Neurocomputing ∎ (∎∎∎∎) ∎∎∎–∎∎∎

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Multi-software reliability allocation in multimedia systems with budget constraints using Dempster–Shafer theory and improved differential evolution ☆

Feng Yue [a], Guofu Zhang [a,b,*], Zhaopin Su [a], Yang Lu [a,b], Ting Zhang [a]

[a] School of Computer and Information, Hefei University of Technology, Hefei 230009, China
[b] Engineering Research Center of Safety Critical Industrial Measurement and Control Technology, Ministry of Education, Hefei 230009, China

ARTICLE INFO

Article history:
Received 25 April 2014
Received in revised form
14 July 2014
Accepted 8 September 2014

Keywords:
Multi-software
Reliability allocation
Uncertainty
Dempster–Shafer theory
Differential evolution

ABSTRACT

In multimedia platform with many applications, reliability allocation plays an important role in the design of a software and has attracted increasing attention in recent years. Thus far, the issues of software reliability allocation have been discussed from many aspects, such as mathematical models and solutions to maximize the reliability. However, most of this research has concentrated on single software. The goal of this work is to investigate the possibility of solving multi-software reliability allocation in multimedia systems with budget constraints. For this purpose, we first develop an architecture-based multi-software Budget-Constrained Reliability-maximization model. In addition, we introduce Dempster–Shafer theory to identify the relative reliability weights of each element in the proposed model and present a searching algorithm based on differential evolution and encoding repair. Finally, contrast experiments are illustrated to demonstrate the proposed methods.

1. Introduction

In recent years, the rapid evolution of computing system technologies stimulates research and development in the field of multimedia integrating platform. And many multimedia systems have been deployed with the requirement from sectors such as traffic control, safety critical, education and information management. Failure of these systems can lead to big losses. The critical challenges to software engineers, who need to develop systems with, must assured high reliability levels while at the same time keeping the development time and costs low.

Software reliability is the probability of execution of software without failure under specified environment for a specified period of time [1,2]. In order to achieve the reliability goal of software, many researchers have devoted to various software reliability growth models (SRGMs) [3–6]. However, accurate software reliability estimation is not usually available until the software has been tested by a large number of failure data for a longer period of time. Furthermore, for software engineers, estimating the reliability of a software during the early phase of development is an important requirement for achieving an optimal system reliability goal.

Software reliability allocation can be operated during the design phase of a software, and which is a method of allocating a target reliability among subsystem and components. In the past decades, researchers devoted to constructing architecture-based models for the reliability allocation problem. The idea of software reliability allocation was first put forward by Zahedi and Ashrafi [7], who adopted analytic hierarchy process (AHP) [8] for modeling the software architecture with cost as the constraints and proposed a method for the system reliability maximization. Leung [9] used the operational profile to define a software utility function, which reflects a weighted sum of reliability-like measures based on the same AHP process as [7]. Helander et al. [10] described two approaches for reliability and cost planning: Reliability-Constrained Cost-Minimization (RCCM) and Budget-Constrained Reliability-Maximization (BCRM), both of which are multivariate constrained optimization problems. Rani and Misra [11] proposed a cost model for allocation of reliabilities during the design phase by minimizing a cost function, which depends on fixed development costs and on a previously experienced failure decrease cost. Tamura and Yamada [12] proposed a stochastic differential equations based SRGM model to control the software development process in terms of reliability, development effort and version-upgrade time for open source software.

☆This work was supported in part by the National Natural Science Foundation of China under Grants 61100127 and 61174170, the National Research Foundation for the Doctoral Program of Higher Education of China under Grant 20120111110001, and the Anhui Provincial Natural Science Foundation under Grants 1508085MF132 and 1508085QF129.

* Corresponding author at: School of Computer and Information, Hefei University of Technology, Hefei 230009, China.
E-mail addresses: Yuefeng@hfut.edu.cn (F. Yue), zgf@hfut.edu.cn (G. Zhang), szp@hfut.edu.cn (Z. Su), Luyang.hf@126.com (Y. Lu), creyer@163.com (T. Zhang).

http://dx.doi.org/10.1016/j.neucom.2014.09.103
0925-2312/

Meanwhile, many approaches for software reliability allocation have also been proposed in literature. Guan et al. [13] formulated an architecture-based approach for modeling software reliability optimization problem, and illustrated a dynamic programming algorithm to allocate the reliability to each component so as to minimize the cost of designing software while meeting the desired reliability goal. Pietrantuono et al. [14] proposed a reliability and testing resources allocation model called Discrete Time Markov Chain (DTMC)-type state-based model, which aimed to quantitatively identify the most critical components of software architecture in order to best assign the testing resources to them. Chatterjee et al. [15] first established a software system hierarchy, which combines the user's view of the system with that of the software manager and the programmer, then use fuzzy analytic hierarchy process (FAHP) [16] to derive the required model parameters from the hierarchy. Ahmadizar and Soltanpanah [17] dealt with a reliability optimization problem for a series system with multiple-choice and budget constraints, and developed ant colony optimization (ACO) for the problem. Sabbineni and Kurra [18] used Dynamic Algorithm for the purpose of allocation of reliability of components in a software product Design phase. Hou et al. [19] established a fuzzy multi-objective software reliability allocation model, and proposed a Multi-Objective Estimation of Distribution-Bacterial Foraging Algorithm (MEDA-BFA) based on estimation of distribution to solve the model. Kaveh et al. [20] proposed a new dynamic self-adaptive multi-objective particle swarm optimization(DSAMOPSO) method to solve binary-state multi-objective reliability redundancy allocation problems (MORAPs). Tavakkoli-Moghaddama et al. [21] proposed a genetic algorithm (GA) for a redundancy allocation problem for the series–parallel system.

