Int. J. Production Economics 170 (2015) 34–44

Contents lists available at ScienceDirect

Int. J. Production Economics

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

Quantifying the risk of project delays with a genetic algorithm

Jeremy Pfeifer [a], Kash Barker [a,*], Jose E. Ramirez-Marquez [b,c], Nazanin Morshedlou [a]

[a] School of Industrial and Systems Engineering, University of Oklahoma, United States
[b] School of Systems and Enterprises, Stevens Institute of Technology, United States
[c] Tec de Monterrey, School of Science and Engineering, Zapopan, Guadalajara, Mexico

ARTICLE INFO

Article history:
Received 23 July 2013
Accepted 1 September 2015
Available online 10 September 2015

Keywords:
Project network
Risk analysis
Genetic algorithm

ABSTRACT

Of interest in project management is the (i) quantification of the risk associated with project performance and the (ii) identification of the project tasks that contribute most to that risk. Risk in this work addresses delays in project completion. The tasks and precendences are represented with nodes and links, respectively, in a project network whose tasks (i) have stochastic completion times that (ii) are subject to disruptions. An optimization problem is developed to maximize project delay subject to particular stochastic task disruptions, and a genetic algorithm is introduce to identify the critical tasks which lead to the maximum risk of project delay. A small project of 40 tasks and large project of 800 tasks are analyzed. Primary conclusions are (i) that critical tasks need not necessarily be on the critical path if they are subject to considerable uncertainty, and (ii) that project complexity (network topology) matters more in the performance of the algorithm than the number of tasks (network size). In fact, the genetic algorithm solution works well for large-scale projects whose schedules cannot be resolved with conventional techniques. Focus is given to the performance of the algorithm for this project risk context.

1. Introduction and motivation

Managing and making decisions in large-scale projects, which can be viewed as networks of tasks and precedences, in uncertain environments is a complex undertaking. Underlying the management of such projects is the risk of failure from several perspectives: time, cost, and operational performance. And often the failure of these projects, when large-scale in nature, can have significant economic and social consequences.

In project management, the different tasks leading to the completion of a project are typically organized according to their precedence constraints, and generally a schedule minimizes the overall completion time based on the time and resources needed to perform each task (Blanchard, 2008). The project scheduling problem (Ke and Liu, 2005) addresses the scheduling of the different tasks of a project, where each task has a given completion time and is subject to several precedence constraints. An optimization problem can be devised to determine a schedule that minimizes total project completion time, for example. A resulting schedule provides the order of each task to be performed, as well as which tasks can be done in parallel. This schedule can be used to identify the critical path, or the path composed of the tasks that cannot be subject to any disruption without incurring delays in the overall project. Non-critical tasks, however, can be delayed up to a certain point without consequences to the overall completion time of the project.

Often, project scheduling problems assume point estimates for task attributes, but in real projects, each task can be subject to many disruptions (e.g., to unavailability of raw materials or delays in labor contracting). As such, the attributes for each task are subject to uncertainty: task completion times, among other quantitative descriptors of task and project performance, are more logically described by random variables. Therefore, the critical tasks in this case are not only the tasks belonging to the critical path, but they can also include tasks for which completion time is (highly) variable. The attribute of interest in this work is task delay, and ultimately project completion delays.

There exists a need to quantify the risk of project failure, as a disruption of one task may impact the further downstream tasks of the project. To describe project risk, we adopt the triplet concept of Kaplan and Garrick (1981), who define risk as the combination of scenario, likelihood, and consequence. We describe *scenario* generally as a disruption to a project task or set of tasks, the *likelihood* of which is described by their completion time random variable. The resulting *consequence* would be a delay in the completion of the project, potentially a project failure depending on the significance of the delay. The collection of these three attributes describes *project risk*, primarily described with the distribution of delay in the project (as this delay distribution

* Corresponding author.
E-mail address: kashbarker@ou.edu (K. Barker).

http://dx.doi.org/10.1016/j.ijpe.2015.09.007
0925-5273/

inherently represents the combination of likelihood and consequence for a given disruptive scenario).

While most work in the area of project scheduling deals with the development of an optimization formulation for finding an optimal schedule and with exploring solution algorithms to solve such problems, we address project risk: analyzing the tasks and combinations thereof that, when disrupted, result in the largest project delay. Useful for a project manager, a prioritized list of tasks could be identified to allocate more resources to prevent disruptions from occurring. We provide a novel means to model task disruption with (i) a new project network formulation that addresses uncertainty that (ii) lends itself to the computationally efficient implementation of a genetic algorithm to find the tasks most impactful on project completion. We provide results for small and large projects generated from a set of topological indicators, gathering insights for each. The remainder of the paper is as follows: Section 2 of this paper provides some background on project networks and genetic algorithms. Section 3 develops the optimization problem and genetic algorithm solution, while Section 4 provides some numerical results for an illustrative example. Conclusions follow in Section 5.

