

PROJECT MANAGEMENT IN A COMPETITIVE ENVIRONMENT: INTERDICTING A CPM BASED PROJECT AND ITS IMPLICATIONS

FATİH KASIMOĞLU^{1,*} AND İBRAHİM AKGÜN²

Abstract. There are two opponents in a classic network interdiction problem, network owner/defender and interdictor/attacker. Each side has enough information about the other's possible courses of action. While the network user wishes to run the network in an optimal way, the attacker with the limited resources tries to prevent the optimal operation of the network by interdicting the arcs/nodes of the network. In this study, we investigate project management in a competitive environment using a network interdiction approach. We assume that the project owner/manager strives to minimize the completion time of a Critical Path Method (CPM) based project while an opponent attempts to maximize the minimum completion time by inflicting some delays on project activities with available interdiction resources. Considering both discrete and continuous delay times, we develop two bi-level mixed-integer programming models for the interdictor. Using duality, we then convert the bi-level models to standard single-level models, which are solvable through standard optimization packages. We extend these models to find efficient solutions in terms of project completion time and interdiction resources from the interdictor's perspective. In this respect, we develop an algorithm to find an efficient solution set for the interdictor. Next, from project manager's standpoint, we discuss the earliest and latest scheduling times of activities in case of interdiction. Finally, we apply the developed techniques in a marketing project aiming at introducing a new product. The findings may enhance a better project management in an environment where an opponent can adversely affect the project management process by delaying some activities.

Mathematics Subject Classification. 90B50, 90B10.

Received February 10, 2019. Accepted September 6, 2019.

1. INTRODUCTION

Typically, there are two opponents in a network interdiction problem; network user/defender and attacker/interdictor. Network user tries to run the network in an optimal way while the interdictor attempts to prevent the optimal operation of the network by attacking on the arcs in some certain ways using limited interdiction resources (*e.g.*, aerial sorties, missiles). An interdiction model can be considered as a game played by two players consecutively, in which each player has enough information about the other's strategy and chooses the

Keywords. Bi-level programming, project management, project scheduling, network modeling, network interdiction, integer programming.

¹ Turkish Military Academy Defense Sciences Institute, 06654 Bakanlıklar, Ankara, Turkey.

² Abdullah Gul University, Department of Industrial Engineering, 38080 Kocasinan, Kayseri, Turkey.

*Corresponding author: ffatihkasimoglu@gmail.com

best course of action at each move. Such a successively played two-person zero-sum game is called a Stackelberg game [41, 45] and is usually modeled with bi-level linear programming (BLP) techniques. Wen and Hsu [47] and Kalashnikov *et al.* [29] provide a detailed information about BLP procedure.

In this study, we address the interdiction problem in the context of project management where the network user runs a Critical Path Method (CPM) based project while an opponent tries to disrupt the project by extending the duration of some activities using limited resources. The project manager (PM) as the network user is to complete the project in minimum time whereas the interdictor tries to maximize the minimum completion time.

The study of network interdiction dates back to 1960s. Wood [52] gives a chronological evolvment of interdiction models. The initial studies related to the topic emerge in maximum flow problems, a well-studied area in which interdictor tries to minimize maximum flow. In this regard, the research made by Ford and Fulkerson [20] exploits the maximum flow-minimal cut theory. Wollmer [49, 50], Ratliff *et al.* [36], and Lubore *et al.* [31] focus on the most important arc(s) in the network. Some other studies that provided remarkable contribution to this area are McMasters and Mustin [34], Ghare *et al.* [23], Helmbold [25], and Wood [51]. Being a milestone in the field, Wood [51] develops a more generic and flexible approach for the solution of the problem, whereas all the other prior studies are specific to a certain application. Wood [51] formulates the problem as a bi-level min-max model and then transforms it into a single-stage mixed-integer programming model. Akgün *et al.* [2] generalizes the max-flow interdiction problem to multiple terminals. Cormican [16] and Cormican *et al.* [17] are among the first studies presenting a stochastic version of the problem. Janjarassuk and Nakrachata [28] develops a simulated annealing algorithm for stochastic network interdiction. Bertsimas *et al.* [7] introduces randomized network interdiction problem in which interdictor uses randomness in choosing the arcs to be removed from the network. Zhang and Fan [53] develops an interdiction model for multicommodity networks. Royset and Wood [38] study the bi-objective version of max-flow network interdiction problem in which minimizing maximum flow and minimizing total interdiction cost are treated as two distinct objectives.

Another category of network interdiction, which has also attracted a lot of attention, is maximizing the shortest path in which an interdictor attacks on arcs in order to maximize the length of the shortest path. Fulkerson and Harding [21], Golden [24], Corley and Shaw [15], Malik *et al.* [33], Israeli [26], and Israeli and Wood [27] have notable contributions to the field in this respect. Rocco and Ramirez [37] use a bi-objective approach in which maximizing shortest path and minimizing interdiction cost are two separate objectives. In a more recent study Wei *et al.* [46] develops a solution procedure taking into account some threshold values for the shortest path.

Having been studied with either a deterministic or stochastic approach, the two main streams in the literature, network interdiction problem has found application in many other different fields. As for some interesting examples to mention, Assimakopoulos [4] applies it in control of hospital infections, Anandalingam and Apprey [3] in conflict resolution problems, Church *et al.* [14] in identifying most critical facility assets in a service/supply system, Salmeron *et al.* [39, 40] and Kim *et al.* [30] in analysis of electric grid security, Brown *et al.* [8] in positioning of defensive missile interceptors, and Brown *et al.* [10] in defending critical infrastructure. In more recent studies Fang *et al.* [19] uses the concept in water resources allocation, Camacho-Vallejo *et al.* [12] and Ashtiani *et al.* [5] in facility location and Lusby *et al.* [32] in railway transportation problems. In another notable study, Sundar *et al.* [43] introduces a system reliability modeling approach and considers probabilistic generalization of the $N-k$ failure-identification problem in power transmission networks where the goal is to find a set of k components that maximizes system disfunction. Chestnut and Zenklusen [13] and Disser and Matuschke [18] deal with the hardness and complexity of the problem.

Even though the literature in network interdiction is rich, the studies relating to interdiction of a project network is very limited. Actually, we have only come by two studies in this area. Using an activity on arc (AoA) representation for the project network, Brown *et al.* [9] provide an analysis on the complexity of project interdiction models and shows that the problem is NP-hard. In another study, Brown *et al.* [11] develop an interdiction model to delay an enemy's CPM-based nuclear weapons project and use a decomposition technique to solve the problem. In both of these studies, discrete delay times of activities are considered and inflicted

TABLE 1. Main contributions to the field.

| Contribution | Remarks |
|--|--|
| Partial interdiction | The idea of partial interdiction of a project activity is introduced in comparison with binary interdiction |
| Efficient solutions for interdictor | An algorithm to find efficient solutions for interdictor in terms of project completion time and resource usage is developed for the first time using a bi-objective approach |
| Implications of interdiction on project scheduling | Bringing in the Project Manager's perspective in a project interdiction problem, a new scheme for project scheduling (<i>i.e.</i> , earliest and latest start times of activities) and critical path has been presented |
| Practical CPM project interdiction models | Applicable CPM project interdiction models that can be easily solved through standard optimization packages have been made available for the convenience of researchers and practitioners trying to achieve a better project management in a competitive environment |

upon an activity on a binary basis. That is, an activity is either delayed for a certain amount of time or not, a case we call binary interdiction. Partial interdiction of an activity is not considered. The interdiction models in these studies use a single objective, which is maximizing project completion time from the interdictor's perspective, but do not take into account the interdictor's effective resource usage. In this regard, there may occur solutions that unnecessarily use more resources. Moreover, the implications of interdiction from PM's perspective, especially on project scheduling, are not addressed.

In this study, using an activity on node (AoN) network representation for the project network and adopting a deterministic modeling approach, we develop two bi-level max–min interdiction models. In the first model, we consider discrete delay times, *i.e.*, binary interdiction case. In the second model, we investigate continuous delay times allowing interdictor to partially delay an activity within an allowable range. We use duality in order to convert the bi-level models into single-level models, which are solvable through standard optimization packages. We extend both models to generate efficient solutions in terms of maximum project completion time and minimum resource usage from interdictor's perspective. In this respect, we present an algorithm to find an efficient interdiction solution set. We also discuss implications of interdiction on project scheduling from PM's standpoint once the project is interdicted. Finally, we apply our models in a project to introduce a new product to the market. The study fills a gap in project network interdiction introducing some new concepts and bringing a new dimension to project management in a competitive environment as summarized in Table 1.

The remainder of the paper is organized as follows. In Section 2, we introduce the problem with the associated assumptions and notation. In Section 3, we develop our project interdiction models. In Section 4, we extend the interdiction models to find efficient solutions and provide an algorithm to find efficient solution sets. In Section 5, we discuss the problem from PM's perspective to schedule activities in case of interdiction. In Section 6, we provide an application and computational study for the developed models. In Section 7, we conclude the paper.

