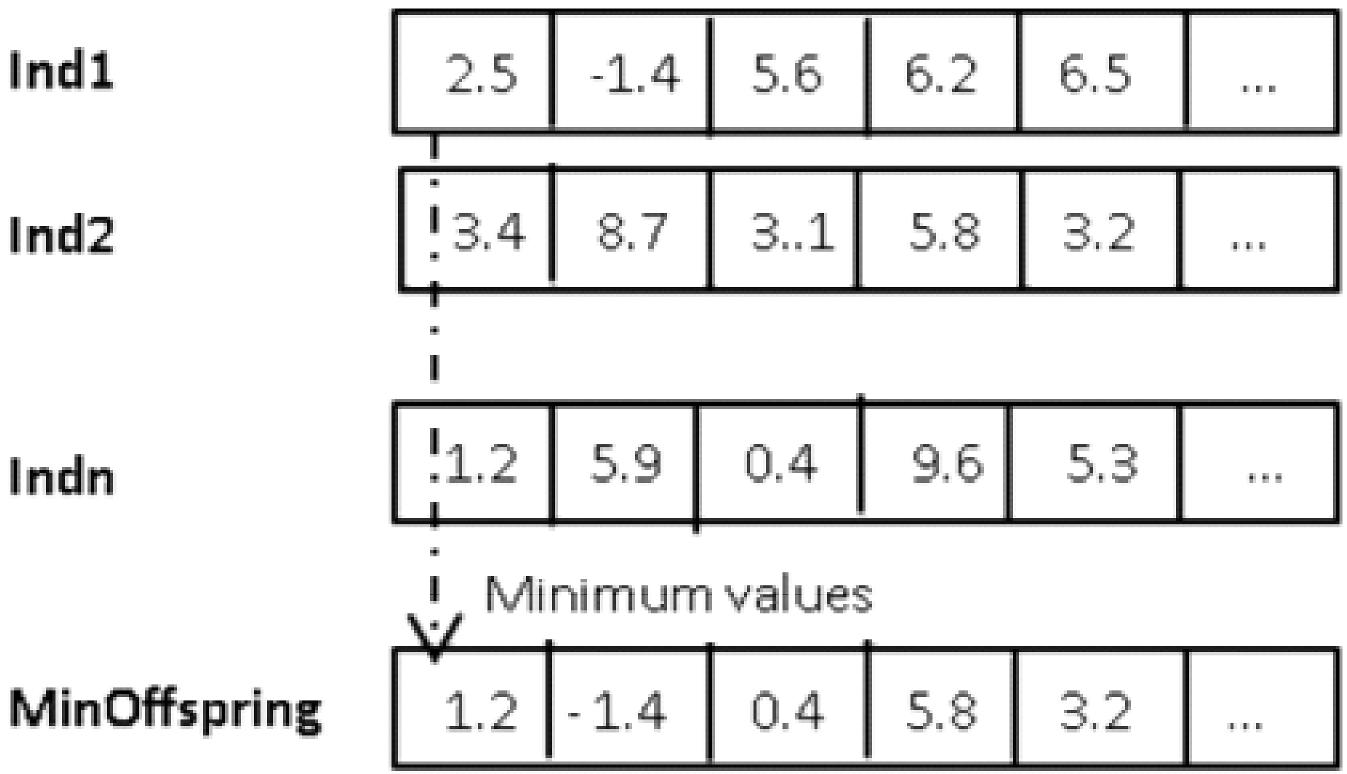
Statistical Mutation Operator

- The statistical mutation operator generates three individuals during each generation: minimum offspring, maximum offspring and average offspring.

Minimum Offspring

- The statistical mutation operator generates three individuals during each generation.
- The first individual (*minimum offspring*) is created so that the value of each of its variables is the minimum value of the respective variables of all individuals in the population.

