OPERATIONS RESEARCH
Vol. 57, No. 4, July–August 2009, pp. 866–877
ISSN 0030-364X | EISSN 1526-5463 | 09 | 5704 | 0866

DOI 10.1287/opre.1080.0643

Interdicting a Nuclear-Weapons Project

Gerald G. Brown, W. Matthew Carlyle
Operations Research Department, Naval Postgraduate School, Monterey, California 93942
{gbrown@nps.edu, mcarlyle@nps.edu}

Robert C. Harney
Systems Engineering Department, Naval Postgraduate School, Monterey, California 93942,
harney@nps.edu

Eric M. Skroch
Northrop Grumman Corporation, Virginia Beach, Virginia 23452,
eskroch@mac.com

R. Kevin Wood
Operations Research Department, Naval Postgraduate School, Monterey, California 93942,
kwood@nps.edu

A "proliferator" seeks to complete a first small batch of fission weapons as quickly as possible, whereas an "interdictor" wishes to delay that completion for as long as possible. We develop and solve a max-min model that identifies resource-limited interdiction actions that maximally delay completion time of the proliferator's weapons project, given that the proliferator will observe any such actions and adjust his plans to minimize that time. The model incorporates a detailed project-management (critical path method) submodel, and standard optimization software solves the model in a few minutes on a personal computer. We exploit off-the-shelf project-management software to manage a database, control the optimization, and display results. Using a range of levels for interdiction effort, we analyze a published case study that models three alternate uranium-enrichment technologies. The task of "cascade loading" appears in all technologies and turns out to be an inherent fragility for the proliferator at all levels of interdiction effort. Such insights enable policy makers to quantify the effects of interdiction options at their disposal, be they diplomatic, economic, or military.

Subject classifications: government: defense, foreign policy; military: targeting; programming: integer, linear; project management: CPM.
Area of review: Military and Homeland Security.
History: Received May 2007; revision received December 2007; accepted February 2008. Published online in *Articles in Advance* April 24, 2009.

From this session interdict, every fowl of tyrant wing.
Shakespeare, *The Phoenix and the Turtle*

1. Introduction

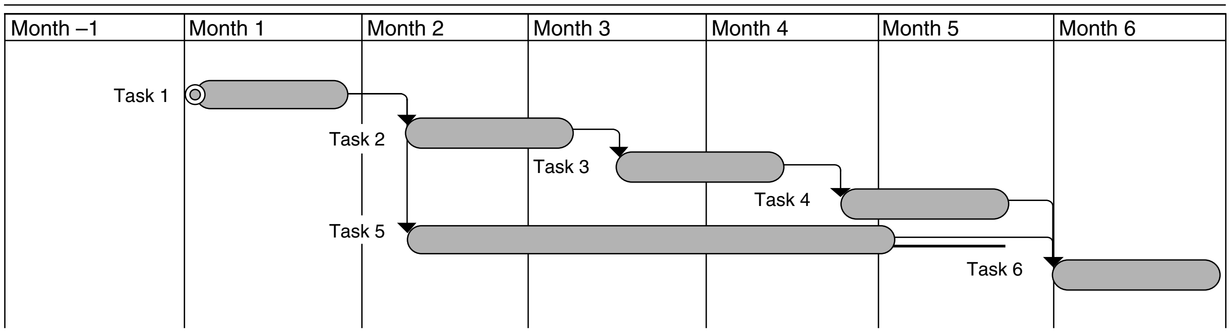
Sixty years after the United States detonated the first nuclear weapon, preventing the proliferation of such weapons is an international priority. One hundred eighty-eight nations have ratified the 1968 *Treaty on the Non-proliferation of Nuclear Weapons* (NPT), more than any other international arms-control agreement (IAEA 1970). Despite this fact, some nonsignatories, and even some signatories, of the NPT are currently developing nuclear weapons or are suspected of developing nuclear weapons covertly, e.g., Iran and Syria. The international community is seeking diplomatic, economic, and perhaps even military means to halt or at least hinder such development. This paper describes and demonstrates a new bilevel mathematical programming model that can help identify the most effective means to accomplish this task.

Production of nuclear weapons attracts intense global scrutiny. Because of this, and because of current NPT safeguards such as on-demand compliance inspections of civilian power reactors (IAEA 1970, Article III), any new *proliferator* would almost certainly run a covert production program. Outside discovery of such a program, or *project*, would depend on intelligence gathering or on a public announcement by the proliferator intended to intimidate other nations into political or economic concessions.

Upon learning of the probable existence of a weapons project, we would likely seek diplomatic means to halt it. Because diplomacy may take months or years to succeed, we can try to delay the project's completion through actions less severe than war such as embargoes on key materials and economic sanctions. The techniques described in this paper can help quantify the effects that such actions would have on the project. They also apply to more severe actions, such as the sabotage or bombing of key facilities, should such become necessary.

A *task* (also called an "activity" in the literature) represents a well-defined step in a project that duplicates no effort with any other step. With exceptions that arise because of "alternate technologies," described later, all tasks that define a project must be completed to complete the project. For example, "acquire metallurgical furnace" is a necessary step in building an enriched-uranium weapon, and it does not overlap any other tasks such as "acquire nitric acid storage

Figure 1. The optimal interdiction target can be a noncritical task.



Notes. This Microsoft Project display shows a project with standard precedence relationships between tasks, but with a one-week lag in each case. Each task on the critical path (Tasks 1, 2, 3, 4, and 6) has a normal duration of four weeks and an interdiction delay of two weeks. Task 5, not on the critical path, has normal duration of 12 weeks, interdiction delay of six weeks, and slack of two weeks (indicated by the dark narrow bar). Interdicting any single critical task delays overall project completion time by two weeks, but interdicting Task 5 delays the overall completion time by four weeks.

tank" or "fabricate uranium components"; however, completion of the last task may depend on the prior completion of "acquire metallurgical furnace," whereas the completion of "acquire nitric acid storage tank" may not.

An *interdiction* of a task represents an action that delays the completion of that task by a prespecified length of time. For example, interdiction of the "acquire metallurgical furnace" task could be achieved through an international agreement that hinders the purchase of any metallurgical furnace that could be used in weapons production; it could be achieved by a military strike that destroys that furnace so that another must be acquired, or it could be achieved through subtle acts of sabotage.

Using a set of task interdictions that is limited by "interdiction resources," such as money, labor, and diplomatic goodwill, we seek to maximally delay the completion of a proliferator's weapons project. We will plan these interdictions by formulating and solving a special, max-min "project-interdiction model." A project-management submodel (Malcolm et al. 1959, Kelley 1961) forms the "min" part of the interdiction model; this submodel integrates all of the tasks necessary to produce a small batch of gun-type, enriched-uranium weapons in the minimum amount of time (Harney et al. 2006). Integer programming constructs surround the submodel to enable maximization of that minimum through resource-limited interdiction. (In the remainder of the paper, we assume the reader's familiarity with standard project-management concepts such as "precedence relationship," "critical path," and "slack." Texts such as Moder et al. 1983 describe these concepts.)

Does optimal project interdiction warrant a formal model? Critical tasks, i.e., tasks on the project's critical path, seem like the obvious targets for interdiction. In fact, military planners to whom we present this problem typically assume that the optimal interdiction plan involves critical tasks with the longest interdiction delays. But, as Figure 1 shows, a noncritical task with sufficiently large delay can make a better target than any critical task. Indeed, the combinatorial nature of this problem makes a formal model imperative (Brown et al. 2005).

Reed (1994) is the first to suggest a project-interdiction model for optimal disruption of a nuclear-weapons project. He does not model the proliferator's full range of options, however, overlooking the proliferator's ability to use different methods to enrich uranium ("alternate technologies") and the ability to expedite project completion by applying extra resources to certain tasks ("crashing the project"). Brown et al. (2005) investigate the complexity of various project-interdiction models and show that even a simpler version of the model described in the current paper is NP-hard. In fact, they point out that this problem may not even belong to the problem class NP because simply evaluating the objective function for a fixed interdiction plan requires the solution of an NP-hard problem.