2. PROBLEM ASSUMPTIONS AND NOTATION

Suppose that there are two opponents in a project management process; namely, the PM who tries to complete a project in minimum time and the interdictor who attempts to deliberately delay the project as much as possible with limited total interdiction resources.

PM uses a CPM-based project network to schedule the activities of the project. Suppose that there are n activities in the project and a standard finish-to-start relation is defined between activity i and activity j . Activity i is said to be an immediate predecessor of activity j and unless activity i is completed, activity j cannot start. Activity 0 and activity $n + 1$ are dummy activities only used to denote the start and completion of the project respectively. Time needed for activity i is denoted by t_i with the dummy activities having $t_i = 0$.

Let $G = (N, A)$ be a directed graph denoting project network with AoN representation where N represents the node set (activities) and A represents set of activity pairs having immediate predecessor relationship.

The interdictor has the capability to obtain information about the PM's project network (through intelligence or some other means) and can interdict the project using some interdiction resources, the total of which is denoted with R (budget).

The following Stackelberg Game assumptions apply for the problem [42, 52]:

- The interdictor acts first to achieve his/her goal before the project starts.
- The game is played in a sequential manner.
- Each time, players choose their optimal courses of action.

Throughout the study the notation that we use in general is as follows.

Indices and sets

- i, j Activities $0, 1, 2, \dots, (n + 1)$.
- N Node set (activities).
- A Arc set (activity pairs having immediate predecessor relationship) $\{(0, 1), (0, 2), \dots, (n, n + 1)\}$.

Parameters

- t_i Time needed for activity i .
- d_i Delay time inflicted on activity i by interdictor.
- r_i Resource needed by interdictor to delay activity i .
- r'_i Resource needed by interdictor to delay activity i by one-unit time.
- \bar{d}_i Allowable delay upper bound for activity i .
- R Total interdiction resource available.

Decision variables

- s_i Start time for activity i (for project manager).
- y_i Binary variable for interdictor (1 if activity i is interdicted; 0 otherwise).
- y'_i Amount of delay time to inflict on activity i by interdictor.
- w_{ij} Dual variable associated with arc $(i - j)$.

3. FORMULATING CPM-BASED PROJECT INTERDICTION MODELS

In formulation of our models we consider two main cases for the interdictor, one with discrete delay times and the other with continuous delay times. In the former case, the interdictor either inflicts a certain delay time, d_i , on activity i or not, thus we call this case as also binary interdiction case. The resource needed (cost) to inflict d_i on activity i is denoted by r_i .

In the latter one, activity delay times are continuous and the interdictor has the opportunity to partially delay an activity up to an upper delay bound, \bar{d}_i , designated for each activity i . How much an activity is delayed is a variable depending on the amount of resource allocated for this purpose. The relationship between the interdiction resource usage and the activity times is linear. As an example we depict this kind of relationship in Figure 1. The interdictor does not pay any cost if he/she decides not to delay an activity but pays a linearly increasing price in accordance with the length of delay that he/she wants to impose on an activity.

Let r'_i be the amount of resource needed (cost) to inflict one-unit delay on activity i . We introduce following parameters to describe r'_i more formally.

- TN_i : normal activity time for activity i .
- TI_i : activity time when maximum interdiction (delay) is inflicted on activity i .
- \bar{d}_i : upper bound of delay for activity i ($\bar{d}_i = TI_i - TN_i$).
- CNI_i : normal cost of activity i for interdictor (this value is normally 0, since the interdictor does not pay anything if he chooses not to delay the activity and allows its completion in a normal way).
- CI_i : the cost of maximally interdicting activity i for interdictor.

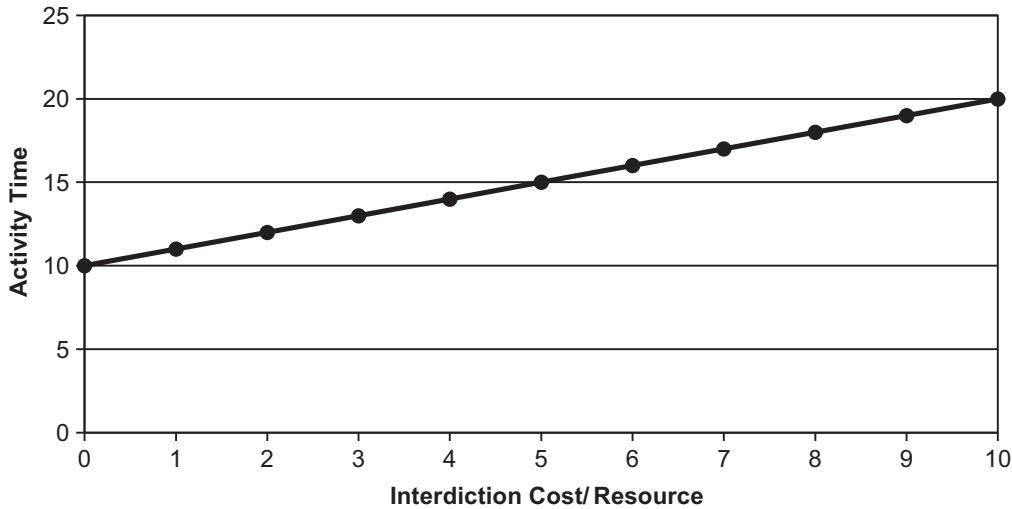


FIGURE 1. Relationship between interdiction cost and activity times.

Thus, $r'_i = (C_i - CNI_i)/(TI_i - TN_i)$, which gives actually the slope in Figure 1.

Note that a similar type of relationship (*i.e.*, a linearly decreasing one) is used in well-known project crashing, in which PM tries to expedite the process by paying additional cost to complete an activity in a shorter time.

In this regard, our purpose is to develop interdiction models to find out interdictor’s best course of action for both of our cases. In doing that, we first give the PM’s model to find minimum project completion time and then build up associated interdiction models.

3.1. PM’s CPM-based model to find minimum project completion time and its dual

Let s_i be the decision variable that indicates the start time of activity i . The PM is to solve the following linear programming CPM model, LPCPM, to find the minimum project completion time and the start time of each activity i .

Model LPCPM: PM’s CPM model to find minimum project completion time

$$\text{Min } s_{n+1} \tag{3.1}$$

$$s_j - s_i \geq t_i \quad \forall (i, j) \in A \quad w_{ij} \tag{3.2}$$

$$s_i \geq 0 \quad \forall i. \tag{3.3}$$

In LPCPM, objective function (3.1) minimizes the project completion time denoted by s_{n+1} . In other words, it finds the earliest possible time to finish the project. Constraints (3.2) define the relationship between the activities having immediate predecessor relationship. That is, activity j can only start after activity i is completed. Constraints (3.3) define decision variables.

Defining w_{ij} as the dual variable associated with constraint set (3.2), we give the dual model of the LPCPM, D-LPCPM.

Model D-LPCPM: The dual model of LPCPM

$$\text{Max } \sum_{(i,j) \in A} t_i w_{ij} \tag{3.4}$$

$$\sum_{j|(i,j) \in A} w_{ji} - \sum_{j|(i,j) \in A} w_{ij} \leq 0 \quad i \neq (n+1) \tag{3.5}$$

$$\sum_{j|(i,j) \in A} w_{ji} \leq 1 \quad i = (n+1) \tag{3.6}$$

$$w_{ij} \geq 0 \quad \forall (i, j) \in A. \tag{3.7}$$

In D-LPCPM, objective function (3.4) gives the project completion time. Constraints (3.5) and (3.6) are dual constraints associated with the primal decision variables s_i . Constraints (3.7) define dual variables.

It is known that the dual of a project network model is equivalent to the longest path network model (Ahuja *et al.* [1], pp. 733, 734). The objective function (3.4) is just the length of longest (critical) path. Even though w_{ij} is defined as a positive real variable, it takes the value of 1 or 0. By definition, the value of a dual variable gives the rate of change in the objective function value when the associated right-hand-side value in the primal model is increased one unit (Bazaraa *et al.* [6], pp. 266, 267). Thus, when we increase a right-hand-side value by one unit in constraints (3.2), the associated w_{ij} value will be either 1 or 0 depending on whether it is on critical path or not. That is, by increasing the right-hand-side value by one unit in constraints (3.2), we increase the duration of an activity either on the critical path or on a non-critical path. Adding one unit to the length of the critical path, means increasing the objective function value by one, whereas adding one unit to the length of a non-critical path does not change the objective function value.

3.2. Interdictor’s models

The interdictor’s problem can be formulated as a bi-level programming model. In this section, we first develop bi-level, max–min type interdiction models for both binary and partial interdiction cases and then convert the models to single-level models by using duality and variable transformations.

3.2.1. Interdiction with discrete delay times

Defining y_i as the binary decision variable of the interdictor (1 if activity i is interdicted; 0 otherwise), the interdictor’s bi-level model for discrete delay times, CPMI-DDT, can be given as follows.

