Available online at www.sciencedirect.com
ScienceDirect
Computers & Operations Research 35 (2008) 1905–1923
computers & operations research
www.elsevier.com/locate/cor

A bilevel mixed-integer program for critical infrastructure protection planning

Maria P. Scaparra[a,*], Richard L. Church[b]

[a]Kent Business School, University of Kent, Canterbury, Kent, UK
[b]Department of Geography, University of California, Santa Barbara, CA, USA

Available online 25 October 2006

Abstract

Vulnerability to sudden service disruptions due to deliberate sabotage and terrorist attacks is one of the major threats of today. In this paper, we present a bilevel formulation of the r-interdiction median problem with fortification (RIMF). RIMF identifies the most cost-effective way of allocating protective resources among the facilities of an existing but vulnerable system so that the impact of the most disruptive attack to r unprotected facilities is minimized. The model is based upon the classical p-median location model and assumes that the efficiency of the system is measured in terms of accessibility or service provision costs. In the bilevel formulation, the top level problem involves the decisions about which facilities to fortify in order to minimize the worst-case efficiency reduction due to the loss of unprotected facilities. Worst-case scenario losses are modeled in the lower-level interdiction problem. We solve the bilevel problem through an implicit enumeration (IE) algorithm, which relies on the efficient solution of the lower-level interdiction problem. Extensive computational results are reported, including comparisons with earlier results obtained by a single-level approach to the problem.

Keywords: Bilevel programming; Protection models; Median location problems

1. Introduction

A crucial issue in today's distribution, supply and emergency response systems is to guarantee continuity and efficiency in service provision in the face of natural and man-made threats. If limited protective resources are available to increase system robustness and resiliency, a key question is to identify which facilities to protect or fortify in order to preserve the functionality of the system as much as possible in case of sabotage or external disruptions. This paper presents a new formulation and a new solution approach for solving the r-interdiction median problem with fortification (RIMF), a problem which was originally introduced by Church and Scaparra [1] to identify the most cost-effective way of protecting facilities in an existing distribution network against worst-case, long-term disruptions. The model assumes the existence of a system with p operating facilities and a set of system users who receive service from their nearest facility. Overall system efficiency is measured in terms of total weighted distance between users

* Corresponding author.
E-mail address: M.P.Scaparra@kent.ac.uk (M.P. Scaparra).

0305-0548/$ - see front matter
doi:10.1016/j.cor.2006.09.019

and their closest facilities, with weights defining user demands for goods or service. This kind of efficiency measure, which characterizes the well-known p-median location problem [2], has been largely used in application settings where maximizing consumer access to supply centers is the primary objective. Access is thus directly determined by the distances between customers and their nearest facilities. If some of the facilities are lost due to an attack on the system, customers need to travel to or receive shipments from facilities which are further away. This reduction in service accessibility can have costly and, sometimes, harmful implications, especially if the facilities provide essential services or goods (e.g., hospitals, drugs, vaccines). The RIMF involves identifying the optimal allocation of a limited amount of protective resources among the p facilities of a distribution network in such a way that the accessibility reduction due to a worst-case loss of $r(r < p)$ unprotected facilities is minimized. The model assumes that exactly q facilities can be protected (with $q + r \leqslant p$) and that an attack on a protected facility has no effect.

Church and Scaparra [1] formulate the problem just described as a mixed-integer problem (IMF) and solve it using a commercial branch-and-bound solver. Unfortunately, the number of variables and constraints in their model is directly determined by taking an explicit enumeration of all possible ways of losing r out of the p facilities. It is easy to see that the size of this model grows very rapidly as p and r increase so that only problems of very modest size can be solved through the IMF formulation. In a subsequent work, Scaparra and Church [3] develop an alternative mixed-integer optimization model for solving the problem, called MCPC. The new model is based upon a maximum covering type formulation and is solved to optimality after a specialized model reduction process, involving the iterative solution of set covering problems. Although this model can be solved significantly faster than the former one, it still presents the limitation of requiring a complete enumeration of all possible ways of interdicting r of the p facilities.

In this paper, we propose an alternate solution methodology based upon a bilevel formulation of the RIMF that does not face the same size restrictions as the previous approaches. In the bilevel formulation, the top level problem involves the system planner or *defender* making decisions about which facilities to secure or harden. The defender's objective is to minimize the worst-case sum of weighted distances among customers and facilities that can occur following the loss of r unprotected facilities. Worst-case scenario losses in response to a given fortification strategy are modeled in the lower-level *interdiction* problem, where a potential attacker or *interdictor* decides which unprotected facilities to hit in order to cause a maximal reduction in system efficiency (i.e., to maximize the sum of weighted distances). Interdiction models have been used extensively over the past few years as a tool for assessing network vulnerabilities to linkage or node disruptions [4,5]. Interdiction problems are inherently bilevel in nature: they involve the conflicting decisions of an interdictor, who tries to degrade system performances by disabling key components, and those of the system users, who try to operate the system in an optimal way after interdiction. In our specific case, the *attacker–user* problem can be modeled as a single-level mixed-integer problem [6], so that what is in principle a trilevel *defender–attacker–user* problem [7] can be reduced to a bilevel *min–max* problem.

Bilevel programming problems fall into the class of NP-hard problems, even in their simplest form with only continuous variables [8]. In order to solve the bilevel RIMF, we propose a specialized tree search algorithm. The computational effort of the proposed approach largely depends on the difficulty of solving the lower-level interdiction problem (RIM). We hence test several alternative formulations of RIM and show empirically that significant computational gains can be obtained through a process of variable reduction and consolidation. As a result, we are now able to solve RIMF problems of significant size, which could not be solved with previous approaches.

The main contribution of this paper centers on the development of an efficient algorithmic approach to solve a discrete bilevel fortification/interdiction problem. Whereas interdiction problems have received vast attention in the literature, the more complex problem which includes an additional level of protection has only been fleetingly mentioned by a few authors [9,7]. To the best of the authors' knowledge, no efficient algorithms have been proposed to deal with three-stage *defender–attacker–user* problems. The IMF and MCPC problems were the first attempts in this direction but they relied on some simplifying assumptions. The new approach relaxes these assumptions and greatly enhances the model's scalability and scope.

The reminder of this paper is organized as follows. In the next section we briefly review the relevant literature on reliability, interdiction and fortification modeling. In Section 3, we formulate RIMF as a bilevel mixed integer program and discuss the difficulties involved in solving problems with nested min–max structures. In Section 4, we present a tailored tree search technique for solving RIMF to optimality. In Section 5, we discuss how to streamline the solution approach by solving special condensed formulations of the lower-level interdiction problem. Computational results are presented in Section 6, followed by a set of conclusions and recommendations for future research.

2. Background

The need for systematic and analytical tools to address issues of system vulnerability and security investment has been widely recognized among academics and practitioners [10]. Nonetheless, the study of mathematical models and techniques for improving system security is still largely unexplored. Prior research in this area has mainly focused on the analysis of risk sources and has outlined general guidelines for mitigating the disruptive impact of offensive strikes on a system with regard to its ability to operate efficiently (see, for example, [11,12]). Comparatively little attention has been paid to actually optimizing security and reliability in logistic systems.

Traditionally, the analysis of system reliability pertains to the study of telecommunication networks and involves the design of network topologies which are robust to link or node failures [13]. In network theory, reliability is defined as the system's ability to preserve node connectivity in the event of random failures and its study is predominantly based upon pathset and cutset approaches for connectivity analysis [14,15]. Within the field of production and distribution systems, the notion of reliability is somewhat different, since the main focus is no longer on ensuring node-to-node connectivity in the face of component failures, but rather on delivering supply and services to customers in an efficient, timely and economic manner, even in the event of intentional disruptions.

Only a few papers have addressed features of reliability and security in logistics systems. Drezner [16] and Lee [17] examine the unreliable p-median location problem, where p unreliable facilities must be located so as to minimize expected travel distances between customers and facilities. Bundschuh et al. [18] present several mathematical models for improving reliability and robustness in supply chains through the optimal choice of suppliers. Snyder and Daskin [19] propose several reliability models to find the optimal location of facilities so as to minimize regular operation costs as well as expected costs incurred when some of the facilities are unavailable. Finally, O'Hanley [20] presents a model for the design of robust, coverage-type service systems. O'Hanley's model finds the optimal location of a set of facilities in order to maximize a combination of initial demand coverage and the minimum coverage level following the loss of one or more facilities.

In all of the above models, the authors demonstrate that the impact of facility loss can be mitigated in the initial design of a system. However, redesigning an entire system is not always a viable option given the potentially large expenses involved with relocating facilities or changing suppliers. Instead, methods for protecting existing infrastructure may not only be preferable over the short term, but may also produce considerably greater gains in reliability improvements over and above passive design strategies. Additionally, most of the above models are more suitable to model situations in which natural or accidental failures are a major concern. It is evident that modeling protection strategies against intentional attacks is fundamentally different from optimizing protection against acts of nature. In fact, whereas nature hits at random, intelligent adversaries try to inflict maximum harm and may adjust their offensive strategies to circumvent protection measures. In this paper, we restrict our attention to the latter type of problem.

