



## Management Science

Publication details, including instructions for authors and subscription information:  
<http://pubsonline.informs.org>

### Interdiction Games on Markovian PERT Networks

Eli Gutin, Daniel Kuhn, Wolfram Wiesemann

To cite this article:

Eli Gutin, Daniel Kuhn, Wolfram Wiesemann (2015) Interdiction Games on Markovian PERT Networks. Management Science 61(5):999-1017. <http://dx.doi.org/10.1287/mnsc.2014.1973>

Full terms and conditions of use: <http://pubsonline.informs.org/page/terms-and-conditions>

This article may be used only for the purposes of research, teaching, and/or private study. Commercial use or systematic downloading (by robots or other automatic processes) is prohibited without explicit Publisher approval, unless otherwise noted. For more information, contact [permissions@informs.org](mailto:permissions@informs.org).

The Publisher does not warrant or guarantee the article's accuracy, completeness, merchantability, fitness for a particular purpose, or non-infringement. Descriptions of, or references to, products or publications, or inclusion of an advertisement in this article, neither constitutes nor implies a guarantee, endorsement, or support of claims made of that product, publication, or service.

Copyright © 2015, INFORMS

Please scroll down for article—it is on subsequent pages



INFORMS is the largest professional society in the world for professionals in the fields of operations research, management science, and analytics.

For more information on INFORMS, its publications, membership, or meetings visit <http://www.informs.org>

# Interdiction Games on Markovian PERT Networks

Eli Gutin

Operations Research Center, Massachusetts Institute of Technology, Cambridge, Massachusetts 02139, [gutin@mit.edu](mailto:gutin@mit.edu)

Daniel Kuhn

Risk Analytics and Optimization Chair, École Polytechnique Fédérale de Lausanne, 1015 Lausanne, Switzerland, [daniel.kuhn@epfl.ch](mailto:daniel.kuhn@epfl.ch)

Wolfram Wiesemann

Imperial College Business School, Imperial College London, London SW7 2AZ, United Kingdom, [ww@imperial.ac.uk](mailto:ww@imperial.ac.uk)

In a stochastic interdiction game a proliferator aims to minimize the expected duration of a nuclear weapons development project, and an interdictor endeavors to maximize the project duration by delaying some of the project tasks. We formulate static and dynamic versions of the interdictor's decision problem where the interdiction plan is either precommitted or adapts to new information revealed over time, respectively. The static model gives rise to a stochastic program, whereas the dynamic model is formalized as a multiple optimal stopping problem in continuous time and with decision-dependent information. Under a memoryless probabilistic model for the task durations, we prove that the static model reduces to a mixed-integer linear program, whereas the dynamic model reduces to a finite Markov decision process in discrete time that can be solved via efficient value iteration. We then generalize the dynamic model to account for uncertainty in the outcomes of the interdiction actions. We also discuss a crashing game where the proliferator can use limited resources to expedite tasks so as to counterbalance the interdictor's efforts. The resulting problem can be formulated as a robust Markov decision process.

**Keywords:** interdiction game; PERT network; Markov decision process; robust optimization

**History:** Received April 15, 2013; accepted March 28, 2014, by Dimitris Bertsimas, optimization. Published online in *Articles in Advance* September 23, 2014.

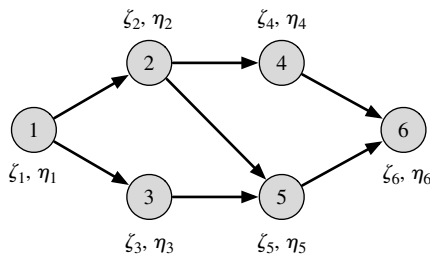
## 1. Introduction

Since its inception in 1968, a total of 190 states have joined the Treaty on the Non-Proliferation of Nuclear Weapons, whose objective is to prevent the spread of nuclear weapons, to promote nuclear disarmament, and to regulate the peaceful uses of nuclear technology for energy production. Despite its overwhelming worldwide acceptance, some nonsignatory and even some signatory states of the treaty are suspected to pursue clandestine nuclear weapons development programs Sagan (2011). The international community perceives this as a threat to global security and therefore endeavors to discourage and inhibit such initiatives by diplomatic means. However, preventive diplomacy requires tenacity and patience. To give diplomacy enough time to succeed, it has repeatedly been attempted to delay nuclear weapons development programs upon their discovery.

Devising an effective strategy to delay the buildup of a nuclear arsenal requires specific knowledge of the tasks a potential proliferator would likely perform. Harney et al. (2006) give a detailed description of a project to produce a first nuclear weapon. They identify about 100 tasks that must be completed to finish the project. Key activities include the diversion of yellowcake uranium oxide, the enrichment of

weapons-grade uranium, the design and construction of the gun device components, the acquisition of a weapons-delivery method such as ballistic missiles, etc. These tasks must obey about 200 temporal precedence relations that describe technological constraints. To lead such a project to success, a proliferator would have to expend significant time and resources. Harney et al. (2006) estimate that a proliferator with a budget of \$190 million could complete the project within six and a half years. By doubling the resource input, the project duration could be reduced to about five years.

To delay nuclear weapons programs, the international community may consider a variety of *interdiction actions*, ranging from economic sanctions and embargoes on key materials to acts of sabotage and military operations. In practice, interdiction actions must be applied with moderation because they utilize scarce resources such as capital, manpower, and political goodwill. Assuming a finite budget of interdiction resources, we thus face the following two key questions. Which tasks should we interdict in order to maximally delay the project, and what is the resulting project duration? Although the answer to the first question is crucial to inform political and military decision making, estimates of the resulting project duration are needed to predict

**Figure 1** PERT Network Consisting of  $n = 6$  Tasks

Notes. Any task  $v \in V = \{1, \dots, 6\}$  has a standard duration  $\zeta_v \geq 0$  and an interdicted duration  $\eta_v \geq \zeta_v$ . The arcs  $A = \{(1, 2), (1, 3), (2, 4), (2, 5), (3, 5), (4, 6), (5, 6)\}$  encode temporal precedences. For example, task 5 can only be initiated once tasks 2 and 3 are both completed.

the time window that is available for a diplomatic settlement with the proliferator.

Brown et al. (2009) formalize the strategic interaction between the proliferator and the interdictor as a deterministic Stackelberg game. They represent the proliferator's project as a project evaluation and review technique (PERT) network, that is, a directed acyclic graph  $G = (V, A)$  whose nodes  $V = \{1, \dots, n\}$  correspond to the project tasks and whose arcs  $A \subseteq V \times V$  encode the technological precedence relations. All precedences are of finish-start type, that is,  $(u, v) \in A$  implies that task  $v$  can only start after task  $u$  has been completed. We assume that task 1 is the unique source and task  $n$  the unique sink of the network. Task  $v$  is assumed to have a standard duration  $\zeta_v \geq 0$  and an interdicted duration  $\eta_v \geq \zeta_v$ . Figure 1 visualizes an example PERT network. The interdictor acts as the leader and chooses binary interdiction decisions  $\theta \in \Theta = \{\theta \in \{0, 1\}^n : \mathbf{1}^\top \theta \leq b_0\}$  with the goal to maximize the project's completion time. By convention,  $\theta_v = 1$  iff task  $v$  is interdicted, in which case its duration increases from  $\zeta_v$  to  $\eta_v$ . The integer  $b_0$  represents the interdiction budget, that is, the maximum number of tasks that can be delayed. The proliferator acts as the follower and chooses the task start times  $s \in \mathbb{R}_+^n$  with the goal to minimize the project duration. Thus, the interdiction game gives rise to the following deterministic bilevel program:

$$\begin{aligned} \max_{\theta \in \Theta} \min_{s \in \mathbb{R}_+^n} \quad & s_n + \zeta_n + \theta_n(\eta_n - \zeta_n) \\ \text{s.t.} \quad & s_v \geq s_u + \zeta_u + \theta_u(\eta_u - \zeta_u) \quad \forall (u, v) \in A. \end{aligned} \quad (1)$$

Formulation (1) implicitly assumes that the interdictor can observe which tasks are being processed at any point in time. Monitoring the state of the project requires effective intelligence gathering, which is facilitated by the fact that a nuclear weapons development program constitutes an industry-scale project that requires significant manpower and is therefore difficult to conceal. In addition, the proliferator may even intentionally disclose the state of the project via public announcements or through ostentatious display of

technological and military capability aimed at intimidating other nations or extorting political concessions. Throughout this paper we will stick to the assumption that the interdictor can observe which tasks are being processed.

The main thrust of this paper is the observation that the activity durations in a nuclear weapons development project are highly uncertain. Indeed, the tasks involve the operation of fragile and sensitive equipment and require tremendous amounts of expertise and knowledge that may be difficult to acquire. This necessitates numerous trial-and-error attempts with uncertain prospects of success. Difficulties in procuring critical raw materials further contribute to the uncertainty. In the presence of uncertain task durations, the estimation of the project's completion time is a challenging problem even if the interdiction plan is fixed; see Hagstrom (1988). In particular, a deterministic model using nominal task durations fails to capture the accumulation of delays across the project network induced by fluctuating task durations and therefore severely underestimates the project's completion time. Unlike deterministic project networks, which decompose into critical and noncritical paths, any path in a stochastic network may become critical with a positive probability. Hence, the interdictor cannot confine herself to delaying tasks that are (almost) critical.

In this paper we introduce "nature" as a fictitious third player that chooses random task durations. We formulate static and dynamic versions of the interdiction game, where the interdiction plan is either precommitted or may adapt to new information revealed over time, respectively. In the static case, the interdictor's decision problem gives rise to a two-stage stochastic program, whereas in the dynamic case it is formulated as a multiple optimal stopping problem in continuous time and with decision-dependent information. In either case, a dominant strategy for the proliferator is to adhere to the early-start policy, whereby each task is initiated as soon as all of its predecessors have been completed. To facilitate an algorithmic analysis, we impose a Markovian structure on the underlying PERT network, that is, we assume that the task durations are mutually independent and exponentially distributed. This assumption can later be relaxed to allow for general phase-type distributions (which are dense in the set of all distributions on the nonnegative reals) or to allow for ambiguous exponential distributions with uncertain rate parameters. Under exponentially distributed task durations, the interdictor's *static* decision problem reduces to a finite mixed-integer linear program (MILP), while the interdictor's *dynamic* decision problem can be reformulated as a finite Markov decision process (MDP) in discrete time.

To showcase the flexibility of our approach, we extend the interdictor's dynamic decision problem to

account for implementation uncertainty. In this model any interdiction attempts only succeed with a certain probability. We also discuss a crashing game where the proliferator can use limited renewable resources to expedite tasks in order to counterbalance the interdictor's efforts. The resulting multiple optimal stopping problem can be reformulated as a *robust MDP*. To gain insights into the mechanics of interdiction games, we further examine analytically tractable special cases based on symmetrical PERT networks.

In view of its practical relevance, the literature on project interdiction games is surprisingly sparse. Apart from the aforementioned contributions by Harney et al. (2006) and Brown et al. (2009), we are only aware of two papers that directly relate to our setting. In the first one, Brown et al. (2005) study the computational complexity of different classes of project interdiction games. Among others, they show that problem (1) can be solved in polynomial time via dynamic programming, but the problem becomes strongly  $\mathcal{NP}$ -hard if the proliferator can expedite project tasks. Pinker et al. (2013) consider a proliferator that manages a covert project so as to minimize the interdictor's reaction time window. The authors assume that the interdictor only acts once she is sufficiently confident about the malicious intentions of the project manager. They model the problem as a Stackelberg game in which the interdictor first selects the information threshold that triggers a response, and afterward the proliferator manages the project so as to minimize the exposure time between the triggering event and the project completion.

There is a rich literature relating to the broader topic of deterministic network interdiction games, which are Stackelberg formulations of network optimization problems such as the shortest path, maximum flow, or minimum cost flow problem. In those games, the interdictor moves first by removing or modifying some network components (i.e., nodes and/or arcs). Afterward, the operator responds with a recourse operation, which typically amounts to a rerouting decision. The resulting max-min problems can be solved by dualizing the inner optimization problem or using Benders' decomposition. Network interdiction games have important military applications, for example in orchestrating attacks on critical infrastructure or supply chains (or increasing the resilience of those networks against such attacks), as well as placing sensors to detect contamination of water systems or smuggling of illegal substances. For reviews of the network interdiction literature, we refer to Dimitrov and Morton (2012), Smith (2010), and Wood (2010). Stochastic extensions of network interdiction games are discussed by Morton (2010).

Most of the literature on network interdiction games focuses on two-stage models. Smith (2010) reviews network fortification games, which are three-stage network

interdiction games where the operator can take actions both before and after the interdictor selects a decision. Zheng and Castañón (2012a, b) discuss three-stage and multistage versions of the maximum flow interdiction problem where the interdictor acquires additional information over time. Lunday and Sherali (2010) consider a multistage and multiobjective variant of a maximum flow interdiction problem where the leader precommits to a static interdiction policy. All of these models have in common that they are considerably more difficult to solve than their two-stage counterparts, and they either require specific modeling assumptions or they are limited to small problem sizes.

To our knowledge, this is the first paper that investigates the stochastic project interdiction game. We demonstrate that stochastic interdiction schedules can significantly outperform their deterministic counterparts, and we show that dynamic policies offer further benefits over static decisions. In contrast to the existing literature, we consider truly adaptive interdiction policies that are not restricted to a small number of decision stages. The proposed methods are scalable enough to provide insights into practically relevant project networks, and we expect our results to have ramifications for other classes of network interdiction problems. Methodologically, our paper spans the often segregated fields of robust optimization, stochastic programming, and dynamic programming, and we contribute to the contemporary research on robust MDPs.

Although we present our contributions in the context of nuclear weapons projects, our findings have several other applications. For example, they can be used to delay the plot of a terrorist cell in order to gather sufficient evidence for a court case. Also, our models can help to protect development or reconstruction projects from acts of sabotage by identifying project tasks that are most vulnerable to attacks. In a nonmilitary context, our formulations may help companies to delay a competitor's new product development project in order to gather further intelligence about the project and to reposition their own product range. Because of the equivalence of project scheduling problems and shortest path problems in acyclic graphs, the proposed models can also help to generalize some of the network interdiction problems discussed earlier.

The remainder of this paper is organized as follows. Section 2 discusses *static* interdiction games where all interdiction decisions are precommitted before the project start, and §3 introduces *dynamic* interdiction games that admit adaptive interdiction policies. We show that the interdictor's decision problem reduces to an MILP in the static and to an MDP in the dynamic case. Section 4 extends the basic dynamic model to account for implementation uncertainty, crashing decisions taken by the project manager and nonexponential task durations. We close with numerical results in §5.



An extended version of the paper that contains technical background material as well as some of the proofs can be found online (Gutin et al. 2014).