Model BI-CPMI-DDT: Bi-level CPM-based interdiction model with discrete delay times

$$\text{Max}_{y_i \in \Omega} \text{Min } s_{n+1} \tag{3.8}$$

$$s_j - s_i \geq t_i + d_i y_i \quad \forall (i, j) \in A \quad w_{ij} \tag{3.9}$$

$$s_i \geq 0 \quad \forall i \tag{3.10}$$

where

$$\Omega \equiv \left\{ y_i \mid y_i \in \{0, 1\} \quad \forall i, \sum_i r_i y_i \leq R \right\}. \tag{3.11}$$

In BI-CPMI-DDT, the inner part belongs to PM, whereas the outer model belongs to the interdictor. In the objective function (3.8), PM tries to minimize project completion time, s_{n+1} , while the interdictor attempts to maximize it with his/her decision plan, vector \mathbf{y} . Note that vector \mathbf{y} is a combination of y_i ’s, which is a subset of set of all possible courses of action, Ω , defined by (3.11). (3.11) consists of y_i ’s that satisfy interdiction resource constraint, $\sum_i r_i y_i \leq R$. Constraints (3.9) require that a delay d_i be imposed on the activity duration if the activity is interdicted.

In BI-CPMI-DDT, the interdictor searches for a combination of y_i ’s that solve the model optimally. Theoretically, this is possible if we define all the possible combinations of y_i ’s and choose the best one among them. However, this means to solve a model with n activities around 2^n times, which is hardly possible even for projects with reasonable number of activities. In this regard, next we present how the model can be converted into a model solvable through off-the-shelf software.

We can transform the model BI-CPMI-DDT into an integer programming (IP) model using an approach similar to that of Wood [51]. Let w_{ij} be the dual variable corresponding to the constraints (3.9). Fixing y_i 's in BI-CPMI-DDT and then taking the dual of the inner model, we obtain the following non-linear max-max model, which is essentially a maximization model.

Model NLP-CPMI-DDT: Non-linear CPM-based interdiction model with discrete delay times

$$\text{Max}_{y_i \in \Omega} \text{Max} \sum_{(i,j) \in A} (t_i + d_i y_i) w_{ij} \quad (3.12)$$

$$\sum_{j|(i,j) \in A} w_{ji} - \sum_{j|(i,j) \in A} w_{ij} \leq 0 \quad i \neq (n+1) \quad (3.13)$$

$$\sum_{j|(i,j) \in A} w_{ji} \leq 1 \quad i = (n+1) \quad (3.14)$$

$$w_{ij} \geq 0 \quad \forall (i,j) \in A \quad (3.15)$$

where

$$\Omega \equiv \left\{ y_i \mid y_i \in \{0,1\} \quad \forall i, \quad \sum_i r_i y_i \leq R \right\}. \quad (3.16)$$

NLP-CPMI-DDT is a nonlinear MIP model, which is difficult to solve due to nonlinear objective function (3.12). However, it can be linearized by using the observation that the variables w_{ij} take only values of 0 and 1.

Let $\gamma_{ij} = y_i w_{ij}$. Because y_i and w_{ij} are both binary variables, γ_{ij} is also a binary variable. If either y_i or w_{ij} takes the value of 0, γ_{ij} is 0. When both y_i and w_{ij} takes the value of 1, γ_{ij} is 1. This situation can be expressed by the following constraints.

$$\gamma_{ij} \leq y_i \quad \forall i, j \quad (3.17)$$

$$\gamma_{ij} \leq w_{ij} \quad \forall i, j \quad (3.18)$$

$$\gamma_{ij} \geq (y_i + w_{ij} - 1) \quad \forall i, j \quad (3.19)$$

By setting $\gamma_{ij} = y_i w_{ij}$ and adding constraints (3.17)–(3.19), we obtain the following IP model, CPMI-DDT.

Model CPMI-DDT: CPM-based interdiction model with discrete delay times

$$\text{Max} \sum_{(i,j) \in A} (t_i w_{ij} + d_i \gamma_{ij}). \quad (3.20)$$

In addition to (3.17)–(3.19),

$$\sum_{j|(i,j) \in A} w_{ji} - \sum_{j|(i,j) \in A} w_{ij} \leq 0 \quad i \neq (n+1) \quad (3.21)$$

$$\sum_{j|(i,j) \in A} w_{ji} \leq 1 \quad i = (n+1) \quad (3.22)$$

$$\sum_i r_i y_i \leq R \quad (3.23)$$

$$w_{ij} \in \{0,1\} \quad \forall i, j \quad (3.24)$$

$$y_i \in \{0,1\} \quad \forall i \quad (3.25)$$

$$\gamma_{ij} \in \{0,1\} \quad \forall i, j. \quad (3.26)$$

In CPMI-DDT, linearized objective function (3.20) is still the project completion time. Constraints (3.21) and (3.22) are the dual constraints. Constraint (3.23) defines the total amount of resource the interdictor has for interdiction. Constraints (3.24)–(3.26) define decision variables.

3.2.2. Interdiction with continuous delay times

Defining y'_i as the decision variable of the interdicator indicating the amount of delay time to inflict on activity i , the interdicator’s bi-level model for continuous delay times, BI-CPMI-CDT, can be given as follows.

Model BI-CPMI-CDT: Bi-level CPM-based interdiction model with continuous delay times

$$\text{Max}_{y'_i \in \Omega} \text{Min } s_{n+1} \tag{3.27}$$

$$s_j - s_i \geq \text{TN}_i + y'_i \quad \forall (i, j) \in A \tag{3.28}$$

$$s_i \geq 0 \quad \forall i \tag{3.29}$$

where

$$\Omega \equiv \left\{ y'_i \mid y'_i \geq 0 \quad \forall i, \quad \sum_i r'_i y'_i \leq R, y'_i \leq \bar{d}_i \forall i \right\}. \tag{3.30}$$

BI-CPMI-CDT is similar to BI-CPMI-DDT except that the interdiction variable y'_i is defined as a continuous variable. In the objective function (3.27), PM tries to minimize project completion time, s_{n+1} , while the interdicator attempts to maximize it with his/her decision plan, vector \mathbf{y}' . \mathbf{y}' is a combination of y'_i 's, which is a subset of interdicator’s decision space Ω defined by (3.30). In (3.30), bounded by an upper limit \bar{d}_i , y'_i has to satisfy interdiction resource constraint $\sum_i r'_i y'_i \leq R$. Note also that y'_i in constraints (3.28) exists as the associated partial delay added to the normal time of an activity i (TN_i).

We can convert the bi-level max–min model into a single-level model by using an approach similar to the one in Section 3.2.1. Fixing y'_i in (3.28) and then taking the dual of the inner model, we get the following model.

Model NLP-CPMI-CDT: Non-linear CPM-based interdiction model with continuous delay times

$$\text{Max}_{y'_i \in \Omega} \text{Max} \sum_{(i,j) \in A} (\text{TN}_i + y'_i) w_{ij} \tag{3.31}$$

$$\sum_{j|(i,j) \in A} w_{ji} - \sum_{j|(i,j) \in A} w_{ij} \leq 0 \quad i \neq (n+1) \tag{3.32}$$

$$\sum_{j|(i,j) \in A} w_{ji} \leq 1 \quad i = (n+1) \tag{3.33}$$

$$w_{ij} \geq 0 \quad \forall (i, j) \in A \tag{3.34}$$

where

$$\Omega \equiv \left\{ y'_i \mid y'_i \geq 0 \forall i, \sum_i r'_i y'_i \leq R, y'_i \leq \bar{d}_i \forall i \right\}. \tag{3.35}$$

The resulting NLP-CPMI-CDT is a nonlinear model and can be linearized as follows.

Let $\gamma_{ij} = y'_i w_{ij}$. Remember that w_{ij} only takes the value of 0 or 1. Thus, when w_{ij} takes the value of 0, γ_{ij} is 0. When w_{ij} takes the value of 1, γ_{ij} is equal to y'_i . In the latter case, since the objective function is maximization, γ_{ij} has to be as big as possible. M being a big positive number, the mentioned situation can be described by constraints (3.36) and (3.37).

$$\gamma_{ij} \leq M w_{ij} \quad \forall i, j \tag{3.36}$$

$$\gamma_{ij} \leq y'_i \quad \forall i, j. \tag{3.37}$$

By setting $\gamma_{ij} = y'_i w_{ij}$ and adding constraints (3.36) and (3.37), we obtain the following MIP model.

Model CPMI-CDT: CPM-based interdiction model with continuous delay times

$$\mathbf{Max} \sum_{(i,j) \in \mathbf{A}} (TN_i w_{ij} + \gamma_{ij}). \quad (3.38)$$

In addition to (3.36) and (3.37),

$$\sum_{j|(i,j) \in \mathbf{A}} w_{ji} - \sum_{j|(i,j) \in \mathbf{A}} w_{ij} \leq 0 \quad i \neq (n+1) \quad (3.39)$$

$$\sum_{j|(i,j) \in \mathbf{A}} w_{ji} \leq 1 \quad i = (n+1) \quad (3.40)$$

$$\sum_i r'_i y'_i \leq R \quad (3.41)$$

$$y'_i \leq \bar{d}_i \quad \forall i \quad (3.42)$$

$$y'_i \geq 0 \quad \forall i \quad (3.43)$$

$$\gamma_{ij} \geq 0 \quad \forall i, j \quad (3.44)$$

$$w_{ij} \in \{0, 1\} \quad \forall i, j. \quad (3.45)$$

In CPMI-CDT, linearized objective function (3.38) gives the maximum project completion time in case of interdiction. Constraints (3.39) and (3.40) are dual constraints obtained after fixing y'_i in (3.28). Constraint (3.41) limits total interdiction resource available to the interdictor. Constraints (3.42) set an upper bound on the possible delay for each activity. Constraints (3.43)–(3.45) define decision variables.

