Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 47 (2015) 118 – 127

Procedia Computer Science

An effective approach to support multi-objective optimization in Software reliability allocation for improving quality

Dr. M. Sangeetha[a], Dr. C.Arumugam[b], Dr.K.M.Senthil Kumar[c], S.Hari Shankar[d]

[a]*Department of Computer Science and Engineering, CIT, Coimbatore 641014, India*
[b]*Department of Mechanical Engineering, CIT, Coimbatore 641014, India*
[c]*Mechanical Engineering, KCT, Coimbatore 641035, India*
[d]*Department of Computer Science and Engineering, CIT, Coimbatore 641014, India*

Abstract

Software development life cycle involves requirement analysis, design, coding, testing and implementation. Software Quality is the most important one, since the success of a Software Engineer relies on the development of failure free software. One of the main quality factors is reliability. Reliability can improve through Software Reliability Models, analyze of reliability data, proper utilization of debuggers, internal quality factors and evaluating the measurement results. Currently, there are several software reliability allocation techniques, but schedule planning and effective resource use are unavailable because they do not support multi-objective optimization. It is an effective software reliability allocation technique that supports multi objective optimization. This approach assists schedule planning as well as effective resources allocation. This paper can aid to increase the effectiveness of resource uses and to establish schedule planning. It will lead to improve Software Reliability.

Peer-review under responsibility of organizing committee of the Graph Algorithms, High Performance Implementations and Applications (ICGHIA2014)

Keywords: Reliability; Fault, Failure Rate; Genetic Algorithm; Multi-Objective function.

1. INTRODUCTION

This paper provides an introduction to the context of the research themes explored in the thesis. The paper introduces the reader to the issues, challenges and existing techniques in Software Reliability through Models. It gives a high-level overview of various approaches to improve quality through Reliability Model, Intelligent Code Evaluator, and Reliability Factor Analysis and discusses various Enhancement Techniques. The paper ends with an outline and contributions of the research work. Software Engineering J. D. Musa (1975) is the application of systematic, disciplined and quantifiable approach to the development,

Peer-review under responsibility of organizing committee of the Graph Algorithms, High Performance Implementations and Applications (ICGHIA2014)
doi:10.1016/j.procs.2015.03.190

operation, and maintenance of Software. It is a direct subfield of Computer Science and has some relations with Management Science. Software development life cycle involves requirement analysis, design, coding, testing, and implementation. Software Quality is the most important one, since the success of a Software Engineer relies on the development of failure free software

1.1Reliability Model

An effective approach is to identify optimal software reliability allocation with consideration of multiple constraints such as reliability, cost, and schedule. It supports the multi-objective optimization, which maximizes software reliability and minimizes cost and schedule. The multi-objective optimization enables not only to establish schedule plan but also to use resources effectively.

2. Genetic Algorithm for Optimization

Genetic Algorithms (GAs) are computer programs that mimic the processes of biological evolution in order to solve problems and to model evolutionary systems. This algorithm helps us to solve many computational problems, which have uncountable search space. It is very difficult for this kind of problems to obtain a global optimal solution because the search space is too large to spend infinite time to obtain the solution. It consists of 3 operations: initialization, selection, crossover, and mutation. During initialization, a population is generated randomly. More fit chromosomes are selected to be delivered to the next generation during the selection. Selected chromosomes are changed during crossover and mutation operations. These two operations make the population more diverse. Genetic algorithm has several strengths – little trends of local optima and easy applicability to any problem Gokhale et al (2006). For this reason, genetic algorithm is frequently applied in optimization problems. Many researchers from different area have been tried to adapt genetic algorithm to their research domain and they got the eye-opening result. As the optimization problems have been considered more complex factors, it is necessary to solve multi-objective optimization problem.

Multi-Objective Genetic Algorithms (MOGAs) have been suggested to support multi-objective optimization. There are many MOGAs, but NSGA-II is one of the most famous MOGAs because of their performance and simplicity. They evaluate the fitness of a chromosome by Pareto rank and crowding distance value. Pareto rank Dwyer and D'Onofrio (2011) is used to select non- dominant chromosomes, which cannot be compared with other chromosomes because their dimensions are different. Crowding distance value is used for guarantee of diversity for chromosomes. Elitism is also another operation to select more fit chromosomes to the next generation Goel and Yang (1997). GA with out multi-objective factors is very difficult to solve computational problems which have uncountable search space .Proposed system uses NSGA-II to identify optimal solutions because of the need for the consideration of software reliability, cost, and schedule simultaneously. Schemes for encoding a chromosome and fitness functions are also used for the application of NSGA-II.

Figure 1Genetic Process