From these literatures above, the area of software reliability allocation has taken the dual and often conflicting constraints of maximizing reliability and minimizing cost into account, and there are two aspects of views: one is searching an optimal reliability allocation method to achieve the given reliability such that the development cost can be as small as possible; the other is on the premise of the given cost so that the software reliability can be maximized. However, the existing work is mainly for a single software, and not suitable to deal with the case of multiple softwares. Moreover, the existing approaches mostly adopted AHP method to derive the required model parameters, and not consider the uncertainty and incompleteness in practical engineering.

Therefore, the aim of this paper is to propose a model of architecture-based multi-software reliability allocation, address before developing the software searching an optimal reliability allocation method on the premise of the given budget so that the multi-software utility can be maximized. On this condition, considering the uncertainty and incompleteness, we adopt Dempster–Shafer theory (DST) [29,22,31,24] to identify relative reliability weights, and develop an efficient differential evolution (DE) algorithm [25–27] to solve the model.

The remaining part of this paper is organized as follows: Section 2 formalizes the architecture-based multi-software Budget-Constrained Reliability-maximization (MSBCRM) model. Section 3 introduces how to identify relative reliability weights based on Dempster–Shafer Theory. In Section 4, we present the solution algorithm for the model. In Section 5, we evaluate its performance by contrast experiments. Finally, Section 6 concludes the paper.

2. The model

In this section, we discuss the hierarchy of multi-software reliability, give the definition of software utility, and formulate the architecture-based multi-software Budget-Constrained Reliability-maximization model.

2.1. Hierarchy of multi-software

In Fig. 1, the hierarchy of multi-software reliability is mainly based on the work of [7,15]. The hierarchy is a top-down approach, which starts from the top with the user's view, which has been defined as the overall reliability goal of each software S_l, denoted by $R_l (l = 1, 2, \ldots, p)$.

The user of each software bases his assessment on the functionality and attributes of the software, which are represented at the second level of the hierarchy. The user expects the software to perform a set of functions and produce the desired result. This is the users view about each software. Considering the repetitive functions among similar softwares, the second level of the hierarchy denotes f functions that the users of all softwares have enumerated, denotes them by $F_k (k = 1, 2, \ldots, f)$.

The third level of the hierarchy is the computer program written by software engineers to accommodate the functions specified by users, and denotes them by $P_i (i = 1, 2, \ldots, n)$. This level denotes the software engineers (SE) view of the software. Generally, each user-specified function would be programmed into more than one programmes.

The fourth level of hierarchy contains the independent modules of which the programmes are composed. In the formulation, it is assumed that SE adheres strictly to the concept of structured programming, which has become an inevitable programming approach for medium and large systems, which in turn, are prime candidates for using a reliability allocation model. As same as [7] and [15], in this paper, it is assumed that the modules are independent units which themselves may have submodules, but each submodule belongs to only one module. The independent modules are denoted by $M_j (j = 1, 2, \ldots, m)$. Hierarchical structure is stopped at the level of the independent modules.

On the whole, the hierarchy can well link the user's view about reliability to the software manager's and programmer's view of the software.

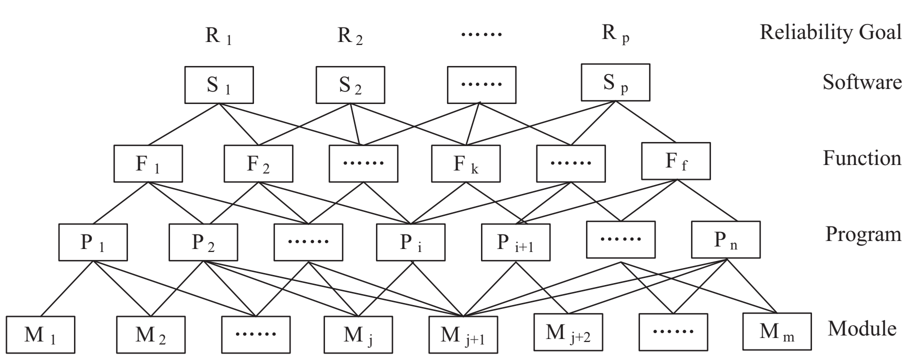

Fig. 1. Hierarchy of multi-software reliability.

Please cite this article as: F. Yue, et al., Multi-software reliability allocation in multimedia systems with budget constraints using Dempster–Shafer theory and improved differential evolution, Neurocomputing (2015), http://dx.doi.org/10.1016/j.neucom.2014.09.103