A first step into modeling deliberate and rational attacks is through the use of *interdiction* models. Interdiction models, initially studied for military applications, have been mainly used to assess the impact of losing critical links or nodes in transportation networks. A variety of models, which differ in terms of objectives and underlying network structures, have been proposed in the literature. For example, the effect of interdiction on the maximum flow through a network is studied in Wollmer [4] and Wood [5]. Cormican et al. [21] propose a stochastic variation of this problem. Fulkerson and Harding [22] and Israeli and Wood [23] analyze the impact of arc removals on the shortest path length between two nodes. Lim and Smith [24] treat the multicommodity version of the shortest path problem, with the objective of assessing shipment revenue reductions due to arc interdictions. A review of early interdiction models is provided in Church et al. [6]. Church et al. [6] also developed two facility interdiction models, called the median facility interdiction model (RIM) and the covering facility interdiction model (RIC), which identify the set of supply or emergency response facilities which, if lost, disrupt service delivery the most.

Interdiction models can help reveal potential weaknesses in a system. However, they do not explicitly address the issue of optimizing security. It is not hard to demonstrate that securing those facilities that are identified as critical in an optimal interdiction solution will not necessarily provide the greatest protection against a malicious attack ([25]; Church and Scaparra, 2006). Optimal interdiction is a function of what is fortified, so it is important to capture this interdependency within a modeling framework. In this paper, we address the issue of optimizing security or fortification of a set of existing facilities in order to thwart to the greatest degree possible the effects of interdiction. The resulting fortification/interdiction problem can be described within a game theoretic framework as a leader–follower or Stackelberg game [26] and formulated as a bilevel programming problem, as detailed in the next section.

3. Formulating RIMF as a bilevel programming program

We assume that a system exists which is comprised of p facilities. We denote by F the set of p operating facilities in the system and by N the set of n demand nodes. The elements in these sets are indexed by j and i, respectively. The demand for service at each node i is a_i, and the shortest distance (or unit shipping cost) between the facility at j and demand node i is given by d_{ij}. We assume that in the initial configuration, the demand at each node is entirely supplied by the closest facility to that node and that, if that facility is lost due to interdiction, the demand is reassigned to the next closest facility among the non-interdicted ones. We assume that both offensive and protective resources are limited so that at most r facilities can be attacked, and at most q facilities can be hardened against interdiction.

RIMF can be seen as a game involving the sequential decisions of two players: a facility planner (the *leader*) first decides which q facilities to fortify so that the system operates as efficiently as possible in case of interdiction; an interdictor (the *follower*) then attempts to reduce the system efficiency as much as possible by hitting r unprotected facilities. Note that the interdictor problem is used to compute the value of any fortification plan on the basis of worst-case losses. In this sense, we can assume that when the interdictor selects the facilities to hit, he has perfect information about which facilities are protected. If we relax this assumption, the interdictor may waste offensive resources by attacking fortified facilities, and his attack would not represent worst-case losses.

The bilevel formulation of RIMF uses the following sets of decision variables:

$$z_j = \begin{cases} 1 & \text{if a facility located at } j \text{ is fortified,} \\ 0 & \text{otherwise,} \end{cases}$$

$$s_j = \begin{cases} 1 & \text{if a facility located at } j \text{ is eliminated, i.e., interdicted,} \\ 0 & \text{otherwise} \end{cases}$$

$$x_{ij} = \begin{cases} 1 & \text{if demand point } i \text{ is served by a facility at } j \text{ after interdiction,} \\ 0 & \text{otherwise.} \end{cases}$$

The variables z_j and s_j represent the upper level fortification variables and the lower-level interdiction variables, respectively. The evaluation of the system efficiency also requires the definition of the assignment variables, x_{ij}, which represent user choices. Additionally, the formulation uses the set $T_{ij} = \{k \in F | k \neq j \text{ and } d_{ik} > d_{ij}\}$, i.e., the set of existing sites (not including j) that are as far or farther than j is from demand i.

RIMF can be stated mathematically as follows:

$$\min \quad H(\mathbf{z}) \tag{1}$$

$$\text{s.t.} \quad \sum_{j \in F} z_j = q, \tag{2}$$

$$z_j \in \{0, 1\} \; \forall j \in F, \tag{3}$$

where

$$H(\mathbf{z}) = \max \quad \sum_{i \in N} \sum_{j \in F} a_i d_{ij} x_{ij} \tag{4}$$

$$\text{s.t.} \quad \sum_{j \in F} x_{ij} = 1 \quad \forall i \in N, \tag{5}$$

$$\sum_{j \in F} s_j = r, \tag{6}$$

$$\sum_{h \in T_{ij}} x_{ih} \leqslant s_j \quad \forall i \in N, \quad j \in F, \tag{7}$$

$$s_j \leqslant 1 - z_j \; \forall j \in F, \tag{8}$$

$$s_j \in \{0, 1\} \quad \forall j \in F, \tag{9}$$

$$x_{ij} \in \{0, 1\} \quad \forall i \in N, \quad j \in F. \tag{10}$$

In the bilevel formulation (1)–(10), the leader and the follower have conflicting objectives. Namely, the leader allocates exactly q fortification resources (2) to minimize an efficiency function $H(1)$, which represents the highest possible level of weighted distance or costs deriving from the loss of r of the p facilities. The meaning of H is enforced in the follower problem, whose objective involves maximizing the weighted distance or service cost after the removal of r facilities.

The lower-level program (4)–(8) is simply the interdiction problem RIM formulated by Church et al. [6], with the additional constraints (8) which prevent the interdiction of any site chosen to be fortified in the upper level problem. In the interdiction problem, constraints (5) state that each demand point must be assigned to a facility after interdiction. Constraint (6) specifies that only r facilities can be interdicted. Constraint (7) maintains that each demand point must be assigned to its closest open facility after interdiction. More specifically, these constraints state that if a given facility j is not interdicted ($s_j = 0$), a customer i cannot be served by a facility which is further than j from i. This kind of closest assignment (CA) constraints was previously employed by Church and Cohon [27] for siting energy facilities and by Hanjoul and Peeters [28] in plant location models. Alternative constructs that have been proposed in the literature to force CA will be discussed in Section 5.3.

Constraints (8) link the upper and lower-level problem. In the remainder of the paper, we will refer to the lower-level problem which includes constraints (8) as conditional RIM (CRIM), due to the conditional nature of what has been fortified. Finally, constraints (3), (9) and (10) represent the integrality requirements for the fortification, interdiction and assignment variables, respectively. In solving this model, integer restrictions are required only for the interdiction variables, s_j , as an optimal solution with fractional x_{ij} variables only occurs when there is a tie in which remaining facility is closest. Such cases, although interesting, do not affect the optimality of the solution.

Note that this formulation does not necessarily require the customers to be initially allocated to their closer facility and it can be easily adjusted to model situations in which customers are free to choose which facility to patronize. Modeling this case only requires that the customer's preference ordering of the facilities is known to the system planner and to the assumed interdictor. CA constraints can still be used to capture preference orderings (see, for example, [28]).

Details on bilevel programming can be found in Bard [29] and Dempe [30]. Although several applications of bilevel programming can be found in the literature when all variables are continuous, few applications have been published involving discrete variables (see for example [33]). The difficulties encountered with the presence of integer restrictions in a bilevel format are explained in Moore and Bard [31] and Vicente et al [32]. Overall, the difficulty in solving such a problem depends on: (1) the class of discrete bilevel programs and (2) the position of the upper level variables in the lower-level problem. The RIMF problem can be classified as especially complex as integer restrictions appear at both levels, and the upper level variables parameterize the constraints of the lower-level problem. Research on solving discrete bilevel programming problems is very limited (see [31,32] for examples). Morton et al. [34] identify three main categories of approaches that can be used to obtain computationally tractable models from difficult bilevel problems: decomposition, duality and reformulation. Decomposition involves the use of cutting plane algorithms and is usually indicated to solve problems with parameter-independent constraints in the lower-level problem [30]. A first attempt at solving RIMF by Benders decomposition demonstrated that this approach is very slow for handling problems with the parametric structure of RIMF. Duality involves taking the dual of the inner problem in order to obtain a nested min–min or max–max problem that can then be solved as a single level problem. This approach has been frequently used to solve bilevel network flow problems where the parameterized inner flow problems are linear [5,22]. Unfortunately, RIM does not have this property so that taking its dual is not a feasible option. Reformulation entails finding equivalent model structures which are easier to handle. As an example, penalties can be added in the follower's objective in order to eliminate the leader's variables from the follower's constraints [20,23,34]. To the best of the authors' knowledge, similar transformations cannot be applied to RIMF. However, the single-level models for RIMF given in Church and Scaparra [1] and Scaparra and Church [3] can be cast as reformulation strategies since they aim at obtaining formulations solvable by mixed-integer optimizers. In the next section, we propose a more efficient approach to solve RIMF, tailored to the specific bilevel structure of the problem.

4. Solving RIMF as a bilevel problem

To solve the bilevel formulation (1)–(10) of the RIMF we propose an IE algorithm. The entire approach is built on a simple observation made by Church and Scaparra [1] and restated below using a leader–follower framework.