The inherent complexity of a nuclear-weapons project adds to the complexity of our interdiction problem. Accordingly, we have developed a complete decision-support system that integrates thousands of project details that cover physics, chemistry, industrial engineering, and materials science, as well as data on the proliferator's ability to marshal personnel and resources to achieve his goal. This system also manages the data pertaining to interdiction options and serves as the interface among the analyst, database, model, and optimizer. End-users, i.e., policy-makers, can use this system to display, and to compare quantitatively, the combined effects that various interdiction plans will have on a nuclear-weapons project.

2. Managing a Nuclear-Weapons Project

The proliferator will need to commit a great deal of material, labor, and technology to his nuclear-weapons project. Managing such a complex and expensive project will entail detailed, centralized coordination, especially if the project is to remain covert. Since the late 1950s, government and industry have widely employed mathematical techniques for managing complex projects. The original, simple models of the program evaluation review technique (PERT; see Malcolm et al. 1959) and the critical path method (CPM; see Kelley 1961) have been extended over the years to incorporate a variety of situations that arise in different types of projects. The proliferator will surely employ such techniques.

Project-management models are universally represented as networks (e.g., Moder et al. 1983, Chapter 1). In such representations, the additive length of the longest path through the network, i.e., the *critical path*, defines the overall duration of the project. (Actually, more than one critical path may exist.) The proliferator wants to minimize the completion time of his weapons project by reducing the duration of *critical tasks*, i.e., tasks on a critical path and tasks that end up on a newly created critical path as the durations of other tasks are reduced. While engaged in the project, the proliferator may also be able to choose among several alternate technologies to reach certain intermediate project objectives, or "milestone events." Modeling such alternatives is crucial because the proliferator will probably not broadcast his technological intentions, and he is free to change technologies in response to our actions.

The proliferator's problem can be represented as a classic project network with the following embellishments:

(1) Completion of any task in a "normal" amount of time consumes a fixed amount of one or more nonrenewable resources (Malcolm et al. 1959).

(2) The duration of an individual task may be expedited, i.e., shortened, by allocation of additional quantities of the required resources. The project is "crashed" when its tasks are expedited; see Kelley (1961) and his references, and see Charnes and Cooper (1962). We assume a linear relationship between the amounts of additional resources provided and a task's expedited duration, but a lower limit on that duration applies no matter the amount of additional resources allocated to it.

(3) Crashing is limited by the availability of each resource and by an overall monetary budget.

(4) Certain milestones may be achieved via alternate courses of action. When one alternative is chosen, the tasks in the other alternative(s) need not be completed. Alternate courses of action diverge at *decision nodes*, and a CPM model that includes one or more such nodes is called a *decision-CPM model* (Crowston and Thompson 1967). In Harney et al. (2006) and in this paper, a single decision node selects one of three uranium-enrichment technologies: *gas diffusion*, *gas centrifuge*, or *aerodynamic*.

(5) Standard finish-to-start (FS) precedence relationships between pairs of tasks are generalized to include start-to-start (SS), finish-to-finish (FF), or start-to-finish (SF). A finite lag time may be associated with each precedence constraint (e.g., Moder et al. 1983, Chapter 2); negative lags represent lead times.

3. Interdicting a Nuclear-Weapons Project

We henceforth use the term *interdictor* to refer to the entity (e.g., nation, group of nations) trying to delay the proliferator's project. The interdictor may exert interdiction effort against tasks in various ways but will be limited by any combination of a monetary budget, a weapons budget, diplomatic constraints, constraints on environmental damage, or constraints on economic consequences. We model the interdiction of the proliferator's project as a *max-min problem* (Danskin 1967). Our max-min problem is an instance of the bilevel programming model described by Bard (1982) and is an instance of a two-stage Stackelberg game, as explained by Israeli and Wood (2002). See also Brown et al. (2006) and that paper's references. The interdictor will first choose a task or set of tasks to interdict that maximizes the length of the resulting critical path. Then, after observing the interdiction plan, the proliferator will choose an enrichment technology to implement and tasks to expedite that minimize his overall project completion time.

In the real world, the interdictor could: interdict some tasks; observe the proliferator's response; interdict additional tasks, later, as the project proceeds; observe the proliferator's response; and so on. However, the two-stage model is solvable—a multistage model might not be—and it provides a valid lower bound on the interdictor's ability to delay the project. That is, the two-stage model is conservative. Of course, if the interdictor does decide to hold back some interdiction effort for later use, he can use the same model, over a rolling horizon, to suggest a sequence of interdictions using unexpended resources. The interdictor can do no worse with this strategy than he would by implementing the plan suggested by the two-stage model.

For ease of explanation, we make the following simplifying assumptions when modeling interdictions. Possible extensions, in parentheses, follow each assumption.

(1) Interdicting a task delays its completion by a constant amount. (This extends easily to discrete levels of delay that depend on level of interdiction effort.)

(2) All interdictions take place before the project begins. (The interdiction of a partially completed project can also be modeled. If interdicted, a partially completed task would just be delayed, but a fully completed task might need to be restarted. For example, if a piece of hardware were destroyed, it would need to be reconstructed or reacquired. All of the successors of such a task would probably be halted until that task is "recompleted," or we might require them to be restarted from scratch. Or, an alternate technology would need to be adopted.)

(3) Interdiction plans are limited by one or more "interdiction-budget constraints." These constraints may represent various limits on interdiction resources, physical or diplomatic (e.g., money, number of military actions, number of embargoes). (Logical constraints are easy to incorporate: for example, "If task i is interdicted, task j must not be.")

4. Case Study

The project-interdiction model will be easier to understand if we first present the foundations of a case study. In this study, we suspect that a rogue nation is developing a fission weapon. The basics of designing such a weapon are now well known and publicly available. Indeed, many of the details from the early weapons programs in the United States and elsewhere have been declassified and appear in the open literature (e.g., FAS 1998). Consequently, the key to developing a weapon will be to obtain the key raw material, viz., weapons-grade uranium or plutonium. The proliferator cannot buy these materials on the open market given current international controls, so he will need to make substantial investments in industrial infrastructure and develop his own, domestic production capability. (For context, see Spears 2001, which traces the life cycle of nuclear materials from raw ore to waste disposal.)

4.1. Case-Study Assumptions

We postulate the proliferator as a medium-sized developing country that:

- Operates several commercial nuclear reactors to generate electric power;
- Has a population that is generally well educated by several modern research universities;
- Has a modern chemical industry (even though many other industries may be underdeveloped by Western standards);
- Has substantial reserves of uranium ore; and
- Is a well-established producer of concentrated uranium ore called "yellowcake" (for the country's own consumption and for the international market).

Furthermore,

- The proliferator has ratified the Nuclear Nonproliferation Treaty (NPT); and
- All safeguards established by the International Atomic Energy Agency (IAEA) are in place. (The United Nations charters IAEA to ensure that signatories of the NPT use nuclear materials only for peaceful purposes. IAEA forges comprehensive safeguard agreements, negotiates additional protocols for locating and rationalizing the presence of any quantity of nuclear material, encourages state systems to account for and control nuclear material, and employs remote sensing to detect undeclared nuclear activities; see IAEA 2005, pp. 67–71.)

The proliferator seeks his own nuclear weapons to counter growing threats from neighboring countries and to gain credibility on the international stage as a key regional player. The proliferator's military and political advisers have come to the consensus that a total nuclear arsenal of several dozen weapons will achieve these goals.