2. Background and literature review

This section describes several areas of the literature related to project management, including (i) formulations of and solution techniques for project scheduling problems, (ii) work on modeling project network risk, and (iii) generating example project networks.

2.1. Project networks

Several techniques have been developed since the 1950s to address planning and controlling large projects, including the Critical Path Method (CPM), the Program Evaluation and Review Technique (PERT), and the Probabilistic Network Evaluation Technique (PNET) (Jun-yan, 2012). These initial techniques enabled the identification of minimum duration schedules for projects under the existence of precedence constraints but assuming that the resources required to complete the projects were unconstrained.

The general class of project network scheduling problems that sufficiently represent the complexity of today's large-scale project networks is the Resource-Constrained Project Scheduling Problem (RCPSP). RCPSPs typically minimize total project duration subject to the more realistic constraints on the required set of resources, in addition to precedence constraints (Herroelen, 1972). The RCPSP is an NP-hard problem. Many heuristic algorithms have been applied to the RCPSP, and given our interest in genetic algorithms, we list a few here (Wang et al., 2005; Li et al., 2007). Valls et al. (2008) develop a hybrid genetic algorithm to the RCPSP, implementing some changes to crossover and local improvement operators. Montoya-Torres et al. (2010) propose a genetic algorithm based on an object-oriented representation of the RCPSP. The use of genetic algorithms has extended to multi-model RCPSP problems, where tasks can be completed in multiple ways, or modes (Mori and Tseng, 1997; Afshar-Nadjafi et al., 2013). None of the above, however, addresses the uncertainty in the completion time of activities.

To control large scale stochastic networks under uncertainty, Greenberg and Golenko-Ginzburg (2010) propose a new method based on existing control techniques by Golenko-Ginzburg and Gonik (1997). Instead of using a control model to describe the entire project network, a control model is used only at certain strategic points of the network. However, these control models are only suited for small or medium-sized networks. Thus, the method consists of using a smaller network, equivalent to the former by aggregating activities. Moreover, Ke and Liu (2005) studied the resolution of the problem with stochastic activity duration times and developed a genetic algorithm to minimize the cost of the project, showing that this type of algorithm is well adapted to solve this problem.

The literature above generally solves for an optimal project schedule with either deterministic or stochastic assumptions. Several solution techniques have been used to solve various variations on constrained project scheduling problems, including genetic algorithms (Hegazy and Petzold, 2003; Chen and Weng, 2009; Proon and Jin, 2011; Liu et al., 2014). Our work differs in two main ways: (i) genetic algorithm are popular in solving NP-hard problems, though no work considers the stochastic nature of large project networks, and (ii) we begin with a project network and determine the tasks that lead to the most risk in project completion (unlike Kilic et al. (2008), who developed a bi-objective genetic algorithm to study the cost of different levels of preventive measures associated with each activity in the project scheduling problem).

2.2. Risk interdependence

In complex projects, the impacts of risk are not measured only by direct consequences. Indeed, delays can also trigger one or more additional delays, and the consequences can be propagated along potential chain reactions. Fang et al. (2012) introduce a method to identify and quantify the risks interactions by means of network theory, building a risk network and identifying the individual links between each risk, as well as the domains in which the risks are classified. Then, indicators define the importance of each risk, according to its likelihood to trigger other disruptions in further activities.

Instead of modeling activities with random durations, Dawood (1998) focuses on the risk factor leading to delays rather than on the activity times themselves. That is, the likely risks are identified, along with their inherent probability, and their impact on different activities is modeled. Thus, a single occurring risk factor can influence more than one activity, and each activity can be subject to one or more risk factor, each with different consequences.

2.3. Project network characteristics

New algorithms and tools for project management lead to a need for comparing their performance, thus requiring a benchmark project network example. New algorithms and tools can be tested either against real world projects or realistic artificially generated projects. Artificially generated projects are best suited for the benchmarking purpose, because the generated instances can be tailored to present relevant characteristics instead of being merely an individual case.

Patterson (1984) proposed a series of problem instances designed to benchmark the proposed solutions for the RCPSP, which the author collected from various sources. Alternatively, many network generators have been designed in order to generate project networks which are realistic according to the characteristics of a project. Kolisch et al. (1995) describe a means to generate networks for the RCPSP according to different characteristics, including the topology of the network and the availability of resources. The resulting generator, ProGen, led to the creation of a Project Scheduling Problem library (Kolisch and Sprecher, 1997). Agrawal et al. (1996) propose a method and software as well, to generate project networks according to a given complexity index and resource utilization. Tavares (1999) propose a set of indicators describing the morphology of a project network. These indicators, including size and topology characteristics, can be used to verify