Observation 1. Let I be the set of r interdictions in the optimal solution to the lower-level RIM problem (4)–(10) without fortification. Then the optimal set of q fortifications selected by the leader must include at least one of the r facilities in I.

This observation can be easily explained by noticing that if none of the facilities in the optimal interdiction set is protected, then it is still possible to interdict all of them and the worst possible case of interdiction is not prevented. Although at least one of the r sites must be a member of I, such a property does not necessarily hold for more than one site of I.

The basic premise of the proposed method is to exploit Observation 1 recursively in order to reduce the number of solutions which need to be evaluated in an enumeration tree. The method can be outlined in simple terms. We start at the root node of the enumeration tree by solving the follower interdiction problem without fortification. We denote by I the resulting set of optimal interdictions. This set represents the candidate sites for fortification associated with the root node. According to Observation 1, the leader must then harden at least one of these facilities. Hence, we randomly chose a site j from this set, and branch on the fortification variable, z_j, by fixing it to 1 and to 0. Each branch leads to a new node in the enumeration tree, which is processed according to one of the two following cases:

1. The node is obtained by fixing a variable z_j to 1. In this case, we proceed as follows:
   (a) We solve a CRIM problem in which we bar the interdiction of all the variables z_j set to 1 along the path from the root to the current node in order to obtain a new optimal solution to the follower problem and the associated optimal interdiction set, I.
   (b) If the path from the root to the current node contains exactly q fortification variables which have been fixed to 1 (meaning that all the fortification resources have been used), the node under consideration is a leaf node and can be excluded from further consideration. Otherwise (i.e., additional fortification resources are still available), we update the set of candidate fortifications according to the new solution to CRIM and branch again on one of the variables associated with a facility in the candidate set.
2. The node is obtained by fixing a variable z_j to 0. This means that none of the facilities in the candidate fortification set of the parent node has been fortified yet. We then need to enforce the fortification of at least one of the remaining facilities (except j). Two cases are possible:
   (a) After the removal of j, the candidate set of fortifications is empty. In this case, the incumbent node is fathomed.
   (b) Otherwise, we select another facility from the candidate set and generate other two child nodes by branching on the variable associated with the selected facility.

The process is iterated until all the nodes are either leaves or fathomed nodes. The leaf with the lowest objective function identifies the optimal solution: by backtracking from that node to the root it is possible to retrieve the optimal fortification set.

An example of how the algorithm works in practice is provided in Fig. 1, which shows the binary tree generated to solve a simple problem with six operating facilities (numbered from 1 to 6), two interdictions and two fortifications. The picture shows the set of candidate fortifications associated with each node in the tree and the branching variables selected at each node. The hatched nodes represent fathomed nodes whereas shaded nodes represent the points where a CRIM problem is solved. Among them, dark-shaded nodes indicate leaf nodes. The optimal solution to the 2-interdiction median problem without fortification at the root node involves interdicting facility 1 and 2. Facility 1 is randomly chosen from the optimal interdiction set and the corresponding variable z_1 is branched on. When the left child is processed ($z_1 = 1$), a new CRIM is solved with the additional restriction that facility 1 cannot be interdicted. The new optimal interdiction set includes facilities 3 and 6. Since the leader has sufficient resource to harden an additional facility, the process is repeated by branching on z_3, and so on. Processing a left child (e.g., the node obtained by fixing $z_1 = 0$) only requires updating the candidate fortification set by removing the facility associated with the variable just fixed to zero. Then two new branches are created by fixing one of the remaining variables (unless the candidate set is empty, in which case the node is fathomed).

Note that this tree search procedure allows the identification of all optimal solutions to the bilevel problem, if more than one exists. In practice, the algorithm can be implemented by using recursion and backtracking. The order in which branching variables are chosen is irrelevant, since all possible fortifications of the candidate sets will eventually be considered during construction of the tree.

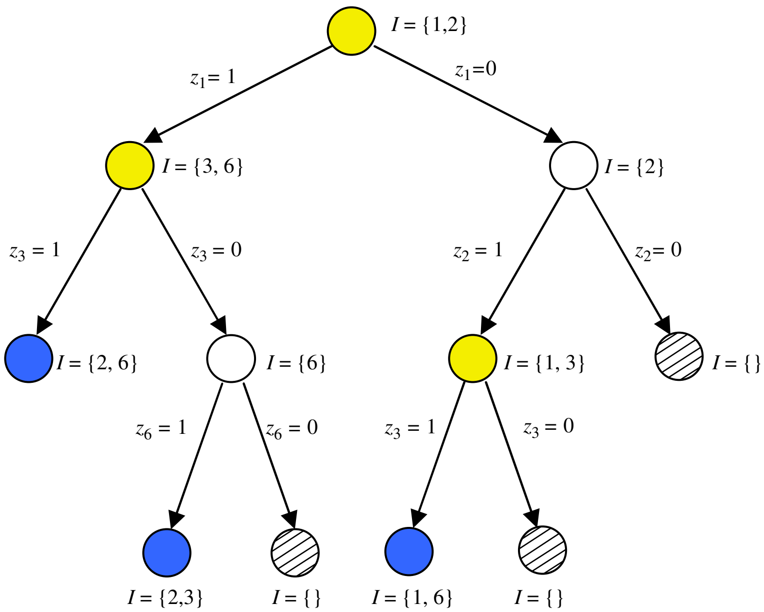

Fig. 1. Binary tree search. Example with $p = 6$, $q = 2$ and $r = 2$.

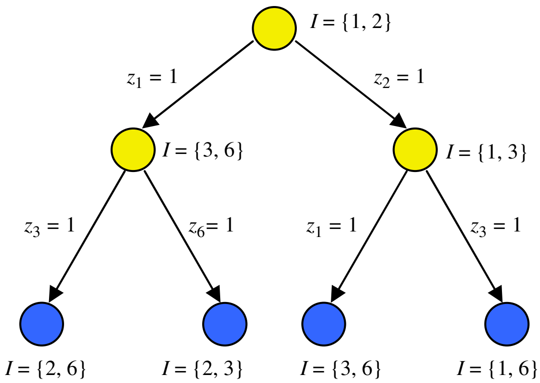

Fig. 2. Tree search. Example with $p = 6$, $q = 2$ and $r = 2$.

The most computationally expensive operation in the procedure is solving the mixed-integer CRIM problems to optimality. In our implementation, the CRIM problems were solved through the general-purpose MIP solver Cplex 9.0. The nice feature of the approach is that the follower problems are not solved from scratch at each iteration. Rather, the CRIM problem at each node is generated from the problem solved at the parent node by simply fixing to zero the interdiction variable associated with the last fortification made. The optimal solution to the CRIM problem at the parent node can then be used as a starting solution for the new problem to save computing time. An upper bound to the number of follower problems which are solved by the enumeration procedure is provided in the following proposition.

Proposition 1. *The tree search IE algorithm solves at most* $(r^{q+1} - 1)/(r - 1)$ *CRIM problems*, *where r is the number of interdictions and q is the number of fortifications*.

Proof. Consider a non-binary implementation of the search strategy in which at each node we create as many branches as the number of interdictions, r. Each branch represents the fortification of one of the interdicted facilities in the optimal set and leads to a node where a CRIM problem is solved to take into account the new fortification made. An example of the tree thus obtained for the same problem illustrated in Fig. 1 is depicted in Fig. 2. It is easy to see that the full enumeration tree built in this fashion has as many levels as the number of fortifications, q. The resulting tree is then a d-heap with $d = r$ and depth q. The number of nodes in such a tree, and consequently the number of CRIMs solved, is $(r^{q+1} - 1)/(r - 1)$ (see, for example, [35]). However, in a non-binary implementation of the search tree, the same fortification patterns may be repeated along different branches of the tree and, consequently, the same CRIM may be solved multiple times. A binary implementation overcomes this problem by avoiding repetitions of the same fortification patterns. Hence, the number $(r^{q+1} - 1)/(r - 1)$ is only an upper bound on the number of CRIMs which are actually solved during a binary search. □

The above proposition demonstrates that the size of the enumeration tree and, consequently, of the number of CRIM problems solved during the search, is independent on p. Obviously, the parameter p affects the size of the CRIM problems and the computing time for solving them. Even though the number of CRIM problems solved is limited by Proposition 1, this number can be relatively large, and therefore every effort should be taken to reduce the time to solve each CRIM. In the next section, we explain several processes by which we can accelerate the solution of the CRIM.

5. Solving RIM and CRIM efficiently

The computational effort of the proposed IE procedure is largely determined by the efficiency with which the lower-level interdiction problem can be solved. It is easy to see that any reduction in computing time for solving CRIMs to optimality may have an amplified effect on the total speed of the algorithm. In this section, we explore the possibility of streamlining the RIM model. The extension to the CRIM is straightforward, since constraints (8) are not affected by the newly introduced modifications. We then show how the new formulation improves the efficiency and scalability of our overall approach. Specifically, we investigate possible model reductions, variable consolidation and alternative formulations of the CA constraints. We also briefly comment on the computational enhancement derived from the introduction of each of these modifications. These modifications are explained within the context of RIM, but apply equally to CRIM as well.

