International Journal of Software Innovation
Volume 11 • Issue 1

A Method for Feature Subset Selection in Software Product Lines

Nahid Hajizadeh, Shiraz University of Technology, Iran
Peyman Jahanbazi, Shiraz University of Technology, Iran
Reza Akbari, Shiraz University of Technology, Iran*

ABSTRACT

Software product line (SPL) represents methods, tools, and techniques for creating a group of related software systems. Each product is a combination of multiple features. So, the task of production can be mapped to a feature subset selection problem, which is an NP-hard problem. This issue is very significant when the number of features in a software product line is huge. This chapter is aimed to address the feature subset selection in software product lines. Furthermore, the authors aim at studying the performance of a proposed multi-objective method in solving this NP-hard problem. Here, a multi-objective method (MOBAFS) is presented for feature selection in SPLs. The MOBAFS is a an optimization algorithm, which is inspired by the foraging behavior of honeybees. This technique is evaluated on five large-scale real-world software product lines in the range of 1,244 to 6,888 features. The proposed method is compared with the SATIBEA. According to the results of three solution quality indicators and two diversity metrics, the proposed method, in most cases, surpasses the other algorithm.

KEYWORDS

Bee algorithm, feature selection, Multi-objective optimization, Search Based Software Engineering, Software Product Lines

1. INTRODUCTION

Nowadays few software products are produced individually. Most organizations tend to develop families of similar software. These products share some common elements, which is named software asset. A software asset is a description of a solution or knowledge that application engineers use to develop or modify products in a software product line (Withey, 1996). By applying the reusability concept, shared software assets can be reused, instead of developing from scratch. The process of software asset reusability is important in the SPL. SPLs are software systems that share a common regulated set of features, which include architecture, design, documents, test cases, and other assets. Also, a standard definition of a software feature has been established by IEEE[1]:" *a distinguishing*

DOI: 10.4018/ijsi.315654 *Corresponding Author

characteristic of a software item (for example, performance, portability, or functionality) " (ANSI/IEEE, 1983). Features are employed to develop products. In other words, each product is a combination of some features, and these features are selected based on the attributes that are assigned to each feature.

The SPL development process is divided into two phases: domain engineering and application engineering (Pohl et al., 2005). Domain engineering is the process of defining and realizing the commonality and variability of the SPL. It refers to the core assets and application engineering process for developing particular applications by employing the variability of the SPL (Pohl et al., 2005). Application engineering is the product generation using the core asset achieved by domain engineering.

A feature model or feature diagram is a compact display of all products in terms of features in a software product line. Indeed, a feature model shows the principal features of a product's family in the domain and the relationships between them (Kang et al., 1990). A feature model is a compressed demonstration of all the products of SPL in respect of "features". A feature model has a tree structure, which defines the relationships between features in a hierarchical pattern. In Figure 1, a feature model for a mobile phone is illustrated. In a feature model, there are two types of relationships between features: 1) the relationship between a parent feature and its children's features, and 2) cross-tree constraints.

A feature model can be expressed using a Boolean expression, as depicted in Figure 2. Software products are created by merging the selected features from a feature model and considering all the constraints. Here the issue is to select and extract a set of features from a feature model and the goal is selecting the features set in a way that not only covers restrictions but also be optimized in terms of the objective functions. Feature models consist of hundreds and thousands of features on an industrial scale so it is practically impossible to use exact algorithms to derive products in a reasonable time. In other words, how to combine a set of features to make a product, optimally, is an NP-hard problem, as White et al. indicated, and is known as a feature subset selection problem in SPLs (White et al., 2008). Therefore, the need for a metaheuristic algorithm for solving such problems is felt. Hence, in this paper, a new method based on the bee algorithm is presented. The method is used to get an optimal solution to the problem in an acceptable time.

Figure 1. A sample feature model for a mobile phone product line (Benavides et al., 2010)

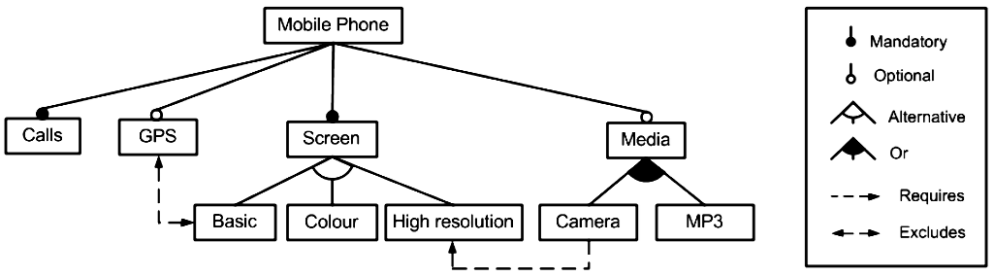

Figure 2. Feature model of mobile phone SPL in a Boolean expression format (Sayyad et al., 2013)

$$
\begin{aligned}
FM = \; & (\text{Mobile Phone} \leftrightarrow \text{Calls}) \\
& \wedge (\text{Mobile Phone} \leftrightarrow \text{Screen}) \\
& \wedge (\text{GPS} \rightarrow \text{Mobile Phone}) \\
& \wedge (\text{Media} \rightarrow \text{Mobile Phone}) \\
& \wedge (\text{Screen} \leftrightarrow \text{XOR (Basic, Color, High resolution)}) \\
& \wedge (\text{Media} \leftrightarrow \text{Camera} \vee \text{MP3}) \\
& \wedge (\text{Camera} \rightarrow \text{High resolution}) \\
& \wedge \neg(\text{GPS} \wedge \text{Basic})
\end{aligned}
$$