European Journal of Operational Research 234 (2014) 308–318

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Interfaces with Other Disciplines

Skill-based framework for optimal software project selection and resource allocation

Fadi A. Zaraket, Majd Olleik, Ali A. Yassine *

Faculty of Engineering and Architecture, American University of Beirut, Beirut, Lebanon

ARTICLE INFO

Article history:
Received 8 November 2012
Accepted 24 September 2013
Available online 9 October 2013

Keywords:
Resource allocation
Project selection
Software development
Software skills
Meta-heuristic

ABSTRACT

This paper presents a conceptual framework and a mathematical formulation for software resource allocation and project selection at the level of software skills. First, we introduce a skill-based framework that considers universities, software companies, and potential projects of a country. Based on this framework, we formulate a linear integer program PMax which determines the selection of projects and the allocation of human resources that maximize profit for a certain company. We show that PMax is NP-complete. Therefore, we devise a meta-heuristic, called *Tabu Select and Greedily Allocate* (TSGA), to overcome the computational complexities. When compared to PMax running on CPLEX, TSGA performs 15 times faster with an accuracy of 98% on small to large size problems where CPLEX converges. On larger problems where CPLEX does not return an answer, TSGA computes a feasible solution in the order of minutes.

For demonstration, the proposed skill-based framework and the corresponding mathematical model are applied to Lebanon by performing two surveys on the Lebanese software industry and academia. The case study shows that the proposed framework and mathematical model can be used in practice to improve project selection and resource allocation decisions in software companies.

© 2013 Elsevier B.V. All rights reserved.

1. Introduction

The project management literature generously addressed *resource allocation* in general (Brucker, Drexl, Mohring, Neumann, & Pesch, 1999; Herroelen & Leus, 2005; Herroelen, Reyck, & Demeulemeester, 1998; Oezdamar & Ulusoy, 1995) and project portfolio selection (Archer & Ghasemzadeh, 1999; Meade, 2002) as two separate problems. However, there is great room for suggesting models that solve the two interdependent problems simultaneously. In particular, the authors in Gutjahr, Katzensteiner, Reiter, Stummer, and Denk (2010), Yoshimuraa, Fujimia, Izuia, and Nishiwakia (2006) present models to solve the two problems one after the other assuming that the profitability of a chosen portfolio of projects is totally independent from the resources allocated on each one of them. We argue that in reality, the cost of the development of a project highly depends on the human workforce that works on it (Acuna, Juristo, & Moreno, 2006).

Additionally, *software* specific studies concerning project selection and resource allocation are still scarce (Otero, Centeno, Ruiz-Torres, & Otero, 2009). The problem of resource allocation in *software project development* is a unique challenge due to specific characteristics of software projects and software developers (Kan, 1994). The work of Acuna et al. (2006) finds that human resource allocation on software projects is generally left to the judgment of experts such as software team leaders and project managers. Although judgments are educated guesses that work in practice, however, managers lack mathematical tools to develop and assess project schedules and associated human resource allocation (Padberg, 2001). In particular, Plekhanova (1999) and Otero et al. (2009) observe that project managers map each software task to one skill when allocating human resources to tasks. Otero suggests that tasks should be mapped to several skills (Otero et al., 2009). Ngo-The maps a task into a set of skills and optimally allocates resources in release planning (Ngo-The & Ruhe, 2009).

In this paper, we propose a *skill-based framework* (SBF) that considers software projects, software companies and software related academia at the level of basic skills. That is, skills are the common thread that cuts across these three domains and are therefore at the center of our proposed framework. Educational processes, represented by *universities*, generate skills and form software development human resources that can be categorized into *talent classes*. Each talent class: $S \mapsto L$ is characterized as a map from skills S to a strength level $L = \{none, weak, average, good, excellent\}$. We characterize *software projects* by collective strength levels of required skills. The required skills and their associated levels are based on expert estimations where experts are project managers and

* Corresponding author. Tel.: +961 1 350 000.
E-mail addresses: fz11@aub.edu.lb (F.A. Zaraket), mbo01@aub.edu.lb (M. Olleik), ay11@aub.edu.lb (A.A. Yassine).

0377-2217/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.ejor.2013.09.035

software team leaders. We denote by *potential projects* those projects that can be developed in a given country. The process of developing software *requires* skills. Companies in a given country select from a set of potential projects. Each company then allocates its human resources in the form of capita per talent class to develop the selected projects. The company does the selection and allocation with maximum profit as a target. Companies in turn enrich their human resources via (1) strengthening existing skills and (2) producing skills that are not covered in current curricula. We assume that skills produced during development work will eventually make it into curricula due to the interaction between the industry and the academia. We refine our skill set S to include the skills produced by the industry S_I and assume *none* as the strength level for fresh graduates in such skills.