**Notation.** Random objects are denoted by symbols with tildes, and their realizations are denoted by the same symbols without tildes. For any subset  $\mathcal{A} \subseteq V = \{1, \dots, n\}$  we define  $\mathbf{1}_{\mathcal{A}} \in \mathbb{R}^n$  through  $(\mathbf{1}_{\mathcal{A}})_v = 1$  if  $v \in \mathcal{A}$ ;  $= 0$  otherwise. We also use the shorthand  $\mathbf{1}$  to denote  $\mathbf{1}_V$ . For  $z \in \mathbb{R}^n$  we let  $z_+ = \max\{z, 0 \times \mathbf{1}\}$ , where “max” denotes component-wise maximization.

## 2. Static Interdiction Games

A project is represented by a directed acyclic graph  $G = (V, A)$  whose nodes  $V = \{1, \dots, n\}$  correspond to the project’s activities or tasks and whose arcs  $A \subseteq V \times V$  encode the immediate technological precedence relations among the tasks. The precedence relations induce a strict partial order “ $<$ ” on  $V$ , where  $u < v$  exactly if  $u \neq v$  and there exists a directed path from  $u$  to  $v$ . Without loss of generality we assume that 1 is the unique source and  $n$  is the unique sink node of the graph, that is,  $1 < u < n$  for all  $u \in V \setminus \{1, n\}$ .

At any point in time  $t \geq 0$ , some tasks are *idle*, that is, they have not yet been initiated, some tasks are *active*, that is, they are currently being processed, and some other tasks are *completed*. The state of the project can therefore be encoded by a partition  $x = (I_x, A_x, C_x)$  of  $V$ , where  $I_x, A_x$  and  $C_x$  represent the sets of all idle, active, and completed tasks, respectively. We refer to  $x_0 = (V \setminus \{1\}, \{1\}, \emptyset)$  as the initial and to  $x_\infty = (\emptyset, \emptyset, V)$  as the terminal state. Moreover, we define  $X$  as the set of all admissible states  $x$  that satisfy the relation

$$v \in A_x \cup C_x \iff u \in C_x \quad \forall u < v. \quad (2)$$

The “if” implication in (2) can be viewed as a causality condition that requires all predecessor activities of active or completed tasks to be completed. The “only if” implication reflects our assumption that the project is managed under the *early start policy*, whereby task  $v$  is initiated (and thus becomes nonidle) as soon as all of its predecessor activities  $u < v$  are completed.

The state of the project changes whenever an active task terminates. Let  $E_x(v) = \{u \in I_x: v < u, w \in C_x \cup \{v\} \forall w < u\}$  be the set of all emerging tasks that are activated if task  $v \in A_x$  terminates in state  $x = (I_x, A_x, C_x)$ . In that case, the project enters a new state  $y = (I_y, A_y, C_y)$  defined through  $I_y = I_x \setminus E_x(v)$ ,  $A_y = (A_x \setminus \{v\}) \cup E_x(v)$  and  $C_y = C_x \cup \{v\}$ . We let  $Y_x$  be the set of all immediate successor states of  $x$ . Note that  $Y_x = \emptyset$  iff  $x = x_\infty$ . Moreover, we denote by  $X'$  the set of all states  $x$  that can be reached from  $x_0$  in finitely many transitions. Thus,  $X'$  is the smallest subset of the state space containing  $x_0$  and satisfying  $x \in X' \implies Y_x \subseteq X'$ .

**PROPOSITION 1.**  $X' = X$ .

**PROOF.** The inclusion  $X' \subseteq X$  can be proved by induction. Indeed,  $x_0$  satisfies (2) since 1 is the only source

in  $G$ , while all  $y \in Y_x$  satisfy (2) if  $x$  does so. To prove the converse inclusion, we define  $X_y = \{x \in X: y \in Y_x\}$  as the set of all states from which  $y$  can be reached in one transition. Equivalently, we can think of  $X_y$  as the set of all states  $x$  that can be reached from  $y$  in one *inverse* transition. Note first that  $X_y = \emptyset$  for  $y = x_0$ . We will now argue that  $X_y \neq \emptyset$  for all  $y \in X \setminus \{x_0\}$ . Because  $|X|$  is finite and the number of completed states decreases by one in each inverse transition, this implies that  $x_0$  is reachable from any  $y \in X$  in a finite number of inverse transitions, and hence  $X \subseteq X'$ . Choose an arbitrary  $y \in X \setminus \{x_0\}$ . As the set  $C_y$  is finite and nonempty for all  $y \in X \setminus \{x_0\}$ , it has a maximal element  $v$  with respect to the partial order “ $<$ .” Then, define the set  $E = \{u \in V: v < u, w \in C_y \forall w < u\}$ . As state  $y$  satisfies condition (2), we have  $E \subseteq A_y \cup C_y$ . By the maximality of  $v$  in  $C_y$ , we may further conclude that  $E \subseteq A_y$ . Defining state  $x$  through  $C_x = C_y \setminus \{v\}$ ,  $A_x = (A_y \setminus E) \cup \{v\}$  and  $I_x = I_y \cup E$ , it is clear that the project migrates from  $x$  to  $y$  if task  $v \in A_x$  terminates, that is,  $y \in Y_x$ . Moreover, it can be verified that  $x$  satisfies (2), that is,  $x \in X$ . Thus,  $X_y \neq \emptyset$  for all  $y \in X \setminus \{x_0\}$ . Note that  $E$  coincides with the set of tasks  $E_x(v)$  that are activated upon termination of task  $v \in A_x$  in the newly constructed state  $x$ .  $\square$

As condition (2) is easy to verify, Proposition 1 provides a simple and efficient means to enumerate all reachable states in the algorithms developed below. As any task can be in one of three modes, the total number of states  $|X|$  is bounded above by  $3^n$ . If the geometry of the project graph implies that at most  $m$  tasks can be active simultaneously, then  $|X|$  is bounded above by  $\sum_{i=1}^m \binom{n}{i} \leq m \cdot n^m$  and is thus polynomial in  $n$  for any fixed  $m$ . Note that a state is uniquely described by its active tasks because any task that is a transitive ancestor (successor) of an active task must be completed (idle); see also (2). Thus,  $\binom{n}{i}$  estimates the number of states in which exactly  $i$  tasks are active. In the remainder of the text, we denote the uncertain duration of task  $v$  by  $\tilde{\zeta}_v$  in the nondelayed mode and by  $\tilde{\eta}_v$  in the delayed mode, respectively, and we impose the following distributional assumption.

**MARKOV PROPERTY.** The task durations  $\tilde{\zeta}_v$  and  $\tilde{\eta}_v$ ,  $v \in V$ , follow stochastically independent exponential distributions with rates  $\nu_v$  and  $\delta_v$ , respectively, where  $\nu_v > \delta_v > 0$ .

By definition, a PERT network is called Markovian if its (standard and delayed) task durations satisfy the Markov property. The requirement  $\nu_v > \delta_v$  implies that  $\tilde{\eta}_v$  first-order stochastically dominates  $\tilde{\zeta}_v$ . For ease of notation we define the  $n$ -dimensional random vectors  $\tilde{\zeta} = (\tilde{\zeta}_v)_{v \in V}$  and  $\tilde{\eta} = (\tilde{\eta}_v)_{v \in V}$ . We also introduce the  $2n$ -dimensional random vector  $\tilde{\xi} = (\tilde{\zeta}, \tilde{\eta})$ . By assumption,  $\tilde{\xi}$  is supported on  $\Xi = \mathbb{R}_+^{2n}$ . Elements of  $\Xi$  are termed *scenarios*.

In the static game under consideration, the interdictor decides here and now (that is, before the project starts)

which activities to delay. Thus, the interdictor's decisions are representable by a binary vector  $\theta = (\theta_v)_{v \in V}$ , where  $\theta_v = 1$  if task  $v$  is delayed;  $= 0$  otherwise. Below we denote by  $c_v$  the completion time of task  $v$  and by  $c = (c_v)_{v \in V}$  the corresponding completion schedule. For any given state  $x \in X$ , interdiction plan  $\theta \in \{0, 1\}^n$  and scenario  $\xi \in \Xi$ , we define  $F_x(\theta, \xi)$  as the set of all completion schedules satisfying the following constraints:

$$c_v = 0 \quad \forall v \in C_x, \quad (3a)$$

$$c_v = \max_{u \prec v} c_u + (1 - \theta_v)\zeta_v + \theta_v \eta_v \quad \forall v \in V \setminus C_x. \quad (3b)$$

Note that, because of the early start policy, the start time of task  $v$  coincides with the maximum of the completion times of its predecessor activities  $u \prec v$ . Without loss of generality, constraint (3a) sets the completion time of any completed task to zero. Constraint (3b) sets the completion time of any other task to its start time plus its duration. By the acyclicity of the project graph, constraints (3a) and (3b) have a unique solution for all  $x \in X$ . Thus,  $F_x(\theta, \xi)$  is a degenerate singleton set.

The completion schedules inherit via constraints (3a) and (3b) the uncertainty of the task durations. Thus, we formally introduce the notion of a *random time*.

**DEFINITION 1 (RANDOM TIMES).** Let  $\mathcal{T}$  be the sets of all Borel measurable functions from  $\Xi$  to  $[0, \infty]$ . We will refer to the elements of  $\mathcal{T}$  as random times.

If the project is currently in state  $x \in X$ , then its expected residual completion time  $\varphi_x$  under the interdiction plan  $\theta$  can be expressed as the optimal value of an optimization problem over random times and with a degenerate feasible set:

$$\varphi_x = \max_{c \in \mathcal{T}^n} \{\mathbb{E}(c_n(\xi)) : c(\xi) \in F_x(\theta, \xi) \quad \forall \xi \in \Xi\}.$$

The definition of  $\varphi_x$  as the optimal value of an optimization problem may seem artificial because of the degenerate feasible set. It is adopted to facilitate a symmetrical treatment of static and dynamic interdiction games. Under the Markov property, the seemingly difficult expected makespan problem has an elegant recursive solution discovered by Kulkarni and Adlakha (1986).

**PROPOSITION 2 (KULKARNI AND ADLAKHA 1986).** The expected completion time  $\varphi_x$  satisfies the Bellman equation

$$\begin{aligned} \varphi_x &= r_x(\theta) + \sum_{y \in Y_x} p_{y|x}(\theta) \varphi_y \quad \forall x \in X \setminus \{x_\infty\} \quad \text{and} \\ \varphi_{x_\infty} &= 0, \end{aligned} \quad (4)$$

where for  $x \in X \setminus \{x_\infty\}$ ,  $y \in Y_x$  and  $v \in C_y \setminus C_x$ ,

$$\begin{aligned} r_x(\theta) &= \frac{1}{\sum_{u \in A_x} (1 - \theta_u) \nu_u + \theta_u \delta_u} \quad \text{and} \\ p_{y|x}(\theta) &= \frac{(1 - \theta_v) \nu_v + \theta_v \delta_v}{\sum_{u \in A_x} (1 - \theta_u) \nu_u + \theta_u \delta_u}. \end{aligned}$$

An optimal interdiction plan can be obtained by solving the optimization problem

$$\max_{\theta \in \Theta} \max_{c \in \mathcal{T}^n} \{\mathbb{E}(c_n(\xi)) : c(\xi) \in F_{x_0}(\theta, \xi) \quad \forall \xi \in \Xi\}, \quad (\mathcal{P})$$

where  $\Theta = \{\theta \in \{0, 1\}^n : \mathbf{1}^\top \theta \leq b_0\}$ , and  $b_0 \in \{0, \dots, n\}$  denotes the *interdiction budget*, that is, the maximum number of tasks that can be delayed. Note that problem  $\mathcal{P}$  can be interpreted as a Stackelberg game that is played between the interdictor and the proliferator, and as such it is always optimized by a deterministic interdiction strategy (Fudenberg and Tirole 1991).

Proposition 2 indicates that the evaluation of the objective function in  $\mathcal{P}$  requires the solution of the Bellman equation (4), which seems computationally costly. We will now show that  $\mathcal{P}$  is in fact equivalent to an MILP, for which powerful off-the-shelf solvers are available.

**THEOREM 1.** Problem  $\mathcal{P}$  is equivalent to the MILP

$$\begin{aligned} \max \quad & \varphi_{x_0} \\ \text{s.t.} \quad & \theta \in \Theta, \quad \varphi \in \Phi, \quad \alpha \in \mathbb{R}_+^{[X] \times n}, \quad \beta \in \mathbb{R}_+^{[X] \times n} \\ & \sum_{v \in A_x} \nu_v (\varphi_x - \varphi_{y_x^v}) + (\beta_{xv} - \alpha_{xv}) \leq 1 \quad \forall x \in X \setminus \{x_\infty\} \\ & \alpha_{xv} \leq (\nu_v - \delta_v) \varphi_x, \quad \alpha_{xv} \leq M \theta_v \\ & \quad \quad \quad \forall x \in X \setminus \{x_\infty\}, v \in A_x \\ & \beta_{xv} \geq (\nu_v - \delta_v) \varphi_{y_x^v} - M(1 - \theta_v) \\ & \quad \quad \quad \forall x \in X \setminus \{x_\infty\}, v \in A_x, \end{aligned}$$

where  $\Phi = \{\varphi \in \mathbb{R}_+^{[X]} : \varphi_{x_\infty} = 0\}$  and  $M$  is a large constant, and  $y_x^v$  denotes the successor state of  $x$  that emerges if  $v \in A_x$  is the first active task that terminates.

**PROOF.** By using Proposition 2, problem  $\mathcal{P}$  can be rewritten as follows:

$$\begin{aligned} \max \quad & \varphi_{x_0} \\ \text{s.t.} \quad & \theta \in \Theta, \quad \varphi \in \Phi \\ & \varphi_x = r_x(\theta) + \sum_{y \in Y_x} p_{y|x}(\theta) \varphi_y \quad \forall x \in X \setminus \{x_\infty\}. \end{aligned} \quad (5)$$

Here, the expected completion times  $\varphi_x$  are reinterpreted as decision variables constrained by the Bellman equation (4). Note that the Bellman constraints can be relaxed to inequalities as all transition probabilities are nonnegative and  $\varphi_{x_0}$  is maximized. Substituting the formulas for  $r_x(\theta)$  and  $p_{y|x}(\theta)$  into (5) then yields

$$\begin{aligned} \max \quad & \varphi_{x_0} \\ \text{s.t.} \quad & \theta \in \Theta, \quad \varphi \in \Phi \\ & \varphi_x \leq \frac{1 + \sum_{v \in A_x} [\nu_v - (\nu_v - \delta_v) \theta_v] \varphi_{y_x^v}}{\sum_{v \in A_x} [\nu_v - (\nu_v - \delta_v) \theta_v]} \\ & \quad \quad \quad \forall x \in X \setminus \{x_\infty\}. \end{aligned} \quad (6)$$

The claim now follows if we multiply the constraints in (6) by the denominators of their right-hand sides and linearize the resulting inequalities by introducing new decision variables  $\alpha, \beta \in \mathbb{R}_+^{|X| \times n}$  that satisfy the relations<sup>1</sup>

$$\alpha_{xv} \leq (\nu_v - \delta_v) \theta_v \varphi_x \iff \alpha_{xv} \leq (\nu_v - \delta_v) \varphi_x, \quad \alpha_{xv} \leq M \theta_v,$$

$$\beta_{xv} \geq (\nu_v - \delta_v) \theta_v \varphi_{y_x^v} \iff \beta_{xv} \geq (\nu_v - \delta_v) \varphi_{y_x^v} - M(1 - \theta_v).$$

This concludes the proof.  $\square$

The MILP derived in Theorem 1 bears some similarity to a model developed for generic two-stage interdiction games (Bailey et al. 2006). The key differences are that Bailey et al. use a scenario discretization for the uncertain problem parameters, and their formulation assumes that the transition probabilities of the MDP do not depend on the interdiction decisions. Note that both the number of variables and the number of constraints in the MILP grow with the size of  $X$ , which can be very large in practice. We will revisit the solution of problem  $\mathcal{P}$  in §3.