The proliferator's economy depends heavily on nuclear power generation, and this dependence will only grow over time. The proliferator will not jeopardize this economic resource by openly violating the NPT, so he must operate a covert nuclear-weapons program. To remain covert, the program must operate independent of the proliferator's existing nuclear facilities, which are monitored by the IAEA. (History shows that most nations seeking nuclear weapons establish separate military programs, rather than divert enriched nuclear material from their safeguarded civilian facilities; see NERAC 2001.) Even if discovered, continued covert development lends itself to official denials and obfuscation. That will help the proliferator avoid quick reprisals from the international community and enable him to reach the ultimate goal of a nuclear arsenal.

Because NPT safeguards do not pertain to uranium mining and yellowcake production (U.S. Congress OTA 1993, p. 137), the surest path to a nuclear-weapons program leads through the development of a uranium-enrichment capability fed by existing yellowcake production; producing and using the alternative bomb-making material, plutonium, would be more difficult (e.g., EPA 2007). Highly enriched uranium (HEU) can be used in a gun-type or implosion-type fission weapon, but it can also fuel the proliferator's commercial reactors. Thus, if the proliferator must abandon his weapons program for any reason, he knows that no HEU he has time to produce will go to waste.

Given the proliferator's goal of completing a first weapon as soon as possible, we assume that he will pursue a gun-type weapon, the same design used in the "Little Boy" bomb dropped on Hiroshima, Japan in 1945. That design is simple but reliable: Little Boy was crude, but its designers were so confident in its construction that Hiroshima was its first full-scale test (Rhodes 1995, pp. 17–18).

The proliferator can hide his nuclear research-and-production facilities in existing industrial parks, where they should escape notice because of the concomitant growth of legitimate industry.

4.2. Case-Study Data

To reduce the chance of detection while still achieving sufficient capacity to meet the arsenal goal in just a few years, the proliferator will design production facilities to produce six weapons per year. This goal requires an annual output of 250 kilograms of HEU, which requires an annual input of approximately 68 metric tons of yellowcake (Harney et al. 2006). Because IAEA safeguards for yellowcake cover only imports and exports, covert diversion of this relatively small quantity from the existing production facilities should be easy: 5.6 metric tons per month suffices, and 5.6 tons of yellowcake will fit on a small truck.

Designing a nuclear weapon and constructing and operating the sophisticated support facilities required to build six weapons per year make for a complicated project. Necessary achievements include:

- Covert diversion of 68 metric tons of yellowcake annually;
- Production of enrichment-plant feed material (uranium hexafluoride, UF_6) from yellowcake;
- Uranium enrichment;
- Conversion of highly enriched UF_6 to uranium metal; and
- Design and construction of the actual weapons.

Harney et al. (2006) assess these requirements and model the proliferator's weapons project using a project network

Table 1. Cost and availability of resources in the case study.

Resource	Units	Unit cost ($/unit)	Total units available
Energy	MWhr	100	3,100,000
Materials	$k	1,000	190,000
Professional labor	Mmo	48,000	10,000
Skilled labor	Mmo	24,000	10,000
Unskilled labor	Mmo	6,000	6,000

Notes. The project is further constrained by a budget of $380 million. Costs are fictitious.

MWhr, megawatt-hours; $k, thousands of dollars; Mmo, man-months.

Table 2. Interdiction delay and cost indexing for tasks in Harney et al. (2006).

Delay index	A	B	C	D	E	F	G	H	I	J	K	L
Delay (months)	4	8	12	16	20	24	28	36	40	48	56	60
Cost index	a	b	c	d								
Cost ($M)	0.20	0.45	1.20	1.70								

Notes. Each uppercase letter code denotes an interdiction delay in months, and each lowercase letter codes an interdiction cost in millions of dollars. For example, the index "Fa" for task 131, "(Acquire) Hafnia crucibles," indicates that interdicting this task would inflict a 24-month delay on the task and would cost the interdictor 0.20 million dollars. Costs are fictitious.

containing approximately 200 nodes (tasks) and 600 arcs (direct precedence relationships between tasks). The proliferator manages five nonrenewable resources: energy, materials, professional labor (e.g., scientists, engineers), skilled labor (e.g., machinists, electronics technicians), and unskilled labor. Table 1 summarizes the costs and availability of these resources. Figure 2 displays a small part of the complete Gantt chart associated with one candidate project plan.

Our case study employs only mild, nonmilitary interdictions of tasks: we assume that the interdictor prefers to avoid the *casus belli* that, say, strategic strikes would imply. Table 2 shows how to interpret each task's "index" in Harney et al. (2006), to recover data that define the cost to interdict the task and the delay resulting from interdiction. Our computational examples also place limits on the total number of tasks interdicted.

5. A Model to Maximally Delay a Project: DCPMI

Here, we formulate a model to interdict a project that is defined using generalized decision-CPM constructs. This interdiction model, denoted DCPMI, will identify interdiction actions, subject to interdiction-resource constraints, that maximally delay a proliferator's nuclear-weapons project, given that the proliferator will observe any such actions and adjust his plans (i.e., choice of enrichment technology and task-expediting efforts) to minimize the project's completion time. We use an activity-on-node formulation of the project network in which a node represents a project task and an arc represents the partial order between a predecessor- and successor-task pair (e.g., Ahuja et al. 1993, pp. 732–734). For simplicity of exposition, we (a) make assumptions for the interdictor as described in §3 and (b) measure the consumption of each task-expediting resource r, whether monetary or physical, in units called "r-dollars."

Model: Delaying a Decision-CPM Project, DCPMI

Indices and Index Sets [cardinality]

$i, j \in N$ tasks (nodes) [~200].

$\{\text{start}, \text{end}\} \subset N$ project start and end tasks, respectively.

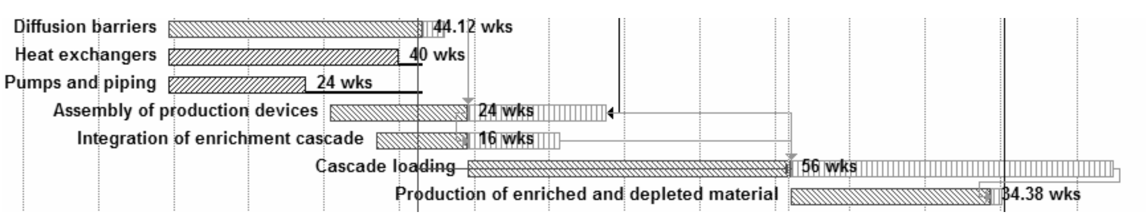
$(i, j) \in A$ precedence relationships: task i directly precedes task j (arcs) [~600].

$A_{FS} \subseteq A$ $(i, j) \in A_{FS}$ if task i must finish before task j can start.

$A_{FF} \subseteq A$ $(i, j) \in A_{FF}$ if task i must finish before task j can finish.

$A_{SF} \subseteq A$ $(i, j) \in A_{SF}$ if task i must start before task j can finish.

Figure 2. Portion of a Gantt chart, produced with Microsoft Project™, that displays seven of 196 tasks in the case-study project.



Notes. "(Acquire) Diffusion barriers" constitutes one task in the gas diffusion uranium-enrichment process; it is represented by a two-part horizontal bar. The left part, with diagonal hashing, represents the task's implemented length of 44.12 weeks, as labeled. The direction of the hashing, from top-left to bottom-right, indicates that the task is critical. The second part of the bar, on the right and with vertical hashing, indicates that the task has been expedited by the length of the hashed extension (an unlabeled 3.88 weeks). "(Acquire) Pumps and piping" is another task represented by a two-part bar. The left part, with top-right to bottom-left diagonal hashing, denotes a noncritical task; the right part (a thick line) indicates the amount of *slack* in this task (20.12 weeks, unlabeled), which represents the delay that the task can incur before becoming critical. The figure also shows an "FF" relationship between "Cascade loading" and "Production of enriched and depleted material."