5.1. Model reduction

The formulation of RIM can be streamlined by eliminating certain variables. For a given problem involving p existing facilities and the interdiction of r facilities, one can observe that the worst case for a given demand point will occur if the r closest facilities to that demand point have been interdicted. This means that the worst case for a given demand point i will occur when that demand point must be served by its $r + 1$st closest facility. This observation can be used to reduce the size of RIM by defining the following additional sets:

G_i the set of $r + 1$st closest facilities to demand point i before interdiction
U_{ij} $\{k \in G_i | k \neq j$ and $d_{ik} > d_{ij}\}$, the set of existing sites (not including j) that are as far or farther than j is from customer i, but not further than the $r + 1$st closest site from i
F_i the set of r closest sites to customer i before interdiction.

RIM can then be reformulated as

$$\max \sum_{i\in N} \sum_{j\in G_i} a_i d_{ij} x_{ij} \tag{11}$$

s.t.

$$\sum_{j\in F} s_j = r, \tag{12}$$

$$\sum_{j\in G_i} x_{ij} = 1 \quad \forall i \in N, \tag{13}$$

$$\sum_{h\in U_{ij}} x_{ih} \leqslant s_j \quad \forall\, i \in N, \quad j \in F_i, \tag{14}$$

$$s_j \in \{0, 1\} \quad \forall j \in F, \tag{15}$$

$$x_{ij} \in \{0, 1\} \quad \forall\, i \in N, \quad j \in G_i. \tag{16}$$

This revised formulation contains fewer constraints and variables: the number of assignment variables (x_{ij}) is reduced from np to $n(r+1)$; the number of constraints of type (14) is reduced from np to nr. This straightforward reduction proved to be very effective in practice and significantly reduced the computational time in all the preliminary tests we attempted. Hence, when we refer to RIM throughout the remainder of the paper, we refer specifically to this condensed formulation.

5.2. Variable consolidation

Church [36] recently proposed a new model formulation for the p-median location problem, called COBRA. The COBRA model is associated with identifying and consolidating redundant assignment variables, under special proximity conditions. More specifically, Church [36] demonstrated that two demands may be assigned to a given facility site, if such a site has the same order of closeness for both demands and if the set of closer sites than the site in question for both demands is equivalent. These "equivalent assignment conditions" are formalized in the following theorem, whose proof is provided in Church [36].

Theorem. *If facility j is the k closest site for both demand s and demand t, and if the set of $k-1$ closest sites for s and for t is the same, then at optimality $x_{sj}=x_{tj}$.*

The above property makes it possible to consolidate some of the variables, thus allowing a reduction in the size of the overall problem. This variable consolidation process was found to reduce the size of p-median models considerably. Furthermore, the extent of the reduction was more remarkable in those problems where the demand nodes appreciably outnumbered the facility sites. This is precisely the case of the RIM problem, given that the interdictions are restricted to the p sites where facilities already exist and that p is in general much smaller than the number of nodes, n. The properties of the COBRA model apply directly to the RIM model, which can then be reduced as explained below.

Assume that all the variables which are equivalent according to the COBRA construct have been identified by inspection of the order of site closeness for any pair of demands. Based upon this equivalency, the original assignment variables can be replaced by a smaller set of variables. We index this new set of variables by v and denote each new substitution variable as x_v. Additionally, let A be the set of indices of the new variables, i.e., $A=\{v|x_v \text{ is included in the model after the substitution}\}$. The mapping between the old variables and the new variables can be formalized and included in the mathematical formulation through the introduction of a new set of parameters, α_{ijv}, defined for each $i \in N$, j in G_i and $v \in A$ as follows:

$$\alpha_{ijv} = \begin{cases} 1 & \text{if the variable } x_{ij} \text{ is replaced by the new assignment variable } x_v, \\ 0 & \text{otherwise.} \end{cases}$$

RIM can then be reformulated in terms of the new variables:

$$\max \sum_{i\in N}\sum_{j\in G_i}\sum_{v\in A}\alpha_{ijv}a_i d_{ij}x_v \tag{17}$$

s.t.

$$\sum_{j\in F} s_j = r, \tag{18}$$

$$\sum_{j\in G_i}\sum_{v\in A}\alpha_{ijv}x_v = 1 \quad \text{for all } i\in N, \tag{19}$$

$$\sum_{h\in U_{ij}}\sum_{v\in A}\alpha_{ihv}x_v \leqslant s_j \quad \text{for all } i\in N \text{ and for all } j\in F_i, \tag{20}$$

$$s_j \in \{0,1\} \quad \text{for all } j\in F, \tag{21}$$

$$x_v \in \{0,1\} \quad \text{for all } v\in A. \tag{22}$$

The computational benefits of the new condensed formulation have been tested on two benchmark data sets, frequently used in location analysis: the 150 node London, Ont.[1] data set [37] and the 316 node Alberta[2] data set [38]. These two data sets will be used throughout the paper to test different algorithmic implementations of our approach. The two sets contain real geographical data of two Canadian regions. The London, Ont. data set has been used for the location of gasoline stations as well as fire stations, where distances are based upon a road network. In the Alberta data set, the nodes represent Alberta population centers, the distances are shortest path lengths on the actual road network and the demand at each node is given by the population at the corresponding center. In the analysis, we assume that p facilities (e.g., warehouses) are initially located at the optimal p-median sites of the two data sets. This choice facilitates the replication of our results by other researchers. However, any other initial configuration can be used as well. All results reported in this and in the next sections were run on a PC with a Pentium 4, 2.8 Ghz processor and 1 GB of RAM. Each RIM was solved with the branch-and-bound-based solver Cplex 9.0, supplied with specific directives to improve performance. We solved both data sets for different values of the parameters p and r. More specifically, we let p vary between 20 and 50 and r between 1 and 15. For any problem instance solved with the parameters ranging in the stated intervals, the number of variables was reduced significantly. The reduction extent for the London data set varied between a minimum of 19% to a maximum of 65% of the total number of variables, with the largest reductions obtained for small values of r. Tests on the Alberta data set showed the same kind of behavior, but this time the impact of the consolidation process was even more pronounced: the number of variables was reduced by up to 80% for small values of r and never by less than 50% for values of r in the upper range. Additional computational details related to the COBRA implementation will be provided in the next section, to study their effect in combination with the use of different CA constraints.

5.3. CA constraints

It was mentioned in Section 3 that there are several ways of enforcing CA. A comprehensive discussion of the structural properties of CA constraints in location problems is provided in Gerrard and Church [39]. Their study demonstrates that the choice of CA constraints is problem specific and no dominance can be established among them for all problems. In this section, we discuss a different kind of CA constraints for RIM, similar to the ones first introduced by Rojeski and ReVelle [40] in the context of the budget constrained median problem. The Rojeski and ReVelle constraints are among the most widely cited CA constraints and, as the Church and Cohon constraints (7), (14) and (20) used in our RIM formulations, have the nice property of inherently yielding integral assignment variables. Other CA constructs (see Gerrard and Church, 1996, for a review) allow fractional assignments and, consequently, increase the complexity of solving the IP formulation through solvers based on branch and bound. For this reason, we restrict our analysis to the Rojeski and ReVelle and Church and Cohon constraints only. Throughout the discussion, we will refer to these two types of CA constraints as RR and CC constraints, respectively.

The CA constraints for the RIM problem can be expressed in an RR-type form as follows:

$$x_{ij} \geqslant (1 - s_j) - \sum_{h \in C_{ij}} (1 - s_h) \quad \text{for all } i \in N \text{ and for all } j \in F_i, \tag{23}$$

where the set C_{ij} represents the set of all the facilities h which are closer to demand i than facility j, but not further than the $r+1$ closest site to i. Constraints (23) simply establish that if a facility at j is not interdicted ($s_j = 0$) but all the facilities which are closer to i are interdicted ($\sum_{h \in C_{ij}} (1 - s_h) = 0$), then demand point i must be assigned to j. However, if any of the closer facilities is operational, the right-hand side of (23) is always less or equal to zero and, hence, relation (23) has no effect on the assignment. Constraints (7) and (23) can be used interchangeably in the RIM formulation. When the COBRA version of RIM is considered, the CC constraints (20) can be replaced by the following modified version of the RR constraints (23), which take into account the variable substitutions:

$$\sum_{v \in A} \alpha_{ijv} x_v \geqslant (1 - s_j) - \sum_{h \in C_{ij}} (1 - s_h) \quad \text{for all } i \in N \text{ and for all } j \in F_i. \tag{24}$$

[1] The London, Ont. data set can be obtained upon request from the corresponding author.
[2] The Alberta data set is available at the web site: http://www.bus.ualberta.ca/eerkut/testproblems/.