To gain deeper insights into the mechanics of interdiction games and to understand what delays can be inflicted with static interdiction plans, it is instructive to take a closer look at some well-structured network topologies.

**DEFINITION 2 (IID, PARALLEL, AND SERIAL NETWORKS).** A Markovian PERT network is an independent and identically distributed (IID) network if all task durations are independent and identically distributed in the sense that there exist  $\nu, \delta \in \mathbb{R}_+, \nu > \delta > 0$ , with  $\nu_v = \nu$  and  $\delta_v = \delta \forall v \in V \setminus \{1, n\}$ , and the dummy tasks 1 and  $n$  have zero duration. A parallel network is an IID network, where  $A = \cup_{v \in V \setminus \{1, n\}} \{(1, v), (v, n)\}$ . Similarly, a serial network is an IID network, where  $A = \cup_{v \in V \setminus \{1\}} \{(v-1, v)\}$ .

The following proposition describes a simple bound on the maximum delay that can be inflicted on an IID network with *any* static interdiction plan.

**PROPOSITION 3.** *Any static interdiction plan can delay the expected completion time of a project on an IID network at most by a factor  $\nu/\delta > 1$ .*

**PROOF.** The delay associated with any fixed interdiction plan is bounded above by the delay that would result from interdicting *all* tasks (assuming that we could inflate the interdiction budget to  $b_0 = n$ ). Thus, the claim follows if we can prove that interdicting all tasks amounts to scaling the expected project completion time by a factor  $\nu/\delta$ . This, however, is an immediate consequence of the formula for  $r_x(\theta)$  and the Bellman equation in Proposition 2.  $\square$

<sup>1</sup> We can set  $M > \varphi_{x_0} \max\{\nu_v - \delta_v : v \in V\}$ . A simple upper bound on  $\varphi_{x_0}$  is  $\sum_{v \in V} 1/\delta_v$ .

A question of primary practical importance is whether we can inflict delays of the order of  $\nu/\delta$  also if the interdiction budget is limited. The following result provides evidence that the theoretical upper bound  $\nu/\delta$  can actually be attained asymptotically even with an arbitrarily small interdiction budget as long as the project network exhibits significant parallelism.

**PROPOSITION 4.** *Consider a parallel network of size  $n$  and set  $b_0 = \lfloor \varrho(n-2) \rfloor$  for some  $\varrho \in (0, 1)$ . As  $n$  tends to infinity, the expected project completion time is given by*

- (i)  $\log n/\nu + \mathcal{O}(1)$  if no tasks are interdicted;
- (ii)  $\log n/\delta + \mathcal{O}(1)$  if any  $b_0$  tasks are interdicted.

**PROOF.** Without interdictions, the expected project completion time amounts to

$$\mathbb{E}[\max\{\tilde{\zeta}_2, \dots, \tilde{\zeta}_{n-1}\}] = \frac{1}{\nu} \sum_{v=1}^{n-2} \frac{1}{v} = \frac{\log n}{\nu} + \mathcal{O}(1),$$

where we use a well-known formula for the expectation of the maximum of independent and identically distributed exponential random variables; see e.g., Cox and Hinkley (1974), p. 468. Thus, (i) follows. As for (ii), assume, without any loss of generality, that the first  $b_0$  parallel tasks are interdicted. The expected project duration of this interdiction plan admits the following lower bound:

$$\begin{aligned} & \mathbb{E}[\max\{\tilde{\eta}_2, \dots, \tilde{\eta}_{b_0+1}, \tilde{\zeta}_{b_0+2}, \dots, \tilde{\zeta}_{n-1}\}] \\ & \geq \mathbb{E}[\max\{\tilde{\eta}_2, \dots, \tilde{\eta}_{b_0+1}\}] = \frac{1}{\delta} \sum_{v=1}^{b_0} \frac{1}{v} = \frac{\log n}{\delta} + \mathcal{O}(1). \end{aligned} \quad (7)$$

The last equality in the above expression uses the identity  $\log b_0 = \log \varrho n + \mathcal{O}(1) = \log n + \mathcal{O}(1)$ . To estimate the expected project duration from above, we note that

$$\begin{aligned} & \mathbb{E}[f(\tilde{\xi}) = \max\{\tilde{\eta}_2, \dots, \tilde{\eta}_{b_0+1}, \tilde{\zeta}_{b_0+2}, \dots, \tilde{\zeta}_{n-1}\}] \\ & = \int_0^\infty \mathbb{P}[f(\tilde{\xi}) \geq \tau] d\tau \\ & = \int_0^{\log n/\delta} \mathbb{P}[f(\tilde{\xi}) \geq \tau] d\tau + \int_{\log n/\delta}^\infty \mathbb{P}[f(\tilde{\xi}) \geq \tau] d\tau \\ & \leq \frac{\log n}{\delta} + \int_{\log n/\delta}^\infty \left[ \sum_{v=2}^{b_0+1} \mathbb{P}(\tilde{\eta}_v \geq \tau) + \sum_{v=b_0+2}^{n-1} \mathbb{P}(\tilde{\zeta}_v \geq \tau) \right] d\tau \\ & = \frac{\log n}{\delta} + \frac{b_0}{\delta n} + \frac{n-2-b_0}{\nu n^{\nu/\delta}} = \frac{\log n}{\delta} + \mathcal{O}(1), \end{aligned}$$

where the fourth line exploits Boole's inequality, and the fifth line uses the distributional assumptions for task durations in IID networks and the definition of  $b_0$ . As both bounds share the same leading term, the claim follows.

Intuitively, for  $n \gg 1$  the maximum in (7) is attained by one of the interdicted task durations with high probability. The result then follows as the expectation



of the maximum of  $b_0$  exponential random variables with rate  $\delta$  amounts to  $\log b_0/\delta = \log n/\delta + \mathcal{O}(1)$ .  $\square$

Proposition 4 implies that the best possible delay factor  $\nu/\delta$ , which is attainable with unlimited interdiction resources, remains asymptotically attainable in parallel networks even if only a constant fraction of all tasks may be interdicted.

In contrast to parallel networks, the maximum delay factor attainable in serial networks depends on the relative interdiction budget even when the number of tasks tends to infinity.

**PROPOSITION 5.** *Consider a serial network of size  $n$  and set  $b_0 = \lfloor \rho(n-2) \rfloor$  for some  $\rho \in (0, 1)$ . As  $n$  tends to infinity, the expected project completion time is given by  $n/\nu + \mathcal{O}(1)$  if no tasks are interdicted and by  $(1-\rho)n/\nu + \rho n/\delta + \mathcal{O}(1)$  if any  $b_0$  tasks are interdicted.*

**PROOF.** The claim holds due to the linearity of expectation and because the expected value of an exponential random variable coincides with its inverse rate parameter.  $\square$

Proposition 5 implies that the delay factor attainable in a serial network is asymptotically equal to  $1 + \rho \cdot (\nu - \delta)/\delta$ . Thus, it grows with  $\rho$  and the delay factor  $\nu/\delta$  of any single task.

**REMARK 1 (RISK AVERSION).** Since the development of a nuclear weapon is a nonrecurring project, a judicious interdictor may want to use a performance criterion that penalizes downside risk, thereby encouraging positively skewed makespan distributions. This could be achieved by maximizing the expectation of a concave utility function of the project makespan. Our static model can be generalized to this setting if the utility function used is a degree- $d$  polynomial, in which case the expected utility is representable as a linear combination of the first  $d$  moments of the makespan distribution. In fact, all moments of the makespan distribution can be computed using similar recursions as in Proposition 2; see Kulkarni and Adlakha (1986), §3.1. Thus, the interdictor's decision problem can still be reformulated as an MILP by the same reasoning as in Theorem 1.

### 3. Dynamic Interdiction Games

We now relax the requirement that the interdiction decisions must be precommitted. Instead, we give the interdictor the flexibility to defer the interdiction decision for any particular task to its execution period. This additional flexibility allows the interdictor to use online information about the project's progress to inflict maximum harm to the opponent. For example, when several tasks with identically distributed durations are executed in parallel, the interdictor could wait and see which tasks drag on before deploying any

resources. There is substantial benefit in interdicting exactly those tasks that—by coincidence—happen to last longest. In this section we will thus allow the tasks to be interdicted anytime between their start and completion times.

In the following we denote by  $c_v \in \mathbb{R}_+$  the *completion time* and by  $i_v \in \mathbb{R}_+ \cup \{\infty\}$  the *interdiction time* of task  $v$ . The interdiction time represents the time at which the interdictor takes measures to delay task  $v$ , thereby changing the task duration from  $\zeta_v$  to  $\tilde{\eta}_v$ . In the remainder we will refer to  $c = (c_v)_{v \in V}$  as the *completion schedule* and to  $i = (i_v)_{v \in V}$  as the *interdiction schedule* of the project. For any given scenario  $\xi \in \Xi$ , we define  $F_0(\xi)$  as the set of all completion and interdiction schedules satisfying the following constraints:

$$c_v = \begin{cases} \max_{u < v} c_u + \zeta_v & \text{if } i_v \geq \max_{u < v} c_u + \zeta_v, \\ i_v + \eta_v & \text{else,} \end{cases} \quad \forall v \in V, \quad (8a)$$

$$i_v \geq \max_{u < v} c_u \quad \forall v \in V, \quad (8b)$$

$$|\{v \in V : i_v < \infty\}| \leq b_0. \quad (8c)$$

Note that under the early start policy the start time of task  $v$  coincides with the maximum of the completion times of its predecessor activities  $u < v$ . Constraint (8a) sets the completion time of any task to its start time plus *nondelayed* duration, if the task is *not* interdicted while active, or to its interdiction time plus *delayed* duration, otherwise. This implies that any interdiction of task  $v$  at a time  $i_v \geq \max_{u < v} c_u + \zeta_v$  has no impact on the duration of  $v$ . Moreover, constraint (8b) forces the interdiction times of all tasks to exceed their start times, respectively. One can show that this constraint is unnecessary to include in the formulation, in the sense that this constraint would automatically be satisfied at optimality even if it were removed. Nevertheless, it helps us later to contain the sizes of the state and action spaces when we reformulate the interdictor's decision problem as a Markov decision process. The resource constraint (8c) ensures that not more than  $b_0$  tasks can be interdicted in finite time, where  $b_0$  denotes the interdiction budget.

The completion and interdiction schedules are coupled to the random task durations via constraints (8a)–(8c) and must therefore be interpreted as random times in the sense of Definition 1. The interdiction schedule must further obey causality, that is, an interdiction decision taken at time  $t$  may only depend on information about the uncertain task durations that is in fact available by monitoring the project up to time  $t$ . This requirement prompts us to model the game's information structure as a filtration and to introduce the notion of stopping times.



For a given completion schedule  $c \in \mathcal{T}^n$  the information available at time  $t \geq 0$  can be represented by the  $\sigma$ -algebra  $\mathcal{F}_t(c) = \sigma(\min\{c(\xi), t\})$ , where the minimization operator truncates the completion times beyond  $t$  and ensures that only the observable portion of the completion schedule impacts the current information set. The inflow of new information over time is therefore captured by the filtration  $\mathbb{F}(c) = \{\mathcal{F}_t(c)\}_{t \geq 0}$ . We remark that  $\mathbb{F}(c)$  is indeed a filtration because  $\mathcal{F}_s(c) \subseteq \mathcal{F}_t(c)$  for all  $s \leq t$ . A richer filtration is the right-continuous extension of  $\mathbb{F}(c)$ , which is defined as  $\mathbb{F}^+(c) = \{\mathcal{F}_t^+(c)\}_{t \geq 0}$ , where  $\mathcal{F}_t^+(c) = \bigcap_{s>t} \mathcal{F}_s(c)$ . Whereas  $\mathbb{F}(c)$  provides only information about past task completions, its right-continuous extension  $\mathbb{F}^+(c)$  provides information about both past and present task completions. Moreover, under  $\mathbb{F}^+(c)$  every new piece of information has a definite first time of arrival. In light of this, it is more reasonable to model the interdictor's information structure through  $\mathbb{F}^+(c)$  instead of  $\mathbb{F}(c)$ . Causal interdiction times are thus modeled as stopping times with respect to  $\mathbb{F}^+(c)$ .

**DEFINITION 3 (STOPPING TIMES).** Let  $\mathbb{F}^+(c)$  be the right-continuous filtration induced by some  $c \in \mathcal{T}^n$ . A random time  $\tau \in \mathcal{T}$  is an  $\mathbb{F}^+(c)$ -stopping time if

$$\{\xi \in \Xi: \tau(\xi) \leq t\} \in \mathcal{F}_t^+(c) \quad \forall t \geq 0. \quad (9)$$

Condition (9) implies that at any time  $t \geq 0$  it is possible to decide whether or not  $\tau(\xi)$  exceeds  $t$  solely on the basis of the information reflected by  $\mathcal{F}_t^+(c)$ . It is easy to verify that every task completion time  $c_v$ ,  $v \in V$ , as well as every deterministic time  $t \geq 0$  is an  $\mathbb{F}^+(c)$ -stopping time. Lemma 1 in Gutin et al. (2014) presents useful equivalent characterizations of stopping times.

We are now ready to formalize the interdictor's decision problem. The goal is to find a causal interdiction schedule  $i$  and corresponding completion schedule  $c$  that maximize the expected project completion time. Each component of  $i$  must be an  $\mathbb{F}^+(c)$ -stopping time, and constraints (8a)–(8c) must hold scenario-wise for each  $\xi \in \Xi$ . Thus, the interdictor solves the following maximization problem:

$$\begin{aligned} \max \quad & \mathbb{E}(c_n(\xi)) \\ \text{s.t.} \quad & c_v \text{ is a random time} \quad \forall v \in V, \\ & i_v \text{ is an } \mathbb{F}^+(c)\text{-stopping time} \quad \forall v \in V, \\ & (c(\xi), i(\xi)) \in F_0(\xi) \quad \forall \xi \in \Xi. \end{aligned} \quad (\mathcal{P}_0)$$

Problem  $\mathcal{P}_0$  constitutes a multiple optimal stopping problem in continuous time with decision-dependent information. Indeed, the game's information structure is captured by the filtration  $\mathbb{F}^+(c)$  that depends explicitly on the completion schedule  $c$  and—through the coupling of  $c$  and  $i$  in (8a)–(8c)—implicitly on the

vector of stopping times  $i$  chosen by the interdictor. Problems of this type are usually hard to solve even approximately. Here, however, we can exploit the memoryless property of the exponentially distributed task durations to reduce  $\mathcal{P}_0$  to a finite-horizon Markov decision process in discrete time.