Based on the above framework, we build an integer linear program to optimally select a portfolio of projects and allocate resources (i.e. talents) to them such that the allocated talents satisfy the skills required by the selected projects. PMax differs from previous project selection and resource allocation methods (Gutjahr et al., 2010; Yoshimuraa et al., 2006) in that it treats the resource allocation and project selection as a single problem and presents an integer linear program to solve it. PMax also provides a more realistic cost estimation formula as it considers a company to pay all its talents regardless of whether a talent is allocated or not. Additionally, PMax differs from existing models (Gutjahr et al., 2010; Otero et al., 2009; Yoshimuraa et al., 2006) in that it introduces the concept of critical skills. A critical skill j for a project p is a skill with a minimum level of expertise $\mathbf{\Theta}[p][j]$, where at least one allocated talent must possess to satisfaction a strength in $j \geqslant \mathbf{\Theta}[p][j]$, while other skills can be satisfied by strength levels of several talents.

In this paper, we make several contributions to the software management and operations research literature.

- We present SBF, a skill-based framework, to formalize the relationship between the software academia, the software industry of a given country and the potential projects at the level of skills. We formulate project selection and resource allocation as a mathematical program, PMax, which allows for critical skills and estimates cost more accurately.
- We introduce TSGA, a Tabu-based meta-heuristic, to overcome the computational complexity of PMax since PMax is shown to be NP-complete. We compare the performance of TSGA to CPLEX. TSGA performs 15 times faster than CPLEX and reaches an optimal solution 64% of the time. On average, the profit value obtained from TSGA is 98% of the optimal profit obtained by CPLEX.
- We conduct surveys covering the Lebanese software industry and academia to demonstrate how SBF can be implemented.

2. Literature review

The paper draws upon various streams of research form project management, analyses of skills and competencies required for software product development, and software estimation models and techniques. From the project management literature, we only focus on the literature that combines project portfolio selection, activity scheduling, and resource allocation. Although relevant, the voluminous literature on resource constrained project scheduling problem, including multi-project and multi-mode versions, will be ignored. Additionally, since our mathematical model is built on a new skill-based framework, we also review literature describing various skills and competencies required in a software development environment. Finally, since our proposed model requires the estimation of various input parameters relating to cost and duration of development activities, we also discuss these various software estimation techniques in this section.

2.1. Resource allocation

Gutjahr et al. (2010) present a mixed integer non-linear programming model for project selection and resource allocation while focusing on increasing the competences of the staff through experience. They decomposed the problem by applying a meta heuristic for project selection and then a greedy priority based heuristic for project scheduling and staffing. The main problem resides in the assumption that the cost of development of a certain project is assumed to be given independently of the resources that the model allocates on it. PMax differs by incorporating the cost of the allocated resources in the calculation of the project development cost.

Yoshimuraa et al. (2006) tackle the problems of project selection and resource allocation. They start by selecting the portfolio of projects that maximizes profit. Then they allocate a project leader for each project to end up with allocating the other human resources. In this paper, we consider that project selection and resource allocation are two interdependent problems that should be solved together.

Otero et al. (2009) presented a method that associates a set of required skills with each software task for the completion of the task. The method assigns available human resources to complete the required tasks. Otero's work addresses the situation when the available resources fall short of covering the required skills and minimizes the learning time based on rhetorical relations between missing and available skills. PMax differs in that it maximizes profit, extends skills to projects instead of tasks, and considers project selection concurrently with resource allocation.

Xiao et al. (2009) consider the time and cost optimization problem in project scheduling and present a near optimal genetic algorithm. Ngo-The and Ruhe (2009) consider the release planning problem in software development. They present an optimal allocation of resources that maximizes the value gained from the released features. Their solution does not isolate software release planning from resource allocation across several releases to solve the problem globally. They first compute an optimal solution for a relaxed version of the problem, which they use with a genetic algorithm to compute a near optimal solution for the original problem.

Finally, Barreto et al. present a project manager with utility functions to form a team that fits desired needs using constraint satisfaction approach. The desired needs could optimize several aspects such as expense, performance, or size (Barreto, de O. Barros, & Werner, 2008).

2.2. Cost estimation

There are two main approaches to software development effort estimation: judgment-based methods using group consensus techniques, and model-based methods using formal mathematical models (Boehm, Abts, & Chulani, 2000). Expert judgment techniques involve consulting with a group of software cost estimation experts to use their past experiences and arrive at an estimate (or to a consensus) for the cost and duration of the proposed project (Jorgensen, 2005). Formal models, on the other hand, are designed to provide some mathematical equations to perform effort estimation. These mathematical models could be based on rules-of-thumb, historical data, and analogies, and use inputs such as lines of code, number of functions to perform, and other cost drivers such as language, design methodology, skill-levels, risk assessments. The algorithmic methods have been largely studied and there are a lot of models that have been developed, such as

Download English Version:

https://daneshyari.com/en/article/479886

Download Persian Version:

https://daneshyari.com/article/479886

Daneshyari.com