Remember that the dual of CPM model is equivalent to longest path model. In both interdiction models CPM-DDT and CPM-CDT, what is done is essentially finding longest path, the length of which is the project completion time. However, note that arc lengths do not have fixed values as they are variables. Thus, the problem turns out to finding the longest path after some activities are delayed for an amount of time to be decided.

4. EFFICIENT SOLUTIONS FOR INTERDICTION MODELS: A BI-OBJECTIVE APPROACH

The proposed interdiction models CPMI-DDT and CPMI-CDT may produce solutions in which the interdiction resources are unnecessarily used. Having minimum project completion times as the single objective, these models do not take into account efficient resource usage. As long as the associated resource constraints (3.23) and (3.41) are satisfied, the models may end up with any feasible amount of resource usage, which may result in wasting the interdiction resources. To get over this deficiency, we adopt a bi-objective approach from the interdictor's perspective and extend the models CPMI-DDT and CPMI-CDT to guarantee efficient solutions for the interdictor in terms of project completion time and resource usage.

To extend CPMI-DDT, we introduce objective function (4.1) that represents the total amount of resource used by the interdictor.

$$\mathbf{Min} \sum_i r_i y_i \equiv \mathbf{Max} - \sum_i r_i y_i. \quad (4.1)$$

The interdictor is to maximize both objective functions (3.20) and (4.1) simultaneously. We can prioritize (weight) and combine (3.20) and (4.1) to get a single objective function (Winston [48], pp. 194–197) as given in (4.2).

$$\mathbf{Max} \sum_{(i,j) \in \mathbf{A}} (t_i w_{ij} + d_i \gamma_{ij}) - \varepsilon \sum_i r_i y_i \quad (4.2)$$

In (4.2), ε is an arbitrarily small positive number to describe resource usage as a second prioritized objective. In this regard, after project completion time is maximized, the resource usage is minimized. In fact, a similar approach is commonly used to find earliest and latest start time of activities for PM.

The resulting model to find efficient solutions, E-CPMI-DDT, can be stated as follows.

Model E-CPMI-DDT: CPM-based interdiction model with discrete delay times to find efficient solutions

Objective function (4.2)

Constraints (3.17)–(3.19), (3.21)–(3.26).

Similarly, to extend CPMI-CDT, we introduce the objective function (4.3) to minimize total resource usage.

$$\text{Min} \sum_i r'_i y'_i \equiv \text{Max} - \sum_i r'_i y'_i. \quad (4.3)$$

This time the interdicator is to maximize the objective functions (3.38) and (4.3) simultaneously. The combined objective function is given in (4.4).

$$\text{Max} \sum_{(i,j) \in A} (\text{TN}_i w_{ij} + \gamma_{ij}) - \varepsilon \sum_i r'_i y'_i \quad (4.4)$$

The resulting efficient solution model, E-CPMI-CDT, follows next.

Model E-CPMI-CDT: CPM-based interdiction model with continuous delay times for efficient solutions

Objective function (4.4)

Constraints (3.36) and (3.37), (3.39)–(3.45).

Efficient solution models obtained in this section, E-CPMI-DDT and E-CPMI-CDT can be considered as more generalized versions of models obtained in Section 3, CPMI-DDT and CPMI-CDT respectively and are superior to them in that they provide an efficient solution.

In multiple objective decision making, an efficient (or pareto-optimal) solution is a solution in which no improvement is possible in an objective without inflicting a disimprovement (deterioration) in one or more other objectives (Tabucanon [44], pp. 10–13). E-CPMI-DDT and E-CPMI-CDT guarantee an efficient solution from the interdicator's perspective in terms of resource usage and project completion time. However, each model only gives one single efficient solution in accordance with the designated total interdiction resource R . It would be beneficial for the interdicator to know the effect of more or less interdiction resource on the project completion time and thus to figure out efficient solution set. The following algorithm to find an efficient interdiction solution set, EISS, achieves this.

Algorithm EISS: Efficient interdiction solution set finding algorithm

- Δr : Increment of interdiction resource at each iteration.
- Iter : Number of iterations (1,2,3, ...).
- RR_{iter} : Total amount of interdiction resource used at each iteration.
- $s_{(n+1, \text{iter})}$: Project completion time at each iteration.
- SS_{iter} : Solution set obtained at each iteration, $SS_{\text{iter}} = \{(RR_0, s_{(n+1,0)}) \cup (RR_1, s_{(n+1,1)}) \cup \dots\}$.
- Start** : Read the data belonging to model E-CPMI-DDT/E-CPMI-CDT
iter = 0, $R = 0$, Δr = a reasonable integer number defined by user,
 $SS_{\text{iter}} \leftarrow \emptyset$, $s_{(n+1, \text{iter})} = 0$

- Step-1** : Solve E-CPMI-DDT/E-CPMI-CDT
 $RR_{\text{iter}} = \sum_i r_i y_i$ (applies for E-CPMI-DDT)
 $RR_{\text{iter}} = \sum_i r'_i y'_i$ (applies for E-CPMI-CDT)
- Step-2** : If $s_{n+1} = s_{(n+1, \text{iter})}$ then go to end (Step-4).
 $s_{(n+1, \text{iter})} = s_{n+1}$
 $SS_{\text{iter}} = SS_{\text{iter}} \cup (RR_{\text{iter}}, s_{(n+1, \text{iter})})$
- Step-3** : $R = R + \Delta r$
 $\text{iter} = \text{iter} + 1$
 go to Step-1.
- Step-4** : End

Algorithm EISS solves E-CPMI-DDT/E-CPMI-CDT iteratively. At each iteration, R is incremented by Δr . The algorithm terminates when there is no improvement in the project completion time even though the interdiction resource is increased by Δr . It allows us to see the relationship between project completion time and interdiction cost and hence gives a better idea about the worst and best case scenarios as well as the intermediate ones related to the completion time of the project.

So far, the problem is addressed from the interdictor's perspective. PM may also use the results to find critical activities that requires specific attention in case of interdiction aiming to increase his/her situational awareness and take necessary precautions. In the next section, we specifically address a need for PM, how to schedule the activities in case of interdiction.

5. EARLIEST START TIMES (ESTs)/LATEST START TIMES (LSTs) OF ACTIVITIES IN CASE OF INTERDICTION

Once the project is interdicted, the project completion time gets longer and the activity scheduling times need to be calculated accordingly. From a scheduling perspective, it is important to find the earliest and latest start times of the activities. This provides flexibility to the PM in planning project activities. Earliest start time (EST) of an activity is the soonest time an activity can be started depending on the completion of its predecessors. Likewise, latest start time (LST) of an activity is the latest time an activity can be started without causing a delay in the minimum project completion time. The difference between EST and LST of an activity is called a slack. PM can schedule an activity between EST and LST without causing a delay in minimum project completion time. However, there is no slack time for an activity on a critical path, which means that associated EST/LST values and activity start times (s_i) are equal to each other (Meredith and Mantel [35], pp. 396–399). Thus, an activity on a critical path has to start exactly on s_i obtained through LPCPM so that the project finalizes in minimum completion time.

We can find ESTs and LSTs by solving LPCPM-EST and LPCPM-LST obtained by replacing the objective function (3.1) in LPCPM with (5.1) and (5.2), respectively, which is commonly used in project scheduling. ε is an arbitrarily small positive number as before. The given structure first guarantees minimum project completion time and then finds ESTs/LSTs.

Model LPCPM-EST: PM's CPM model to find ESTs

$$\text{Min } s_{n+1} + \varepsilon \sum_i s_i. \quad (5.1)$$

Along with the constraints (3.2) and (3.3).

Model LPCPM-LST: PM's CPM model to find LSTs

$$\text{Min } s_{n+1} - \varepsilon \sum_i s_i. \quad (5.2)$$

Along with the constraints (3.2) and (3.3).

The aforementioned models can also be used to find earliest and latest times of the activities when the project is interdicted. However, as given in (5.3) and (5.4), activity times need to be updated in accordance with the best decision plan of interdictor, vector y^* /vector y'^* , obtained through models CPMI-DDT/CPMI-CDT respectively.

$$t_i = t_i + d_i y_i^* \quad \forall i \tag{5.3}$$

$$TN_i = TN_i + y_i'^* \quad \forall i. \tag{5.4}$$

Having made these adjustments and using LPCPM-EST/LPCPM-LST, PM can find ESTs/LSTs to plan project activities as well as the new critical path in case of interdiction.

6. APPLICATION AND COMPUTATIONAL STUDY

In this section, we present an application in a marketing project and then give a computational study for projects in different sizes (*i.e.*, different number of activities).

