Foreword

The importance of using evolutionary computation and optimization algorithms in software engineering domain is evident by the great number of books and papers 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 - Evolutionary Computation and Optimization Algorithms in Software Engineering: Applications and Techniques - focus on these two dimensions, and has presented some new evolutionary techniques in these directions.

The present book is one which presented some interesting idea of applying evolutionary techniques in software engineering domain.

From my point of view, the book 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 for their research.

The 12 chapters of the book have been written by a number of researcher and practitioners in software engineering and evolutionary computation and optimization from all over the word.

The book starts with a short introduction in some applications of evolutionary and bio-inspired techniques in the software engineering. This step was important because have been tried to identify the key points of this subject and to point the importance of the subject. The readers can start using this book as preliminary information in order to study more other subjects. This step is essential because it the contact with the general subject. Another very important chapter is the usage of evolutionary approaches in the software development effort estimation. A very interesting a state of the art in the field by reporting on the most significant empirical studies is presented in the first chapter. Another interesting topic discussed in this book is the application of Genetic Algorithms to the evaluation of software reliability. It is very interesting the way in which Genetic Algorithms are used for optimization.

One of the biggest provocations for the developer of object-oriented software is modeling and developing of the objects themselves, so that they are easily reusable in complex systems. The final quality of the software depends mostly on the quality of the modeling developed for it. Modeling and specification of software are fundamental steps for making the software development an activity of engineering. Design is the activity in which software behavior and structure are elaborated. During this phase many models are developed anticipating several views of the final product and making software evaluation possible even before the software is implemented. The work studies a possibility to synthesize higher quality modeling through the evolution of Genetic Algorithms, a technique that has proved to be efficient in dealing with problems involving large search spaces. The study of object-oriented software engineering involved the establishment of software development phases and the characterization of the representation used in modeling phase and, in particular, the characterization of class diagrams based on UML. The study also investigated software quality metrics such as Reutilization, Flexibility, Understandability, Functionality, Extensibility and Effectiveness.

Fault prediction is one of the key important issues in software development. The idea of applying Artificial Immune Systems Paradigm for Developing Software Fault Prediction Models it is another very important topic of the book.

In this context the idea of using Genetic Programming (GP) for cross-release fault count predictions in large and complex software projects presented in this book is very useful for application in software development.

Software fault prediction can play an important role in ensuring software quality through efficient resource allocation. This could, in turn, reduce the potentially high consequential costs due to faults. Predicting faults might be even more important with the emergence of short-timed and multiple software releases aimed at quick delivery of functionality. Previous research in software fault prediction has indicated that there is a need: i) to improve the validity of results by having comparisons among number of data sets from a variety of software; ii) to use appropriate model evaluation measures and iii) to use statistical testing procedures. Moreover, cross-release prediction of faults has not yet achieved sufficient attention in the literature. In an attempt to address these concerns, this paper compares the quantitative and qualitative attributes of 7 traditional and machine-learning techniques for modeling the cross-release prediction of fault count data. The comparison is done using extensive data sets gathered from a total of 7 multi-release open-source and industrial software projects. These software projects together have several years of development and are from diverse application areas, ranging from a web browser to robotic controller software. Our quantitative analysis suggests that Genetic Programming tends to have better consistency in terms of goodness of fit and accuracy across majority of data sets. It also has comparatively less model bias. Qualitatively, ease of configuration and complexity are fewer strong points for GP even though it shows generality and gives transparent models. Artificial Neural Networks did not perform as well as expected while linear regression gave average predictions in terms of goodness of fit and accuracy. Support Vector Machine regression and traditional software reliability growth models performed below average on most of the quantitative evaluation criteria while remained on average for most of the qualitative measures.

An interesting topic presented in this book was Multi Agent System (MAS) approach, for service discovery process to consider the user in the service discovery process involving his interactions under constraints. Self organization within the MAS is adopted by the recourse to a self organizing protocol conceived from bacteria colony and evolutionary computation paradigm.

Another interesting chapter refers to innovative hybrid genetic algorithms and line search method for industrial production management which can be very useful in software engineering.

Software testing is a key part of software development life cycle. The book contains a chapter on Ant Colony Optimization technique for automated and fully coverage state-transitions in the system is presented. This is a very useful theme for the practitioners and researchers.

Testing is a difficult and costly activity in the development of object-oriented programs. Very useful chapters in which has been evaluated genetic programming as a heuristic search algorithm which is suitable to evolve object-oriented test programs automatically to achieve high coverage of a class have been included in the present book.

Software Quality Assurance consists of monitoring the software engineering processes and ensuring the highest quality. A fuzzy multi-criteria approach to measure the total software quality and to identify the best alternative from a set of software products have been presented in the book. This is a very interesting topic for companies which work for software quality assurance. The fact that there are new techniques which can be applied is very useful.

The book does not cover the entire topic in the software engineering. Parts of some interesting subject are presented together and can be useful for the topics selected for presentation in the university curricula. Also it can be useful for the people involved in software engineering in the companies because can offer another interesting approaches for software development.

It is clear that this book is valuable and useful to all practitioners and researchers working on different fields in software engineering, evolutionary computation or optimization. Therefore it is recommended to anyone who has interest in new techniques in software engineering. It is hoped that others will follow the example and the topics presented in this book and present more studies in this field.

Consider that the book can be useful.

Crina Groşan
Babeş Bolyai University Cluj-Napoca
Romania

Crina Groşan *received her MS degree in Mathematics and PhD degree in Computer Science from Babes-Bolyai University, ClujNapoca, Romania in 2005. She is a Lecturer of Artificial Intelligence at Babes-Bolyai University. Her research focuses on different global optimization techniques and applications. She has been researching such topics as multiobjective optimization, global optimization, operations research, numerical analysis, computational intelligence and swarm intelligence. She has published over 100 research articles in peer reviewed international journals, book chapters and conference proceedings. She is co-author of two books on programming languages and is also a co-editor of four volumes titled Stigmergic Optimization, Swarm Intelligence in Data Mining, Hybrid Evolutionary Algorithms and Engineering Evolutionary Intelligent Systems, published by Springer Verlag, Germany. She guest edited a special issue on Soft Computing for Modeling and Simulation for the International Journal of Simulation Systems, Science & Technology, published by the UK Simulation Society. She is the managing editor of International Journal of Computational Intelligence Research and also serves the editorial board of few other Journals. She co-founded the Evolutionary Multiobjective Optimization: Design and Applications (EMODA) International workshop series in 2005.*