Computers & Operations Research 63 (2015) 1–6

Contents lists available at ScienceDirect

Computers & Operations Research

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

# An adaptive crashing policy for stochastic time-cost tradeoff problems

Changmuk Kang [a], Byung-Cheon Choi [b,*]

[a] Department of Industrial and Information Systems Engineering, Soongsil University, 369 Sangdo-ro, Dongjak-Gu, Seoul 156-743, South Korea
[b] Department of Business Administration, Chungnam National University, 79 Daehangno, Yuseong-gu, Daejeon 305-704, South Korea

ARTICLE INFO

Available online 1 May 2015

*Keywords:*
Stochastic project scheduling
Dynamic programming
Adaptive policy

ABSTRACT

We consider a stochastic time-cost tradeoff problem that determines how much to crash activities with the purpose of minimizing the expected summation of crashing and tardiness costs. We propose a threshold policy that makes a crashing decision contingent on a project's current status; crashing an activity to compensate delayed starting time from a predetermined threshold. First, we present a dynamic programming (DP) formulation to find the threshold values, and prove that the resulting threshold policy is optimal for a serial-graph project. Since the above optimality does not hold generally, we develop a DP-based procedure to construct a threshold policy for arbitrary-graph projects.

We show through the computational experiments that our threshold policy is rapidly constructed by this procedure, and its cost is not very far from the theoretical lower bound.

© 2015 Elsevier Ltd. All rights reserved.

## 1. Introduction

We consider the following stochastic time-cost tradeoff problem (STCTP) in project scheduling. A project consists of activities, each of which has a stochastic distribution. Each activity can be constrained to be finished by a given due date; otherwise a certain penalty is charged that is proportional to the tardiness. The baseline schedule is to start an activity as soon as its preceding activities are completed and to *do nothing* with its duration. This can result in missing due dates and incurring tardiness penalties. To avoid such costs, a project manager can expedite an activity's completion by *crashing* the duration. Since the crashing also involves a cost, however, it should be optimally planned to trade off with the reduced time. The decision also needs to consider a finite capacity of the crashing time.

While time-cost tradeoff problems with deterministic activity durations (in short, DTCTP) have been thoroughly studied in the literature, studies on the STCTP are relatively rare [16]. An earlier stream of the literature tries to solve STCTPs with a critical path method (CPM) under uncertainty [27]. Their solutions are *static*, since a crashing schedule is fixed and never changed upon a project's progress (a detailed review of static solutions is presented in Section 2). In contrast, recent studies suggest *adaptive* policy that specifies how much to crash contingent on a current status. Since crashing decisions are sequentially made as the project progresses, the adaptive policy provides the project manager with a realistic course of action, and generally yields better results than the static policy. In spite of the salient advantages, the study on the adaptive policy is still rare.

To fill this gap, this paper suggests an adaptive policy to minimize the expected summation of the crashing and tardiness costs. Under the resulting policy, each activity has a predefined threshold value for its starting time. If a preceding activity finishes after this threshold time and thus delays the starting time of a succeeding activity, the succeeding activity is crashed to compensate the delayed time. Thus, the threshold value can be considered as a *reserved starting time* or an internal milestone from the operational perspective. We present a dynamic programming (DP) formulation, through which a threshold policy is constructed, and prove that this policy is optimal for a serial-graph project. For an arbitrary-graph project, we develop an algorithm that integrates multiple DP formulations for individual paths and finds a unique threshold value for each activity. Although the resulting threshold values cannot guarantee its optimality, we verify that its performance is much better than previously suggested policies, as presented in the computational experiments in Section 5.

The paper is organized as follows. Section 2 reviews the related literature, primarily on STCTPs. Section 3 defines our problem formally. Section 4 develops the DP-based algorithm to determine a unique threshold value for each activity based on some dominance rules, and shows optimality of the resulting threshold policy for series-graph projects. Section 5 discusses the performance of our threshold policy for an arbitrary-graph project through the numerical and computational analysis. Section 6 concludes this paper.

* Corresponding author.
*E-mail address:* polytime@cnu.ac.kr (B.-C. Choi).

http://dx.doi.org/10.1016/j.cor.2015.04.021
0305-0548/© 2015 Elsevier Ltd. All rights reserved.

## 2. Literature review

As mentioned above, the DTCTP has been thoroughly studied in the literature (an extensive review on the deterministic version can be found in [7, chap. 8]). Where the crashing cost function is linear, the optimal crashing schedule can be found using linear programming (LP); more efficient algorithms also exist [11,18]. These algorithms have been extended to convex [17,19], concave [9], and discrete [22] function forms. Especially for DTCTPs with discrete cost functions, for which finding optimal crashing schedule is proven to be strongly NP-hard [6], many heuristic methods exist [26].

Meanwhile, studies on STCTPs have been conducted relatively recently [16]. Existing solutions to STCTPs can be classified as *static* or *adaptive* approaches (corresponding to the proactive and reactive approaches of Herroelen and Leus [16], respectively). Static approaches are to construct a fixed crashing schedule under uncertainty of activity duration. To tackle the uncertainty, Monte Carlo (MC) simulation, first proposed by Van Slyke [25], has been widely utilized for evaluating schedules. A series of studies combines this MC simulation with well-known heuristic methods, such as greedy [3,14,4], branch and bound [15], and electromagnetism [24] heuristics, to determine the optimal schedule. Another stream of studies adopts probability models to avoid computational burden of simulation. Mitchell and Klastorin [20] and Mokhtari et al. [21] analytically derive makespan distributions of paths and formulate crashing schedules. Azaron et al. [1] model crashing as a control variable in a Markov network and find the optimal control at each time period that minimizes the multiple objectives of direct cost, expected completion, and variance of completion.