6.1. Application in a marketing project

6.1.1. Problem setting

A company decides to initiate a project in order to introduce a new product in marketplace in a competitive environment. The project activities, activity durations, and immediate predecessor relationships are given in Table 2. AoN representation of the network is given in Figure 2.

It is a common practice in business that the project activities are undertaken by some subcontractors in accordance with an agreement made between the subcontractor and the company that owns the project as it is the case in our problem. It is also a fact that a new product in the market means sharing the market with other companies and loss in revenues of the rival companies already at the stage marketing the same kind of product.

It is quite possible that a cunning rival company might try to delay any such project as much as possible with its available resources in an effort to keep its own profit at the highest level. The rival can achieve this by manipulating the process with either an overt or covert agreement with the subcontractors taking part in the project activities so that the subcontractors do not fulfill their commitments with respect to the project on time and a delay is incurred. Even though for a subcontractor this means a failure in commitment to the project, it is not an unusual course of action in a competitive environment as long as the money offered by the rival is satisfactory enough and the punishment in case of not fulfilling the project commitment is affordable.

We describe five cases for our application. Case 0 is the one in which there is no interdiction. Cases 1–3 are interdiction cases in which the interdictor has the opportunity to delay the project with discrete delay times.

TABLE 2. Details of a project for marketing a new product.

| Activity | Immediate predecessor | Duration (in weeks) |
|---|-----------------------|---------------------|
| a(1)-Designing the product | – | 7 |
| b(2)-Market research | – | 10 |
| c(3)-Define appropriate store sites | b | 7 |
| d(4)-Produce prototype | a | 8 |
| e(5)-Supply raw material | a | 6 |
| f(6)-Set up for mass production | e | 5 |
| g(7)-Delivering the product to stores | c, d, f | 10 |
| h(8)-Advertising campaign for the product | e | 11 |

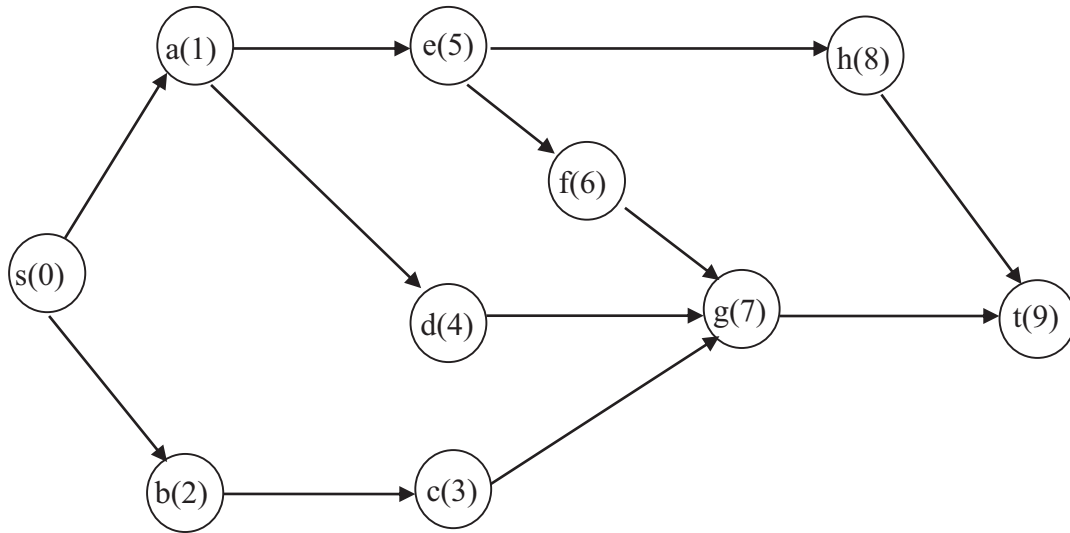


FIGURE 2. AoN representation of the project for marketing a new product.

TABLE 3. Adversary's data for 3 interdiction cases with discrete delay times.

| Activity | Delay/Interdiction (in weeks) (d_i) | | | Resource needed for Delay/Interdiction (r_i) | | |
|----------|--|--------|--------|---|--------|--------|
| | Case-1 | Case-2 | Case-3 | Case-1 | Case-2 | Case-3 |
| a(1) | 1 | 1 | 1 | 1 | 1 | 1 |
| b(2) | 1 | 3 | 3 | 1 | 1 | 5 |
| c(3) | 1 | 2 | 2 | 1 | 1 | 3 |
| d(4) | 1 | 1 | 4 | 1 | 1 | 2 |
| e(5) | 1 | 1 | 1 | 1 | 1 | 4 |
| f(6) | 1 | 1 | 2 | 1 | 1 | 3 |
| g(7) | 1 | 1 | 3 | 1 | 1 | 2 |
| h(8) | 1 | 1 | 6 | 1 | 1 | 5 |
| | | | | Total amount of resource | | |
| | | | | 5 | 5 | 15 |

CPMI-DDT/E-CPMI-DDT is used to solve these cases. Case 4 is the case in which the interdictor delays the project with continuous delay times. CPMI-CDT/E-CPMI-CDT is used to solve Case 4.

Table 3 provides data for Cases 1–3 with different d_i and/or r_i values. Notice that d_i and r_i values are 1 for all activities in Case 1. In Case 2, d_i values are slightly different from those of Case 1 while r_i values remain the same as in Case 1. In Case 3, d_i and/or r_i values are remarkably different from those in Cases 1 and 2. The total amount of resource is 5 units in Cases 1 and 2 while it is 15 units in Case 3.

The associated data for Case 4 is given in Table 4. Note that Case 4 is an adaptation of Case 3 to partial interdiction occasion.

We $\varepsilon = 0,01$, in all cases. The models are run on a PC with 1.8 GHz Intel Core 2 Duo processor and 2 GB of RAM by using GAMS/CPLEX Version 10 [22].

TABLE 4. Adversary’s data for interdiction with continuous delay times (Case 4).

| Activity | Normal time (in weeks) (TN _{<i>i</i>}) | Interdicted time (in weeks) (TI _{<i>i</i>}) | Interdicted cost (CI _{<i>i</i>}) | Unit interdiction cost ($r'_i = \frac{C_i - \text{CNI}_i^*}{\text{TI}_i - \text{TN}_i}$) |
|----------|---|--|---|---|
| a(1) | 7 | 8 | 1 | 1 |
| b(2) | 10 | 13 | 5 | 1.66 |
| c(3) | 7 | 9 | 3 | 1.5 |
| d(4) | 8 | 12 | 2 | 0.5 |
| e(5) | 6 | 7 | 4 | 4 |
| f(6) | 5 | 7 | 3 | 1.5 |
| g(7) | 10 | 13 | 2 | 0.66 |
| h(8) | 11 | 17 | 5 | 0.83 |

Notes. *CNI_{*i*}(normal cost of activity *i* for interdictor) is 0.

TABLE 5. Results for binary interdiction through CPMI-DDT.

| | Project completion time (in weeks) | Activities interdicted ($y_i^* = 1$) | Critical path | Total interdiction resource used |
|--------|---------------------------------------|---|---------------|-------------------------------------|
| Case 0 | 28 | – | s–a–e–f–g–t | – |
| Case 1 | 32 | a–e–f–g | s–a–e–f–g–t | 4 |
| Case 2 | 33 | b–c–e–g | s–b–c–g–t | 4 |

6.1.2. Binary and Partial Interdiction of the Marketing Project

The results for Cases 0 through 2 are given in Table 5. In Case 0 (no interdiction), the project is completed in 28 weeks with the critical path of a–e–f–g. In Case 1, the project is extended 4 more weeks resulting with a completion time of 32 weeks and 4 units of interdiction resource. Activities a, e, f, and g are chosen for interdiction, which means that each one of these activities are delayed by 1 week (associated d_i . values are 1 in Tab. 3). Observe that the critical path in Case 1 is the same as the one in Case 0. That is, in Case 1, interdiction occurs on the original critical path of the project. Since the activities have no advantage on each other in terms of delay and interdiction resource usage, it is natural that original critical path is selected for interdiction in this particular case. On the other hand, in Case 2, project completion time is 33 weeks and the interdicted activities are b, c, e, and g. In this case, resulting critical path s–b–c–g–t chosen for interdiction is different from that of Case 0. The reason w this path is selected lies in the fact that it provides a better interdiction opportunity than all the other paths including the original critical path. Observe that, in Case 2, activity b and c are promising for interdiction since the associated delay times (d_i). are relatively high. So, how much an activity can be delayed and the resource needed for the delay, are critical factors in deciding the activities or the path to interdict.

Note that activity e appears as an interdicted one in Case 2 even though it does not exist on the new critical path s–b–c–g–t resulting from interdiction. In fact, whether activity e is interdicted or not, the maximum project completion time will be 33 weeks as long as the other activities in the solution, *i.e.*, b, c, and g, are interdicted. This is because the activities b, c, and g exist on the emerged critical path after interdiction whereas activity e does not. The interdiction of activity e, which does not influence the project completion time in this case, is only a waste of resources. The model CPMI-DDT (CPMI-CDT as well) guarantees maximum interdicted project completion time but not the most efficient usage of interdiction resources. This deficiency is addressed in Section 6.1.3 by using E-CPMI-DDT.

