Chapter 1

Introduction:
A Survey of the Evolutionary Computation Techniques for Software Engineering

Monica Chiş
Siemens IT Solutions and Services PSE, Romania

ABSTRACT

This chapter aims to present a part of the computer science literature in which the evolutionary computation techniques, optimization techniques and other bio-inspired techniques are used to solve different search and optimization problems in the area of software engineering.

MOTIVATION

Evolutionary algorithms are methods, which imitate the natural evolution process. An artificial evolution process evaluates fitness of each individual, which are solution candidates. The next population of candidate solutions is formed by using the good properties of the current population by applying different mutation and crossover operations.

Different kinds of evolutionary algorithm applications related to software engineering were searched in the literature. Because the entire book presents some interesting information chapter of some evolutionary computation techniques applied into the software engineering we consider necessary to present into this chapter a short survey of some techniques which are very useful in the future research of this field.

The majority of evolutionary algorithm applications related to software engineering were about software design or testing.

Software Engineering is the application of a systematic, disciplined, quantifiable approach to the development, operation, and maintenance of software, and the study of these approaches; that is, the application of engineering to software (Abran and Moore, 2004).

The purpose of this book is to open a door in order to find out the optimization problems in different software engineering problems. The idea of putting together the application of evolutionary computation and evolutionary optimization techniques in software engineering problems provided to the researchers the possibility to study some existing
works in this fields and to found approaches and methodologies that will combine advances in computational methods to model and analyze complex problems with powerful optimization techniques based on evolutionary principles in order to create more general, robust, accurate and computationally methods for evolutionary computation applications.

Software Engineering

Software engineering describes the collection of techniques that apply an engineering approach to the construction and support of software products. Software engineering activities include managing, costing, planning, modeling, analyzing, specifying, designing, implementing, testing and maintaining software products. Whereas computer science provides the theoretical foundations for building software, software engineering focuses on implementing software in a controlled and scientific way (Fenton et al. 1997).

Measurement is the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules.

An entity is an object or an event in the real world. An attribute is a feature or property of an entity. We measure attributes of entities. Rules are used to determined the values for an attribute in order to made the process repeatable not subjective. It is very important to consider the representation form of each attributes (measurement unit).

Measurement is considered to be fundamental to any engineering principle. Software projects well managed have to be measured first. There are a lot of applications of evolutionary computation and optimization in the field of software engineering problems.

Defining the project estimated cost, duration and maintenance effort early in the development life cycle is a valuable goal to be achieved for software projects. There are a lot of researches in this which have used evolutionary computation in the applications of software engineering.

Evolutionary Computation

Evolutionary computation is emerging as a new engineering computational paradigm, which may significantly change the present structural design practice. A great number of evolutionary computational models have been developed, including evolutionary algorithms, genetic algorithms, the evolution strategy, evolutionary programming.

Evolutionary Algorithm (EA) incorporates aspects of natural selection or survival of the fittest. An evolutionary algorithm maintains a population of structures (initially randomly generated) that evolves according to rules of selection, recombination, mutation and survival, referred to as genetic operators. EAs are useful for optimization when other techniques such as gradient descent or direct, analytical discovery unsuccessful.

The background on various implementations of evolutionary algorithms is provided in Fonseca and Fleming (1995), Bäck (1996), Coello (1999), and Van Veldhuizen and Lamont (2000).

Genetic Programming creates computer programs in the scheme computer languages as the solution. A genetic algorithm creates a string of numbers that represent the solution.

Genetic programming consists of the following four steps:

1. Generate an initial population of random compositions of the functions and terminals of the problem (computer programs).
2. Execute each program in the population and assign it a fitness value according to how well it solves the problem.
3. Create a new population of computer programs.
4. The best computer program that appeared in any generation, is selected as the result of genetic programming (Koza 1992; Benzhaf, Nordin, Keller and Francone, 1998).
Related Content

Risk Factors of Breast Cancer in Indian Context: A Systematic Review
[www.igi-global.com/article/risk-factors-of-breast-cancer-in-indian-context/79913?camid=4v1a](www.igi-global.com/article/risk-factors-of-breast-cancer-in-indian-context/79913?camid=4v1a)

A Hybrid Meta-Heuristic to Solve a Multi-Criteria HFS Problem
[www.igi-global.com/chapter/hybrid-meta-heuristic-solve-multi/69888?camid=4v1a](www.igi-global.com/chapter/hybrid-meta-heuristic-solve-multi/69888?camid=4v1a)

Page Number and Graph Treewidth
[www.igi-global.com/chapter/page-number-graph-treewidth/63818?camid=4v1a](www.igi-global.com/chapter/page-number-graph-treewidth/63818?camid=4v1a)

A Non-Permutation Flow Shop Manufacturing Cell Scheduling Problem with Part's Sequence Dependent Family Setup Times