$A_{SS} \subseteq A$	$(i, j) \in A_{SS}$ if task i must start before task j can start (A_{FS}, A_{FF}, A_{SF}, and A_{SS} are mutually exclusive and exhaustive subsets of A).
$p \in P$	decision outcomes (in our example, the three enrichment technologies) [3].
$N_0 \subseteq N$	tasks not associated with any decision (in all cases, $\{\text{start}, \text{end}\} \subset N_0$).
$N_p \subset N$	all tasks associated with decision outcome $p \in P$ (N_0 and the sets N_p, $\forall p \in P$, form a partition of N).
$N_D \subset N$	decision nodes.
$r \in R$	nonrenewable project resources (for the proliferator) [5].
$r' \in R'$	nonrenewable interdiction resources (for the interdictor) [2].

Data [units]

$\bar{d}_i$, $\underline{d}_i$	duration of task $i \in N$ with, respectively, no expediting effort and maximum expediting effort [weeks]. Note: $\bar{d}_i = \underline{d}_i \equiv 0$ for $i \in \{\text{start}, \text{end}\}$.
lag_{ij}	time lag for a precedence constraint indexed by $(i, j) \in A$ [weeks].
f'_{ij}	1 if $(i, j) \in A_{FF} \cup A_{FS}$, else 0. Indicates that task i's duration affects the start of task j.
f''_{ij}	1 if $(i, j) \in A_{FF} \cup A_{SF}$, else 0. Indicates that task j's duration affects task i's start.
c_{ir}	the cost, in terms of project resource $r \in R$, to complete task $i \in N$ with no expediting effort [r-dollars].
a_{ir}	the per-week cost, in terms of project resource $r \in R$, for expediting task $i \in N$ [r-dollars/week].
b_r	total budget for decision-and-expediting resource $r \in R$ [r-dollars].
delay_i	amount that task $i \in N$ is delayed if interdicted [weeks].
$d_{\max}$	a large number used to relax precedence constraints for tasks that are not actually completed (e.g., $d_{\max} = \max_{i \in N}(\bar{d}_i + \text{delay}_i) + \max_{(i,j) \in A} \lvert\text{lag}_{ij}\rvert$) [weeks].
$v_{ir'}$	cost to interdict task $i \in N$ in terms of interdiction resource $r' \in R'$ [r'-dollars].
$w_{r'}$	total budget for interdiction resource $r' \in R'$ [r'-dollars].

Variables [units if applicable]

X_i	1 if task $i \in N$ is interdicted, 0 otherwise (vector $\mathbf{X}$ is the "interdiction plan").
D_i	1 if task $i \in N$ is completed, 0 otherwise (vector $\mathbf{D}$ is the "decision plan").
E_i	amount task $i \in N$ is expedited (vector $\mathbf{E}$ is the "expediting plan") [weeks].
S_i	earliest start time of task $i \in N$ (vector $\mathbf{S}$) [weeks].

Formulation: DCPMI

$$Z^* = \max_{\mathbf{X} \in \Xi} \Big[\min_{(\mathbf{D}, \mathbf{E}, \mathbf{S}) \in \Delta(\mathbf{X})} S_{\text{end}} \Big], \tag{1}$$

where the set Ξ is defined by

$$\sum_{i \in N_0 \cup N_p} v_{ir'} X_i \leqslant w_{r'} \quad \forall r' \in R',\ p \in P, \tag{2}$$

$$X_i \in \{0, 1\} \quad \forall i \in N, \tag{3}$$

and the set $\Delta(\mathbf{X})$ is defined by

$$S_j - S_i \geqslant f'_{ij}(\bar{d}_i - E_i + \text{delay}_i X_i) - f''_{ij}(\bar{d}_j - E_j + \text{delay}_j X_j) + \text{lag}_{ij} - d_{\max}(1 - D_i) \quad \forall (i, j) \in A, \tag{4}$$

$$\sum_{i \in N \setminus \{\text{start, end}\}} c_{ir} D_i + \sum_{i \in N \setminus \{\text{start, end}\}} a_{ir} E_i \leqslant b_r \quad \forall r \in R, \tag{5}$$

$$D_j \geqslant D_i \quad \forall (i, j) \in A \mid i \notin N_D, \tag{6}$$

$$D_i = \sum_{j \mid (i,j) \in A} D_j \quad \forall i \in N_D, \tag{7}$$

$$D_i \equiv 1 \quad \forall i \in N_0, \tag{8}$$

$$0 \leqslant E_i \leqslant \bar{d}_i - \underline{d}_i \quad \forall i \in N, \tag{9}$$

$$D_i \in \{0, 1\} \quad \forall i \in N \setminus N_0, \tag{10}$$

$$S_i \geqslant 0 \quad \forall i \in N, \tag{11}$$

$$S_{\text{start}} \equiv 0. \tag{12}$$

Discussion

The objective function (1) represents the interdictor's desire to chose an *interdiction plan* $\mathbf{X}$ that increases individual task durations and thereby maximizes the project's overall completion time; simultaneously, it also represents the proliferator's desire to minimize that time by choosing which tasks to complete in a *decision plan* $\mathbf{D}$, and by choosing which tasks to expedite with *expediting plan* $\mathbf{E}$, and by solving for earliest start times $\mathbf{S}$. The interdictor's actions are restricted to the set Ξ, as defined by constraints (2)–(3), and for any specific interdiction plan $\mathbf{X}$ chosen by the interdictor—the reader should view $\mathbf{X}$ as data now—the proliferator's resulting decisions are restricted to the set $\Delta(\mathbf{X})$, as defined by constraints (4)–(12).

Each constraint (2) asserts a budget restriction for one of the decisions the proliferator can make: the total expenditure of budget resource r' cannot exceed $w_{r'}$ for any realization of the proliferator's decision plan. Specifically, for each enrichment technology a separate set of budget constraints limits overall interdiction expenditures for nonenrichment tasks together with the tasks for that specific enrichment technology. For example, given a single "budget" of one interdiction, the interdictor could attack one task that is not part of any specific enrichment technology, or he could interdict three tasks, one from each technology. This modeling assumption allows the interdictor to develop interdiction plans that are contingent upon the proliferator's choice of technology but not upon his application of expediting resources. This prevents situations in which the interdictor commits to interdictions of tasks specific to one technology, and the proliferator simply chooses a different technology and avoids interdiction altogether. We are therefore assuming that, by the time enrichment-specific tasks must be interdicted, the interdictor will have become aware of the proliferator's choice of technology and will interdict only the relevant tasks.

We express constraints (4) in a nonstandard fashion to emphasize the roles played by f'_{ij} and f''_{ij}. Each such constraint enforces a lower limit on the difference between the earliest start time of a task j and one of its predecessor tasks i, based on the nature of the precedence relationship between i and j. The term $(\bar{d}_i - E_i + \text{delay}_i X_i)$ is included if that precedence relationship relates to the finish time of task i, and the term $-(\bar{d}_j - E_j + \text{delay}_j X_j)$ is included if that precedence relationship relates to the finish time of task j. Each of these constraints also accounts for the lag between tasks i and j, specified by lag_{ij}.

Constraints (5) enforce monetary and/or physical resource limitations on the proliferator's actions. Constraints (6) require the completion of each successor of any completed task, except for certain successors of the decision node. Constraints (7) require that exactly one of the successors of the (single) decision node be completed, thereby guaranteeing the selection of exactly one enrichment technology. Constraints (8) require the completion of all non-decision tasks. Constraints (9) limit the amount by which each task i can be expedited.