We will demonstrate that problem  $\mathcal{P}_0$  can be solved recursively by decomposing it into finitely many subproblems that differ from  $\mathcal{P}_0$  with regard to their initial conditions at time  $t = 0$ . To construct the subproblems, we first define the (time-dependent) state of the interdiction game similarly as in §2. Although tasks can still be idle, active, or completed, we now have to subdivide the set of active tasks into those that are *active delayed* (i.e., active and interdicted) and *active nondelayed* (i.e., active and not interdicted), respectively. Moreover, we must track the residual interdiction budget. The state of the dynamic game is therefore encoded by a tuple  $x = (b_x, I_x, N_x, D_x, C_x)$ , where  $b_x \in \{0, \dots, n\}$  denotes the residual interdiction budget, and  $\{I_x, N_x, D_x, C_x\}$  constitutes a partition of  $V$  with  $I_x, N_x, D_x$  and  $C_x$  containing all tasks that are idle, active nondelayed, active delayed, and completed in state  $x$ , respectively.

We can identify  $x_0 = (b_0, V \setminus \{1\}, \{1\}, \emptyset, \emptyset)$  as the initial state of problem  $\mathcal{P}_0$ , where  $b_0$  represents the initial interdiction budget. Any state with  $C_x = V$  is referred to as a terminal state, and the set of terminal states is denoted by  $X_\infty$ . Note that terminal states can have different interdiction budgets. In analogy to §2, we let  $X$  be the set of all admissible states  $x$  that satisfy the relations

$$v \in V \setminus I_x \iff u \in C_x \quad \forall u < v, \quad (10a)$$

$$|D_x| \leq b_0 - b_x \leq |C_x \cup D_x|. \quad (10b)$$

Condition (10a) reflects causality as well as the project manager's adherence to the early start policy. Condition (10b) requires that the interdiction budget spent to date amounts at least to the number of delayed tasks (if none of the completed tasks were interdicted) and at most to the number of delayed and completed tasks (if all of the completed tasks were interdicted).

The state of the project changes over time, and a transition from state  $x$  to a new state  $y$  occurs whenever one of the active tasks  $v \in N_x \cup D_x$  terminates or when some active nondelayed tasks  $\mathcal{A} \subseteq N_x$  are interdicted. We denote by  $Y_x$  the set of all states  $y$  that can follow state  $x$  in a single transition. The events triggering a state transition as well as the resulting successor states are listed in Table 1. For brevity, we will henceforth identify successor states  $y \in Y_x$  by their triggering events instead of their precise definitions in the columns of Table 1. Note that  $Y_x = \emptyset$  if and only if  $x$  is a terminal state. We denote by  $X'$  the set of all states  $x$  that can be reached in finitely many transitions starting from  $x_0$ . Thus,  $X'$  is the smallest subset of the state space containing  $x_0 \in X'$  and satisfying  $x \in X' \implies Y_x \subseteq X'$ .

**Table 1** Possible Successor States of  $x$

| Triggering event    | $v \in N_x$<br>terminates                 | $v \in D_x$<br>terminates    | $\mathcal{A} \subseteq N_x$<br>interdicted |
|---------------------|---|------------------------------|--|
| Interdiction budget | $b_y = b_x$                               | $b_y = b_x$                  | $b_y = b_x -  \mathcal{A} $                |
| Idle tasks          | $I_y = I_x \setminus E_x(v)$              | $I_y = I_x \setminus E_x(v)$ | $I_y = I_x$                                |
| Nondelayed tasks    | $N_y = (N_x \setminus \{v\}) \cup E_x(v)$ | $N_y = N_x \cup E_x(v)$      | $N_y = N_x \setminus \mathcal{A}$          |
| Delayed tasks       | $D_y = D_x$                               | $D_y = D_x \setminus \{v\}$  | $D_y = D_x \cup \mathcal{A}$               |
| Completed tasks     | $C_y = C_x \cup \{v\}$                    | $C_y = C_x \cup \{v\}$       | $C_y = C_x$                                |

*Notes.* Different events trigger transitions to different successor states  $y$ . The set of emerging tasks is defined as  $E_x(v) = \{u \in I_x : v < u, w \in C_x \cup \{v\} \forall w < u\}$ , where  $v \in N_x \cup D_x$ .

**PROPOSITION 6.**  $X' = X$ .

**PROOF.** The proof parallels that of Proposition 1 and is therefore omitted.  $\square$

Since any task can be in one of four modes, the total number of states  $|X|$  is bounded above by  $b_0 \cdot 4^n$ . If the geometry of the project graph implies that at most  $m$  tasks can be active simultaneously, then  $|X|$  is bounded above by  $b_0 \sum_{i=1}^m \binom{n}{i} 2^i \leq b_0 \cdot m \cdot (2n)^m$  and is thus polynomial in  $n$  for any fixed  $m$ . Note that a state is uniquely described by its budget and its active delayed and nondelayed tasks because any task that is a transitive ancestor (successor) of an active task must be completed (idle); see also (10a) and (10b). Thus,  $\binom{n}{i} 2^i$  estimates the number of states in which exactly  $i$  tasks are active (each of which can be in the delayed or nondelayed mode).

We now introduce subproblems of  $\mathcal{P}_0$  that model auxiliary interdiction games starting in some state  $x \in X$  at time  $t = 0$ . By a slight abuse of notation, we let  $c$  and  $i$  again denote the underlying completion and interdiction schedules, respectively. For any fixed state  $x \in X$  and scenario  $\xi \in \Xi$ , we define  $F_x(\xi)$  as the set of all completion and interdiction schedules satisfying the following constraints:

$$c_v = 0 \quad \forall v \in C_x; \quad (11a)$$

$$c_v = \begin{cases} \max_{u < v} c_u + \zeta_v & \text{if } i_v \geq \max_{u < v} c_u + \zeta_v, \\ i_v + \eta_v & \text{else,} \end{cases} \quad \forall v \in V \setminus C_x; \quad (11b)$$

$$i_v = 0 \quad \forall v \in D_x; \quad (11c)$$

$$i_v \geq \max_{u < v} c_u \quad \forall v \in V \setminus D_x \quad (11d)$$

$$|\{v \in N_x \cup I_x : i_v < \infty\}| \leq b_x. \quad (11e)$$

These constraints generalize the definition of  $F_0(\xi)$  in (8a)–(8c). Without any loss of generality, constraints (11a) and (11c) set the completion and interdiction times of all tasks that are already completed and interdicted at  $t = 0$  to zero, respectively. Note

that, by construction,  $F_x(\xi)$  does not depend on  $\{\zeta_v\}_{v \in C_x}$  and  $\{\eta_v\}_{v \in C_x}$ . Note also that the interdiction times of completed tasks have no physical meaning and can vary freely within  $[0, \infty]$ . They are only carried along for ease of notation and to avoid that the dimension of the interdiction schedule becomes state dependent. We remark that (11a)–(11e) reduces to (8a)–(8c) for  $x = x_0$ , that is,  $F_0(\xi) = F_{x_0}(\xi)$ .

As before, the completion and interdiction schedules must be interpreted as random times. Causality further dictates that all interdiction times must be modeled as  $\mathbb{F}^+(c)$ -stopping times, where the filtration  $\mathbb{F}^+(c)$  is defined in the usual way. Thus, if the project is in state  $x \in X$  at time  $t = 0$ , the interdictor aims to solve the following auxiliary optimization problem:

$$\begin{aligned} \varphi_x = \max \quad & \mathbb{E}(c_n(\tilde{\xi})) \\ \text{s.t.} \quad & c_v \text{ is a random time} \quad \forall v \in V, \\ & i_v \text{ is an } \mathbb{F}^+(c)\text{-stopping time} \quad \forall v \in V, \\ & (c(\xi), i(\xi)) \in F_x(\xi) \quad \forall \xi \in \Xi. \end{aligned} \quad (\mathcal{P}_x)$$

The optimal value  $\varphi_x$  quantifies the expected residual makespan under an optimal interdiction strategy. We note that  $\mathcal{P}_x$  is feasible for all  $x \in X$  as it is always possible to pursue a static interdiction strategy that satisfies  $i_v(\xi) = 0$  for  $v \in D_x$ ;  $i_v = \infty$  for  $v \in V \setminus D_x$ , meaning that no further interdictions are carried out. The results of §2 thus imply that  $0 \leq \varphi_x < \infty$ . We also remark that, by construction, problem  $\mathcal{P}_x$  reduces to  $\mathcal{P}_0$  for  $x = x_0$ .

We now demonstrate that the computation of  $\varphi_x$  for all  $x \in X$  can be reduced to the solution of a finite MDP in discrete time. This reduction leverages a structural characterization of the optimal interdiction schedules in  $\mathcal{P}_x$ .

**DEFINITION 4 (FIRST COMPLETION AND INTERDICTION TIMES).** For  $(c, i)$  feasible in  $\mathcal{P}_x$ ,  $x \in X$ , the first completion time  $\underline{c} \in \mathcal{T}$  and the first interdiction time  $\underline{i} \in \mathcal{T}$  are defined through  $\underline{c}(\xi) = \inf\{c_v(\xi) : v \in N_x \cup D_x\}$  and  $\underline{i}(\xi) = \inf\{i_v(\xi) : v \in N_x\}$ .

Note that  $\underline{c}(\xi)$  represents the time when the first active task terminates in scenario  $\xi$ . We can omit idle tasks in the definition of  $\underline{c}$ . In fact, each idle task has an active predecessor under the early start policy, which implies via (11b) that  $c_v(\xi) \geq \underline{c}(\xi)$  for all  $\xi \in \Xi$ ,  $v \in I_x$ . Similarly,  $\underline{i}(\xi)$  denotes the time when the first active nondelayed task is interdicted in scenario  $\xi$ . The first interdictions prior to  $\underline{c}(\xi)$  will later correspond to the actions in state  $x$  of the MDP that is to be constructed. Idle tasks can thus be disregarded in the definition of  $\underline{i}$  by virtue of (11d). We remark that the first completion time  $\underline{c}$  and the first interdiction time  $\underline{i}$  are  $\mathbb{F}^+(c)$ -stopping times whenever  $(c, i)$  is feasible in  $\mathcal{P}_x$  for  $x \in X$ . This is an immediate consequence of the observation that minima of finitely many stopping times are stopping times.

**PROPOSITION 7.** Let  $(c, i)$  be feasible in  $\mathcal{P}_x$  for some  $x \in X$ . Then, for each  $v \in N_x$  there is  $\alpha_v \in [0, \infty]$  such that  $i_v(\xi) = \alpha_v$  for all  $\xi \in \Xi$  with  $i_v(\xi) < c(\xi)$ . Moreover,  $\alpha = \inf_{v \in N_x} \alpha_v$  satisfies  $i(\xi) = \alpha$  for all  $\xi \in \Xi$  with  $i(\xi) < c(\xi)$ , and the cardinality of the set  $\mathcal{A} = \{v \in N_x: \alpha_v = \alpha, \alpha_v < \infty\}$  does not exceed  $b_x$ .

**PROOF.** See Gutin et al. (2014).  $\square$

Proposition 7 provides a partial structural characterization of implementable interdiction schedules. It implies that the first interdictions executed before the first completion time  $c(\xi)$  occur at some deterministic time  $\alpha$ . If  $\alpha$  is finite, the set  $\mathcal{A}$  comprises all tasks to be interdicted at time  $\alpha$ . Otherwise,  $\mathcal{A}$  is empty. In other words, whenever the project enters a new state  $x$ , the interdictor selects a set of active tasks  $\mathcal{A}$  and a target interdiction time  $\alpha$  with  $\alpha = \infty$  iff  $\mathcal{A} = \emptyset$ . If no task that is active in state  $x$  ends earlier, all tasks in  $\mathcal{A}$  are interdicted at time  $\alpha$ .

**DEFINITION 5 (ACTIONS).** We refer to  $A_x = \{(\alpha, \mathcal{A}) \in [0, \infty] \times 2^{N_x}: |\mathcal{A}| \leq b_x\}$  as the action space in state  $x \in X$ . Elements of  $A_x$  are referred to as actions.

Recall that  $Y_x$  denotes the set of all possible immediate successor states of  $x \in X$  that can emerge if any active task terminates or if any subset of the active nondelayed tasks is interdicted. However, not all states  $y \in Y_x$  are reachable in one transition from  $x \in X$  if a particular action  $a \in A_x$  is implemented.

**DEFINITION 6 (SUCCESSOR STATES).** For any  $x \in X$  and  $a \in A_x$  we denote by  $Y_{xa} \subseteq Y_x$  the set of all possible immediate successor states of  $x$  under action  $a$ .

If  $\alpha = 0$  for action  $a = (\alpha, \mathcal{A}) \in A_x$ , then  $Y_{xa}$  contains only the state that emerges from  $x$  by interdicting the tasks in  $\mathcal{A}$ . If  $\alpha > 0$ , then  $Y_{xa}$  contains also all those states that emerge if an active task terminates before  $\alpha$ ; see Table 1. By construction, we have  $Y_x = \bigcup_{a \in A_x} Y_{xa}$ . Next, we characterize the (random) time of the first state transition under a given action.

**DEFINITION 7 (FIRST STATE TRANSITION TIME).** Let  $\tau_{y|xa} \in \mathcal{T}$  be the time of the event triggering a transition from  $x \in X$  to  $y \in Y_{xa}$  under action  $a = (\alpha, \mathcal{A}) \in A_x$ ,

$$\tau_{y|xa}(\xi) = \begin{cases} \zeta_v & \text{if } y \text{ is triggered by the completion} \\ & \text{of some } v \in N_x, \\ \eta_v & \text{if } y \text{ is triggered by the completion} \\ & \text{of some } v \in D_x, \\ \alpha & \text{if } y \text{ is triggered by the interdiction} \\ & \text{of all } v \in \mathcal{A}. \end{cases}$$

The first state transition time  $\tau_{xa} \in \mathcal{T}$  is then defined via  $\tau_{xa}(\xi) = \inf_{y \in Y_{xa}} \tau_{y|xa}(\xi)$ .

In scenario  $\xi$  the interdiction game remains in state  $x \in X$  until time  $\tau_{xa}(\xi)$  and then moves to some state  $y \in Y_{xa}$ . We remark that  $\tau_{xa}$  admits an alternative characterization as  $\tau_{xa}(\xi) = \min\{c(\xi), i(\xi)\}$  and that  $\tau_{xa}$  is an  $\mathbb{F}^+(c)$ -stopping time. For the sake of concise terminology we further define random successor states.