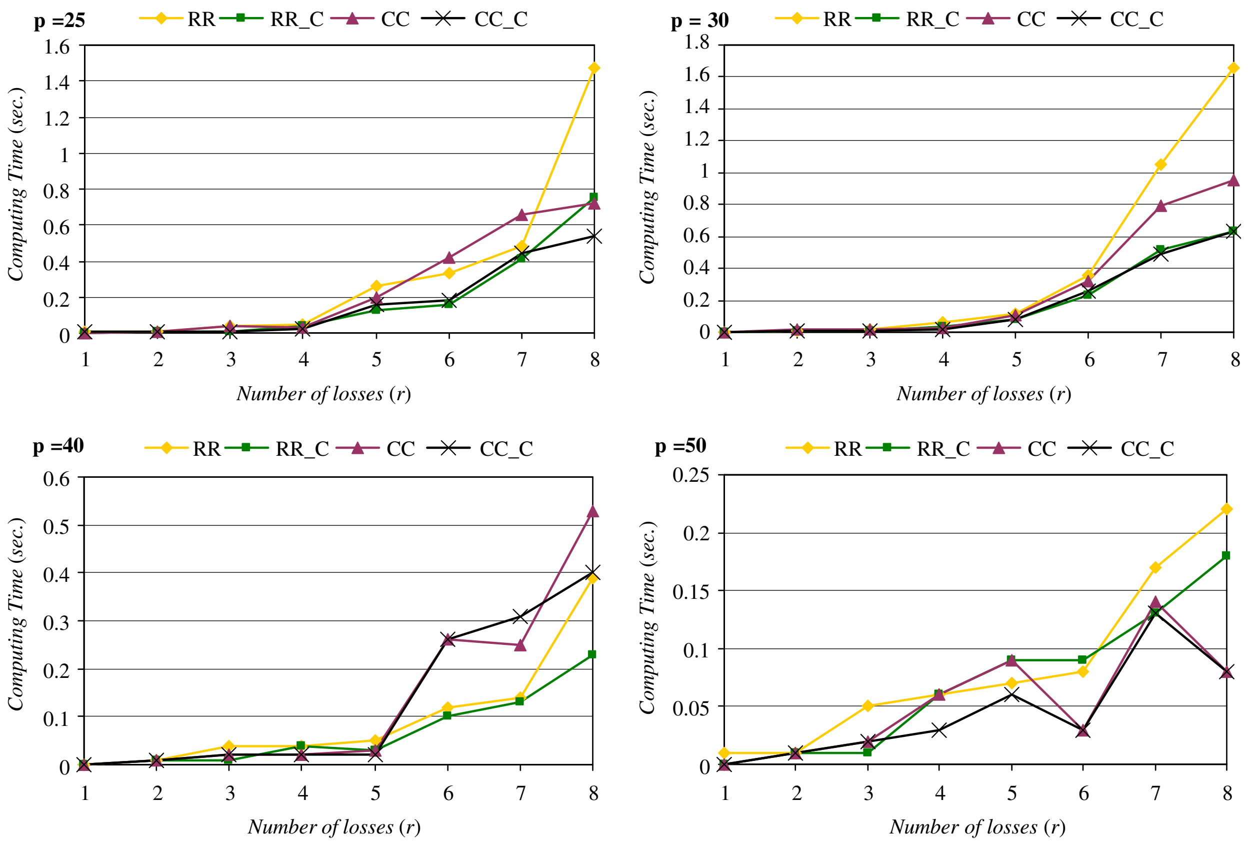

Fig. 3. Impact of COBRA reduction and CA constraints on computing time for the London, Ont. data set.

5.4. Computational comparison among different RIM implementations

In this section, we provide additional computational evidence of the benefits derived from the application of COBRA variable reduction to the RIM formulation. It is shown that this efficiency gain is independent of the specific type of CA constraints used. We also compare the relative efficiency of the two forms of CA constraints (RR and CC). The results are summarized in Figs. 3 and 4, which illustrate the impact of the different formulation options on the computing time needed to solve the London, Ont. problem and the Alberta problem, respectively. The options considered include RR and CC constraints with and without COBRA reduction. The resulting four combinations are denoted as RR, CC, RR_C and CC_C, where the C after the underscore indicates the incorporation of COBRA in the formulation. In each figure we compare the computing times in seconds for solving problem instances with four different values of p (namely, $p = 25$, 30, 40 and 50) and values of r ranging between 1 and 8.

From the analysis of the graphs, it is easy to see that the use of COBRA is always beneficial. The impact is especially remarkable in combination with the RR constraints: whereas the RR version is decidedly dominated by the CC version, when the COBRA reduction is implemented the two formulations show somewhat similar behaviors. RR_C is the best option in a few cases (e.g., London data set with $p = 40$ and Alberta data set with $p = 25$ and large values of r). Even though a clear dominance cannot be established between RR_C and CC_C, the CC_C version seems to outperform RR_C for the largest values of p and r (i.e., $p = 50$ and r between 6 and 8). This tendency is confirmed in the results reported in the next section, where we compare the efficiency of the two forms of CA constraints within the tree search procedure in solving RIMF. There are only a few cases, usually occurring when the CC constraints are used and for small values of r, in which the COBRA consolidation does not produce significant time reductions. This behavior can be attributed to the time needed for identifying the replacement variables. In small problems, in fact, the time savings derived from solving a reduced problem does not offset the amount of time needed to perform the variable substitutions.

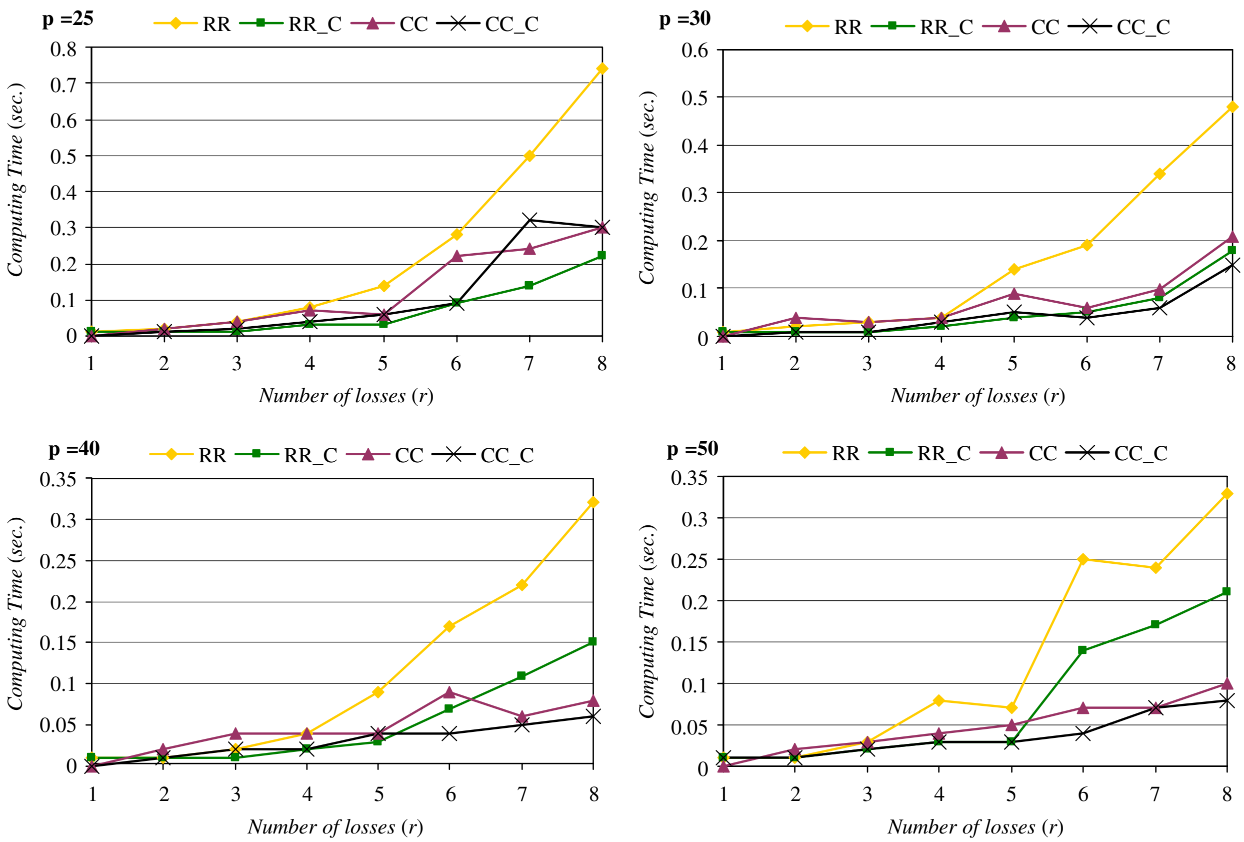

Fig. 4. Impact of COBRA reduction and CA constraints on computing time for the Alberta data set.

This minor limitation of COBRA consolidation, however, is completely overcome when RIM is used within the tree search procedure. In this case, in fact, the COBRA time saving is propagated throughout the tree exploration, whereas the variable replacement is performed only once, when RIM is solved for the first time at the root node. Overall, the introduction of COBRA generated enormous time savings in the IE algorithm. Therefore, in the next section we will only present the results for the implementation which include this option.

6. Computational results for solving RIMF using IE

Our computational experience is aimed at: (1) validating the findings outlined in the previous section through further investigation of the impact of the CA constraints on the overall performance of the tree search procedure; (2) comparing the performance of the tree search procedure with the MCPC approach described in Scaparra and Church [3] with the specific purpose of identifying relative strengths and weaknesses of the two approaches; (3) analyzing the impact of increasing the fortification resources on the level of protection achieved. Finally, we will briefly discuss our initial assumption of fixing the number of possible facility losses to r and show how the results obtained with this restriction may provide useful information to cope with a random number of possible losses.

