



Software requirement optimization using a multiobjective swarm intelligence evolutionary algorithm



José M. Chaves-González*, Miguel A. Pérez-Toledano, Amparo Navasa

Computer Science Department, University of Extremadura, Cáceres, Spain

ARTICLE INFO

Article history:

Received 24 July 2014

Received in revised form 12 February 2015

Accepted 14 March 2015

Available online 24 March 2015

Keywords:

Next release problem

Multiobjective evolutionary algorithm

Software requirement selection

Search-based software engineering

Swarm intelligence

Artificial bee colony

ABSTRACT

The selection of the new requirements which should be included in the development of the release of a software product is an important issue for software companies. This problem is known in the literature as the Next Release Problem (NRP). It is an NP-hard problem which simultaneously addresses two apparently contradictory objectives: the total cost of including the selected requirements in the next release of the software package, and the overall satisfaction of a set of customers who have different opinions about the priorities which should be given to the requirements, and also have different levels of importance within the company. Moreover, in the case of managing real instances of the problem, the proposed solutions have to satisfy certain interaction constraints which arise among some requirements. In this paper, the NRP is formulated as a multiobjective optimization problem with two objectives (cost and satisfaction) and three constraints (types of interactions). A multiobjective swarm intelligence meta-heuristic is proposed to solve two real instances generated from data provided by experts. Analysis of the results showed that the proposed algorithm can efficiently generate high quality solutions. These were evaluated by comparing them with different proposals (in terms of multiobjective metrics). The results generated by the present approach surpass those generated in other relevant work in the literature (e.g. our technique can obtain a HV of over 60% for the most complex dataset managed, while the other approaches published cannot obtain an HV of more than 40% for the same dataset).

© 2015 Elsevier B.V. All rights reserved.

1. Introduction

The number and difficulty of the tasks to be performed by current software systems are increasing ever more rapidly. One consequence is a growth in the complexity and the extension of modern software systems, and a concomitant increase in development effort (both time and cost). Software development companies have to efficiently satisfy large sets of requirements by minimizing the production costs of software projects. In most cases, it is not possible to develop all the new features originally suggested.

Software requirement optimization is an important task in Software Engineering, and is especially relevant when managing incremental software development approaches, such as the agile group of methods.

In these methods, the software product is developed by generating releases which are produced in short iterative cycles. In each iteration, a new set of requirements is proposed, tailored to fit the clients' needs and the development costs. In this context,

the challenge consists of defining which requirements should be developed taking into consideration several complex factors (priorities given to different clients which have different levels of importance to the company, development efforts, cost restrictions, interactions between different requirements, etc.). This complex problem, called the Next Release Problem (NRP, [1] in the related literature, has no simple solution.

NRP is an NP-hard problem [14] which simultaneously manages two conflicting and independent objectives: development cost (effort) and clients' satisfaction. Thus, it cannot be managed by traditional exact optimization methods. In these kinds of cases, multiobjective evolutionary algorithms (MOEAs) are very appropriate strategies [6,7] because they take into account simultaneously several conflicting objectives without the artificial adjustments which form part of classical single-objective optimization methods. However, most of the related work in the literature takes a simplified approach by using a kind of aggregation function, and tackles the problem as if there were a single objective. Other work does not address the interactions that arise between the requirements in real NRP instances of the problem. In this present communication, we propose a novel technique

* Corresponding author.

E-mail addresses: jm@unex.es (J.M. Chaves-González), toledano@unex.es (M.A. Pérez-Toledano), amparonm@unex.es (A. Navasa).

corresponding to the Search-Based Software Engineering (SBSE) research field [16] to deal with a real multiobjective version of the NRP (MONRP).

Specifically, we propose an adapted version of the artificial bee colony algorithm (ABC, [18]), in which several multiobjective features have been included in order to obtain high-quality results for a realistic MONRP. Such swarm intelligence approaches have recently been applied to a wide range of complex optimization problems with very satisfactory results [23,24,5,21]. As will be described below, our technique provides better results for MONRP than other approaches published in the literature.

The remainder of this paper is organized as follows. Section 2 discusses related work. Section 3 describes the basic background of the problem, and the multiobjective formulation which we adopted. Section 4 presents our proposed approach: a multiobjective artificial bee colony (MOABC) algorithm for the software requirement selection problem. The experiments performed and their results are presented and analyzed in Section 5. Finally, Section 6 summarizes the conclusions and future lines of work.

2. Related work

The requirement selection problem consists of selecting a certain number of requirements that will be developed for the next release of a specific software package such that those requirements minimize the development costs of the project and maximize the clients' satisfaction. The problem involves two conflicting objectives which have to be considered on equal terms. In the literature, Karlsson [20] proposed two kinds of method for selecting and prioritizing software requirements: an Analytical Hierarchy Process (AHP) and Quality Function Deployment (QFD). In AHP the requirements are classified by a pairwise cost-value, and in QFD they are prioritized on an ordinal scale. However, neither kind of method supports requirement interdependencies, and they have to perform a very large number of comparisons for projects with large sets of requirements.