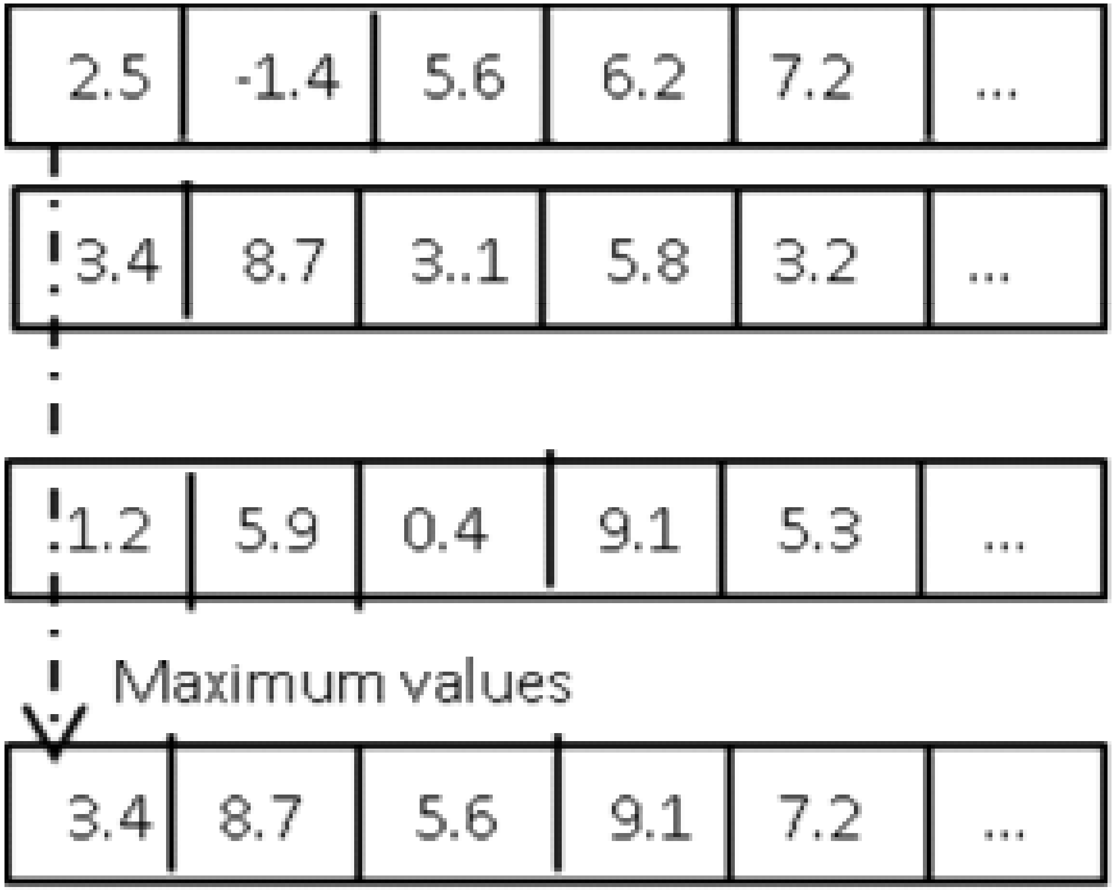
Minimum Offspring



Maximum Offspring

- The second individual (*maximum offspring*) is created so that the value of each of its variables is the maximum value of the respective variables of all individuals in the population.

