

Preface

The idea of this book was conceived a long time ago when I started to research activities in the field of software engineering and I tried to apply evolutionary computation techniques. In that moment, I realized that there is a lot of research in this field and it would be better to put together some research in this field. Of course it is impossible to keep in a single book all the topics in this field.

The book “Evolutionary Computation and Optimization Algorithms in Software Engineering: Applications and Techniques” is a collection of techniques and applications which try to solve software engineering problems by using evolutionary computation and optimization techniques.

THE MOTIVATION OF SOFTWARE ENGINEERING AND EVOLUTIONARY COMPUTATION

According to the Software Engineering Body of Knowledge (SWEBOK), 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. Software engineering means the application of engineering to software. Software Engineering is related to the disciplines of computer science, project management and systems engineering. It is considered to be a subfield of Computer science. The building of a software system is usually considered a project and the management of it borrows many principles from the field of Project management. A lot of problems from complexity of large systems knowledge are applied to many software engineering problems.

Evolutionary Computation (EC) is the general term for several computational techniques which are based to some degree on the evolution of biological life in the natural world. Evolutionary Computation methods have been successfully extended to solve multi-objective optimization tasks. Several evolutionary computational models have been developed, including evolutionary algorithms, genetic algorithms, the evolution strategy, evolutionary programming.

The importance of using evolutionary computation and optimization algorithms in software engineering domain is evident by the great number of books and paper dedicated to this subject. Executing any software project requires skills in two key dimensions— engineering and project management. While engineering deals with issues of architecture, design, coding, testing etc., project management deals with planning, monitoring, risk management etc. Consequently, this book focuses on these two dimensions, and has presented some new evolutionary techniques in these directions.

The present book is one which presents some interesting ideas for applying evolutionary techniques in some problems from software engineering.

For my point of view can be useful for those who study software engineering because they can optimize the existing information and processing and both for people which work on the evolutionary computation and optimization field because they can discover new directions form their research.

A GUIDED TOUR OF THE CHAPTERS

The book is composed of 11 chapters, plus an introduction chapter which is a short survey of some research activities in the field of the application of evolutionary computation in software engineering. The chapters are authored by different groups of researchers and practitioners in software engineering and evolutionary computation and optimization from all over the word.

The book starts with a short introduction in some application of evolutionary and bio-inspired techniques in the software engineering. The introduction chapter has the role to identify the key points of this subject and to point the importance of the subject. The readers can start use this as a preliminary information in order to study more about the applications of evolutionary computation and optimization techniques into the software engineering problems.

In Chapter 1, **Introduction: A Survey of the Evolutionary Computation Techniques for Software Engineering**, *Monica Chiş* aims to present a part of the computer science literature in which the evolutionary computation techniques are used to solve different search and optimization problems in the area of software engineering.

Software development effort estimation is a critical activity for the competitiveness of a software company. Effort estimation is one of the most important parts of the project development for planning and monitoring project development and for delivering the product on time and within budget. In Chapter 2, **Using Evolutionary Based Approaches to Estimate Software Development Effort**, *Filomena Ferrucci, Carmine Gravino, Rocco Oliveto, Federica Sarrohave* present a state-of-the-art in the field of using evolutionary computation techniques for effort estimation. Some suggestions for future research directions are also provided.

Chapter 3, **The Application of Genetic Algorithms to the Evaluation of Software Reliability** is written by *Angel Fernando Kuri-Morales*. The author shown that evolutionary computation (illustrated in this chapter with genetic algorithms) is a tool enabling the designer to tackle the software reliability problem by solving the complex optimization involved in evaluating the status of the systems and taking the appropriate decisions in line with the development of the system.

The possibility to synthesize higher quality modeling through the evolution of Genetic Algorithms was presented in Chapter 4, **Synthesis of Object-Oriented Software Structural Models Using Quality Metrics And Co-Evolutionary Genetic Algorithms**. The authors *Dilza Szwarcman, Thiago S. M. Guimarães, André Vargas Abs da Cruz and Marco Aurélio C. Pacheco* elaborated and evaluated a model able to combine the technique of the Co-evolutionary Genetic Algorithm to the object oriented software engineering and to the metrics proposed in the QMOOD model (Quality Models for Object-Oriented Modeling), in order to synthesize software modeling with better quality. The research evaluated the performance of the model before the random search by its application, in order to optimize the different quality attributes in simple and more complex modeling problems.

Artificial Immune Systems, a biologically inspired computing paradigm such as Artificial Neural Networks, Genetic Algorithms, and Swarm Intelligence, embody the principles and advantages of vertebrate immune systems. It has been applied to solve several complex problems in different areas

such as data mining, computer security, robotics, aircraft control, scheduling, optimization, and pattern recognition. An interesting and very useful work was presented in Chapter 5 **Application of Artificial Immune Systems Paradigm for Developing Software Fault Prediction Models**. The authors, *Cagatay Catal and Soumya Banerjee*, have showed that the Artificial Immune Systems (AISR) is an emerging computational intelligence technique that provides promising results for many problems. They have investigated the use of Artificial Immune Systems paradigm for software fault prediction problem during the research implementation. The study showed the elaborated working steps for such algorithm with a software engineering dataset.