The requirement selection problem was originally formulated in a single-objective form in the Search-Based Software Engineering (SBSE) field by Bagnall et al. [1]. SBSE is the research field in which search-based optimization algorithms are proposed and tested to tackle problems in Software Engineering [16]. The problem formulated by Bagnall et al. [1] was solved by applying different metaheuristic algorithms. All the proposals were single-objective evolutionary algorithms which combined the objectives by using an aggregation function. The same is the case with the works of Greer and Ruhe [15], and Baker et al. [2] which also adopted a single objective formulation for different evolutionary algorithms. None of these works considered the interactions which arise among requirements. Moreover, the single-objective formulation has the drawback of performing a biased search in the search space of solutions because the objectives have to be artificially aggregated in some way (for example, with a weighted sum or a weighted product).

More recently, the NRP was formulated as a multiobjective optimization problem (MOOP). Zhang et al. [36] proposed the first multiobjective formulation of the original NRP (MONRP). This version was based on Pareto dominance [6]. The method tackles each objective separately (without any combination function), thus allowing the algorithm to explore non-dominated solutions (the solutions of greater quality). The work of Finkelstein et al. [13,12] also used multiobjective optimization for the analysis of trade-offs among multiple clients with potentially conflicting requirement priorities, but the authors did not consider the interactions that arise among requirements. The same is the case with the work of Durillo et al. [11], Jiang et al. [17], and Charan

Kumari et al. [4], in which different multiobjective evolutionary algorithms are proposed for solving NRP, but without addressing dependencies among requirements. In Durillo et al. [11], the authors used the well-known algorithms PAES (Pareto Archived Evolution Strategy, [22]), NSGA-II (fast Non-dominated Sorting Genetic Algorithm, [8]), and MOCell (MultiObjective Cellular genetic algorithm, [25]). Jiang et al. [17] solved NRP by using an Ant Colony Optimization (ACO) algorithm [10]. Finally, Charan Kumari et al. [4] proposed the use of a hybrid differential evolution strategy [26].

The only two studies in which, to the best of our knowledge, the MONRP was tackled considering the requirement interactions are those of Sagrado et al. [28] and Souza et al. [34]. They both propose the use of Ant Colony Optimization [10] to solve the problem, but only the work of Sagrado et al. [28] published the datasets used, so that, in Section 5, we shall compare our results with those of that study. In this respect, it is worth mentioning that many of the related studies do not make all the information about the datasets used public (probably for commercial reasons), so it is not possible to make any numerical comparison with their results.

Here, we present a multiobjective search-based approach based on the artificial bee colony (ABC) algorithm [18]. We have adapted the algorithm to work with an MONRP formulation in which different types of requirement interactions and effort constraints are considered. Our technique searches for high quality sets of solutions that balance the clients' priorities and the cost limitations while keeping the requirement interactions. Recent publications have shown the applicability of this approach to different domains. Thus, in Chaves-Gonzalez et al. [5], MOABC is applied to the generation of DNA sequences for reliable DNA computing. In Li et al. [23], the algorithm is used in machine learning to optimize boiler efficiency. [27] propose the use of MOABC to solve the routing problem in optical networks. Silva-Maximiano et al. [30] apply the algorithm to solving the frequency assignment problem. In [29], Phylogenetic Inference was solved with MOABC. [32] successfully solves the minimum spanning tree problem with an adapted version of the artificial bee colony, and [19] use ABC to solve constrained optimization problems. In those studies, the results obtained with the artificial bee colony algorithm are compared with those results obtained from other approaches (NSGA-II, SPEA2, PSO, DE, etc.). We shall here apply MOABC to another research field, and we shall show in Section 5 that the results given by our technique are better than those obtained by other published multiobjective approaches.

3. The multiobjective next release problem

This section explains the MONRP for the selection of software requirements. As mentioned above, the NRP problem was originally formulated by Zhang et al. [36], but here we shall update the formulation in order to handle real instances of the problem in which different types of interactions occur among the requirements to be managed. But first we shall introduce some multiobjective concepts needed for a clearer understanding of what follows.

3.1. Multiobjective background

Two of the most important concepts in multiobjective optimization (MOO) are Pareto dominance and Pareto front. In MOO, a problem does not have a unique optimal solution, but a Pareto front of solutions [6]. The Pareto front is a vector of decision variables which satisfy the problem constraints and optimize the objective functions being considered. Thus, the Pareto front contains a set of Pareto solutions which are not dominated by any

Download English Version:

<https://daneshyari.com/en/article/402305>

Download Persian Version:

<https://daneshyari.com/article/402305>

[Daneshyari.com](https://daneshyari.com)