6.1. Impact of the CA constraints on the overall approach

Empirical tests conducted to study the impact of the CA constraints on the IE algorithm performance demonstrated that the two constructs are equally efficient for solving problems with modest offensive resources (i.e., when $r = 1, 2, 3, 4$). The difference in computing time between the two formulations was practically negligible for these values of r and varied values of p and q, with the RR version running slightly faster in a few cases. However, as r increases, the tree

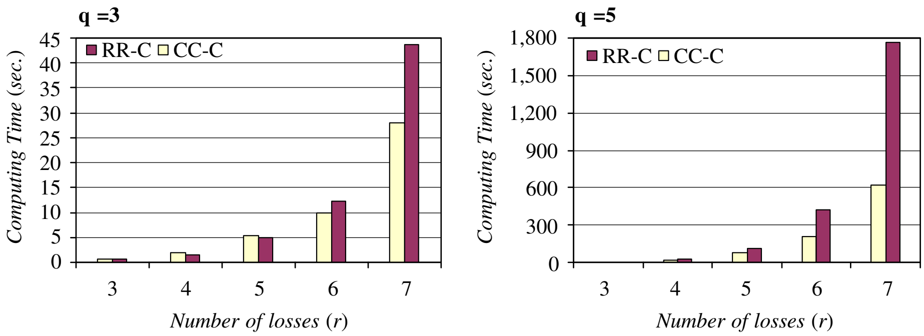

Fig. 5. Time comparison between CC and RR constraints for the London, Ont. data set with $p = 25$.

search version which uses the CC constraints becomes steadily better, and the computing time difference between the two approaches boosts rapidly with each increment of the r value.

An example of this behavior is depicted in Fig. 5, for the London data set with 25 initial facilities and two different values of fortification resources (namely, $q = 3$ and 5). The graph shows the computing times obtained with the two constructs for different values of r. From the graph, it is evident that, although the RR version is a competitive approach for small r, the CC formulation is to be preferred when a large number of possible losses are considered. The same tendency was observed in the results obtained for several different combinations of the parameters p, q and r, which are not reported for the sake of brevity. In light of these results, we restricted the subsequent analysis to the RIM formulation that uses the CC constraints.

6.2. Tree search vs. MCPC approach

In this section we compare the computational performances of the implicit enumeration approach and the maximum covering with precedence constraints approach (MCPC) proposed in Scaparra and Church [3]. The experiments were conducted on the larger set of problem instances used in Scaparra and Church [3], with the only difference that we allow up to eight interdictions instead of seven. The experiments include tests on the London and Alberta data sets with 25 and 30 existing facilities, and up to seven fortifications. The results are displayed in Table 1 where, for each of the two data sets, we show the optimal objective function value and the computing times of the two approaches, under different combinations of the parameters p, q and r. The analysis of the results for the London data set (columns 3–5) shows that MCPC is generally much faster than IE when the interdiction resources are small. However, the MCPC approach is significantly more sensitive to variations of the parameter r. It is important to remember that the MCPC approach requires enumerating all possible interdiction patterns. Therefore, its performance and applicability are firmly tight to the total number of interdiction patterns. The MCPC performance is usually quite good when this number (p choose r) does not exceed a few million, but it starts deteriorating when this amount is exceeded. As an example, the MCPC computing time is still competitive when $p = 30$ and $r = 7$, resulting in a total number of 2,035,800 interdiction patterns, but it rises dramatically when r is increased to 8 (see Table 1, fifth column). With five fortification resources, MCPC required almost 17 h to solve the London problem, whereas IE solved it in around 10 min. With seven fortifications available, MCPC could not solve the problem. The limitation of the MCPC approach in solving problems with a large number of interdiction patterns is even more accentuated on the Alberta data set. As already noted in Scaparra and Church [3], this difficulty is due to the heterogeneous distribution of the demands in this set, which makes the set covering problems solved at intermediate steps of the MCPC algorithm much more difficult. On the other hand, a simple greedy algorithm is usually able to find good approximate solutions to the Alberta problems, making their study less interesting from a theoretical point of view (see [3], for an in-depth discussion of this point). Table 2 provides information about the size of the problems solved. For the two data sets, it shows the number of variables and constraints in the lower-level problem. The table clearly shows that the model reduction process outlined in Section 5 significantly reduces the size of the problems.

Given the IE robustness in scaling to bigger problems, we extended the empirical investigation by solving problem instances with larger numbers of facilities in the initial configuration. Table 3 displays results for the London data set

Table 1
Computational results for the London, Ont. and the Alberta data sets solved with different combinations of the parameters p, q and r

p	q	r	London			Alberta		
			Objective value	MCPC time	IE time	Objective value	MCPC time	IE time
25	3	4	153,638.54	0.08	1.86	57,178,804	0.28	0.59
25	3	5	164,458.35	0.30	4.99	62,785,799	1.89	1.69
25	3	6	174,942.60	1.89	9.91	67,647,091	14.03	3.50
25	3	7	188,282.97	9.67	26.89	73,255,693	161.68	6.49
25	3	8	205,611.14	163.95	41.53	79,218,189	–	11.05
25	5	4	143,058.40	0.14	7.13	43,265,666	0.59	2.11
25	5	5	151,559.19	1.16	22.61	46,077,394	7.19	8.41
25	5	6	162,485.24	12.97	55.71	49,452,379	97.88	22.94
25	5	7	171,987.27	85.56	170.76	52,315,922	409.87	47.11
25	5	8	181,881.35	321.72	246.03	55,254,840	–	196.17
25	7	4	137,307.81	0.89	19.11	31,746,528	2.89	6.38
25	7	5	147,589.13	7.14	80.89	34,993,344	37.59	34.80
25	7	6	156,685.61	57.56	209.96	37,921,475	334.39	109.90
25	7	7	164,595.84	353.88	616.69	40,381,720	1819.89	260.14
25	7	8	172,623.60	1428.89	980.67	42,886,753	–	896.69
30	3	4	121,378.81	0.16	2.69	46,518,667	0.61	0.59
30	3	5	132,032.99	1.13	9.36	52,028,642	4.47	2.11
30	3	6	140,618.50	8.63	24.34	56,889,934	26.81	4.59
30	3	7	152,969.85	56.84	43.97	60,993,485	516.12	8.37
30	3	8	164,159.70	821.67	72.31	65,509,587	–	19.19
30	5	4	118,060.47	0.25	16.31	36,974,712	1.05	2.34
30	5	5	128,667.35	1.70	70.70	40,150,939	12.31	9.71
30	5	6	137,061.54	16.36	195.29	42,611,184	410.13	27.52
30	5	7	146,299.89	199.84	436.55	45,728,325	–	70.06
30	5	8	155,709.62	60,614.35	693.39	48,904,552	–	182.76
30	7	4	114,789.52	0.70	61.00	27,604,698	3.98	7.81
30	7	5	121,953.59	16.44	357.04	30,064,943	144.86	40.52
30	7	6	130,678.87	121.89	994.23	32,569,976	1862.97	132.12
30	7	7	136,730.79	1529.27	2,382.11	34,767,116	–	425.46
30	7	8	144,073.46	–	4255.58	36,853,397	–	1333.44

with 40, 50 and 60 operating facilities. More specifically, Table 3 shows the objective function values and the CPU times obtained when r ranged between 2 and 5 and the fortification budget was chosen to be equal to constant proportions of the total number of facilities (namely, $q = 10\%$, 15% and 20% of p). All these instances were solved to optimality in a reasonable amount of computational time (ranging from fractions of seconds for the smallest parameter values to less than 4 h for the largest parameter combination). The number of problems solved during the tree search varied between 22 for the smallest instance ($p = 40$, $q = 4$ and $r = 2$) and 272,460 for the biggest instance ($p = 60$, $q = 12$ and $r = 5$) (Fig. 6).

The proposed approach was able to solve problems with even larger values of the parameter r, which we do not report for the sake of brevity. Just as an indication of the computational effort derived from using bigger values of r, solving the London problem with 50 initial facilities and seven interdictions required between less than a minute for small values of q to around 6 h when the number of fortifications was increased to around 15% of the total number of facilities (i.e., $q = 8$). The size of these problem instances could not have been handled by the MCPC approach.

