European Journal of Operational Research 237 (2014) 448–453

Contents lists available at ScienceDirect

European Journal of Operational Research

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

Discrete Optimization

On the complexity of project scheduling to minimize exposed time

Edieal Pinker [a], Joseph Szmerekovsky [b,*], Vera Tilson [c]

[a] School of Management, Yale University, New Haven, CT 06520, United States
[b] Department of Management and Marketing, College of Business, North Dakota State University, Fargo, ND 58108, United States
[c] Simon School of Business, University of Rochester, Rochester, NY 14627, United States

ARTICLE INFO

Article history:
Received 28 November 2012
Accepted 4 February 2014
Available online 11 February 2014

Keywords:
Complexity theory
Project scheduling
Secret project
NP-complete
Polynomial algorithm

ABSTRACT

We consider project scheduling where the project manager's objective is to minimize the time from when an adversary discovers the project until the completion of the project. We analyze the complexity of the problem identifying both polynomially solvable and NP-hard versions of the problem. The complexity of the problem is seen to be dependent on the nature of renewable resource constraints, precedence constraints, and the ability to crash activities in the project.

1. Introduction

Pinker, Szmerekovsky, and Tilson (2013) (here after PST2013) introduced problem STEALTH where a project manager schedules a project to minimize the exposed time, i.e. the time from when an adversary initiates a response to the project until project completion. Theorem 1 in PST2013 establishes that the version of STEALTH they study is strongly NP-hard even for a simplified case. In this paper we expand the problem of STEALTH to include the potential for a single renewable resource and provide a detailed analysis of the complexity of PST2013 including both polynomially solvable and NP-hard cases.

Problem STEALTH is motivated by the competitive environment in which most projects are managed. Project scheduling in a competitive environment is considered in a small number of papers in the operations literature. Averbakh and Lebedev (2005) examine the problem of scheduling activities when two firms compete to perform the same project, and the payoff depends on the difference in activity completion time relative to the competitor. In (Averbakh, 2010) efficient algorithms for finding Nash equilibrium activity sequences are derived when two competing firms work on the same set of activities but only the first firm to complete an activity receives a reward or in the case of simultaneous completion either a dominant firm receives the reward or the reward is shared equally. Brown, Carlyle, Royset, and Wood (2005) discuss the problem of project interdiction. They model interaction between two adversaries as a leader–follower game. The leader seeks to maximally delay a project managed by the follower. The leader selects which project activities are interfered with subject to leader resource constraints. The follower, aware of which activities will be interdicted, decides how to "crash" the schedule subject to his own budget constraints. Brown et al. (2005) show that the leader does not always choose to interdict activities on the critical path. The authors also provide a detailed computational complexity analysis for the problem. In a follow on work, Brown, Carlyle, Harney, Skroch, and Wood (2009) look specifically at the interdiction of a nuclear weapons development project. Using data from (Harney, Brown, Carlyle, Skroch, & Wood, 2006) they develop and solve the interdiction model from the point of view of the leader, identifying which project tasks should be interdicted to cause maximal delay in the project completion.

As Brown et al. (2009) acknowledge, interdiction is limited by available resources such as "money, labor, and diplomatic goodwill." Interdiction is also limited by the project information available. However, a key assumption in all the above mentioned papers is that the project is scheduled *to be completed as quickly as possible* and the interdictor *can freely choose the point in time to interdict*. To our knowledge, other than PST2013 who introduced problem STEALTH, we are the first to consider the complexity of scheduling with the *exposed time* objective.

Section 2 discusses the formulation of problem STEALTH along with the relevant notation. Section 3 provides detailed results concerning the complexity of problem STEALTH. Section 4 concludes the paper.

* Corresponding author. Address: College of Business, NDSU Dept. 2420, PO Box 6050, Fargo, ND 58108-6050, United States. Tel.: +1 701 231 8128.

E-mail addresses: edieal.pinker@yale.edu (E. Pinker), Joseph.Szmerekovsky@ndsu.edu (J. Szmerekovsky), vera.tilson@simon.rochester.edu (V. Tilson).

http://dx.doi.org/10.1016/j.ejor.2014.02.013
0377-2217/

2. Problem definition and model formulation

A full discussion of STEALTH along with the MILP formulation can be found in PST2013. In this section we briefly review the problem and introduce the notation. We consider a project manager managing a project $\mathcal{P}$ with $n + 1$ activities labeled 0 through n. Activities labeled 0 and n are "dummy" activities, the former corresponding to the project's start, and the latter to the project's completion. Let s be an $(n + 1)$-dimensional vector, with s_i, representing the starting time of project activity i, and d be an $(n + 1)$-dimensional vector, with d_i, representing activity i's duration.

2.1. Task dependencies

PST2013 allows for generalized precedence relations: "finish-to-start" precedence constraints between activities, as well as the "start-to-start", "finish-to-finish", and "start-to-finish" constraints with the corresponding lead or lag times (Elmaghraby & Kamburowski, 1992). Here we limit our discussion to the classical "finish-to-start" precedence constraints that can be written as

$$s_i + d_i \leqslant s_j \tag{1}$$

for all $(i, j) \in E$ where E is a binary relation on the set of activities.