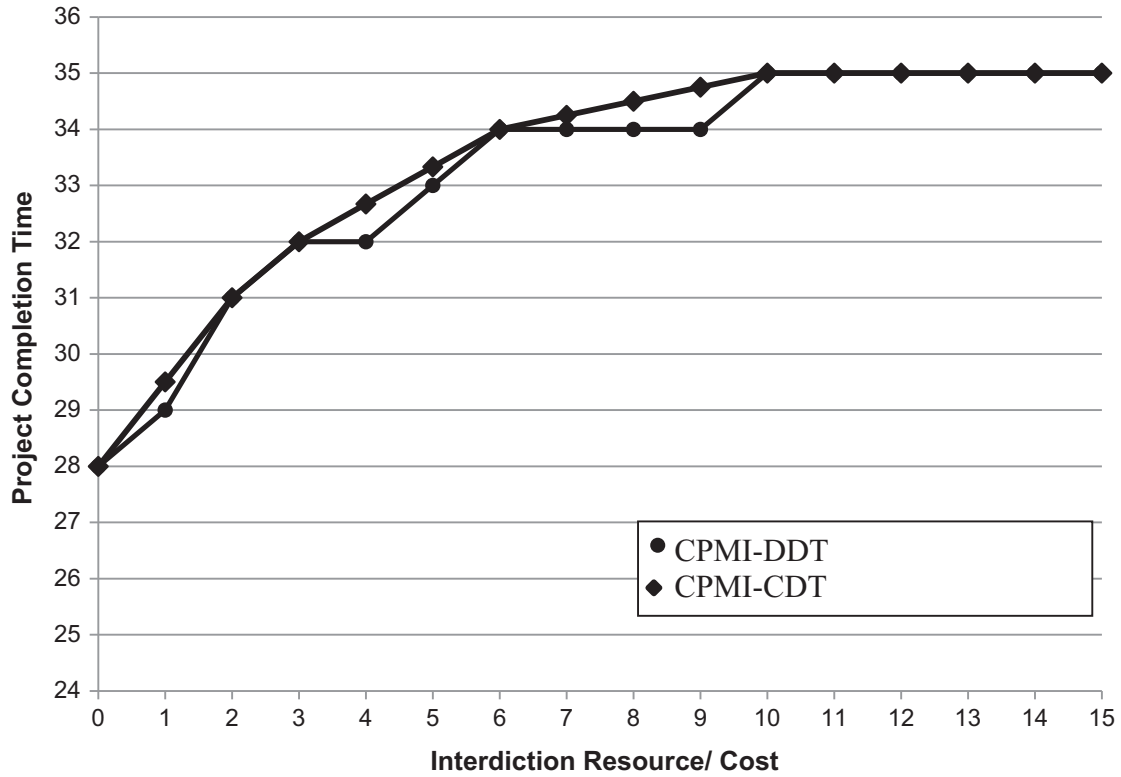


FIGURE 3. Results of CPMI-DDT and CPMI-CDT.

The results for Cases 3 and 4 are summarized in Figure 3. Note that the results for Cases 3 and 4 are obtained by solving CPM-DDT and CPM-CDT, respectively. Figure 3 indicates that the results of CPM-CDT are the same or better than those of CPM-DDT in terms of project completion time for the same level of interdiction resource. For example, for $R = 9$, the completion times for CPM-DDT and CPM-CDT are 34 and 34.75, respectively. By allowing partial interdiction, average delay in project completion time is increased from 4.36 to 4.64 weeks with an improvement of 6.42%.

6.1.3. Efficient interdiction solutions

As discussed earlier, models CPMI-DDT and CPMI-CDT do not consider optimal resource usage and do not necessarily produce efficient solutions. In this regard, we now use E-CPMI-DDT to get efficient solutions for Case 3.

Choosing $\Delta r = 1$, we apply algorithm EISS to find efficient solutions. Figure 4 depicts the associated efficient and non-efficient solutions obtained through E-CPMI-DDT and CPMI-DDT, respectively. In Figure 4, efficient solutions are represented by filled circles and non-efficient solutions by blank ones. E-CPMI-DDT only generates efficient solutions eliminating those that are non-efficient, whereas CPMI-DDT generates both efficient and non-efficient solutions as seen in Figure 4. As an example, to extend project completion time from 28 up to 34 we need a minimum resource of 6, which is guaranteed by E-CI-DDT. However, it is quite possible to get alternative solutions for the same project completion time using more resources (*i.e.*, 7–9) when we use CPMI-DDT as seen in Figure 4. Observe also that the project can be extended up to 35 weeks, for which the interdiction resource used is 10.

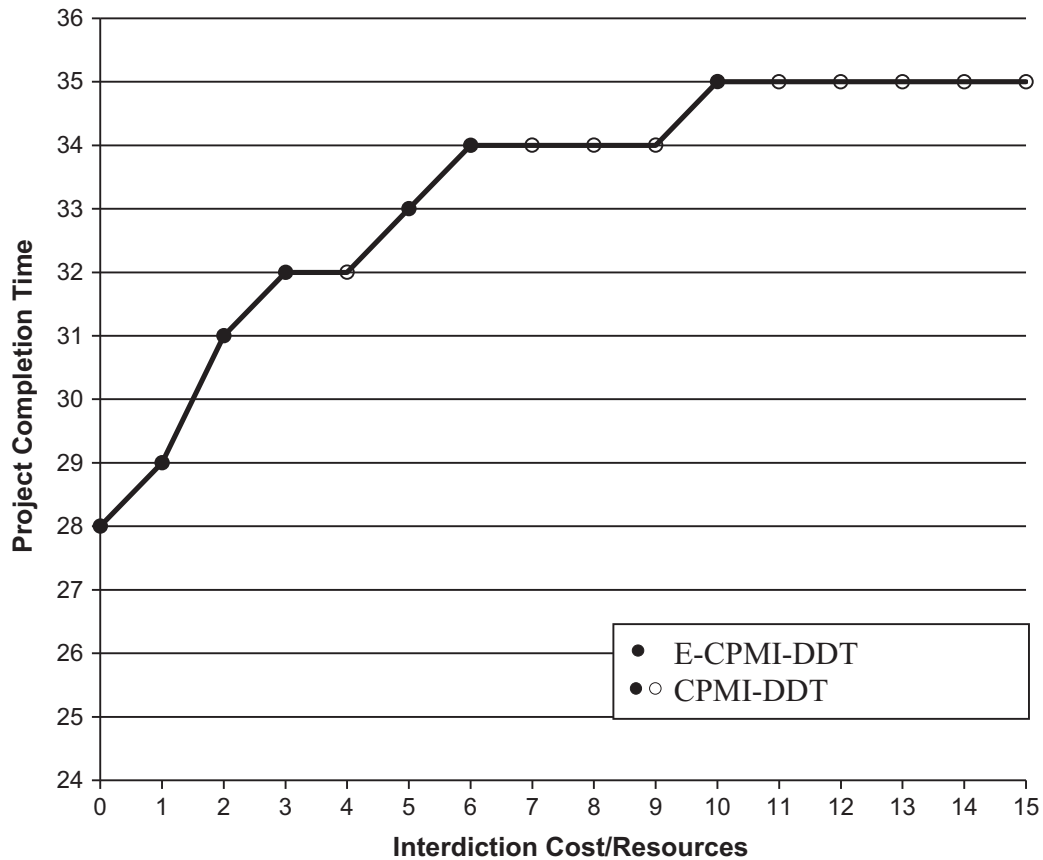


FIGURE 4. Comparison of results through CPMI-DDT and E-CPMI-DDT.

6.1.4. ESTs/LSTs in Case of interdiction

Updating t_i values after interdiction as discussed in Section 5 for Case 1 and 2 as an example and then using LPCPM-EST and LPCPM-LST, we obtain ESTs and LSTs given in Table 6.

Notice that the highlighted ESTs and LSTs are equal to each other for an activity in each case. These activities are the ones on the critical path. The critical path in Case 1 (s-a-e-f-g-t) is the same as in Case 0. However, ESTs/LSTs in Table 6 in Case 1 differ substantially from those in Case 0. In Case 2, critical path (s-b-c-g-t) is different from that in Case 0. There is a noticeable difference between ESTs/LSTs of Case 2 and Case 0.

6.2. Computational study

The models developed in our study are similar to each other when we consider the factors having influence on computational complexity, such as types and numbers of constraints/variables in the models. Our goal in this section is to have an idea about the performance of our models in different-size projects. We chose model E-CPMI-DDT and generate data for projects in different sizes.

We set $d_i = 1$ and $r_i = 1$ for an activity i . We randomly assign activity times from a uniformly distributed interval [5, 10].