Software fault prediction can play an important role in ensuring software quality through efficient resource allocation. Chapter 6, **Genetic Programming for Cross-Release Fault Count Predictions in Large and Complex Software Projects**, written by *Wasif Afzal, Richard Torkar, Robert Feldt, Tony Gorschek* have used the genetic programming for fault prediction.

The paper evaluates genetic programming (GP) for cross-release prediction of fault counts on data set from large real world projects. The authors evaluate the created models on fault data from several large and real world software projects, some from open-source and some from industrial software systems. Some very useful part of this work is the part in which multiple different fault count modeling techniques, both traditional and several machine learning approaches are compared.

The service discovery has become an emerging phenomenon in software engineering and process engineering as well. In Chapter 7, **Exploring a Self Organizing Multi Agents Approach for Service Discovery**, the authors, *Hakima Mellah, Soumya Banerjee, Salima Hassas, Habiba Drias*, have been presented the service discovery referring bio-inspired and evolutionary agents. The emphasis has been initiated a model for service failure detection and recovery in distributed environment with the help of the proposed self organized Multi Agent. The positioning of the self organized agents and its protocol has been elaborated in the chapter with a brief service case model. The relevance and potential applications of self organized agents in service discovery paradigm is an emerging field of research in different flavors of software engineering and recommender system. The information of this chapter can be use very useful for different area of work. Collective intelligence from nature has already been envisaged effectively in the service response and failure coverage; hence service discovery through self organized agents could be treated as major mile stone of software engineering research.

Another useful chapter for the software engineering domain is the Chapter 8, **Innovative Hybrid Genetic Algorithms and Line Search Method for Industrial Production Management**, written by *Pandian Vasant*. In this chapter a hybrid optimization approach was presented. The hybrid genetic algorithm with line search approach has improved the fitness value obtained by genetic algorithms alone. This technique is a robust and excellent efficient method for non-linear industrial production planning problems. In particular this is a very useful techniques for planning in software engineering problems. The author strongly believes that there should be another hybrid approach such as particle swarm optimization, ant colony optimization, artificial immune system, neural network, fuzzy logic, tabu search and evolutionary computation (Genetic programming, Evolutionary strategy & Genetic algorithms) to improve the solution for the decision variables in real world situation. The hybrid optimization techniques of meta-heuristic approaches with classical optimization approaches will be great added advantages for solving production management problems in a fuzzy environment.

Chapter 9, **Automatic Test Sequence Generation for State Transition Testing via Ant Colony Optimization**, written by *Praveen Ranjan Srivastava and Baby* have proposed an Ant Colony Optimi-

zation technique for automated and fully coverage state-transitions in the system. Through proposed algorithm all the transitions are easily traversed at least once in the test-sequence.

According to the *Arjan Seesing and Hans-Gerhard Gross* Testing is a difficult and costly activity in the development of object-oriented programs. The challenge is to come up with a sufficient set of test scenarios, out of the typically huge volume of possible test cases, to demonstrate correct behavior and acceptable quality of the software. For this they have presented in Chapter 10, **Object Oriented Software Testing with Genetic Programming and Program Analysis**, genetic programming as a heuristic search algorithm which is suitable to evolve object-oriented test programs automatically to achieve high coverage of a class. The chapter introduced a method for generating test cases automatically according to a coverage criterion for object-oriented software, based on the application of genetic programming. The authors have considered the genetic programming as a likely more suitable technique for the generation of test operation invocation sequences, in contrast to applying genetic algorithms. They have presented the principles of Genetic Programming-based test case generation and shown how it can be improved with program analysis techniques and a two-stage execution.

Chapter 11, **Assessment of Software Quality: A Fuzzy Multi-Criteria Approach** is written by *Praveen Ranjan Srivastava, Ajit Pratap Singh and Vageesh K.V.* The authors have proposed a fuzzy multi-criteria approach to quantify the abstract software quality attributes. This approach gives a method to identify the best software product from a set of products by accommodating the perspectives of a manager, user and developer involved with the product. The model can be further enhanced by increasing the number of quality attributes, and linguistic variables for example by increasing the degrees of importance to eleven linguistic values.

Chapter 12, **Verification of Attributes in Linked Lists Using Ant Colony Metaphor**, describes the validation of the attributes of linked list using modified pheromone biased model (of Ant Colony) under complex application environment mainly for kernel configuration and device driver operations. The authors *Soumya Banerjee and P.K. Mahanti* incorporate into the presented work the idea of pheromone exploration strategy with small learning parameter associated while traversing a linked list. This process of local propagation on loop and learning on traversal is not available with the conventional validation mechanism of data structure using predicate logic.

We consider that the book can be very useful for researchers and teachers in the field of evolutionary computation and optimization techniques for software engineering.

I wanted to thank again to all the people who contribute to this project in order to develop and to put together different researches. The fact that the book presents different applications and techniques it is very important for us to contribute to the development of new applications. The presented book chapters can represent some important information for the future works in software engineering in which the evolutionary computation techniques have to be applied.