2.2. Renewable resources

We assume that every task requires the services of a single renewable resource, similar to a processor. A total of $m \in [1, \ldots, \infty]$ renewable resources can be available at any particular time. The results in this paper deal with the cases where $m \geqslant n - 1$, i.e. renewable resources do not constrain the project, and $m = 1$, which implies that activities can only be executed in series. When $m = 1$ the problem is more like scheduling precedence constrained activities on a single machine than it is like scheduling in a traditional project environment. However, the scenario of a single bottleneck resource can occur in project environments and has been studied by Kavadias and Loch (2003) and Coolen, Wei, Talla Nobibon, and Leus (2012) for new product development (NPD) organizations and research and development (R&D) projects. Note, the version of STEALTH in PST2013 did not include renewable resources, but we include them in our analysis to show their impact on the complexity of the scheduling problem.

2.3. Non-renewable resources and duration crashing

We assume the project manager has a non-renewable resource available to him, with a budget b. Following PST2013 we assume that the duration of activities in the project can be shortened (crashed) – from a nominal value of $\bar{\delta}_i$ to duration $d_i \geqslant \underline{\delta}_i$ for activity i. Since 0 and n are "dummy" activities, $\bar{\delta}_0 = \underline{\delta}_0 = \bar{\delta}_n = \underline{\delta}_n = 0$. The reduction in duration of non-dummy activities requires expenditure of the non-renewable resource. We define γ_i as the amount of the non-renewable resource required to reduce the duration of activity i by one time unit.

2.4. Detection and deception

When an activity is initiated, we assume that it is observed by the adversary. Very generally, if s is the activity initiation schedule for project $\mathcal{P}$ the project will be detected by some time t determined by s. To represent this relationship each activity i is assigned a non-negative detection weight w_i. The detection weight model combines two phenomena: the ease with which the activity can be detected, and the probability (from the point of view of an adversary) that the activity is related to a "nefarious" objective. We assume that the adversary will act when his belief that the project is nefarious is strong enough. To represent the idea that combinations of activities indicate project intention we define $F(t, s, w)$ to be the likelihood the adversary will act to counter the project at time t given the activity start times s and activity detection weights w. In this paper we use the form introduced in PST2013:

$$F(t, s, w) = \begin{cases} 1, & \text{if } \sum_{i,\text{s.t.} s_i \leqslant t} w_i > \upsilon, \text{ where } 0 \leqslant \upsilon \leqslant \sum_{i=1}^{n} w_i \\ 0, & \text{otherwise} \end{cases} \tag{2}$$

With this functional form, there is some threshold of suspicion that once reached will cause the adversary to act. The threshold υ is project specific and partly reflects the costs to the adversary of interfering with the project.

In PST2013 the project manager is also allowed to invest in deception as follows: similar to project crashing, the project manager can reduce the detection weight for activity i from a nominal value of $\bar{\omega}_i$ to $w_i \geqslant \underline{\omega}_i$ through expenditure of the same non-renewable resource used for crashing. The expenditure associated with deception is proportional to the amount by which the detection weight is reduced. We define α_i as the amount of the non-renewable resource required per unit reduction in the detection weight.

2.5. Scheduling objective

The project manager's problem is as follows: given a project completion due date τ, non-renewable resource constraint, renewable resource constraint, and a detection threshold, the objective is to complete the project by the due date while minimizing the exposed time, i.e. if T_D is the detection time, then minimize $s_n - T_D$ by selecting the starting times s, durations d, and detection weights w for project activities. A full discussion of STEALTH along with the MILP formulation can be found in PST2013. As mentioned previously we include renewable resources in our analysis which are not considered in PST2013. For a variety of special cases of STEALTH we can derive specialized solution algorithms for this problem. In Section 3 we present the results of a computational complexity analysis of different special cases of the problem.

3. Time complexity of problem stealth

In this section we analyze the time complexity of problem STEALTH. A version of STEALTH is defined by four parameters: the number of processors, the type of precedence constraints, the type of crashing allowed, and the type of deception allowed. Table 1 indicates the parameter values for the special cases of STEALTH which we consider.

We denote the special case of STEALTH under consideration by listing the parameter values in parenthesis. For example, the version of STEALTH with a single-processor, precedence constraints, crashing allowed, and deception allowed is denoted as STEALTH($m = 1, prec, \bar{\delta} \geqslant \underline{\delta}, \bar{\omega} \geqslant \underline{\omega}$). Our complexity results are summarized in Table 2.

We now provide a brief discussion of the complexity results. A complete proof for each result can be found in Section 3.1 for cases with $m = \infty$ or Section 3.2 for cases with $m = 1$.

STEALTH($m = \infty, prec, \bar{\delta} = \underline{\delta}, \bar{\omega} = \underline{\omega}$) is very similar to basic project scheduling to satisfy precedence constraints with the only exception being that the objective is to minimize the exposed time. Indeed, Proposition 1 of PST2013 establishes that the well known late-start schedule which can be found in $O(n^2)$ time is optimal for STEALTH($m = \infty, prec, \bar{\delta} = \underline{\delta}, \bar{\omega} = \underline{\omega}$). Further, Proposition 1 in this work asserts that given any schedule for the activity start and finish times (such as the late-start schedule), a corresponding