Engineering Applications of Artificial Intelligence 43 (2015) 89–101

Contents lists available at ScienceDirect

Engineering Applications of Artificial Intelligence

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

Teaching learning based optimization with Pareto tournament for the multiobjective software requirements selection

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

Computer Science Department, University of Extremadura, Cáceres, Spain

ARTICLE INFO

Article history:
Received 2 October 2014
Received in revised form
1 April 2015
Accepted 7 April 2015
Available online 29 April 2015

Keywords:
Software requirements selection
Multi-objective evolutionary algorithm
Teaching learning based optimization
Search Based Software Engineering
Next Release Problem
Swarm intelligence

ABSTRACT

Software requirements selection is a problem which consists of choosing the set of new requirements which will be included in the next release of a software package. This NP-hard problem is an important issue involving several contradictory objectives which have to be tackled by software companies when developing new releases of software packages. Software projects have to stick to a budget, but they also have to satisfy the highest number of customer requirements. Furthermore, when managing real instances of the problem, the requirements tackled suffer interactions and other restrictions which make the problem even harder. In this paper, a novel multi-objective teaching learning based optimization (TLBO) algorithm has been successfully applied to several instances of the problem. For doing this, the software requirements selection problem has been formulated as a multiobjective optimization problem with two objectives: the total software development cost and the overall customer's satisfaction. In addition, three interaction constraints have been also managed. In this context, the original TLBO algorithm has been adapted to solve real instances of the problem generated from data provided by experts. Numerical experiments with case studies on software requirements selection have been carried out in order to prove the effectiveness of the multiobjective proposal. In fact, the obtained results show that the developed algorithm performs better than other relevant algorithms previously published in the literature.

1. Introduction

The complexity and extension of modern software systems have been increased in the last decade. In addition, software products have to be usually developed in limited periods of time and with severe cost restrictions. Thus, software development companies have to satisfy in an efficient way large sets of requirements by minimizing the production efforts (in time and cost). In fact, in most of cases it is not possible to develop all the new features suggested by the clients when the new release of a software product has to be produced. Software requirements optimization is an important task in Software Engineering, and especially relevant within the incremental approaches of software development, e.g. agile methodologies. In these kinds of methodologies, the software product is developed by generating releases which have to be produced in short iterative cycles and a new set of requirements, tailored to fit the needs of the clients and the development costs, is proposed in each iteration. In this context, the challenge of Software Engineering consists of defining which requirements should be developed by considering several complex factors (different clients' priorities with different importance, development efforts, cost restrictions, interactions between different requirements, etc.). There is not a simple solution to this complex problem, which is also called in the related literature the Next Release Problem, NRP (Bagnall et al., 2001).

The NRP is an NP-hard problem (Garey and Johnson, 1990) which simultaneously manages two independent and conflicting objectives which have to be simultaneously optimized: the development effort (cost), and the clients' satisfaction. Thus, the problem cannot be managed by traditional exact optimization methods. In this case, multi-objective evolutionary algorithms (MOEAs) are the most appropriate strategies (Coello et al., 2007; Deb, 2001) because MOEAs tackle simultaneously several conflicting objectives without the artificial adjustments included in classical single-objective optimization methods. However, most of related works in the bibliography are simplified by using an aggregation function and they manage the problem as a single objective version of the problem. Furthermore, there are others works that do not tackle the interactions produced between the requirements in real NRP instances of the problem.

In this paper, a novel technique within the Search-Based Software Engineering (SBSE) research field (Harman et al., 2012) has been proposed to deal with a real multiobjective version of the NRP (MONRP). Specifically, in this paper we introduce an adapted