6.3. Impact of protective resources

We now discuss the effect of adding additional protective resources on the total efficiency. To this end, Fig. 6 shows the percentage marginal improvements in efficiency (or distance or cost) derived from any individual fortification. The graph summarizes the results for the London data set with 40, 50 and 60 facilities in the initial configuration. We let r vary between 1 and 5, and consider the marginal contributions of up to 10 fortifications. This information sheds light on possible tradeoffs between the cost of protecting additional facilities and the efficiency gain in case of worst-case system

Table 2
Summary of the lower-level interdiction problem sizes in terms of number of variables and constraints for the two data sets

p	q	r	London				Alberta			
			Before reduction		After reduction		Before reduction		After reduction	
			#Vars.	#Constrs.	#Vars.	#Constrs.	#Vars.	#Constrs.	#Vars.	#Constrs.
25	3	4	3775	3901	391	752	7925	8217	401	1566
25	3	5	3775	3901	493	902	7925	8217	504	1894
25	3	6	3775	3901	601	1052	7925	8217	609	2209
25	3	7	3775	3901	696	1202	7925	8217	707	2526
25	3	8	3775	3901	797	1352	7925	8217	798	2841
25	5	4	3775	3901	391	752	7925	8217	401	1566
25	5	5	3775	3901	493	902	7925	8217	504	1894
25	5	6	3775	3901	601	1052	7925	8217	609	2209
25	5	7	3775	3901	696	1202	7925	8217	707	2526
25	5	8	3775	3901	797	1352	7925	8217	798	2841
25	7	4	3775	3901	391	752	7925	8217	401	1566
25	7	5	3775	3901	493	902	7925	8217	504	1894
25	7	6	3775	3901	601	1052	7925	8217	609	2209
25	7	7	3775	3901	696	1202	7925	8217	707	2526
25	7	8	3775	3901	797	1352	7925	8217	798	2841
30	3	4	4530	4651	443	752	9510	9797	489	1571
30	3	5	4530	4651	556	902	9510	9797	608	1882
30	3	6	4530	4651	668	1052	9510	9797	733	2207
30	3	7	4530	4651	781	1202	9510	9797	861	2523
30	3	8	4530	4651	886	1352	9510	9797	979	2840
30	5	4	4530	4651	443	752	9510	9797	489	1571
30	5	5	4530	4651	556	902	9510	9797	608	1882
30	5	6	4530	4651	668	1052	9510	9797	733	2207
30	5	7	4530	4651	781	1202	9510	9797	861	2523
30	5	8	4530	4651	886	1352	9510	9797	979	2840
30	7	4	4530	4651	443	752	9510	9797	489	1571
30	7	5	4530	4651	556	902	9510	9797	608	1882
30	7	6	4530	4651	668	1052	9510	9797	733	2207
30	7	7	4530	4651	781	1202	9510	9797	861	2523
30	7	8	4530	4651	886	1352	9510	9797	979	2840

disruptions. Usually, most of the protection benefit is achieved with the first two or three fortifications (they typically contribute more than 50% of the overall improvement), whereas subsequent security investments produce progressively lower efficiency gains. In general, the fortification of the second facility still yields significant improvements. This is the case, for instance, of the problem instances with 50 operating facilities. In the specific case of $r = 3$, for example, the protection of only one facility, albeit the optimal one, only increases the efficiency level by less then 0.5% as compared to the worst-case loss when no protective measures are taken (the worst-case total weighted distance is reduced from approximately 70,249 units to 69,923 units). However, by only hardening one extra facility, the total weighted distance can be improved by an additional 4%, dropping to nearly 67,343 units. In any case, there is always a benefit in increasing protection expenditure by hardening additional facilities. Even the last fortification ($q = 10$) sometimes results in a 2% efficiency enhancement. This could represent a significant gain considering the order of magnitude of the service costs sustained in many distribution systems. As expected, the impact on system efficiency of each individual fortification generally tends to increase with the extent of a possible attack (i.e., as r increases) and to diminish as we consider larger systems (i.e., for larger values of p).

6.4. Fixed vs. probabilistic losses

Our model assumption of fixing the offensive resources to exactly r interdictions might seem quite questionable given that the extent of terrorist and man-made attacks is always characterized by a large degree of uncertainty. Nevertheless,

Table 3
Computational results for the London, Ont. data set solved for larger values of p

p	q[a] (%)	Objective value				Running time (s)			
		$r=2$	$r=3$	$r=4$	$r=5$	$r=2$	$r=3$	$r=4$	$r=5$
40	10	75,676.41	81,766.35	88,495.16	94,687.71	0.20	2.67	15.30	58.71
	15	75,418.05	81,424.85	87,170.59	93,286.72	0.52	11.10	110.76	476.28
	20	74,847.58	80,370.63	86,182.77	91,664.38	1.30	53.23	797.86	3467.25
50	10	60,168.50	65,160.41	69,918.29	74,694.85	0.25	4.42	22.86	117.28
	15	59,225.56	63,552.59	68,302.73	73,055.22	1.14	26.51	197.98	1665.66
	20	58,553.08	62,261.17	67,026.53	71,140.40	2.11	77.23	664.77	7441.07
60	10	46,563.64	50,809.54	54,621.16	58,615.76	0.70	9.53	45.11	204.12
	15	45,889.14	49,697.61	53,509.22	56,932.06	1.44	65.67	351.38	2229.49
	20	45,310.07	48,814.47	52,011.79	55,469.28	2.55	254.38	1568.08	13,088.10

[a]q is given as a percentage of the total number of facilities, rounded to the nearest integer.

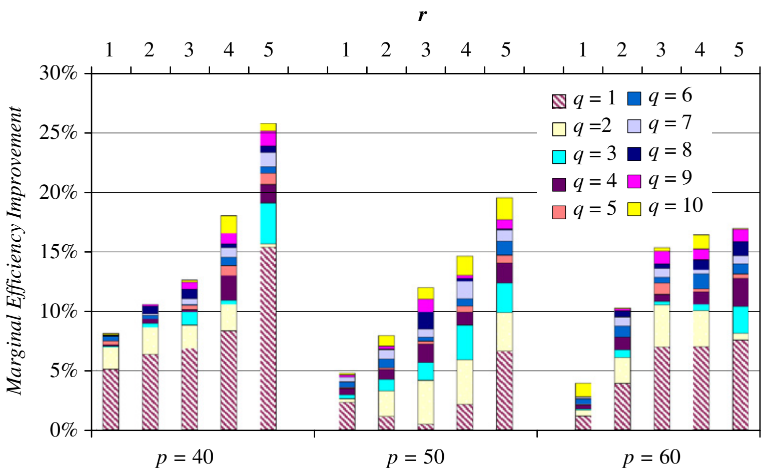

Fig. 6. Marginal percentage improvement in efficiency due to any additional fortification.

Table 4
Optimal set of six fortifications for the London, Ont. data set with 30 facilities and various numbers of interdictions

r	Optimal fortification set ($q=6$)					
1	83	91	92	100	141	149
2	41	83	91	92	141	149
3	41	83	91	92	141	149
4	41	83	91	92	103	149
5	41	83	91	92	103	149
6	20	83	92	116	141	149

our approach provides a powerful tool for identifying best possible fortification strategies in response to attacks of variable size. From the analysis of the results obtained with different values of r, in fact, we can infer which system components need to be protected under different scenarios. As an example, consider the solutions to the bilevel problem for the London data set with 30 facilities and protection resources to harden six of them. Table 4 shows the optimal fortification sets when the bilevel program is solved for different values of r, ranging between 1 and 6. It is easy to see that the optimal fortification patterns are very similar to each other, with some key facilities occurring in all of them.

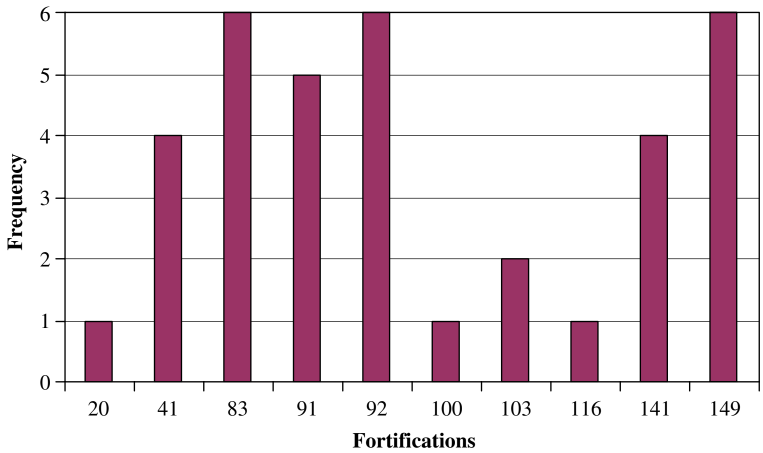

Fig. 7. Frequency with which each facility appears in a fortification set in Table 4.

Table 5
Optimal set of six fortifications for the London, Ont. data set with 60 facilities and various numbers of interdictions

r	Optimal fortification set ($q = 6$)					
1	8	13	19	26	99	144
2	13	19	26	89	92	99
3	13	19	47	73	99	141
4	13	26	36	92	103	144
5	13	26	36	92	103	144
6	13	26	36	47	56	144

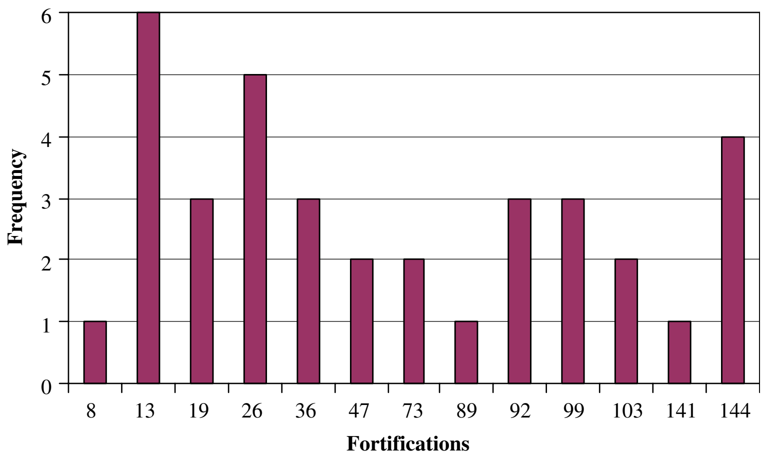

Fig. 8. Frequency with which each facility appears in a fortification set in Table 5.