5.1. EMIN: Optimal Decision-CPM After Interdiction

For a fixed interdiction plan $\widehat{\mathbf{X}} \in \Xi$, we denote the resulting decision-CPM model as EMIN($\widehat{\mathbf{X}}$) with objective function

$$Z_{\min}(\widehat{\mathbf{X}}) \equiv \min_{(\mathbf{D},\mathbf{E},\mathbf{S})\in\Delta(\widehat{\mathbf{X}})} S_{\text{end}} \tag{13}$$

and with constraints defined by (4)–(12). Our overall goal is therefore to solve

$$Z^* \equiv \max_{\mathbf{X}\in\Xi} Z_{\min}(\mathbf{X}). \tag{14}$$

In theory, we could solve (14) by enumerating the finite set of interdiction plans $\widehat{\mathbf{X}} \in \Xi$, solving EMIN($\widehat{\mathbf{X}}$) for each, and choosing the plan that results in the largest value of $Z_{\min}(\widehat{\mathbf{X}})$. In practice, Ξ is too large to enumerate, so we solve (14) with the decomposition algorithm described immediately below.

5.2. DMAX: How to Optimally Delay a Project with a Known Decision-and-Expediting Plan

Our procedure for solving DCPMI requires an upper bound on the optimal objective value, i.e., an optimistic value on how long we can expect to delay the proliferator. Accordingly, we formulate an optimization model, denoted DMAX($\widehat{\mathbf{D}}, \widehat{\mathbf{E}}$), that determines an optimal interdiction of any fixed decision-and-expediting plan $(\widehat{\mathbf{D}}, \widehat{\mathbf{E}})$; we assume that this plan always satisfies (4)–(12). A solution to this model yields the upper bound we seek because the proliferator's response is restricted to $(\widehat{\mathbf{D}}, \widehat{\mathbf{E}})$.

We adopt the dual, longest-path point of view in this formulation (e.g., Ahuja et al. 1993, pp. 733–734): the model routes one unit of "flow" along a single, longest path through the activity-on-node network. However, we must define several sets of auxiliary binary variables, and associated constraints, to "cost out" that path correctly.

Model DMAX($\widehat{\mathbf{D}}, \widehat{\mathbf{E}}$): Delaying a Project Having a Fixed Decision-and-Expediting Plan

Only constructs not defined earlier are listed here.

Indices and Index Sets

$\widehat{N}$ $\quad \{i \in N \mid \widehat{D}_i = 1\}$ (Note: $\{\text{start}, \text{end}\} \subset \widehat{N}$).

Variables [units]

- Z duration of the project [weeks].
- Y_{ij} 1 if the precedence relationship (i, j) is on the critical path, 0 otherwise.
- W_i 1 if task i is on the critical path and is not interdicted, 0 otherwise.
- W'_i 1 if task i is on the critical path and is interdicted, 0 otherwise.
- W''_i 1 if task i is on the critical path and its duration must be *subtracted*, 0 otherwise.

Formulation

$$Z_{\max}(\widehat{\mathbf{D}}, \widehat{\mathbf{E}}) \equiv \max_{\mathbf{X},\mathbf{Y},\mathbf{W},\mathbf{W}',\mathbf{W}''} Z \tag{15}$$

$$\text{s.t.}\quad Z \leqslant \sum_{i\in\widehat{N}} (\bar{d}_i - \widehat{E}_i) W_i + \sum_{i\in\widehat{N}} (\bar{d}_i - \widehat{E}_i + \text{delay}_i) W'_i - \sum_{i\in\widehat{N}} (\bar{d}_i - \widehat{E}_i) W''_i + \sum_{(i,j)\in A \mid i,j\in\widehat{N}} \text{lag}_{ij} Y_{ij}, \tag{16}$$

$$\sum_{j\in\widehat{N} \mid (i,j)\in A} Y_{ij} - \sum_{j\in\widehat{N} \mid (j,i)\in A} Y_{ji} = \begin{cases} +1 & \text{for } i = \text{start}, \\ 0 & \forall\, i \in \widehat{N}\setminus\{\text{start}, \text{end}\}, \\ -1 & i = \text{end}, \end{cases} \tag{17}$$

$$W_i + W'_i \leqslant \sum_{j\in\widehat{N} \mid (i,j)\in A_{\text{FS}}\cup A_{\text{FF}}} Y_{ij} \quad \forall\, i \in \widehat{N}\setminus\{\text{end}\}, \tag{18}$$

$$W_i + W'_i \leqslant \sum_{j\in\widehat{N} \mid (j,i)\in A_{\text{FS}}\cup A_{\text{SS}}} Y_{ji} \quad \forall\, i \in \widehat{N}\setminus\{\text{start}\}, \tag{19}$$

$$\sum_{j\in\widehat{N} \mid (j,i)\in A_{\text{SF}}\cup A_{\text{FF}}} Y_{ji} + \sum_{j\in\widehat{N} \mid (i,j)\in A_{\text{SS}}\cup A_{\text{SF}}} Y_{ij} \leqslant 1 + W''_i \quad \forall\, i \in \widehat{N}\setminus\{\text{start}, \text{end}\}, \tag{20}$$

$$W_i \leqslant 1 - X_i \quad \forall\, i \in \widehat{N}\setminus\{\text{start}, \text{end}\}, \tag{21}$$

$$W'_i \leqslant X_i \quad \forall\, i \in \widehat{N}\setminus\{\text{start}, \text{end}\}, \tag{22}$$

$$\sum_{i\in\hat{N}} v_{ir'} X_i \leqslant w_{r'} \quad \forall\, r' \in R', \tag{23}$$

$$X_i \equiv 0 \quad \forall\, i \in N_D \cup \{\text{start}, \text{end}\}, \tag{24}$$

$$W_{\text{start}} \equiv W_{\text{end}} \equiv 1, \tag{25}$$

$$\text{all variables} \in \{0,1\} \text{ except } Z. \tag{26}$$

Discussion

The objective function (15) represents the overall project duration Z. We use a nonstandard formulation to represent the calculation of this objective in constraint (16) to motivate and simplify the description of the decomposition algorithm that follows. Constraint (16) bounds the project duration by the sum of the task durations and the appropriate lag times along a critical path resulting from the fixed decision-and-expediting plan $(\hat{\mathbf{D}}, \hat{\mathbf{E}})$. A task's duration depends on whether or not it has been interdicted, and, based on interactions between the various precedence relations, this task's duration is added to the overall project duration, is subtracted from it, or has no influence. The upper bound implied by the right-hand side of (16) is valid because it computes the length of a critical path for any interdiction plan under the restrictive assumption that the proliferator cannot change his decision-and-expediting plan.

Flow-balance constraints for this activity-on-node network appear in (17). These ensure the consistent routing of a single unit of "critical flow" through the network, from the artificial start task to the artificial end task. Constraints (18) and (19) determine whether a task's duration can add to the critical path's length; the task's duration can be the uninterdicted, expedited duration or the interdicted, expedited duration. The variable W_i is 1 when (a) task i is on the critical path, (b) task i's duration adds to the length of the critical path, and (c) task i is not interdicted; W_i is 0 otherwise. The variable W_i' is 1 when (a) task i is on the critical path, (b) task i's duration adds to the length of the critical path, and (c) task i is interdicted; W_i' is 0 otherwise. Note that task i's duration adds to the critical-path length only if there exists a critical arc $(i, j) \in A_{\text{FS}} \cup A_{\text{FF}}$ and a critical arc $(j', i) \in A_{\text{FS}} \cup A_{\text{SS}}$.

Constraints (20) determine whether a particular task on the critical path has predecessor and successor relationships along that path that require its duration be subtracted from the overall project duration: the variable W_i'' is 1 if and only if task i is on the critical path, and this task's duration subtracts from the length of the critical path. This unusual situation occurs when (a) a sequence of tasks h-i-j lies on the critical path, (b) either the start or the finish time of task h places a lower bound on the finish time of task i, and (c) the start time of task i places a lower bound on the start or finish time of task j. In realistic situations, other tasks would constrain the start time and the end time of a task i, and it is unlikely that the sequence of three tasks required for this unusual situation would be critical simultaneously.