Depending on the amount of total interdiction resources R at hand and the number of project activities, we come up with the solutions given in Table 7 through model E-CPMI-DDT. Note that even for a project with

TABLE 6. Early start times (ESTs)/ Latest start times (LSTs) of activities.

| Activity | Case 0 | | Case 1 | | Case 2 | |
|----------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|-------------------------|
| | EST _{<i>i</i>} | LST _{<i>i</i>} | EST _{<i>i</i>} | LST _{<i>i</i>} | EST _{<i>i</i>} | LST _{<i>i</i>} |
| s(0) | 0 | 0 | 0 | 0 | 0 | 0 |
| a(1) | 0 | 0 | 0 | 0 | 0 | 3 |
| b(2) | 0 | 1 | 0 | 4 | 0 | 0 |
| c(3) | 10 | 11 | 10 | 14 | 13 | 13 |
| d(4) | 7 | 10 | 8 | 13 | 7 | 14 |
| e(5) | 7 | 7 | 8 | 8 | 7 | 10 |
| f(6) | 13 | 13 | 15 | 15 | 14 | 17 |
| g(7) | 18 | 18 | 21 | 21 | 22 | 22 |
| h(8) | 13 | 17 | 15 | 21 | 14 | 22 |
| t(9) | 28 | 28 | 32 | 32 | 33 | 33 |

TABLE 7. Computational study results (E-CPMI-DDT).

| Number of Nodes/ Activities (<i>n</i>) | Total interdiction resource | | | |
|---|-----------------------------|------------------------------------|----------------------|------------------------------------|
| | <i>R</i> = 5 | | <i>R</i> = 20 | |
| | Solution time (s) | Project completion time (Weeks) | Solution time (s) | Project completion time (Weeks) |
| 10 | 0.047 | 32 | 0.042 | 32 |
| 20 | 0.120 | 60 | 0.107 | 62 |
| 40 | 0.147 | 117 | 0.114 | 128 |
| 60 | 0.161 | 173 | 0.144 | 188 |
| 80 | 0.109 | 229 | 0.125 | 244 |
| 100 | 0.112 | 285 | 0.110 | 300 |
| 120 | 0.121 | 341 | 0.126 | 356 |
| 140 | 0.134 | 397 | 0.146 | 412 |
| 160 | 0.136 | 453 | 0.147 | 468 |
| 180 | 0.121 | 509 | 0.162 | 524 |
| 200 | 0.129 | 565 | 0.154 | 580 |

200 activities, which is a good size for a project, the solution has been obtained within a second. The project completion time for the project having 200 activities turns out to be 565 when there is a 5-unit interdiction resource available, whereas it extends to 580 when the mentioned resource is 20.

Figure 5 shows the project completion times against different number of nodes (activities) for $R = 5$ and $R = 20$. It is clearly seen that with more interdiction resources ($R = 20$) we get longer project completion times compared to less resource case ($R = 5$). That is why the line associated with $R = 20$ is above that of $R = 5$.

7. CONCLUSIONS

In this study, we have analyzed the interdiction of a basic CPM-based project network. We have considered two interdiction cases with limited interdiction resources in which delay times are either discrete or continuous. In the former, the interdictor either inflicts some specific delay on an activity or not. In the latter, the interdictor has the chance to partially delay an activity within an allowable range. For both cases, we have developed interdiction models, CPMI-DDT and CPMI-CDT, respectively. In doing that, we have first introduced bi-level interdiction models and then converted them into standard MIP/IP models by means of duality. The results indicate that depending on the resources and the possible delay times to inflict upon an activity, the interdiction

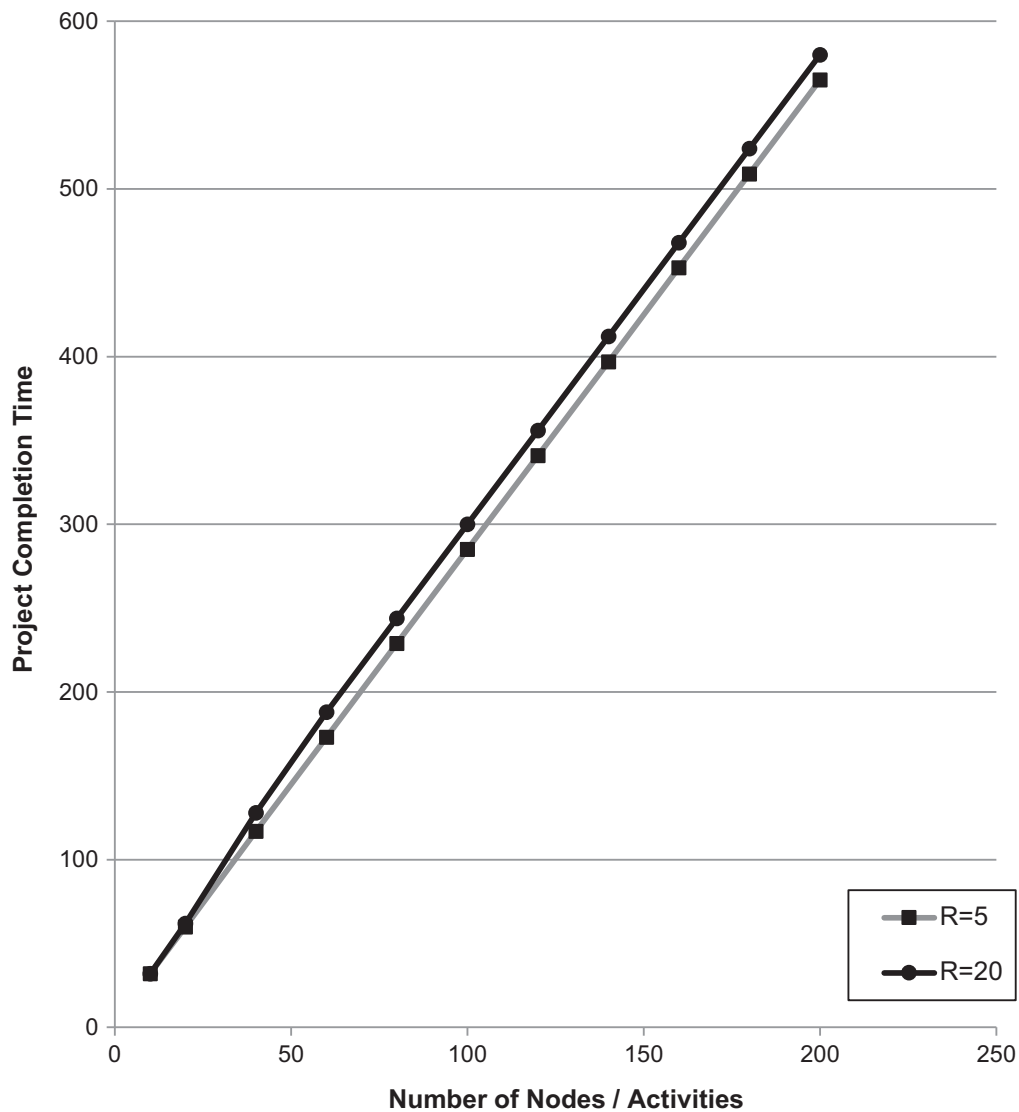


FIGURE 5. Project completion time against number of nodes.

might or might not happen on the original critical path of the project network. The model for continuous delay times (CPMI-CDT) allow the interdictor to obtain better solutions in terms of project completion time with the same amount of resource.

As the original models do not take into account the effective resource usage for the interdictor, we have extended them to find efficient solutions with respect to completion time and interdiction resource usage. E-CPMI-DDT and E-CPMI-CDT enable interdictor a more efficient resource usage because they eliminate the potential non-efficient solutions of the original models CPMI-DDT and CPMI-CDT, respectively. In order to have a better idea about the relationship between project completion time and interdiction resources, we have also developed the algorithm EISS to find efficient interdiction solution set.

We have shown how to find the ESTs and LSTs from the PM's perspective and thus provided a scheme for scheduling activities in case of interdiction.

The proposed models and algorithms have been applied in a problem setting where an interdicator tries to delay a project of a rival company to introduce a new product. The results put forward critical information for both the interdicator and PM to determine their possible courses of action in a competitive environment. Specifically, interdicator has the chance to define his/her best interdiction strategy to delay an adversary's project. He/She can choose the to-be-delayed activities among many others with an efficient resource usage in order to maximize project completion time. On the other hand, PM may use the obtained results to be more vigilant about the critical activities since the original critical path may change in case of interdiction. He/She can schedule the project activities more robustly in the light of the obtained information for a possible interdiction scenario so that the associated project completion time is minimized. The worst case for PM in which the project might be maximally delayed is also unveiled.

Finally, to figure out the performance of our models, we use different size projects increasing the number of activities up to 200. The results indicate that the solution is achievable within seconds for a reasonably large project having 200 activities

Further research is needed to investigate the applicability of the concepts and approaches proposed in this study in different project scheduling models such as crashed projects or resource-constrained projects. Research is also needed to develop new solution methodologies (*e.g.*, decomposition methods) that can be used for the problems (*e.g.*, discrete project network models) where the concept of duality does not work to convert the bi-level models into single-level models. Another future study having potential to contribute to the field would be adopting a stochastic approach and using system reliability modeling concepts to identify the failure risk of a project due to a deliberate intervention based on some probabilistic activity delays.