The frequency with which each facility occurs in the six fortification patterns is depicted in Fig. 7. There are three facilities (83, 92 and 149) which appear in every optimal fortification set and which, consequently, should be fortified independently of the extent of an anticipated attack. Furthermore, the optimal set of facilities to harden is exactly the same if we consider the worst-case loss of two or three facilities (rows $r = 2$ and 3 in Table 3). The optimal protection against the worst-case interdiction of four or five facilities also requires the fortification of the same set of facilities (rows $r = 4$ and 5), and this set differs from the previous one only by one facility (it includes facility 103 instead of 141). This analysis can then be used to find core sets of key facilities to harden for each range of possible losses, and eventually, to identify good tradeoff solutions in view of a random number of losses.

Obviously, the problem becomes more complicated when we consider larger systems (see, for instance, Table 5 and Fig. 8, which provide the same information as Table 4 and Fig. 7 for a system with 60 operating facilities). As the number of facilities increases, there is a greater variability in terms of the fortifications which are needed to thwart attacks of different sizes. Overall, 13 different facilities appear at least in one of the six fortification patterns and only one (facility 13) appears in all of them. Although even for this case we can draw valuable information about which facilities should be protected, new mathematical models need to be developed which explicitly take into account expected numbers of losses in large systems.

7. Conclusions

This paper has presented a new approach for solving the RIMF, utilizing a bilevel programming model formulation solved by an implicit enumeration process. The efficiency of the proposed process rests in part on being able to solve the "interdictor" problem efficiently. Overall, the tree search process has been used to solve relatively large facility system problems, problems that could not be solved optimally by other techniques. Example results indicate that this model can be used to develop a cogent protection strategy for an existing system and that protection, even for a few key facilities, can significantly increase the resilience of a system when attacked.

While this analysis represents an important first step towards understanding and modeling logistics system reliability and protection, a number of important issues still needs to be explored in order to capture additional realism. As an example, future research should be directed towards expanding the model framework to include probabilistic elements. In our analysis, we assume that an attack on a given facility is always successful, that the number of possible interdiction is known to the protector and that a protected facility is immune to interdiction. These assumptions should be relaxed to model situations where attacks are successful only with a given probability or only cause partial disruption, where the number of possible losses is known only probabilistically and where protection may only reduce the probability of successful attacks. Also new models should be developed which include capacitated facilities, multiechelon systems and expected losses in addition to worst-case losses. All these models are currently under investigation.

Acknowledgments

This work was partially supported by the Research and Special Projects Agency of USDOT. We wish to thank Mike Goodchild of UCSB, Val Noronha of DGRC, Inc. and K.T. Thurmali of USDOT for their comments and encouragement. Thanks are due to two referees for their valuable comments.

References

[1] Church RL, Scaparra MP. Protecting critical assets: the r-interdiction median problem with fortification. Geographical Analysis, 2006, forthcoming.
[2] Hakimi SL. Optimum location of switching centers and the absolute centers and medians of a graph. Operations Research 1964;12:450–9.
[3] Scaparra MP, Church RL. An optimal approach for the interdiction median problem with fortification. Working paper no. 78, Kent Business School, 2005. University of Kent, UK (Available at ⟨http://www.kent.ac.uk/kbs/research-information/working-papers/⟩).
[4] Wollmer R. Removing arcs from a network. Operations Research 1964;12:934–40.
[5] Wood RK. Deterministic network interdiction. Mathematical and Computer Modeling 1993;17:1–18.
[6] Church RL, Scaparra MP, Middleton RS. Identifying critical infrastructure: the median and covering facility interdiction problems. Annals of the Association of American Geographers 2004;94(3):491–502.
[7] Brown GG, Carlyle WM, Salmeron J, Wood K. Analyzing the vulnerability of critical infrastructure to attack and planning defenses. In: Greenberg HJ, series editor. Smith JC. volume editor, Tutorial in operations research: emerging theory, methods, and applications, INFORMS 2005. 2005.
[8] Hansen B, Jaumard B, Savard G. New branch and bound rules for linear bilevel programming. SIAM Journal on Scientific and Statistical Computing 1992;13:1194–217.
[9] Salmeron J, Wood K, Baldick R. Analysis of electric grid security under terrorist threat. IEEE Transactions on Power Systems 2004;19(2): 905–12.
[10] Jüttner U, Peck H, Christopher M. Supply chain risk management: outlining an agenda for future research. International Journal of Logistics: Research and Applications 2003;6(4):197–210.

[11] Sheffi Y. Supply chain management under the threat of international terrorism. International Journal of Logistics Management 2001;12(2):1–11.
[12] Rice JB, Caniato F. Building a secure and resilient supply network. Supply Chain Management Review 2003;7(5):22–30.
[13] Ball MO, Colbourn CJ, Provan JS. Network reliability. In: Ball MO, Magnanti TL, Monma CL, Nemhauser GL, editors. Network models, handbook of operations research and management science, vol. 7. Elsevier: Amsterdam; 1995. p. 673–762.
[14] Raghavan S, Magnanti TL. Network connectivity. In: Dell'Amico M, Maffioli F, Martello S, editors. Annotated bibliography in combinatorial optimization. New York: Wiley; 1997. p. 335–54.
[15] Balakrishnan A, Magnanti TL, Mirchandani P. Connectivity-splitting models for survivable network design. Networks 2003;43(1):10–27.
[16] Drezner Z. Heuristic solution methods for two location problems with unreliable facilities. Journal of the Operational Research Society 1987;38(6):509–14.
[17] Lee S. On solving unreliable planar location problems. Computers and Operations Research 2001;28:329–44.
[18] Bundschuh M, Klabjan D, Thurston DL. Modelling robust and reliable supply chains. Working paper, University of Illinois at Urbana-Champaign, 2003.
[19] Snyder LV, Daskin MS. Reliability models for facility location: the expected failure cost case. Transportation Science 2005;39(3):400–16.
[20] O'Hanley JR. Optimizing reserve selection planning in the face of uncertain site loss. PhD thesis, Department of Environmental Science, Policy and Management, University of California, Berkeley, 2005.
[21] Cormican KJ, Morton DP, Wood RK. Stochastic network interdiction. Operations Research 1998;46(2):184–97.
[22] Fulkerson DR, Harding GC. Maximizing the minimum source-sink path subject to a budget constraint. Mathematical Programming 1977;13:116–8.
[23] Israeli E, Wood RK. Shortest-path network interdiction. Networks 2002;40:97–111.
[24] Lim C, Smith JC. Algorithms for discrete and continuous multicommodity flow network interdiction problems, IIE Transactions 2006, to appear.
[25] Israeli E. System interdiction and defense. PhD dissertation, Operations Research Department, Naval Post Graduate School, Monterey, CA, 1999.
[26] Stackelberg H. The theory of market economy. Oxford: Oxford University Press; 1952.
[27] Church RL, Cohon JL. Multiobjective location analysis of regional energy facility siting problems. Report prepared for the US Energy Research and Development Administration (BNL 50567), 1976.
[28] Hanjoul P, Peeters D. A facility location problem with clients' preference orderings. Regional Science and Urban Economics 1987;17:451–73.
[29] Bard JF. Practical bilevel optimization. The Netherlands: Kluwer Academic Publishers; 1998.
[30] Dempe S. Foundations of bilevel programming. The Netherlands: Kluwer Academic Publishers; 2002.
[31] Moore JT, Bard JF. The mixed integer linear bilevel programming problem. Operations Research 1990;38:911–21.
[32] Vicente L, Savard G, Judice J. Discrete linear bilevel programming problem. Journal of Optimization Theory and Applications 1996;89:597–614.
[33] Karlof JK, Wang W. Bilevel programming applied to the flow shop scheduling problem. Computers and Operations Research 1996;23:443–51.
[34] Morton DP, Pan F., Saegar KJ. Models for nuclear smuggling interdiction. In: Higle JL, Römisch W, editors. Stochastic Programming E-Print Series (SPEPS); 2006-5.
[35] Ahuja RK, Magnanti TL, Orlin JB. Networks flows. Englewood Cliffs, NJ: Prentice-Hall; 1993.
[36] Church RL. COBRA: a new formulation of the classic p-median location problem. Annals of Operations Research 2003;122:103–20.
[37] Goodchild M, Noronha V. Location-allocation for small computers. Monograph No. 8., Department of Geography, The University of Iowa, Iowa City, 1983.
[38] Alp O, Drezner Z, Erkut E. An efficient genetic algorithm for the p-median problem. Annals of Operations Research 2003;122:21–42.
[39] Gerrard RA, Church RL. Closest assignment constraints and location models: properties and structure. Location Science 1996;4(4):251–70.
[40] Rojeski P, ReVelle CS. Central facilities location under an investment constraint. Geographical Analysis 1970;2:343–60.