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

http://dx.doi.org/10.1016/j.engappai.2015.04.002
0952-1976/

Acronyms

ACO Ant Colony Optimization
$\varnothing$ Empty set
AHP Analytical Hierarchy Process
$C=\{c_1, c_2,\ldots, c_m\}$ Set of m clients or customers
cl_i Client i
CWPopulation Crowding distance of the individuals in the population
Δ Spread
DE Differential evolution
d_f, d_l Euclidean distance from the first and the last solution in the Pareto front, respectively
d_i Euclidean distance between two consecutive solutions
$E=\{e_1, e_2,\ldots, e_n\}$ Set of n costs associated to the n requirements
$E(X)$, X_{Cost} Overall effort (cost) for X
GRASP Greedy Randomize Adaptive Search Procedure
HV Hypervolume
L_C Cost threshold constraint
meanInd Mean individual
$Mean_{Req}$ Number of requirements in the mean individual
MOCell Multi-objective Cellular genetic algorithm
MOEA Multi-objective evolutionary algorithm
MONRP Multi-objective NRP
MOO Multi-objective optimization
MOOP Multi-objective optimization problem
MO-TLBO Multi-objective TLBO
N Total number of solutions in the Pareto
nd Non-dominated
NDS Non-dominated solutions
NDS_archive Non-dominated solution archive
NRP Next Release Problem
NSGA-II Fast Non-dominated Sorting Genetic Algorithm
$obj1_{max}$, $obj2_{max}$ Highest values for the objectives 1 and 2
$obj1_{min}$, $obj2_{min}$ Minimal values for the objectives 1 and 2
P Population
P_Size Size of the population
PAES Pareto Archived Evolution Strategy
QFD Quality Function Deployment
$R=\{r_1, r_2,\ldots, r_n\}$ Set of n requirements
$r[0,1]$ Random number between 0 and 1
$r_i \otimes r_j$ Exclusion interaction
$r_i \oplus r_j$ Combination interaction
$r_i \Rightarrow r_j$ Implication interaction
r_i, r_j Requirements i and j
$S(X)$, $X_{Satisfaction}$ Overall satisfaction for X
$S=\{s_1, s_2,\ldots, s_n\}$ Set of global satisfaction
SBSE Search-Based Software Engineering
SPEA-2 Strength Pareto Evolutionary Algorithm 2
Std. dev. Standard deviation
$Sum_{Req}(r)$ Addition of the requirements in r
T_{factor} Teaching factor
TLBO Teaching learning based optimization
υ, $X_{ReqsNumber}$ Total number of requirements in X
v_{ij} Importance that a requirement r_j has for a particular client c_i
$W=\{w_1, w_2,\ldots, w_m\}$ Set of clients' weights
X1CW, *X2CW* Crowding distance values for $X1$ and $X2$
X1Rank, *X2Rank* Dominance values for the individuals $X1$ and $X2$
X_{new}, X_a, X_b, $X_{new''}$, P_i, *auxInd* Individuals of the population
xTeacher High-quality individual of the population

version of the teaching learning-based optimization (TLBO) method, a very recent swarm intelligence evolutionary algorithm (Rao et al., 2012), which was adapted to obtain high-quality results of the multiobjective NRP (MONRP). We will name our proposal MO-TLBO, because some multi-objective features of well known MOEAs were wisely included to the original version of the algorithm. In addition, in order to test the accuracy of MO-TLBO, we have compared it with the multi-objective standard NSGA-II (Fast Non-dominated Sorting Genetic Algorithm) proposed by Deb et al. (2002), and other approaches proposed in other works published in the literature. As will be shown in this paper, our proposal provides high quality results, surpassing the results previously published in the literature for several instances of the problem.

The rest of the paper has been organized as follows: Section 2 discusses related work. Section 3 summarizes the basic background on the problem and the multiobjective formulation which has been proposed. Next section presents our proposal: a multi-objective teaching learning based optimization (MO-TLBO) algorithm for the software requirements selection problem. The experiments performed and the results obtained are presented and analyzed in Section 5. Finally, Section 6 summarizes the conclusions of the paper.

2. Related work

Requirements optimization is an NP-hard problem (Garey and Johnson, 1990) which consists of selecting a set of requirements that will be developed for the next release of a software product. The problem evaluates two conflicting objectives, and both objectives have to be equally considered. In the literature, Karlsson (1996) proposed two kinds of methods for selecting and prioritizing software requirements: Analytical Hierarchy Process (AHP) and Quality Function Deployment (QFD). In QFD, requirements are prioritized in an ordinal scale, and in AHP the requirements are classified by a pair cost-value. However, both kinds of methods do not support requirements interdependencies, which are real needs nowadays, and they need to perform huge numbers of comparisons when the project scale is increased.

The requirements selection problem was firstly formulated as a single-objective problem in the Search-Based Software Engineering (SBSE) field by Bagnall et al. (2001). SBSE is the research field in which search-based optimization algorithms are proposed to tackle problems in Software Engineering (Harman et al., 2012). The original problem proposed by Bagnall et al. (2001) has been solved with different metaheuristics along the last years. However, most of the approaches published are single-objective evolutionary algorithms which combine the objectives by using an aggregation function (Baker et al., 2006; Greer and Ruhe, 2004). In all cases, those works did not consider the interactions produced among the requirements. Moreover, single objective formulation has the inconvenient of making a biased search of the solution space, because the objectives have to be artificially aggregated in some way, for example with a weighted sum of the objectives.

The NRP has been recently formulated as a multi-objective optimization problem (MOOP). Zhang et al. (2007) proposed the first multi-objective formulation for the original NRP (MONRP). This formulation tackles each objective separately, without any combination function. This feature allows the algorithm to explore non-dominated solutions (the solutions of more quality) for the problem. The works by Finkelstein et al. (2008, 2009) also include the use of multi-objective optimization for the analysis of trade-offs among multiple clients with potentially conflicting requirements priorities,