Acknowledgements. This research is supported by the Scientific and Technological Research Council of Turkey (TUBİTAK) and Grant No. 214M196.

REFERENCES

- [1] R.K. Ahuja, T.L. Magneti and J.B. Orlin, *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, New Jersey (1993).
- [2] İ. Akgün, B.Ç. Tansel and R.K. Wood, The multi-terminal maximum flow network interdiction problem. *Eur. J. Oper. Res.* **211** (2011) 241–251.
- [3] G. Anandalingam and V. Apprey, Multi-level programming and conflict resolution. *Eur. J. Oper. Res.* **51** (1991) 233–247.
- [4] N. Assimakopoulos, A network interdiction model for hospital infection control. *Comput. Biol. Med.* **17** (1987) 413–422.
- [5] M.G. Ashtiani, A. Makui and R. Ramezani, A robust model for a leader-follower competitive facility location problem in a discrete space. *Appl. Math. Model.* **37** (2013) 62–71.
- [6] M.S. Bazaraa, J.J. Jarvis and H.D. Sherali, *Linear Programming and Network Flows*. John Wiley & Sons, New Jersey (2005).
- [7] D. Bertsimas, E. Nasrabadi, J.B. Orlin, On the power of randomization in network interdiction. *Oper. Res. Lett.* **44** (2016) 114–120.
- [8] G. Brown, M. Carlyle, D. Diehl, J. Kline and R.K. Wood, A two-sided optimization for theater-ballistic missile defense. *Oper. Res.* **53** (2005) 745–763.
- [9] G. Brown, M. Carlyle, J. Royset and R.K. Wood, On the complexity of delaying an adversary's project. In: *The Next Wave in Computing, Optimization and Decision Technologies*. Springer, New York (2005) 3–17.
- [10] G. Brown, M. Carlyle, J. Salmeron and R.K. Wood, Defending critical infrastructure. *Interfaces* **36** (2006) 530–544.
- [11] G. Brown, M. Carlyle, R. Harney, E. Skroch and R.K. Wood, Interdicting a nuclear weapons project. *Oper. Res.* **57** (2009) 866–877.
- [12] J.F. Camacho-Vallejo, A.E. Cordero-Franco and R.G. Gonzalez-Ramirez, Solving the bilevel facility location problem under preferences by a stackelberg-evolutionary algorithm. *Math. Prob. Eng.* **2014** (2014) 430243.
- [13] S.R. Chestnut and R. Zenklusen, Hardness and approximation for network flow interdiction. *Networks* **69** (2017) 378–387.
- [14] R.L. Church, M.P. Scaparra and R.S. Middleton, Identifying critical infrastructure: the median and covering facility interdiction problems. *Ann. Assoc. Am. Geographers* **94** (2004) 491–502.
- [15] H.W. Corley and D.Y. Shaw, Most vital links and nodes in weighted networks. *Oper. Res. Lett.* **1** (1982) 157–160.
- [16] K.J. Cormican, *Computational methods for deterministic and stochastic network interdiction problems*. Masters thesis. Naval Post Graduate School, Monterey, CA (1995).
- [17] K.J. Cormican, D.P. Morton and R.K. Wood, Stochastic network interdiction. *Oper. Res.* **46** (1998) 184–197.
- [18] Y. Disser and J. Matuschke, The complexity of computing a robust flow. Preprint [arXiv:1704.08241](https://arxiv.org/abs/1704.08241) (2017).

- [19] S. Fang, P. Guo, M. Li and L. Zhang, Bilevel multiobjective programming applied to water resources allocation. *Math. Prob. Eng.* **2013** (2013) 837919.
- [20] L.R. Ford and D.R. Fulkerson, *Flows in Networks*. Princeton University, Princeton, NJ (1962).
- [21] D.R. Fulkerson and G.C. Harding, Maximizing the minimum source-sink path subject to a budget constraint. *Math. Prog.* **13** (1977) 116–118.
- [22] GAMS Development Corporation, *General Algebraic Modeling System (GAMS)*. Rev 146 (2006).
- [23] P.M. Ghare, D.C. Montgomery and T.M. Turner, Optimal interdiction policy for a flow network. *Nav. Res. Logist. Q.* **18** (1971) 37–45.
- [24] B. Golden, A problem in network interdiction. *Nav. Res. Logist. Q.* **25** (1978) 711–713.
- [25] R.L. Helmbold, *A Counter Capacity Network Interdiction Model*. Report R-611-PR, RAND Corporation: Santa Monica, CA (1971).
- [26] E. Israeli, *System interdiction and defense*. Doctoral dissertation Naval Postgraduate School, Monterey, CA (1999).
- [27] E. Israeli and R.K. Wood, Shortest-path network interdiction. *Networks* **40** (2002) 97–111.
- [28] U. Janjarassuk and T. Nakrachata-Amon, A simulated annealing algorithm to the stochastic network interdiction problem. In: *Proceedings of IEEE International Conference on Industrial Engineering and Engineering Management* (2016) 230–233.
- [29] V.V. Kalashnikov, S. Dempe, G.A. Pérez-Valdés, N.I. Kalashnykova and J.F. Camacho-Vallejo, Bi-level programming and applications. *Math. Prob. Eng.* **2** (2015) 1–16.
- [30] T. Kim, S.J. Wright, D. Bienstock and S. Harnett, Analyzing vulnerability of power systems with continuous optimization formulations. *IEEE Trans. Network Sci. Eng.* **3** (2016) 132–146.
- [31] S.H. Lubore, H.D. Ratliff and G.T. Sicilia, Determining the most vital link in a flow network. *Nav. Res. Logist. Q.* **17** (1971) 497–502.
- [32] R.M. Lusby, J. Larsen and S. Bull, A survey on robustness in railway planning. *Eur. J. Oper. Res.* **266** (2018) 1–15.
- [33] K. Malik, A.K. Mittal and S.K. Gupta, The k -most vital arcs in the shortest path problem. *Oper. Res. Lett.* **8** (1989) 223–227.
- [34] A.W. McMasters and T.M. Mustin, Optimal interdiction of a supply network. *Nav. Res. Logist. Q.* **17** (1970) 261–268.
- [35] J.R. Meredith and S.J. Mantel, *Project management: a managerial approach*. John Wiley & Sons, Inc., New Jersey (2003).
- [36] H.D. Ratliff, G.T. Sicilia and S.H. Lubore, Finding the n most vital links in flow networks. *Manage. Sci.* **21** (1975) 531–539.
- [37] C. Rocco and J.E. Ramirez-Marquez, A bi-objective approach for shortest path network interdiction. *Comput. Ind. Eng.* **59** (2010) 232–240.
- [38] J.O. Royset and R.K. Wood, Solving the bi-objective maximum flow network interdiction problem. *INFORMS J. Comput.* **19** (2007) 175–184.
- [39] J. Salmeròn, R.K. Wood and R. Baldick, Analysis of electric grid security under terrorist threat. *IEEE Trans. Power Syst.* **19** (2004) 905–912.
- [40] J. Salmeròn, R.K. Wood and R. Baldick, Worst case interdiction analysis of large-scale electric power grids. *IEEE Trans. Power Syst.* **24** (2009) 96–104.
- [41] M. Simaan and J.B. Cruz, On the Stackelberg strategy in nonzero-sum games. *J. Optim. Theory App.* **11** (1973) 533–555.
- [42] J.C. Smith, *Basic interdiction models*. In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Hoboken (2011).
- [43] K. Sundar, C. Coffrin, H. Nagarajan and R. Bent, Probabilistic $N - k$ failure-identification for power systems. *Networks* **71** (2018) 302–321.
- [44] M.T. Tabucanon, *Multiple Criteria Decision Making in Industry*. Elsevier, Amsterdam (1988).
- [45] H. Von Stackelberg, *The Theory of the Market Economy*. William Hodge & Co., London, German (1952).
- [46] U.-P. Wei, C. Zhu, K. Xiao, Q. Yin and Y. Zha, Shortest Path network interdiction with goal threshold. *IEEE Access* **6** (2018) 29332–29343.
- [47] U.-P. Wen and S.-T. Hsu, Linear bi-level programming problems – a review. *J. Oper. Res. Soc.* **42** (1991) 125–133.
- [48] W.L. Winston, *Operations Research Applications and Algorithms*. Brooks/Cole, Cengage Learning, Belmont (2004).
- [49] R. Wollmer, *Some Methods for Determining the Most Vital Link in a Railway Network*. Rand Corporation, Santa Monica, CA (1963).
- [50] R. Wollmer, Removing arcs from a network. *Oper. Res.* **12** (1964) 934–940.
- [51] R.K. Wood, Deterministic network interdiction. *Math. Comput. Model.* **17** (1993) 1–18.
- [52] R.K. Wood, Bi-level network interdiction models: formulations and solutions. In: *Wiley Encyclopedia of Operations Research and Management Science*. John Wiley & Sons, Hoboken (2011).
- [53] P. Zhang and N. Fan, Analysis of budget for interdiction on multicommodity network flows. *J. Global Optim.* **67** (2016) 1–31.