**DEFINITION 8 (RANDOM SUCCESSOR STATES).** The random successor state  $y_{xa}$  of  $x \in X$  under action  $a \in A_x$  is a Borel measurable map from  $\Xi$  to  $X$  such that  $y_{xa}(\xi) = y$  if  $y \in Y_{xa}$  and  $\tau_{y|xa}(\xi) = \tau_{xa}(\xi)$ , with an arbitrary tiebreaker.

Consider an MDP with state space  $X$  and state-dependent action spaces  $A_x$ . Let  $r_{xa} = \mathbb{E}(\tau_{xa}(\tilde{\xi}))$  be the reward of action  $a$  in state  $x$ , and let  $p_{y|xa} = \mathbb{P}(y_{xa}(\tilde{\xi}) = y)$  be the probability of a transition from state  $x$  to state  $y \in Y_{xa}$  under action  $a$ . We now demonstrate that the reward-to-go function of this MDP in state  $x$  coincides with the residual project duration  $\varphi_x$ .

**PROPOSITION 8.** The optimal value  $\varphi_x$  of the multiple optimal stopping problem  $\mathcal{P}_x$  can be computed recursively via the Bellman equation

$$\begin{aligned} \varphi_x &= \max_{a \in A_x} r_{xa} + \sum_{y \in Y_{xa}} p_{y|xa} \varphi_y \quad \forall x \in X \setminus X_\infty \quad \text{and} \\ \varphi_x &= 0 \quad \forall x \in X_\infty. \end{aligned} \quad (12)$$

**PROOF.** The definition of the feasible set  $F_x(\xi)$  implies that  $\varphi_x = 0$  for  $x \in X_\infty$ . Next, fix some  $x \in X \setminus X_\infty$ . We first show that  $\varphi_x$  is bounded above by the right-hand side of the Bellman equation (12) corresponding to state  $x$ . Recall that  $\mathcal{P}_x$  is feasible and has a finite optimal value. We then select  $\varepsilon > 0$  and let  $(c, i)$  be an  $\varepsilon$ -optimal solution for  $\mathcal{P}_x$  with

$$\varphi_x - \varepsilon \leq \mathbb{E}(c_n(\tilde{\xi})). \quad (13)$$

Let  $a = (\alpha, \mathcal{A}) \in A_x$  be the action corresponding to  $(c, i)$  as given by Proposition 7, and construct the first state transition time  $\tau_{xa}$  as well as the random successor state  $y_{xa}$  corresponding to  $a$  as in Definitions 7 and 8, respectively. The law of total probability then ensures that

$$\begin{aligned} \mathbb{E}(c_n(\tilde{\xi})) &= \sum_{y \in Y_{xa}} p_{y|xa} \\ &\cdot \int_0^\alpha \mathbb{E}(c_n(\tilde{\xi}) \mid y_{xa}(\tilde{\xi}) = y, \tau_{xa}(\tilde{\xi}) = \tau) d\phi_{xa}^y(\tau), \end{aligned} \quad (14)$$

where  $\phi_{xa}^y(\tau) = \mathbb{P}(\tau_{xa}(\tilde{\xi}) \leq \tau \mid y_{xa}(\tilde{\xi}) = y)$  represents the distribution function of the first state transition time  $\tau_{xa}(\tilde{\xi})$  conditional on the successor state being  $y$ . By the memoryless property of the exponential task durations, for any  $y \in Y_{xa}$  and  $\tau \in \{\tau_{xa}(\xi): \xi \in \Xi, y_{xa}(\xi) = y\}$  there is a residual completion and interdiction strategy  $(c^{y,\tau}, i^{y,\tau})$  feasible in  $\mathcal{P}_y$  that satisfies

$$\begin{aligned} \mathbb{E}(c_n(\tilde{\xi}) \mid y_{xa}(\tilde{\xi}) = y, \tau_{xa}(\tilde{\xi}) = \tau) \\ = \tau + \mathbb{E}(c_n^{y,\tau}(\tilde{\xi})) \leq \tau + \varphi_y, \end{aligned} \quad (15)$$



where the last inequality holds because of the feasibility of  $(c^y, \tau, i^y, \tau)$  in  $\mathcal{P}_y$ . A formal proof of this statement is provided in Lemma 2 in Gutin et al. (2014). Combining (13), (14), and (15) gives

$$\begin{aligned}\varphi_x - \varepsilon &\leq \sum_{y \in Y_{xa}} p_{y|xa} \int_0^\alpha (\tau + \varphi_y) d\phi_{xa}^y(\tau) \\ &= \sum_{y \in Y_{xa}} p_{y|xa} [\mathbb{E}(\tau_{xa}(\tilde{\xi}) | y_{xa}(\tilde{\xi}) = y) + \varphi_y] \\ &= r_{xa} + \sum_{y \in Y_{xa}} p_{y|xa} \varphi_y \leq \max_{a' \in A_x} r_{xa'} + \sum_{y \in Y_{xa'}} p_{y|xa'} \varphi_y.\end{aligned}$$

Since  $\varepsilon > 0$  was chosen arbitrarily, we conclude that

$$\varphi_x \leq \max_{a' \in A_x} r_{xa'} + \sum_{y \in Y_{xa'}} p_{y|xa'} \varphi_y.$$

We now prove that  $\varphi_x$  is also bounded from below by the right-hand side of the Bellman equation (12) corresponding to state  $x$ . To this end, we choose again a tolerance  $\varepsilon > 0$  and let  $a = (\alpha, \mathcal{A}) \in A_x$  be an  $\varepsilon/2$ -optimal solution for

$$\max_{a' \in A_x} r_{xa'} + \sum_{y \in Y_{xa'}} p_{y|xa'} \varphi_y. \quad (16)$$

We construct the first state transition time  $\tau_{xa}$  as well as the random successor state  $y_{xa}$  corresponding to action  $a$  as in Definitions 7 and 8, respectively. Moreover, we let  $(c^y, i^y)$  be an  $\varepsilon/2$ -optimal solution for problem  $\mathcal{P}_y$  for each successor state  $y \in Y_{xa}$ . This means that

$$\varphi_y - \frac{\varepsilon}{2} \leq \mathbb{E}(c_n^y(\tilde{\xi})) \quad \forall y \in Y_{xa}. \quad (17)$$

By Lemma 3 in Gutin et al. (2014), which holds because of the memoryless property of the exponential task durations, the schedules  $(c^y, i^y)$ ,  $y \in Y_{xa}$ , can be amalgamated into an aggregate interdiction policy  $(c, i)$  that is feasible in  $\mathcal{P}_x$  and satisfies

$$\mathbb{E}(c_n(\tilde{\xi})) = r_{xa} + \sum_{y \in Y_{xa}} p_{y|xa} \mathbb{E}(c_n^y(\tilde{\xi})).$$

From this we conclude that

$$\begin{aligned}\varphi_x &\geq \mathbb{E}(c_n(\tilde{\xi})) = r_{xa} + \sum_{y \in Y_{xa}} p_{y|xa} \mathbb{E}(c_n^y(\tilde{\xi})) \\ &\geq r_{xa} + \sum_{y \in Y_{xa}} p_{y|xa} \varphi_y - \frac{\varepsilon}{2} \\ &\geq \max_{a' \in A_x} r_{xa'} + \sum_{y \in Y_{xa'}} p_{y|xa'} \varphi_y - \varepsilon,\end{aligned}$$

where the first inequality follows from the feasibility of  $(c, i)$  in  $\mathcal{P}_x$ , and the second inequality holds because of (17), and the third inequality exploits the  $\varepsilon/2$ -optimality of action  $a$  in (16). Because this argument holds for all  $\varepsilon > 0$ , we find that

$$\varphi_x \geq \max_{a' \in A_x} r_{xa'} + \sum_{y \in Y_{xa'}} p_{y|xa'} \varphi_y.$$

Thus, the claim follows.  $\square$

Proposition 8 shows that the solution of the multiple optimal stopping problem  $\mathcal{P}_x$  reduces to the solution of an MDP with a finite state space. It is well known that such MDPs are optimized by deterministic policies, which implies that there is always an optimal interdiction schedule that is deterministic (Puterman 1994). The MDP (12) still seems difficult to solve since it has an infinite action space as  $\alpha$  ranges over the interval  $[0, \infty]$ . We will now show that the MDP can be further simplified to one with both finite state and action spaces.

**COROLLARY 1.** *The maximum in the Bellman equation (12) is always achieved at an action  $a = (\alpha, \mathcal{A}) \in A_x$  with  $\alpha \in [0, \infty]$ .*

**PROOF.** See Gutin et al. (2014).  $\square$

Corollary 1 implies that the optimal decision in state  $x \in X$  is to (i) interdict some tasks  $\mathcal{A} \subseteq N_x$  immediately or (ii) wait until one of the active tasks terminates. In case (i) the project evolves instantly to a new state that differs from  $x$  only in that the active nondelayed tasks contained in  $\mathcal{A}$  are delayed. The optimal decision in the new state is to wait until one of the active tasks terminates. Indeed, if interdicting a nondelayed task in the new state reduced the expected project duration, then the action taken in state  $x$  would have been strictly suboptimal. Thus, in case (i) it is optimal to interdict some tasks  $\mathcal{A} \subseteq N_x$  and then wait until one of the active tasks terminates. Because the optimal strategy for case (ii) is to wait, we can interpret (ii) as a special case of (i) in which  $\mathcal{A} = \emptyset$ .

The above reasoning implies that the MDP at hand can be simplified by combining its decision epochs to pairs, where the first epoch in each pair corresponds to an immediate (but potentially void) interdiction action, and the second epoch relates to a waiting period until an active task ends. Each pair of epochs in the original MDP corresponds to an epoch in the new simplified MDP. The action space in state  $x$  now simplifies to the finite set  $A_x^d = \{a \in 2^{N_x} : |a| \leq b_x\}$  that captures all task sets that may be interdicted. The possible successor states of  $x$  under action  $a \in A_x^d$  and the events triggering a state transition are listed in Table 2. The set of all such successor states is denoted by  $Y_{xa}^d$ . We also denote by  $r_{xa}^d = r_{x'a'}$  the reward of action  $a \in A_x^d$  in state  $x$ , and we let  $p_{y|xa}^d = p_{y|x'a'}$  be the probability of a transition from  $x$  to  $y \in Y_{xa}^d$  under action  $a \in A_x^d$ , where the intermediate state  $x'$  is defined through  $b_{x'} = b_x - |a|$ ,  $I_{x'} = I_x$ ,  $N_{x'} = N_x \setminus a$ ,  $D_{x'} = D_x \cup a$  and  $C_{x'} = C_x$ , and action  $a' \in A_{x'}$  is set to  $a' = (\infty, \emptyset)$ .

The above discussion culminates in the following main theorem.



**Table 2** Possible Successor States of  $x$  Under Action  $a$  in the Simplified MDP

| Triggering event    | $v \in N_x \setminus a$ terminates                 | $v \in D_x \cup a$ terminates         |
|---------------------|--|---------------------------------------|
| Interdiction budget | $b_y = b_x -  a $                                  | $b_y = b_x -  a $                     |
| Idle tasks          | $I_y = I_x \setminus E_x(v)$                       | $I_y = I_x \setminus E_x(v)$          |
| Nondelayed tasks    | $N_y = (N_x \setminus (a \cup \{v\})) \cup E_x(v)$ | $N_y = (N_x \setminus a) \cup E_x(v)$ |
| Delayed tasks       | $D_y = D_x \cup a$                                 | $D_y = (D_x \cup a) \setminus \{v\}$  |
| Completed tasks     | $C_y = C_x \cup \{v\}$                             | $C_y = C_x \cup \{v\}$                |

Notes. Different events trigger transitions to different successor states  $y$ . The set of emerging tasks  $E_x(v)$ ,  $v \in N_x \cup D_x$ , is defined as in Table 1.

**THEOREM 2.** The optimal value  $\varphi_x$  of the multiple optimal stopping problem  $\mathcal{P}_x$  can be computed recursively via the Bellman equation

$$\varphi_x = \max_{a \in A_x^d} r_{xa}^d + \sum_{y \in Y_{xa}^d} p_{y|xa}^d \varphi_y \quad \forall x \in X \setminus X_\infty \quad (18)$$

and  $\varphi_x = 0 \quad \forall x \in X_\infty$ .

Here,  $A_x^d = \{a \in 2^{N_x} : |a| \leq b_x\}$  and  $Y_{xa}^d$  represents the set of all possible successor states of  $x$  under action  $a$  as specified in Table 2. Moreover,  $r_{xa}^d = 1/\lambda_{xa}$  and

$$p_{y|xa}^d = \begin{cases} \nu_v / \lambda_{xa} & \text{if } y \text{ arises due to the completion of} \\ & v \in N_x \setminus a, \\ \delta_v / \lambda_{xa} & \text{if } y \text{ arises due to the completion of} \\ & v \in D_x \cup a, \end{cases}$$

where  $\lambda_{xa} = \sum_{v \in N_x \setminus a} \nu_v + \sum_{v \in D_x \cup a} \delta_v$ .

The Bellman equation (18) can be solved with the memory-efficient value iteration proposed by Creemers et al. (2010). Equation (18) bears some resemblance to MDP formulations that have been proposed for the expected net present value optimization of stochastic project scheduling problems; see, e.g., Buss and Rosenblatt (1997) and Sobel et al. (2009). However, our result differs from these formulations in its scope as it captures the interactions of *two* agents. Moreover, our MDP emerges analytically from a fundamental description of the interdiction game as a multiple optimal stopping problem in continuous time. In contrast, the MDP models of Buss and Rosenblatt (1997) and Sobel et al. (2009) are introduced directly, thus representing higher-level descriptions of the underlying scheduling problems. An adaptation of the results in this section shows that the static interdiction game  $\mathcal{P}$  can be reformulated as an MDP where all interdiction decisions are taken a priori, that is, before any of the tasks are executed. This MDP would have a significantly larger state space, however, so that the solution of the MILP in Theorem 1 is preferable.

We close this section with some qualitative insights into the mechanics of the Bellman equation (18) that can be distilled from analytically tractable PERT networks.

In Proposition 3 we established that *static* interdiction plans can delay the expected durations of IID networks at most by a factor  $\nu/\delta$ . Using Theorem 2 it is easy to show that  $\nu/\delta$  remains the best possible delay factor even for *dynamic* policies. In view of Proposition 4, which asserts that static policies with limited interdiction resources already achieve this bound asymptotically, we should ask whether the use of dynamic policies actually provides any tangible added value.

In the following, we first characterize the optimal dynamic interdiction policy for a project on a parallel network. We then show that this policy can indeed substantially outperform any static interdiction plan on medium-sized project networks (i.e., outside the asymptotic regime where static policies are near optimal).