Constraints (17)–(25) are defined only with respect to tasks that are completed by the proliferator, i.e., tasks $i \in \hat{N}$. Therefore, no restrictions apply to $\mathbf{W}$, $\mathbf{W}'$, $\mathbf{W}''$, or $\mathbf{Y}$ for tasks involved in the technologies not chosen by the proliferator.

Constraints (21) and (22) determine whether the interdicted or uninterdicted version of a task can appear on the critical path. Constraints (23) enforce various interdiction-resource constraints on the interdictor, although they could also represent logical constraints such as "Do not place embargoes on more than two of the proliferator's critical resources," or "If task i_1 is interdicted, then task i_2 must not be." Constraints (24) preclude interdiction of the artificial start and end nodes and of any decision node. Constraints (25) maintain consistency of the definition of W_i: the artificial start and end nodes always appear on the critical path.

5.3. Algorithm MAXMIN

We now present a decomposition algorithm, denoted MAXMIN, to solve (14). As a generalization of Benders decomposition (Benders 1962), MAXMIN alternates between (a) a *master problem* that identifies an optimal interdiction plan for a fixed decision-and-expediting plan and (b) a *subproblem* that, for a fixed interdiction plan, chooses an optimal combination of enrichment technology and expediting plan for the weapons program. MAXMIN does look similar to the Benders-decomposition algorithm used by Israeli and Wood (2002) to identify interdictions in a road network that maximize the length of the shortest s-t path. However, unlike standard Benders decomposition, MAXMIN's subproblem is a difficult integer program, not a linear program (and not an integer program that solves directly through its linear programming relaxation, as with a shortest-path model). Furthermore, the master problem exploits a nominally nonconvergent bound, which we augment with special constraints to guarantee the algorithm's convergence.

At iteration K of MAXMIN, a set of decision-and-expediting plans $(\hat{\mathbf{D}}^k, \hat{\mathbf{E}}^k)$, $k = 1, \ldots, K$, will have been generated from the subproblem. The optimal objective to $\text{DMAX}(\hat{\mathbf{D}}^K, \hat{\mathbf{E}}^K)$ would provide a valid upper bound on Z^* at that point. However, to obtain a bound that can improve from iteration to iteration, MAXMIN solves a master problem that defines a restriction of $\text{DMAX}(\hat{\mathbf{D}}^K, \hat{\mathbf{E}}^K)$ by replacing the single constraint (16) in $\text{DMAX}(\hat{\mathbf{D}}^K, \hat{\mathbf{E}}^K)$ with the following constraints:

$$\begin{aligned} Z \leqslant &\sum_{i \mid \hat{D}_i^k = 1} (\bar{d}_i - \hat{E}_i^k) W_i + \sum_{i \mid \hat{D}_i^k = 1} (\bar{d}_i - \hat{E}_i^k + \text{delay}_i) W_i' \\ &- \sum_{i \mid \hat{D}_i^k = 1} (\bar{d}_i - \hat{E}_i^k) W_i'' + \sum_{(i,j) \in A \mid \hat{D}_i^k = \hat{D}_j^k = 1} \text{lag}_{ij} Y_{ij} \\ &\qquad \text{for } k = 1, \ldots, K. \end{aligned} \tag{27}$$

We refer to constraints (27) as "cuts" because they are analogous to Benders cuts; we denote the master problem that includes these cuts as DMAX^K.

The optimal objective value to DMAX^K provides a valid upper bound on Z^* because each cut in (27) is valid for any interdiction: (a) Either the cut at iteration k was generated by the same decision plan as the "active" technology for this instance of the master problem (that is, for $\widehat{\mathbf{D}}^k = \widehat{\mathbf{D}}^K$), and that cut's validity follows from the arguments used to justify $\text{DMAX}(\widehat{\mathbf{D}}, \widehat{\mathbf{E}})$, or (b) a different technology generated that cut, in which case the cut is relaxed. In the latter case, constraints (17)–(20) do not restrict variables W_i, W_i', and W_i'', so that the corresponding cut in (27) is relaxed.

Unfortunately, the upper bound from DMAX^K need not converge to Z^*. This is true because even if DMAX^K covered all decision plans, its solution would find the best interdiction plan assuming that the interdictor could dictate the proliferator's decision plan. To alleviate this difficulty, we simply add what we call *solution-elimination constraints* (SECs). (Brown et al. 1997 discuss SECs without naming them; Israeli and Wood 2002 define a generalization of SECs, called "super-valid inequalities," but use those inequalities to speed convergence of a decomposition algorithm, not to ensure its convergence.) The following SECs prohibit any interdiction plan $\widehat{\mathbf{X}}^k$, $k = 1, \ldots, K$, from being repeated but allow all others:

$$\sum_{i \mid \widehat{X}_i^k = 0} X_i + \sum_{i \mid \widehat{X}_i^k = 1} (1 - X_i) \geqslant 1, \quad k = 1, \ldots, K. \tag{28}$$

The upper bound provided by DMAX^K with constraints (28) may not be valid if some $\widehat{\mathbf{X}}^k$ is actually optimal, but that bound can drop below Z^* only if an optimal solution is in hand, at which point the validity of the bound is moot. Step 7 in the algorithm accommodates this situation. A full description of the algorithm follows.

Algorithm MAXMIN

Input: Data for DCPMI, and optimality tolerance $\varepsilon \geqslant 0$;
Output: ε-optimal interdiction plan $\mathbf{X}^*$, and provable optimality gap;
Note: $\underline{Z}_{\min}(\widehat{\mathbf{X}}^K)$ denotes the optimal objective value from $\text{EMIN}(\widehat{\mathbf{X}}^K)$, and $\bar{Z}_{\max}^K$ denotes the optimal objective value from DMAX^K with SECs added.

(1) Initialize upper bound $Z_{\text{UB}} \leftarrow \infty$, lower bound $Z_{\text{LB}} \leftarrow 0$, define the incumbent, null interdiction plan $\mathbf{X}^* \leftarrow \widehat{\mathbf{X}}^1 \leftarrow \mathbf{0}$ as the best found so far, and set iteration counter $K \leftarrow 1$;

(2) **Subproblem:** Attempt to solve $\text{EMIN}(\widehat{\mathbf{X}}^K)$ to determine the proliferator's optimal decision-and-expediting plan $(\widehat{\mathbf{D}}^K, \widehat{\mathbf{E}}^K)$ given $\widehat{\mathbf{X}}^K$; the bound on the associated project length is $\underline{Z}_{\min}(\widehat{\mathbf{X}}^K)$;

(3) If ($\text{EMIN}(\widehat{\mathbf{X}}^K)$ is infeasible) set $Z_{\text{LB}} \leftarrow \infty$ and go to **End**;

/∗ Above, if the subproblem is infeasible, the proliferator cannot afford to complete the interdicted project. ∗/

(4) If ($\underline{Z}_{\min}(\widehat{\mathbf{X}}^K) > Z_{\text{LB}}$) set $Z_{\text{LB}} \leftarrow \underline{Z}_{\min}(\widehat{\mathbf{X}}^K)$ and record improved incumbent interdiction plan $\mathbf{X}^* \leftarrow \widehat{\mathbf{X}}^K$;

(5) If ($Z_{\text{UB}} - Z_{\text{LB}} \leqslant \varepsilon$) go to **End**;

(6) Append a new instance of cuts (27) and SECs (28) to DMAX^K, and attempt to solve that master problem to obtain the conditionally valid upper bound on the project length, $\bar{Z}_{\max}^K$;

/∗ The bound is valid if we have not identified the optimal solution yet, and the bound's validity is moot if we have. ∗/

(7) If (DMAX^K is infeasible) set $Z_{\text{UB}} \leftarrow Z_{\text{LB}}$ and go to **End**;