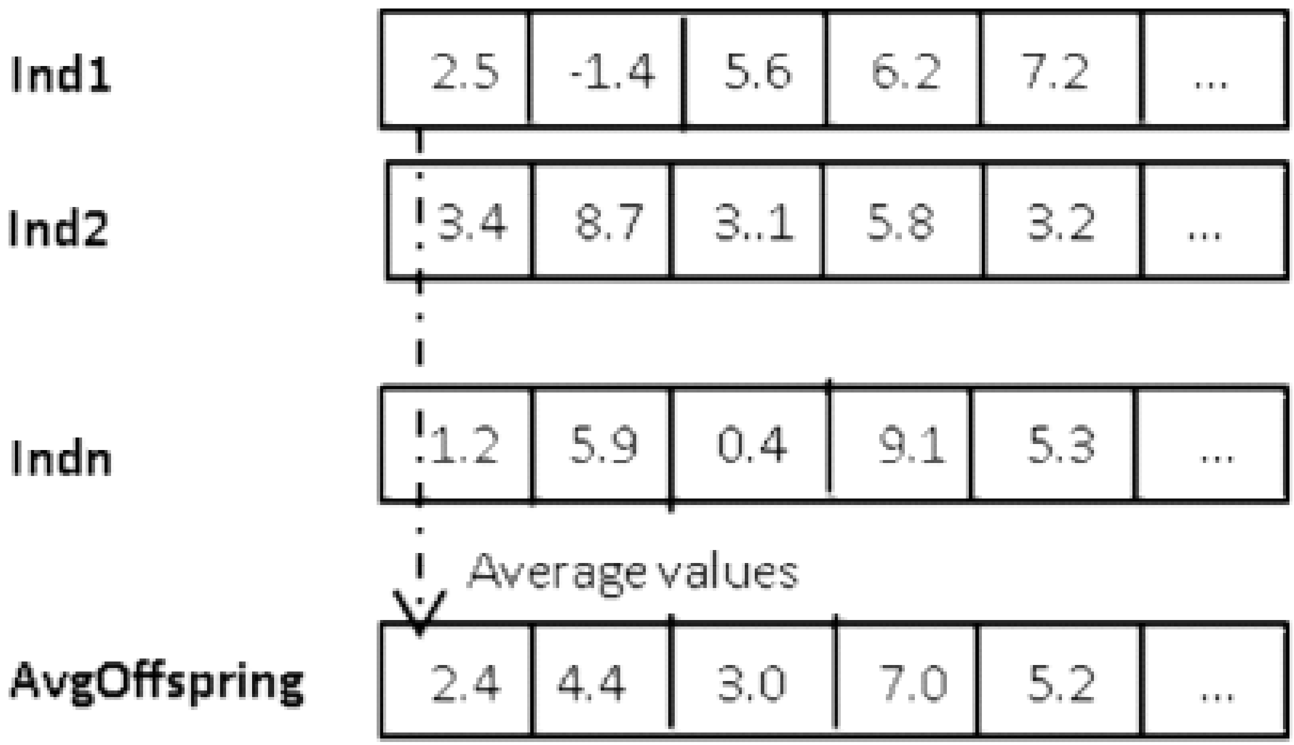
Maximum Offspring



Average Offspring

- The third individual (*average offspring*) is created so that the value of each of its variables is the average value of the respective variables of all individuals in the population.

Average Offspring



Experimentation

- Total 1650 runs on 33 test cases to solve the following minimization problems: Ackley's function, Colville's function, Griewank's function F1, Rastrigin's function, Rosenbrock's function and Schaffer's F6 and F7 functions.
- For multidimensional problems with optional number of dimensions (n), the algorithm was tested for n=1, 2, 3, 5, 10, 50.
- .

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES

Experimentation

- The population size was set to 9.
- Each test case was repeated 50 times.
- For each test case, the performance of the statistical mutation operator combined with polymorphic random building block operator was tested against simple polymorphic random building block operator, single-point mutation operator with 1%, 5% and 8% mutation rates, multipoint mutation operator with 5%, 8% and 15% mutation rates.

Conclusions

- Comparing test results revealed that combining statistical mutation operator with polymorphic random building block operator led to better fitness values within fewer function evaluations compared to other tested alternatives.
- The fascinating feature of statistical mutation and polymorphic random building block operators is that they are dynamic and do not require any pre-set parameter.

Conclusions

- However, for mutation operators the mutation rate and the number of mutation points should be set in advance.
- The statistical mutation and polymorphic random building block operators can be used straight off the shelf without needing to know their best recommended rates.
- Hence, they lack frustrating complexity, which is typical for different versions of the mutation operator.

Conclusions

- The integer and decimal mutation operators proved to outperform the classical crossover and mutation operators in 78% of test cases in terms of the quality of the solution and in 85% of test cases in terms of required function evaluations.
- In 56% of test cases the integer and decimal mutation operators resulted in at least 50% of improvement in the quality of solutions than the classical crossover and mutation operators.

Thank you!

☺

VAASAN AMMATTIKORKEAKOULU
VASA YRKESHÖGSKOLA
UNIVERSITY OF APPLIED SCIENCES