**PROPOSITION 9.** Consider a parallel network consisting of  $n - 2$  parallel tasks as well as two dummy start and end activities; see Definition 2. If  $0 \leq b_0 \leq n - 2$ , the best dynamic interdiction policy awaits the completion of the first  $n - 2 - b_0$  parallel tasks and then interdicts the remaining  $b_0$  parallel tasks simultaneously. The resulting expected project duration is given by  $(1/\delta) \sum_{v=1}^{b_0} 1/v + (1/\nu) \sum_{v=b_0+1}^{n-2} 1/v$ .

**PROOF.** See Gutin et al. (2014).  $\square$

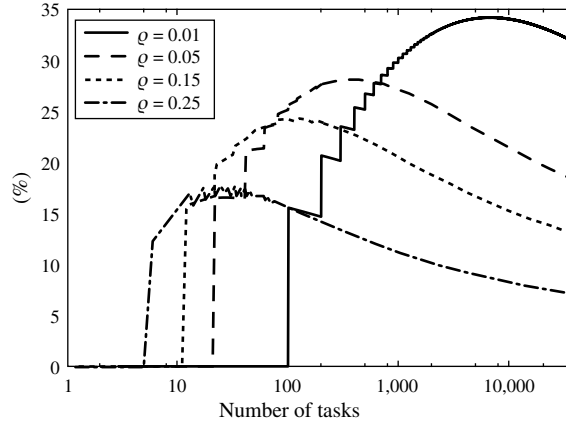
Assume now that the interdiction budget scales with the size of the parallel network, that is, set  $b_0 = \lfloor \varrho(n - 2) \rfloor$  for some  $\varrho \in (0, 1)$ . By Proposition 9, the project duration under an optimal dynamic interdiction policy is then given by

$$\frac{1}{\delta} \sum_{v=1}^{\lfloor \varrho(n-2) \rfloor} \frac{1}{v} + \frac{1}{\nu} \sum_{v=\lfloor \varrho(n-2) \rfloor + 1}^{n-2} \frac{1}{v} = \frac{\log n}{\delta} + \mathcal{O}(1).$$

As expected, the optimal static and dynamic interdiction policies display the same asymptotic performance as  $n$  tends to infinity; see Propositions 3 and 4. Maybe surprisingly, the asymptotic performance is independent of  $\varrho$ . For medium-sized networks, however, dynamic policies can substantially outperform static policies. Figure 2 shows the ratio of the expected project durations of the best dynamic and static policies as a function of the network size  $n$  for  $\nu/\lambda = 2$ . Note that these durations can be computed efficiently using the closed-form expression of Proposition 9 and a symmetry-exploiting recursion based on the Bellman equation of Proposition 2, respectively. We observe that smaller interdiction budgets necessitate a more effective deployment of the limited interdiction resources, and thus the advantage of dynamic over static policies is more pronounced for small values of  $\varrho$ .

We note that the performance ratio of the best dynamic and static policies is bounded by the delay factor of the best dynamic policy, which in turn is bounded by  $\nu/\delta$  for IID networks. The outperformance of the dynamic policies shown in Figure 2 is thus

**Figure 2** Outperformance of Dynamic over Static Policies in Parallel Project Networks



*Note.* The jaggedness of the graphs is caused by the stepwise increase of the interdiction budget  $b_0$ .

uniformly bounded by  $\nu/\delta = 200\%$  across all relative interdiction budgets  $q \in (0, 1)$ . We also remark that dynamic policies offer no advantage over static policies in serial networks. This is a consequence of Corollary 1, which implies that optimal interdiction actions are executed only when an active task ends. Thus, all tasks in a serial network are interdicted at their start times. Since the task durations are independent and identically distributed, it is immaterial which  $b_0$  tasks are actually interdicted, and one can select a static interdiction plan without any loss of optimality.

## 4. Extensions

The dynamic interdiction game of §3 admits several generalizations that enhance its realism. We first consider a setting in which the interdictor's actions are compromised by implementation uncertainty. Afterward, we outline a model in which the project manager can deploy renewable resources in order to speed up the project, and we discuss methods to incorporate nonexponential task durations. We show that the resulting generalized interdiction games are still amenable to dynamic programming formulations.

### 4.1. Implementation Uncertainty

Consider a variant of the standard game in which interdiction actions can fail with a certain probability. Thus, any attempt to interdict a task will decrement the residual interdiction budget but will not necessarily inflict a delay on the task's duration. If the first attempt fails, further attempts may be undertaken until the task is successfully interdicted or further interdiction attempts are no longer affordable. The possible failure of interdiction attempts requires us to redefine  $X$  as the set of all states satisfying (10a) and the first inequality in (10b) only. We let  $\tilde{\omega}_v^k$  be a Bernoulli random variable with success probability  $q_v \in [0, 1]$ , where

$k \in K = \{1, \dots, b_0\}$ , and we assume that  $\tilde{\omega}_v^k$  equals 1 if the  $k$ th attempt to interdict task  $v$  is successful and 0 otherwise. We further assume that all  $\tilde{\omega}_v^k$  are mutually independent and independent of the task durations. For ease of notation, we define  $\tilde{\omega} = (\tilde{\omega}_v^k)_{(v,k) \in V \times K}$ , and we let  $\tilde{\xi} = (\tilde{\zeta}, \tilde{\eta}, \tilde{\omega})$  be the random vector that comprises all exogenous random parameters affecting the interdiction game with implementation uncertainty. Thus, the support of  $\tilde{\xi}$  is  $\Xi = \mathbb{R}_+^{2n} \times \{0, 1\}^{n \times b_0}$ .

Let  $i_v^k$  denote the time of the  $k$ th attempt to interdict task  $v$ . The interdiction time  $i_v$  of task  $v$  can then be defined as the time of the first *successful* interdiction attempt. For a given state  $x \in X$  and scenario  $\xi \in \Xi$ , the generalized interdiction schedule  $i = ((i_v)_{v \in V}, (i_v^k)_{(v,k) \in V \times K})$  thus satisfies the relations

$$i_v^k \geq i_v^{k-1} \quad \forall v \in V, k \in K \setminus \{1\}, \quad (19a)$$

$$i_v^k = 0 \quad \forall v \in D_x, k \in K, \quad (19b)$$

$$i_v^k \geq \max_{u < v} c_u \quad \forall v \in V \setminus D_x, k \in K, \quad (19c)$$

$$i_v = 0 \quad \forall v \in D_x, \quad (19d)$$

$$i_v = \inf\{i_v^k : k \in K, \omega_v^k = 1\} \quad \forall v \in V \setminus D_x, \quad (19e)$$

where the completion schedule  $c = (c_v)_{v \in V}$  is defined as in §3. Constraints (19b) and (19c) can be viewed as the analogs of (11c) and (11d) for the interdiction attempts, respectively, whereas (19d) and (19e) relate the actual interdiction times to the times of the respective interdiction attempts. As in §3, we impose constraints that express the completion schedule in terms of the interdiction decisions,

$$c_v = 0 \quad \forall v \in C_x, \quad (19f)$$

$$c_v = \begin{cases} \max_{u < v} c_u + \zeta_v & \text{if } i_v \geq \max_{u < v} c_u + \zeta_v, \\ i_v + \eta_v & \text{else,} \end{cases} \quad \forall v \in V \setminus C_x, \quad (19g)$$

and we impose the budget constraint

$$|\{(k, v) \in K \times (N_x \cup I_x) : i_v^k < \infty\}| \leq b_x, \quad (19h)$$

which ensures that there are at most  $b_x$  attempts to interdict any active nondelayed or idle tasks in finite time. We now redefine  $F_x(\xi)$  as the set of all completion and interdiction schedules satisfying constraints (19a)–(19h).

Because the completion and interdiction schedules are coupled to the random problem parameters via (19a)–(19h), they must be interpreted as random times, that is, Borel measurable functions from the (redefined) support  $\Xi$  to  $[0, \infty]$ . Moreover, causality dictates that the times of the interdiction attempts must be modeled as stopping times. We thus represent the information

available at time  $t \geq 0$  by monitoring the project and the interdiction attempts through the  $\sigma$ -algebra

$$\mathcal{F}_t(c, i) = \sigma(\min\{c(\tilde{\xi}), t\mathbf{1}\}, \{\tilde{\omega}_v^k: (k, v) \in K \times V, i_v^k(\tilde{\xi}) \leq t\}),$$

which depends on both the completion and the interdiction schedule. The conditioning on  $i_v^k(\tilde{\xi}) \leq t$  ensures that only the results of interdiction attempts undertaken before time  $t$  impact the current information set. We then require the interdiction decisions  $i_v^k$  to be  $\mathbb{F}^+(c, i)$ -stopping times, where the filtration  $\mathbb{F}^+(c, i)$  is the right-continuous extension of  $\mathbb{F}(c, i) = \{\mathcal{F}_t(c, i)\}_{t \geq 0}$ .

If the project is in state  $x \in X$ , the interdictor thus solves a variant  $\mathcal{P}'_x$  of the multiple optimal stopping problem  $\mathcal{P}_x$  that differs from the standard model in §3 only in that it involves the generalized interdiction schedule  $i$  as well as the updated definitions of  $X$ ,  $F_x(\xi)$ , and  $\mathbb{F}^+(c, i)$  discussed above.

**THEOREM 3.** *The optimal value  $\varphi_x$  of problem  $\mathcal{P}'_x$  satisfies the Bellman equation*

$$\varphi_x = \max_{a \in A_x^d} \sum_{a' \subseteq a} q_{aa'} \left[ r_{xa'}^d + \sum_{y \in Y_{xaa'}} p_{y|xa'}^d \varphi_y \right] \quad \forall x \in X \setminus X_\infty, \quad (20)$$

and  $\varphi_x = 0$  for all  $x \in X_\infty$ . Here,  $q_{aa'} = \prod_{v \in a'} q_v \times \prod_{v \in a \setminus a'} (1 - q_v)$  and

$$Y_{xaa'} = \{y \in X: b_y = b_x - |a| \text{ and } \exists y' \in Y_{xa'}^d \text{ with } (I_{y'}, N_{y'}, D_{y'}, C_{y'}) = (I_y, N_y, D_y, C_y)\},$$

whereas  $A_x^d$ ,  $Y_{xa'}^d$ ,  $r_{xa'}^d$ , and  $p_{y|xa'}^d$  are defined as in Theorem 2.

**PROOF.** The proof parallels that of Theorem 2 and is therefore omitted.  $\square$

Note that  $a \in A_x^d$  in (20) represents the set of active tasks that are subjected to an interdiction attempt in state  $x$ , and  $a' \subseteq a$  denotes the subset of those tasks for which the interdiction attempt was successful. Since Theorem 3 reduces the multiple optimal stopping problem  $\mathcal{P}'_x$  to an MDP, we conclude that deterministic interdiction strategies remain optimal for the interdiction game with implementation uncertainty (Puterman 1994). The Bellman equation (20) can again be solved using the memory-efficient dynamic programming algorithm proposed by Creemers et al. (2010). Note that the problem  $\mathcal{P}'_0$  reduces to the standard game from §3 when  $q = 1$ .

#### 4.2. Crashing with Renewable Resources

We now assume that the project manager controls renewable resources (e.g., labor, machinery, or capital) that can be used to expedite the tasks. In this situation the interdictor and the project manager play a leader-follower game where the leader first chooses an interdiction policy with the aim to delay the project completion, in response to which the follower selects a resource allocation schedule with the aim to accelerate

the project. To simplify the exposition, we assume throughout this section that the proliferator controls a single resource.

In contrast to §§2 and 3, the components  $\tilde{\zeta}_v$  and  $\tilde{\eta}_v$  of the random vector  $\tilde{\xi}$  now represent the uncertain work contents of task  $v$  in the normal and delayed mode, respectively. The work content of a task can be understood as a measure of the effort (e.g., man-hours) required for its completion (Wieseemann et al. 2012). This convention leads to an endogenization of the task durations, which not only depend on their exogenous work contents but also on the resource allocation chosen by the follower. Below we let  $E \subseteq \mathbb{R}_+^n$  be a polyhedron that characterizes all possible assignments of the resource to the project tasks at any fixed time, and we denote by  $\mathcal{E}$  the set of all Borel measurable and integrable functions from  $\mathbb{R}_+$  to  $E$ . Thus, any  $e \in \mathcal{E}$  characterizes a particular allocation schedule of the resource over time, where  $e(t) \in E$  represents the allocation at time  $t$ . We let  $\tilde{e}$  denote the identity mapping on  $\mathcal{E}$ , which can be interpreted as a random object on the function space  $\mathcal{E}$  equipped with its cylindrical  $\sigma$ -algebra (Ash 1972, §2.7.4).

If activity  $v$  is initiated at time  $t$  and has work content  $w$ , then its duration under the allocation schedule  $e \in \mathcal{E}$  is set to

$$d_v(t, e, w) = \inf \left\{ \tau \in \mathbb{R}_+ : \int_t^{t+\tau} e_v(s) ds \geq w \right\}.$$

This definition captures our intuition that task  $v$  terminates as soon as its cumulative resource consumption exceeds its work content. For any state  $x \in X$ , scenario  $\xi \in \Xi$ , and allocation schedule  $e \in \mathcal{E}$ , we define  $F_x(e, \xi)$  as the set of admissible completion and interdiction schedules satisfying constraints (11a)–(11e), where  $\zeta_v$  is replaced with  $d_v(\max_{u < v} c_u, e, \xi_v)$  and  $\eta_v$  is replaced with  $d_v(\max_{u < v} c_u, e, \eta_v)$ .

From the leader's perspective, the resource allocation selected by the follower behaves like an exogenous random variable. The interdiction and completion times must therefore be modeled as generalized random times, that is, measurable functions from  $\mathcal{E} \times \Xi$  to  $[0, \infty]$ . Similar to §3, the leader observes the task durations and the follower's resource allocation only indirectly through their implied completion schedule  $c$ . The leader's information at time  $t$  is thus captured by the  $\sigma$ -algebra  $\mathcal{F}_t^l(c) = \sigma\{\min\{c(\tilde{e}, \tilde{\xi}), t\mathbf{1}\}\}$ . As usual, we let  $\mathbb{F}^l(c) = \{\mathcal{F}_t^l(c)\}_{t \geq 0}$  be the corresponding filtration. Causality then dictates that the interdiction schedule  $i$  must be modeled as an  $\mathbb{F}^{l+}(c)$ -stopping time, where the filtration  $\mathbb{F}^{l+}(c)$  is the right-continuous extension of  $\mathbb{F}^l(c)$ . Thus, the leader's policy is chosen from within the set

$$\mathfrak{P}_x^l = \{(c, i):$$

$c_v$  and  $i_v$  are generalized random times  $\forall v \in V$ ;



$i_v$  is an  $\mathbb{F}^{l+}(c)$ -stopping time  $\forall v \in V$ ;  
 and  $(c(e, \xi), i(e, \xi)) \in F_x(e, \xi) \forall (e, \xi) \in \mathcal{E} \times \Xi$ .