/∗ Above, the master problem could be infeasible because all solutions have been eliminated by SECs. In this case, $\mathbf{X}^*$ must be optimal. ∗/

(8) If ($\bar{Z}_{\max}^K < Z_{\text{UB}}$) set $Z_{\text{UB}} \leftarrow \bar{Z}_{\max}^K$;

(9) If ($Z_{\text{UB}} - Z_{\text{LB}} \leqslant \varepsilon$) go to **End**;

(10) Set $K \leftarrow K + 1$ and go to **Subproblem**;

(11) **End:** Print ($\mathbf{X}^*$, "is an ε-optimal solution with objective value," Z_{LB});

(12) Print ("The provable optimality gap is" $\max\{Z_{\text{UB}} - Z_{\text{LB}}, 0\}$);

(13) Halt;

6. Implementation

We have integrated the models and algorithm described above with an off-the-shelf project-management product, Microsoft Project 2003™ ("MS Project"; see Microsoft 2004), to manage data and to display results. We have implemented the algorithm with the GAMS algebraic modeling system (GAMS 2007a) using CPLEX as the optimizer (GAMS 2007b). A custom interface written in VBA (Microsoft 2003) provides the analyst with access to the model and optimizer and connects MS Project's functions with GAMS.

A 2.0 GHz Dell Inspiron 6000 computer serves as our computing platform. Each instance of MAXMIN requires less than 20 minutes to generate (via VBA and GAMS), to optimize (via CPLEX), and to return results to MS Project (via GAMS and VBA). Each subproblem EMIN comprises approximately 200 binary variables, 400 continuous variables, and 1,100 constraints. The master problem DMAX^K comprises approximately 1,200 binary variables and 1,100–1,600 constraints, depending on the number of iterations.

7. Application and Insights

Using MAXMIN, we find the best interdiction plan for a single set of input parameters and recover the resulting project plan used by the proliferator after he sees that interdiction plan. We have found instances in which multiple interdiction plans produce the same maximum expedited project length. A set of such interdiction plans can be presented to policy-makers for evaluation against secondary criteria.

Using the scenario from Harney et al. (2006) that has unlimited resources for the proliferator and no interdictions, the proliferator chooses aerodynamic enrichment and

Table 3. MAXMIN solves DCPMI for the case study, with zero to four interdictions and two different decision-and-expediting budgets for the proliferator.

Number of interdictions	Proliferator's budget = \$380 million				Proliferator's budget = \$480 million			
	Z_{LB}	Z_{UB}	Relgap (%)	Iter.	Z_{LB}	Z_{UB}	Relgap (%)	Iter.
0	260.00	260.00	0.0	1	260.00	260.00	0.0	0
1	316.00	316.00	0.0	13	312.00	316.00	1.3	24
2	350.25	350.5	0.1	54	340.00	340.12	<0.1	5
3	352.00	374.31	6.3	500	352.00	364.12	3.4	17
4	368.00	386.5	5.0	211	368.00	376.50	2.3	488

Notes. The algorithm uses a 5% relative optimality tolerance but is limited to a maximum of 500 iterations. For each problem, the table reports the objective value in weeks for the best solution found Z_{LB}, the best upper bound Z_{UB}, the relative optimality gap ("Relgap") established in the last iteration, and the number of iterations.

completes his project in 260 weeks, i.e., in a little over five years. (Incidentally, the three enrichment technologies have similar completion times when fully crashed but significantly different costs: about 196 weeks and \$108 million for gas centrifuge, 192 weeks and \$318 million for gas diffusion, and 192 weeks and \$258 million for aerodynamic.) Now, suppose that (a) the proliferator plans to complete his project as quickly as possible under the assumption that we will not interdict his plan, (b) we, as interdictor, optimally interdict assuming the proliferator will not notice our actions, and (c) the proliferator, in fact, does not notice. (That is, the proliferator solves EMIN($\mathbf{0}$) to obtain $(\widehat{\mathbf{D}}, \widehat{\mathbf{E}})$, and we solve DMAX$(\widehat{\mathbf{D}}, \widehat{\mathbf{E}})$ and do not have to worry about the proliferator adjusting his subsequent plans.) This is an optimistic situation for us, but it is instructive to analyze because it helps evaluate the importance of keeping our interdiction plan secret. By interdicting the two optimal tasks, we can extend the proliferator's project to 356 weeks (approximately 6.8 years), which amounts to a 37% increase. To achieve this delay, we interdict "cascade loading" and "acquisition of pumps and piping" for the production versions of the enrichment equipment.

MAXMIN then shows that, even if the proliferator is aware of our intention to interdict his project and which tasks we will delay, the best he can do is to complete the project in 348 weeks (approximately 6.7 years). This is still a 34% increase over the length of the project when we do nothing. This significant but not huge increase results from our modest assumptions about the achievable delays from interdictions. Even though the proliferator knows how the project will be interdicted and can compensate, and even if he chooses a different enrichment technology and completely reallocates his monetary resources, he can save only 8 weeks of the project length we achieve by interdicting with complete secrecy. Thus, we have discovered real, unavoidable fragilities in his project.

We have also evaluated a scenario in which the proliferator purchases HEU instead of producing it himself. With a large crashing budget, the proliferator now completes the uninterdicted project in 208 weeks but requires 244 weeks with two interdictions. These results contrast with the belief that some people may have, which is that possessing HEU is essentially equivalent to possessing a nuclear weapon. It is true that if the proliferator produces his own HEU, he needs only approximately $260 - 192 = 68$ weeks to complete a weapon once the HEU is in hand. But those 68 weeks are preceded by approximately $208 - 68 = 140$ weeks of work on infrastructure for exploiting that HEU, effort that goes on in parallel with its production. (The proliferator's ability to fabricate a potent radiological dispersion device, i.e., a "dirty bomb," would pose a more immediate threat; see Magill et al. 2007.)

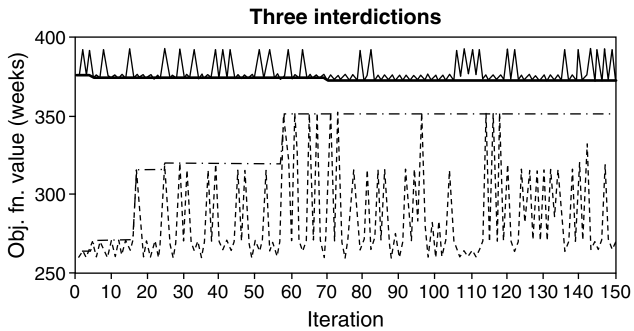
Table 3 illustrates the results of running MAXMIN for various scenarios involving from zero to four interdicted tasks, and with a decision-and-expediting budget of either \$380 million or \$480 million for the proliferator. We modify the algorithm slightly by (a) adding an iteration limit of 500 and (b) replacing the absolute optimality tolerance of ε with a relative optimality tolerance of 5% (that is, the algorithm terminates if $100\% \times (Z_{UB} - Z_{LB})/Z_{LB} \leqslant 5\%$).

The table shows that, for two interdictions, the proliferator can make some modest use of an increased budget. In both budget scenarios, the final optimality gap is less than 1%, and we see that the proliferator can reduce his project-completion time by approximately 10 weeks if he has an extra \$100 million at his disposal.

The algorithm achieves an optimality gap of at most 5% for all scenarios except the one with three interdictions allowed and a \$380 million decision-and-expediting budget. In this case, it terminates at 500 iterations with a 6.3% optimality gap. We use this most-difficult scenario to illustrate, in Figure 3, how MAXMIN progresses through its iterations when faced with a challenging problem. The figure displays lower and upper bounds on Z^*, as well as the individual master- and subproblem-solution values, for the first 150 iterations. At that point, the optimality gap is 6.4%, and the remaining 350 iterations improve only the upper bound, and only fractionally.