In contrast, studies on the adaptive approach are not as comprehensive as those on static approaches. This approach defines a policy that determines how much to crash according to current status of a project and realized durations, rather than a fixed crashing schedule. We review the related literature in detail to clarify contribution of this paper.

Adaptive approaches can be further distinguished by whether a crashing policy requires information on durations for all activities at once at the beginning of a project, or it uses only current status which is sequentially realized as a project progresses. As a policy with all-at-once information, Zhu et al. [28] formulate a two-stage model; the first stage determines target completion times, and the second stage derives the optimal crashing schedule for meeting the target times after all activity durations are realized. Cohen et al. [5] and Goh and Hall [13] utilize a robust counterpart model developed by Ben-Tal and Nemirovski [2] for identifying the crashing schedule that minimizes the worst-case costs and conditional value at risk (CVaR), respectively. The robust counterpart model defines decision rules, which specifies a crashing schedule by a linear combination of realized durations.

If a crashing decision should be made sequentially, without information on future activities, we need another policy that relies only on a project's current status and updates the crashing schedule as it progresses. To the best of our knowledge, no study has been conducted on such policies except Tereso et al. [23] and Godinho and Branco [12]. Tereso et al. [23] apply a DP approach to construct a policy that prescribes crashing schedules for all conceivable states of activity completion. This policy, however, defines actions for only the longest path, conditional on a fixed crashing schedule for other activities. Godinho and Branco [12] suggest a threshold policy. On the grounds that an activity needs more crashing the later its starting time, the authors define starting time thresholds and corresponding crashing levels. The authors obtain optimal threshold values by straightforwardly searching possible threshold values and evaluating them with MC simulation. While they used an electromagnetism heuristic (EMH) method to reduce the search space, it requires substantial burden of computation for practical size projects.

The threshold policy we introduce in this study is similar to Godinho and Branco [12]. This study also defines threshold starting time for each activity to make crashing decisions. While Godinho and Branco [12] adopted this policy without theoretical proof of its optimality, we proved, using DP formulation, that it is optimal for series-graph projects using DP formulation. (The optimality, however, does not hold generally for arbitrary graphs.) We also develop an efficient algorithm to construct a threshold policy for arbitrary graphs, which takes significantly less computation time than the EMH method.

## 3. Problem definition

Our STCTP is represented by a directed activity-on-node graph $G=(N,A)$, where $N=\{0,1,\ldots,n+1\}$ is the set of the activities and $A$ is the set of precedence relations. Relation $(j,k)\in A$ means that activity $j$ should be completed before activity $k$ is started. Nodes 0 and $n+1$ are dummy activities meaning start and termination of a project, respectively. Their durations are set to 0. All activities are numbered to make succeeding activities that have larger index numbers than preceding ones. Non-dummy activity $j$ has a stochastic duration $D_j\in\mathbb{R}_0^+$, which is supported by a random distribution. We call a realized value of $D_j$ as the nominal duration of activity $j$, denoted $d_j$. If activity $j$ is crashed by $x_j\in\mathbb{R}_0^+$, its actual duration is calculated as $d_j-x_j$. In addition, the maximum amount of crashing is bounded by $b_j$. Starting time of $j$ is denoted by $z_j\in\mathbb{R}_0^+$ where $z_0=0$. As an activity is started as soon as all the preceding activities are completed, starting time of an activity is equal to the latest completion time of all its preceding activities. Therefore, $z_k$ is realized as follows:

$$z_k=\max\{z_j+d_j-x_j\,|\,(j,k)\in A\}. \tag{1}$$

The trade-off between time and cost is modeled by two cost elements: crashing and tardiness costs. The crashing cost is $c_jx_j$ where $c_j$ is a unit crashing cost of activity $j$. The tardiness cost of $j$, denoted by $L_j$, is defined as a function of its starting time $z_j$. Since starting time is completion time of preceding activities, tardy completion of $j$ can be penalized by $L_k(z_k)$ where $(j,k)\in A$. In this regard, $L_{n+1}(z_{n+1})$ is the tardiness cost of the project completion. We assume that function $L_j(z_j)$ is convex and non-decreasing, where $\lim_{|z_j|\to\infty}L_j(z_j)=\infty$. The objective is to minimize, by optimally making crashing decisions, the expected summation of the crashing and tardiness cost:

$$\min E\left\{\sum_{j=1}^{n+1}\left(c_jx_j+L_j(z_j)\right).\right\} \tag{2}$$

We assume that this decision can be made activity-by-activity as the project progresses. Then we can adaptively decide how much to crash according to current status of the project. The decision process for activity $j$ is as follows: (1) available starting time $z_j$ is observed, (2) how much to crash $x_j$ is determined, and (3) nominal duration $d_j$ is realized. The same process is repeated for the following activities.

## 4. Construction of threshold policy

In this section, we present a DP formulation for making aforementioned crashing decisions, based on which the threshold policy is constructed. We prove the optimality of this policy for a series-graph project. When this DP formulation is applied to individual in an arbitrary-graph project, however, different threshold policies can be constructed. Thus, we introduce some dominance rules to combine those path-wise threshold values.

Download English Version:

https://daneshyari.com/en/article/472949

Download Persian Version:

https://daneshyari.com/article/472949

Daneshyari.com