The follower also monitors the project, thereby acquiring information about the tasks' work contents. His resource allocation strategy is thus modeled as a stochastic process valued in  $E$ , that is, a Borel measurable function from  $\Xi$  to  $\mathcal{E}$ , adapted to the available information. Thus, the follower's policy is chosen from

$$\mathfrak{P}_x^f(c) = \{e: e \text{ is an } \mathbb{F}^{f+}(e, c)\text{-adapted stochastic process valued in } E\},$$

where the filtration  $\mathbb{F}^{f+}(e, c)$  is the right-continuous extension of  $\mathbb{F}^f(e, c) = \{\mathcal{F}_t^f(e, c)\}_{t \geq 0}$ , and  $\mathcal{F}_t^f(e, c) = \sigma\{\min\{c(e(\tilde{\xi}), \tilde{\xi}), t\}\}$  encodes the follower's information at time  $t$ . Note that there is no need to account for the leader's decisions in the follower's information set as the follower knows the leader's policy and can therefore retrace the interdiction decisions at any time  $t$ .

The leader aims to find a causal interdiction schedule and corresponding completion schedule that maximizes the worst-case expected project makespan. The worst case is taken over all causal resource allocation strategies of the follower. Thus, in state  $x \in X$  the leader solves the robust optimal stopping problem

$$\varphi_x = \max_{(c, i) \in \mathfrak{P}_x^f} \min_{e \in \mathfrak{P}_x^f(c)} \mathbb{E}(c_n(e(\tilde{\xi}), \tilde{\xi})). \quad (\mathcal{P}_x'')$$

Below we will argue that the worst-case expected project makespan  $\varphi_x$  admits an interpretation as the reward-to-go function of a robust MDP.

**THEOREM 4.** *The optimal value  $\varphi_x$  of problem  $\mathcal{P}_x''$  satisfies the Bellman equation*

$$\varphi_x = \max_{a \in A_x^d} \min_{e \in E} r_{xa}^c(e) + \sum_{y \in Y_{xa}^d} p_{y|xa}^c(e) \varphi_y \quad \forall x \in X \setminus X_\infty, \quad (21)$$

with terminal condition  $\varphi_x = 0$  for all  $x \in X_\infty$ . Here,  $A_x^d$  and  $Y_{xa}^d$  are defined as in Theorem 2. Moreover, we set  $r_{xa}^c(e) = 1/\lambda_{xa}(e)$  and

$$p_{y|xa}^c(e) = \begin{cases} \nu_v e_v / \lambda_{xa}(e) & \text{if } y \text{ arises due to the completion} \\ & \text{of } v \in N_x \setminus a, \\ \delta_v e_v / \lambda_{xa}(e) & \text{if } y \text{ arises due to the completion} \\ & \text{of } v \in D_x \cup a, \end{cases}$$

where we use the definition  $\lambda_{xa}(e) = \sum_{v \in N_x \setminus a} \nu_v e_v + \sum_{v \in D_x \cup a} \delta_v e_v$ .

**PROOF.** An important ingredient of the proof is the observation that an optimal interdiction strategy executes interdictions only when tasks start or terminate, and an optimal resource allocation strategy is right continuous and constant between completion times of

active tasks. These properties can be proved by generalizing Proposition 7. The proof then closely follows the reasoning in Proposition 8. Details are omitted for the sake of brevity.  $\square$

Theorem 4 implies that  $\varphi_x$  can be viewed as the reward-to-go function of a robust MDP (Iyengar 2005, Nilim and El Ghaoui 2005, Wiesemann et al. 2013), where the immediate rewards  $r_{xa}^c(e)$  and the transition probabilities  $p_{y|xa}^c(e)$  depend on an "uncertain" parameter  $e$  that cannot be controlled, while the decision maker chooses actions that are optimal in view of the worst-case realization of  $e$  within the "uncertainty set"  $E$ . Since the proliferator can choose a different resource allocation in each state and in response to each action of the interdictor, the robust MDP (21) is  $(s, a)$ -rectangular in the sense of Wiesemann et al. (2013), which implies that the crashing game is optimized by a deterministic interdiction strategy. Note that the crashing game reduces to the standard game of §3 if we set  $E = \{1\}$ .

The algorithm proposed by Creemers et al. (2010) can still be used to solve the Bellman equation (21) in a memory-efficient manner. Evaluating the right-hand side of (21) for a fixed state-action pair, however, involves the solution of a minimization problem and is therefore more expensive than in the basic model. Nevertheless, the inner minimization problem in (21) is readily recognized as a linear fractional program (Charnes and Cooper 1962), which is equivalent to a standard linear program that can be solved efficiently. If the project manager's resource allocations are only subject to an overall budget constraint, then  $E$  reduces to a simplex, and the inner minimization problem in (21) can even be solved in time  $\mathcal{O}(n)$  via vertex enumeration.

### 4.3. Nonexponential Task Durations

The assumption that the task durations are exponentially distributed is not always justifiable. However, any nonexponential task duration can be approximated by a phase-type distribution that emerges from a mixture of exponential distributions (Harchol-Balter 2013). More precisely, one can subdivide each task into several (virtual) subtasks that are arranged in an acyclic precedence network with a unique source and sink, where the subtasks may occur serially, probabilistically, or concurrently. If all subtask durations are independent and exponentially distributed, then the overall task duration follows a phase-type distribution, which can be fitted to any target distribution by adjusting the topology of the precedence network and the rate parameters of the subtasks. The quality of the approximation improves as the number of subtasks grows, but two or three subtasks are sufficient to match the first three moments of an arbitrary task duration; see, e.g., Osogami and Harchol-Balter (2006). We remark that this subtask



model also introduces a dependence between the task durations in the delayed and in the nondelayed mode. Indeed, if a task is interdicted only after some of its subtasks are completed, then both the delayed and the nondelayed task duration depend on the durations of the completed subtasks.

Inserting the subtask networks of all project tasks into the original PERT network yields a refined PERT network of the project. We can then formulate a dynamic interdiction game on the refined PERT network assuming that the interdiction of any active task  $v$  decreases the rate parameters of *all* active and idle subtasks of task  $v$ . Because the subtask durations are independent and exponentially distributed, all results of §3 remain valid in this setting. The additional requirement that all subtasks of a task must be interdicted simultaneously is unproblematic and even prevents the state-wise action spaces from growing too quickly with the number of subtasks. However, this model implicitly assumes that the interdictor can monitor the execution of the subtasks. Because the subtasks merely constitute artifacts of the distributional approximation, they have no physical meaning and remain unobservable. Hence, the interdiction game on the refined PERT network entails an information relaxation and thus overestimates the achievable project makespan. Fortunately, the refined PERT network can also be used to compute a lower bound on the achievable makespan. Indeed, the interdiction policy resulting from the basic model of §3 is obtained under the assumption of exponentially distributed task durations and is therefore suboptimal if the task durations are actually governed by phase-type distributions. The actual makespan of this policy can be evaluated on the refined PERT network because interdiction actions are only executed when one of the project tasks terminates and because each termination of a task in the original PERT network corresponds to the termination of a subtask on the refined PERT network (but not vice versa). Thus, the actual makespan of this policy provides a lower bound on the best makespan that is achievable when the task durations follow phase-type distributions, and the gap between the upper and lower bounds estimates the loss of optimality of this candidate policy.

We remark that the refined PERT network can be substantially larger than the original one as it subdivides each task into several subtasks. When only limited computational resources are available, the modeler may thus be forced to trade off the level of aggregation of the original PERT network against the accuracy of the distributional model for the task durations.

## 5. Numerical Experiments

We apply the formulations of the previous sections to the nuclear weapons development project studied by

**Table 3** Characteristics of the Project Variants Corresponding to the Three Enrichment Technologies

| Enrichment technology       | No. of tasks | No. of precedences | Order strength |
|-----------------------------|--------------|--------------------|----------------|
| Gas centrifuges (GC)        | 99           | 248                | 0.65           |
| Gaseous diffusion (GD)      | 99           | 243                | 0.65           |
| Aerodynamic enrichment (AE) | 97           | 239                | 0.66           |

*Note.* The order strength denotes the fraction of all  $n(n-1)/2$  theoretically possible precedences enforced by the partial order  $<$ , and it is a well-known measure for the difficulty of PERT instances (Demeulemeester et al. 2003).

Harney et al. (2006). The objective of that project is to produce a first batch of fission-type nuclear weapons, and its major steps are (i) the diversion of yellowcake from uranium mines, (ii) the production of uranium hexafluoride from yellowcake, (iii) the construction of uranium enrichment facilities, (iv) the conversion of uranium hexafluoride to highly enriched uranium metal, and (v) the design and construction of the actual weapons. Each step consists of multiple tasks that must be executed subject to various precedence constraints. As in Harney et al. (2006), we assume that the proliferator can choose between three different enrichment technologies in step (iii): gas centrifuges (GC), gaseous diffusion (GD), and aerodynamic enrichment (AE). Contrary to Harney et al. (2006), however, we assume that all precedences are of finish-start type and that the task durations are exponentially distributed. We choose the standard and interdicted rates  $\nu_v$  and  $\delta_v$ ,  $v \in V$ , such that the expected task durations  $\mathbb{E}[\tilde{\xi}_v]$  and  $\mathbb{E}[\tilde{\eta}_v]$  match the deterministic estimates presented in Harney et al. (2006). The characteristics of the three project variants are summarized in Table 3.

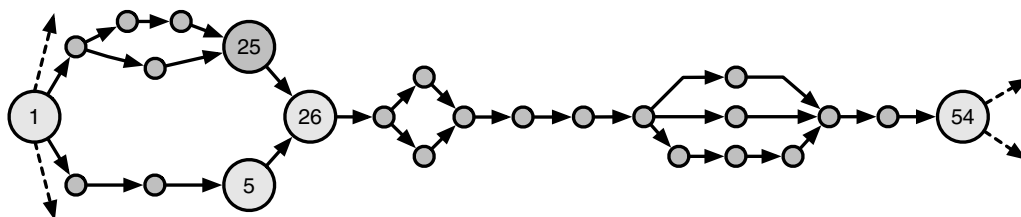
If we replace the uncertain task durations  $\tilde{\xi}_v$  with their expected values, then we obtain a deterministic interdiction game that can be formulated as an instance of problem (1). This problem (henceforth called the “nominal model”) predicts an uninterdicted project duration of 597.0 weeks (GC) and 545.0 weeks (GD and AE). Under technology GC, the project has 42 critical paths, whereas the projects associated with GD and AE both have 21 critical paths. If we assume that the task durations are exponentially distributed, on the other hand, then the Bellman equation (4) reveals that the expected uninterdicted project duration amounts to 729.6 weeks (GC), 636.0 weeks (GD), and 638.9 weeks (AE). Hence, we observe that the nominal model underestimates the project duration by up to two and a half years. In the stochastic model, every path has a strictly positive probability of being critical. Moreover, simulations show that there are no paths that are critical with a probability of more than 0.9% in any of the three projects, and there are 232 (GC), 220 (GD), and 226 (AE) paths that are critical with a probability of at least 0.1%. We thus conclude that the nominal model results

**Table 4** Results for the Standard Game

| Budget | Gas centrifuges (GC) |            |          | Gaseous diffusion (GD) |            |          | Aerodynamic enrichment (AE) |            |          |
|--------|----------------------|------------|----------|------------------------|------------|----------|-----------------------------|------------|----------|
|        | Nominal              | Stochastic | Run time | Nominal                | Stochastic | Run time | Nominal                     | Stochastic | Run time |
| 0      | 729.6                | —          | 0:00     | 636.0                  | —          | 0:00     | 638.9                       | —          | 0:00     |
| 1      | 810.9                | +5.5       | 0:00     | 695.8                  | —          | 0:00     | 698.7                       | —          | 0:00     |
| 2      | 870.9                | +5.5       | 0:01     | 744.2                  | +9.4       | 0:01     | 747.0                       | +9.4       | 0:00     |
| 3      | 919.3                | +15.5      | 0:02     | 800.1                  | +2.1       | 0:02     | 803.0                       | +2.1       | 0:01     |
| 4      | 975.3                | +9.6       | 0:05     | 822.5                  | +6.9       | 0:05     | 834.3                       | +3.2       | 0:04     |
| 5      | 999.3                | +26.2      | 0:10     | 842.4                  | +11.5      | 0:12     | 853.1                       | +10.9      | 0:08     |
| 6      | 1,021.7              | +31.1      | 0:19     | 862.9                  | +14.9      | 0:23     | 882.2                       | +6.2       | 0:15     |
| 7      | 1,040.6              | +36.8      | 0:32     | 886.8                  | +14.4      | 0:39     | 899.5                       | +12.8      | 0:25     |
| 8      | 1,064.6              | +36.7      | 0:47     | 909.3                  | +14.5      | 0:59     | 919.5                       | +16.3      | 0:36     |
| 9      | 1,084.6              | +40.2      | 1:05     | 929.2                  | +15.5      | 1:21     | 941.9                       | +16.4      | 0:50     |
| 10     | 1,101.4              | +45.9      | 1:22     | 946.2                  | +17.7      | 1:43     | 965.9                       | +13.3      | 1:01     |
| 11     | 1,117.4              | +50.6      | 1:38     | 962.2                  | +19.1      | 2:05     | 981.9                       | +15.8      | 1:13     |
| 12     | 1,133.4              | +52.7      | 1:52     | 978.2                  | +19.3      | 2:28     | 997.9                       | +16.2      | 1:25     |
| 13     | 1,149.4              | +53.0      | 2:12     | 994.2                  | +19.3      | 2:51     | 1,013.9                     | +16.8      | 1:37     |

*Notes.* For each enrichment technology and different interdiction budgets  $b_0$ , the table shows the expected project duration in weeks under the optimal interdiction plan for the nominal problem (1), the expected number of weeks gained by implementing the optimal interdiction plan for the dynamic stochastic model  $\mathcal{P}_0$  of §3, as well as the time required to solve  $\mathcal{P}_0$  (hh:mm).

**Figure 3** Detail of the PERT Network for the Nuclear Weapons Project Employing the AE Technology



*Notes.* Task 1 represents the project start, while tasks 5, 25, 26, and 54 correspond to the activities “divert yellowcake,” “assemble and integrate fluoridation plant,” “operate fluoridation plant,” and “cascade loading,” respectively. We use the same numbering as in Harney et al. (2006).

in severely biased estimates of the expected project duration, and that interdicting the critical tasks in the nominal model is unlikely to provide a near-optimal strategy for the stochastic model.

Table 4 compares the optimal interdiction schedules of the nominal problem (1) with those of the dynamic stochastic model  $\mathcal{P}_0$  (see §3) for different interdiction budgets. All tests were run on an Intel Core i7 3.4 GHz CPU with 16 GB of RAM. We solved problem (1) with CPLEX 12.3,<sup>2</sup> and the stochastic model was solved with a C++ implementation of the efficient dynamic programming algorithm proposed by Creemers et al. (2010). Because of the size of the project, we were unable to solve the static stochastic model of §2. In fact, the resulting MDPs have more than 220,000 states, and the corresponding MILPs contain more than 40 million variables and 65 million constraints.