In every variation of the case study, the cascade-loading task in each of the alternate technologies is the most susceptible to interdiction: the three variants of this task appear in every incumbent interdiction plan found by MAXMIN. This is key information for a policy-maker or military planner.

Figure 3. Progress of the MAXMIN algorithm for 150 iterations, given three interdictions and a decision-and-expediting budget of \$380 million for the proliferator.



Notes. At each iteration, the lower, dashed line represents the subproblem objective value; the dashed line directly above represents the global lower bound set by the incumbent interdiction plan; the solid upper line represents the local upper bound $\bar{Z}_{\max}(\widehat{\mathbf{D}}, \widehat{\mathbf{E}})$ obtained from each master-problem solution; and the thicker solid line represents the global upper bound.

8. Conclusions

This paper has presented a max-min model, denoted DCPMI, for the optimal interdiction of a nuclear-weapons project. "Optimal" implies that the project is maximally delayed given limited interdiction resources. Interdictions might involve military strikes, embargoes on key materials, or sabotage of facilities.

At its inner level, DCPMI incorporates a detailed project-management model, specifically, a generalized decision-CPM model. This model shows how the nuclear-weapons developer, or "proliferator," could complete his weapons project as quickly as possible by using limited resources to complete and expedite project tasks and by using alternate technologies to achieve certain intermediate goals. The outer level of DCPMI uses interdiction resources to delay the project's tasks, with the goal being to maximize the proliferator's minimum project-completion time. We develop a special decomposition algorithm to solve DCPMI for an optimal or near-optimal interdiction plan. The algorithm extends Benders decomposition in that (a) its subproblems are difficult integer programs, not linear programs, and (b) it uses a nominally nonconvergent master-problem bound, which is augmented with solution-elimination constraints to ensure convergence.

We have developed a complete decision-support system that implements DCPMI and its solution algorithm using standard software. Computational results from a case study show that optimal or near-optimal solutions can be found for a large-scale, high-fidelity scenario in approximately 20 minutes on a personal computer. Thus, "what-if" exercises can be conducted quickly.

DCPMI is flexible and can easily adapt to a number of modeling nuances not covered in the paper. For example, it is easy to add constraints to limit the political, environmental, or economic impact of candidate interdiction plans (Reed 1994), and DCPMI can easily incorporate multiple types of interdiction resource and/or interdiction actions that affect multiple tasks. DCPMI can also be adjusted to preferentially interdict active or near-term tasks, better to achieve immediate or near-term results. This would also reduce the time that the proliferator has to detect interdictions and to make recovery plans. The model can also be modified to analyze a partially completed project, and even to allow interdictions that set back such a project by "decompleting" certain tasks. For example, a completed enrichment facility could be destroyed by a military strike, or by sabotage, and would need to be rebuilt; see Skroch (2004).

Acknowledgments

The authors thank the Air Force Office of Scientific Research and the Office of Naval Research for their research support.

References

Ahuja, R. K., T. L. Magnanti, J. B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications*. Prentice-Hall, Upper Saddle River, NJ.

Bard, J. F. 1982. An algorithm for solving the general bilevel programming problem. *Math. Oper. Res.* **8** 260–272.

Benders, J. 1962. Partitioning procedures for solving mixed integer variables programming problems. *Numerische Mathematik* **4** 238–252.

Brown, G. G., R. F. Dell, R. K. Wood. 1997. Optimization and persistence. *Interfaces* **27** 15–37.

Brown, G. G., W. M. Carlyle, J. Royset, R. K. Wood. 2005. On the complexity of delaying an adversary's project. B. Golden, S. Raghavan, E.Wasil, eds. *The Next Wave in Computing, Optimization and Decision Technologies*. Springer, New York, 3–17.

Brown, G., M. Carlyle, J. Salmeron, K. Wood. 2006. Defending critical infrastructure. *Interfaces* **36** 530–544.

Charnes, A., W. W. Cooper. 1962. A network interpretation and a directed subdual algorithm for critical path scheduling. *J. Indust. Engrg.* **13** 213–218.

Crowston, W., G. L. Thompson. 1967. Decision CPM: A method for simultaneous planning, scheduling, and control of projects. *Oper. Res.* **15** 407–426.

Danskin, J. W. 1967. *The Theory of Max-Min*. Springer-Verlag, New York.

Environmental Protection Agency. 2007. Radiation information—Plutonium. Accessed April 14, 2007, http://www.epa.gov/radiation/radionuclides/plutonium.htm/#epadoing.

Federation of American Scientists (FAS). 1998. Nuclear weapon design. Accessed October 16, 2006, http://www.fas.org/nuke/intro/nuke/design.htm.

GAMS. 2007a. GAMS homepage. Accessed January 12, 2007, http://www.gams.com/.

GAMS. 2007b. CPLEX 10. Accessed January 12, 2007, http://www.gams.com/dd/docs/solvers/cplex.pdf.

Harney, R., G. Brown, M. Carlyle, E. Skroch, K. Wood. 2006. Anatomy of a project to produce a first nuclear weapon. *Sci. Global Security* **14** 163–182.

International Atomic Energy Agency (IAEA). 1970. Informational circular, INFCIRC/140, 22 April. Accessed March 9, 2009, http://www.iaea.org/Publications/Documents/Infcircs/Others/infcirc140.pdf.

International Atomic Energy Agency (IAEA). 2005. Annual report. Accessed February 14, 2007, http://www.iaea.org/Publications/Reports/Anrep2005/anrep2005_full.pdf.

Israeli, E., R. K. Wood. 2002. Shortest-path network interdiction. *Networks* **40** 97–111.

Kelley, J. E., Jr. 1961. Critical path planning and scheduling: Mathematical basis. *Oper. Res.* **9** 296–320.

Magill, J., D. Hamilton, K. Lützenkirchen, M. Tufan, G. Tamborini, W. Wagner, V. Berthou, A. von Zweidorf. 2007. Consequences of a radiological dispersal event with nuclear and radioactive sources. *Sci. Global Security* **15** 107–132.

Malcolm, D. G., J. H. Roseboom, C. E. Clark, W. Fazar. 1959. Application of a technique for research and development program evaluation. *Oper. Res.* **7** 646–669.

Microsoft Corporation. 2003. Microsoft Visual Basic for Applications, Home Page. Accessed March 9, 2009, http://msdn.microsoft.com/en-us/isv/bb190538.aspx.

Microsoft Corporation. 2004. Microsoft Office. Accessed March 9, 2009, http://office.microsoft.com/en-us/project/FX100649021033.aspx.

Moder, J. J., C. R. Phillips, E. W. Davis. 1983. *Project Management with CPM, PERT and Precedence Diagramming*, 3rd ed. Van Nostrand Reinhold Company, New York.

Nuclear Energy Research Advisory Committee (NERAC). 2001. Technological opportunities to increase the proliferation resistance of global civilian nuclear power systems (TOPS). TOPS Task Force Report, January. U.S. Department of Energy, Washington, D.C.

Reed, B. K. 1994. Models for proliferation interdiction response analysis. Master's thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Rhodes, R. 1995. *Dark Sun*: *The Making of the Hydrogen Bomb*. Simon & Schuster, New York.

Skroch, E. M. 2004. How to optimally interdict a belligerent project to develop a nuclear weapon. Master's thesis, Operations Research Department, Naval Postgraduate School, Monterey, CA.

Spears, D., ed. 2001. Technology R&D for arms control. *Arms Control and Nonproliferation Technologies*. U.S. Department of Energy, National Nuclear Security Administration, Washington, D.C.

U.S. Congress, Office of Technology Assessment (OTA). 1993. Technologies underlying weapons of mass destruction. OTA-BP-ISC-115. U.S. Government Printing Office, Washington, D.C.