Available online at www.sciencedirect.com

ScienceDirect

Procedia Engineering 181 (2017) 634 – 640

Procedia Engineering

www.elsevier.com/locate/procedia

10th International Conference Interdisciplinarity in Engineering, INTER-ENG 2016

Comparative Study between the Improved Implementation of 3 Classic Mutation Operators for Genetic Algorithms

Razvan Cazacu*

Petru Maior University, Nicolae Iorga 1, Tirgu Mures 540088, Romania

Abstract

Mutation is the most important Genetic Algorithms operator, allowing them to thoroughly explore the design space of an optimization problem. If designed correctly it also allows for the exploitation of promising solutions, task usually attributed to crossover. This study compares the performance of three classic mutation operators: uniform, polynomial and Gaussian. The tool used is the OOGA framework which implements an improved and unified variant of the mutation operators. GA performance is evaluated on a benchmark structural optimization problem using three criteria: accuracy, reliability and efficiency. The optimum configuration of each operator is also explored by varying mutation parameters over a range of possible values. Overall the study is aimed at the optimization practitioners, offering them the means to make informed decisions about the right mutation operator and its setting for particular problems.

Peer-review under responsibility of the organizing committee of INTER-ENG 2016

Keywords: Genetic Algorithm; Uniform Mutation; Polynomial Mutation; Gaussian Mutation; performance comparison; MATLAB; OOGA.

1. Introduction

Although the genetic algorithms (GA) are no longer a new optimization technique, their research and application in general and in engineering problems in particular has been constantly increasing over the past 15 years [1]. Inspired by the biological evolution of species, GAs work with groups of solutions (populations) and rely mainly on the principle of selection based on fitness and on two operators: mutation – traditionally responsible for the exploration of the design space – and crossover – responsible for the exploitation of promising solutions.

* Corresponding author. Tel.: +40-749-071521; fax: +40-265-262275.
E-mail address: razvan.cazacu@ing.upm.ro

Peer-review under responsibility of the organizing committee of INTER-ENG 2016
doi:10.1016/j.proeng.2017.02.444

In an attempt to increase the efficiency and reliability of GAs recent studies propose new mutation operators [2,3]. These offer increased performance but are tested solely on synthetic mathematical functions where the computational effort is not an issue and are compared to basic implementations of existing operators. Other research is focused towards innovating classic mutation operators in order to enhance their overall performance [4-8].

The present study compares the performance of three improved classic mutation operators on a benchmark structural optimization problem, offering results and conclusions that can be used to optimally configure and tweak the genetic algorithms in general.

Nomenclature

GA	Genetic Algorithm
p	mutation parameter (*Δ* for uniform mutation, *η* for polynomial mutation, *scale* for Gaussian mutation)
IP	individual mutation probability
GP	gene mutation probability
FPV	final value of the mutation parameter in the evolutionary scheme

2. Methodology

The instrument used for optimization is the OOGA library [7], an object oriented framework developed in MATLAB for the research and implementation of genetic algorithms. The three classic mutation operators compared in this study are:

- Uniform Mutation [9,10];
- Polynomial Mutation [11];
- Gaussian Mutation [12,13].

They are implemented in the framework as a unified concept [8], in the mathematical form given by (1):

$$x_k^{(t+1)} = x_k^{(t)} + s \cdot f(u,p) \cdot R, \qquad k = 1 \ldots n \tag{1}$$

where:

k - the k^{th} gene of the chromosome;
n - length of the chromosome (number of genes);
$x_k^{(t+1)}$ - value of the k^{th} gene in the chromosome at generation t+1;
$x_k^{(t)}$ - value of the k^{th} gene in the chromosome at generation t;
s - sign of the operation (randomly chosen but configurable to favor a direction);
$f(u,p)$ - function of the random number u and the mutation parameter p, with a different form for each operator;
R - range of the mutation, depending on the current value of the gene and its bounds.

For more flexibility, the mutation probability is provided not as a single value but as separate values for the probability of an individual being picked for mutation (*IP*) and the probability of each gene of the selected individuals to actually suffer mutation (*GP*).

The mutation operators are also augmented with an evolutionary scheme which allows the variation of the mutation parameter over generations using a power function. By adjusting the evolutionary scheme mutation can be configured to search the design space more freely in the beginning and to narrow the search range towards the later generations in order to help crossover fine-tune the optimal solution.