Table 4 shows that larger interdiction budgets in the nominal problem lead to longer expected project durations. Although intuitive, this is not a mathematical necessity since the nominal model maximizes the

*nominal* project makespan, which is a biased estimate of the project’s *expected* makespan. The table also reveals that the stochastic interdiction schedules consistently outperform their nominal counterparts, and the outperformance increases with larger interdiction budgets. In fact, the expected project duration can be increased by up to one year if we replace the optimal nominal interdiction schedule with the optimal stochastic interdiction strategy. The table further elucidates how the computation times of the efficient dynamic programming algorithm are affected by the interdiction budget. Larger budgets lead to MDPs with larger state spaces, which in turn make the corresponding interdiction games harder to solve. The nominal model, on the other hand, can be solved within a few minutes for all considered interdiction budgets.

Figure 3 illustrates an example where the stochastic interdiction policy exploits online knowledge that is not available to the nominal interdiction strategy. The figure shows a subset of the tasks in the project employing the AE technology. Tasks 5 and 26 each have an expected duration of 60 weeks, whereas task 54 has a slightly smaller expected duration of 56 weeks. For the interdiction budget  $b_0 = 2$ , the optimal stochastic

<sup>2</sup> IBM CPLEX homepage: <http://www.ibm.com/software/integration/optimization/cplex-optimizer> (accessed September 5, 2014).

**Table 5** Results for the Interdiction Game with Implementation Uncertainty (§4.1)

| Budget | Gas centrifuges (GC) |            |          | Gaseous diffusion (GD) |            |          | Aerodynamic enrichment (AE) |            |          |
|--------|----------------------|------------|----------|------------------------|------------|----------|-----------------------------|------------|----------|
|        | Nominal              | Stochastic | Run time | Nominal                | Stochastic | Run time | Nominal                     | Stochastic | Run time |
| 0      | 729.6                | —          | 0:00     | 636.0                  | —          | 0:00     | 638.9                       | —          | 0:00     |
| 1      | 776.4                | —          | 0:00     | 682.6                  | —          | 0:00     | 677.5                       | —          | 0:00     |
| 2      | 792.1                | +29.0      | 0:01     | 718.5                  | +2.8       | 0:01     | 702.9                       | +7.7       | 0:00     |
| 3      | 848.7                | +11.3      | 0:04     | 740.3                  | +14.5      | 0:04     | 733.5                       | +5.8       | 0:02     |
| 4      | 870.9                | +25.0      | 0:12     | 775.2                  | +9.1       | 0:11     | 746.4                       | +17.9      | 0:04     |
| 5      | 908.8                | +20.7      | 0:30     | 787.4                  | +20.9      | 0:27     | 753.5                       | +32.6      | 0:11     |
| 6      | 920.6                | +39.3      | 1:02     | 792.8                  | +35.4      | 0:59     | 778.9                       | +26.7      | 0:24     |
| 7      | 925.3                | +61.7      | 1:48     | 812.7                  | +33.3      | 1:52     | 800.1                       | +23.5      | 0:44     |
| 8      | 951.2                | +59.7      | 2:47     | 827.4                  | +34.6      | 5:06     | 820.4                       | +19.9      | 1:12     |
| 9      | 961.8                | +70.2      | 3:50     | 850.1                  | +26.9      | 6:27     | 833.9                       | +22.3      | 1:44     |
| 10     | 973.5                | +77.3      | 4:59     | 870.7                  | +20.1      | 7:33     | 850.2                       | +21.3      | 2:19     |
| 11     | 1,001.2              | +66.7      | 6:08     | 865.7                  | +37.9      | 8:46     | 860.8                       | +25.4      | 2:55     |
| 12     | 1,026.8              | +57.0      | 7:18     | 871.4                  | +44.1      | 10:05    | 864.6                       | +35.5      | 3:32     |
| 13     | 1,027.0              | +71.8      | 8:25     | 870.2                  | +56.5      | 11:32    | 879.7                       | +33.6      | 4:08     |

Note. The columns have the same meaning as in Table 4.

interdiction policy always delays task 26, and it interdicts task 5 if task 25 has been completed by the time at which task 5 is started. If, on the other hand, task 25 has not yet been completed when task 5 is started, then task 54 is interdicted instead. In contrast, the optimal nominal interdiction strategy always interdicts tasks 5 and 26, resulting in an expected project duration that is 9.4 weeks shorter. We remark that the expected project duration of the nominal policy could be increased by seven weeks if tasks 5 and 54 were interdicted instead, but the deterministic model fails to recognize this opportunity.

In the next experiment, we assume that each attempt to interdict task  $v \in V$  fails with probability  $1 - q_v$ , where  $q_v$  is chosen uniformly at random from the interval  $[0, 1]$ . Table 5 compares the optimal interdiction schedules of the extended stochastic model of §4.1 with those of an adapted nominal model where we use the deterministic estimates  $\mathbb{E}[\tilde{\zeta}_v]$  and  $q_v \cdot \mathbb{E}[\tilde{\eta}_v] + (1 - q_v) \cdot$

$\mathbb{E}[\tilde{\zeta}_v]$  for the standard and interdicted task durations. Note that the expected project durations are smaller than in the first experiment since the interdictor is at a disadvantage. The stochastic interdiction schedules outperform their nominal counterparts by an even greater margin. We note that the expected project duration can be increased by almost one and a half years if we use the optimal stochastic interdiction strategy instead of the optimal deterministic interdiction schedule. Because of the increased complexity of the Bellman equation (20), the computation times increase by about 252%.

Table 6 compares the optimal interdiction schedules for the crashing game of §4.2. For this experiment, we define the set of admissible resource allocations as

$$E = \{e \in \mathbb{R}_+^n : e \in [1, 1.5 \times \mathbf{1}], \mathbf{1}^\top e = n + 0.5\};$$

that is, at any time, the proliferator can speed up the processing of exactly one task by 50%. We compare the

**Table 6** Results for the Crashing Game (§4.2)

| Budget | Gas centrifuges (GC) |            |          | Gaseous diffusion (GD) |            |          | Aerodynamic enrichment (AE) |            |          |
|--------|----------------------|------------|----------|------------------------|------------|----------|-----------------------------|------------|----------|
|        | Nominal              | Stochastic | Run time | Nominal                | Stochastic | Run time | Nominal                     | Stochastic | Run time |
| 0      | 515.5                | —          | 0:00     | 449.3                  | —          | 0:00     | 450.7                       | —          | 0:00     |
| 1      | 568.1                | +3.1       | 0:00     | 488.1                  | —          | 0:00     | 489.5                       | —          | 0:00     |
| 2      | 607.8                | +3.1       | 0:01     | 519.9                  | +5.9       | 0:01     | 521.3                       | +5.9       | 0:01     |
| 3      | 640.1                | +10.4      | 0:04     | 556.7                  | +0.7       | 0:05     | 558.1                       | +0.7       | 0:03     |
| 4      | 677.3                | +8.7       | 0:13     | 571.1                  | +4.7       | 0:15     | 578.6                       | +1.4       | 0:10     |
| 5      | 693.3                | +22.4      | 0:32     | 583.9                  | +8.7       | 0:39     | 590.4                       | +7.5       | 0:26     |
| 6      | 701.8                | +32.4      | 1:07     | 596.7                  | +11.7      | 1:23     | 610.2                       | +4.4       | 0:53     |
| 7      | 719.6                | +31.5      | 1:59     | 612.9                  | +11.1      | 2:29     | 620.8                       | +9.6       | 1:32     |
| 8      | 732.2                | +34.9      | 3:01     | 627.0                  | +12.1      | 3:53     | 633.1                       | +13.0      | 2:18     |
| 9      | 748.2                | +34.7      | 4:01     | 639.9                  | +13.6      | 5:28     | 651.1                       | +10.2      | 3:10     |
| 10     | 759.0                | +39.0      | 5:24     | 650.6                  | +16.0      | 7:08     | 663.7                       | +12.2      | 4:02     |
| 11     | 769.7                | +42.8      | 6:43     | 661.2                  | +17.3      | 8:56     | 674.3                       | +14.8      | 5:04     |
| 12     | 780.3                | +45.1      | 7:53     | 671.8                  | +17.5      | 10:42    | 684.9                       | +15.8      | 5:51     |
| 13     | 791.0                | +45.5      | 9:10     | 682.5                  | +17.5      | 12:25    | 695.5                       | +16.0      | 6:41     |

Note. The columns have the same meaning as in Table 4.

dynamic policy of §4.2 with a variant of the deterministic model that assumes that the proliferator allocates the resource evenly among all active tasks that are critical in the nominal problem. The expected project durations are smaller than in the standard game since the proliferator is at an advantage. We observe that the outperformance of the stochastic policies over the nominal ones is slightly smaller than in the standard game, but it remains significant. The computation times of the stochastic model increase by about 306%, which is due to the increased complexity of the robust Bellman equation (21).

## 6. Concluding Remarks

In this paper we have proposed models for static and dynamic interdiction games under uncertainty. These games are hard as they involve three players: a proliferator choosing the task start times and resource allocations, an interdictor choosing a subset of tasks to be delayed, and nature choosing random task durations. We have shown that the static game admits an MILP reformulation, and the dynamic game can be reduced to an MDP. These reformulations were facilitated by the Markov assumption, whereby the task durations are independent and exponentially distributed. Using an efficient dynamic programming algorithm we could solve interdiction games involving up to 100 tasks *exactly*. Future work should aim at relaxing the Markov assumption, which may be difficult to justify in reality. Much flexibility could be gained by modeling the task durations using phase-type distributions, for which the methods of this paper remain applicable with minor modifications. An interesting direction for future research would be to design approximate dynamic programming algorithms that find *inexact* but near-optimal solutions for interdiction games with more than 100 tasks and phase-type distributed task durations.

## Acknowledgments

The paper was written while Eli Gutin was an MSc student at Imperial College London. Daniel Kuhn gratefully acknowledges financial support from the Engineering and Physical Sciences Research Council [EP/I014640/1].

## References

- Ash RB (1972) *Real Analysis and Probability, Probability and Mathematical Statistics* (Academic Press, Waltham, MA).
- Bailey M, Shechter S, Schaefer A (2006) SPAR: Stochastic programming with adversarial recourse. *Oper. Res. Lett.* 34(3):307–315.
- Brown G, Carlyle W, Royset J, Wood K (2005) On the complexity of delaying an adversary's project. Golden B, Raghavan S, Wasil E, eds. *The Next Wave in Computing, Optimization, and Decision Technologies* (Springer, New York), 3–17.
- Brown G, Carlyle W, Harney R, Skroch E, Wood K (2009) Interdicting a nuclear-weapons project. *Oper. Res.* 57(4):866–877.
- Buss A, Rosenblatt M (1997) Activity delay in stochastic project networks. *Oper. Res.* 45(1):126–139.
- Charnes A, Cooper W (1962) Programming with linear fractional functionals. *Naval Res. Logist. Quart.* 9(3–4):181–186.
- Cox D, Hinkley D (1974) *Theoretical Statistics* (Elsevier, Amsterdam).
- Creemers S, Leus R, Lambrecht M (2010) Scheduling Markovian PERT networks to maximize the net present value. *Oper. Res. Lett.* 38(1):51–56.
- Demeulemeester E, Vanhoucke M, Herroelen W (2003) RanGen: A random network generator for activity-on-the-node networks. *J. Scheduling* 6(1):17–38.
- Dimitrov NB, Morton DP (2012) Interdiction models and applications. Herrmann JW, ed. *Handbook of Operations Research for Homeland Security* (Springer, New York), 73–104.
- Fudenberg D, Tirole J (1991) *Game Theory* (MIT Press, Cambridge, MA).
- Gutin E, Kuhn D, Wiesemann W (2014) Interdiction games on Markovian PERT networks. Technical report. Optimization Online: [http://www.optimization-online.org/DB\\_HTML/2013/04/3830.html](http://www.optimization-online.org/DB_HTML/2013/04/3830.html).
- Hagstrom JN (1988) Computational complexity of PERT problems. *Networks* 18(2):139–147.
- Harchol-Balter M (2013) *Performance Modeling and Design of Computer Systems. Queueing Theory in Action* (Cambridge University Press, Cambridge, UK).
- Harney R, Brown G, Carlyle M, Skroch E, Wood K (2006) Anatomy of a project to produce a first nuclear weapon. *Sci. Global Security* 14(2–3):163–182.
- Iyengar G (2005) Robust dynamic programming. *Math. Oper. Res.* 30(2):257–280.
- Kulkarni V, Adlakha V (1986) Markov and Markov-regenerative PERT networks. *Oper. Res.* 34(5):769–781.
- Lunday BJ, Sherali HD (2010) A dynamic network interdiction problem. *Informatica* 21(4):553–574.
- Morton DP (2010) Stochastic network interdiction. Cochran JJ, ed. *Wiley Encyclopedia of Operations Research and Management Science* (John Wiley & Sons, Hoboken, NJ).
- Nilim A, El Ghaoui L (2005) Robust control of Markov decision processes with uncertain transition matrices. *Oper. Res.* 53(5):780–798.
- Osogami T, Harchol-Balter M (2006) Closed form solutions for mapping general distributions to quasi-minimal PH distributions. *Performance Evaluation* 63(6):524–552.
- Pinker E, Szmerekovsky J, Tilson V (2013) Technical note—Managing a secret project. *Oper. Res.* 61(1):65–72.
- Puterman ML (1994) *Markov Decision Processes: Discrete Stochastic Dynamic Programming* (John Wiley & Sons, Hoboken, NJ).
- Sagan SD (2011) The causes of nuclear weapons proliferation. *Annual Rev. Political Sci.* 14:225–244.
- Smith JC (2010) Basic interdiction models. Cochran JJ, ed. *Wiley Encyclopedia of Operations Research and Management Science* (John Wiley & Sons, Hoboken, NJ).
- Sobel M, Szmerekovsky J, Tilson V (2009) Scheduling projects with stochastic activity duration to maximize expected net present value. *Eur. J. Oper. Res.* 198(3):697–705.
- Wiesemann W, Kuhn D, Rustem B (2012) Multi-resource allocation in stochastic project scheduling. *Ann. Oper. Res.* 193(1):193–220.
- Wiesemann W, Kuhn D, Rustem B (2013) Robust Markov decision processes. *Math. Oper. Res.* 38(1):153–183.
- Wood RK (2010) Bilevel network interdiction models: Formulations and solutions. Cochran JJ, ed. *Wiley Encyclopedia of Operations Research and Management Science* (John Wiley & Sons, Hoboken, NJ).
- Zheng J, Castañón DA (2012a) Dynamic network interdiction games with imperfect information and deception. *Proc. 51st IEEE Conf. Decision and Control* (IEEE, New York), 7758–7763.
- Zheng J, Castañón DA (2012b) Stochastic dynamic network interdiction games. *Proc. 2012 Amer. Control Conf.* (IEEE, New York), 1838–1844.