Kent
Business
School

ISSN 1748-7595 (Online)

Working Paper Series

A Bi-level Mixed Programme for Critical Infrastructure Protection Planning

Maria P. Scaparra
Richard L. Church
Kent Business School

Working Paper No. 116
Jan. 2006

A bilevel mixed integer program for critical infrastructure protection planning

Maria P. Scaparra

Kent Business School
University of Kent, Canterbury
Canterbury, Kent, UK

Richard L. Church

Department of Geography
University of California, Santa Barbara,
Santa Barbara, California, USA

July , 2005

Abstract

Vulnerability to sudden service disruptions due to deliberate sabotage and terrorist attacks is one of the major threats of today. Whereas the design of reliable supply networks has received considerable attention in the last few decades, few authors have addressed the issue of optimizing security investments for existing, but vulnerable, systems. In this paper we consider a simple service/supply system with p facilities and a set of costumers, where each customer receives service from its closest facility. In this paper, we assume that there is an antagonist who will attempt to do the most harm to the service system and that our objective is to efficiently allocate security investments so that the effects of a worst-case attack are minimized. As this represents a problem with two different and opposing goals (one of protection and one of inflicting harm), we have cast this problem as a bilevel programming model where the top level problem involves the decisions about which facilities to secure or harden and the lower level problem entails the interdictor response on which unprotected facilities to attack. We solve the bilevel problem through an implicit enumeration algorithm, based on a binary tree search. The algorithm involves solving iteratively r-interdiction median (RIM) models. We show how the original RIM formulation can be streamlined through a process of variable reduction and consolidation and this significantly reduces the computing time needed to solve the bilevel fortification/interdiction model. Extensive computational results are reported, including comparisons with earlier results obtained by a single-level approach to the problem.

1. Introduction

Recent world events, including the dramatic terrorist attacks on the World Trade Center and the Pentagon, have raised the issue of service/supply system vulnerability into sharp focus and have posed a new challenge to devise sound procedures for increasing system security. The challenge is even more compelling, when considering the complexity that characterizes today's logistics networks. The close interrelationship and interdependence among a large number of system elements measurably increases the exposure to intentional harm and the level of vulnerability. It also increases the difficulty of assessing the impact of losing some of the system components as well as identifying the most effective protective measures. The need of systematic and analytical tools for addressing the issues of systems vulnerability, security investment and the design of resilient networks has been widely recognized among academics and practitioners (Juttner et al. 2003). Nevertheless, the study of mathematical models and techniques for improving systems 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 Sheffi, 2001, and Rice and Caniato, 2003). Yet, little attention has been paid to the development of quantitative methods and systematic approaches for improving production and distribution system security.

It is important to recognize that several recent papers address features of reliability/security. Among them, Bundschuh et al. (2003) present several mathematical models for improving reliability and robustness in supply chains through the optimal choice of suppliers. In addition Snyder and Daskin (2005) have proposed several reliability models to find the optimal location of facilities so as to minimize regular operation costs as well as the expected costs incurred when some of the facilities are unavailable. Finally, O'Hanley (2005) presents a novel model for the design of robust, coverage-type service networks. 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 be preferable over the short term. A first step in this direction is to identify the critical components of a system, i.e. those elements that, if lost, hamper the system's ability to continue operations the most.

The first mathematical models aimed at identifying network vulnerabilities were developed for military applications. Military planners, in fact, have had a long-term interest in identifying critical elements so that they could allocate strike resources to inflict the greatest harm to an enemy. A review of these early "interdiction" models, which mainly focus on the impact of the loss of a one or more transportation links (or arcs) in transportation networks, is provided in Church et al. (2004). Church et al. (2004) also developed two new spatial optimization models, called the median facility interdiction model and the covering facility interdiction model, which identify the set of supply or emergency response facilities which, if lost, disrupt service delivery the most. In the former model, disruption is measured in terms of the loss of system efficiency and in the later model disruption is measured in terms of the loss of demand coverage.

Interdiction models can help reveal potential weaknesses in a system. However, they do not explicitly address the issue of optimizing security. For example, if it is possible to secure one or more facilities from interdiction, which ones should be fortified or made secure? If the process of securing a facility is inexpensive, then it makes sense to secure all facilities. However, if the costs of security are expensive and protection resources are limited then the question just posed is an important one to address. It is easy to demonstrate that securing those facilities that are identified as critical in an optimal interdiction solution will not necessarily provide the greatest protection against an intelligent antagonist (Church and Scaparra, 2005; Israeli, 1999). Optimal interdiction is a function of what is fortified, so it is important to capture this interdependency within a modeling framework. The remainder of this paper addresses the issue of optimizing security or fortification to a set of existing facilities in order to thwart as much as possible the effects of interdiction. Our work here is based upon the assumption that resources to fortify are limited and only a subset of facilities can be made secure. This paper addresses a system based upon the p-median system framework. We call this problem the *r*-interdiction median problem with fortification (RIMF) as the problem entails optimal fortification to mitigate interdiction losses.

In the next section we define in greater detail the RIMF problem. Following that we present a bilevel programming formulation of this problem and then present a solution approach that involves a specialized tree search algorithm. To make this process efficient, a special formulation of the lower level problem (i.e. the interdictor problem) of the bilevel model has been developed.

We present details of this special condensed formulation. Then computational results are given, followed by a set of conclusions and recommendations for future research.

2. The problem of allocating limited fortification resources against interdiction

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 also assume that hardening a facility will protect a facility to the extent that an interdictor will only select non-fortified ones. Finally, 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. We can then define the overall fortification problem as:

> Identify the set of q facilities to secure or harden, so that after interdiction, the remaining system operates as efficiently as possible.

Fortification or asset hardening is of considerable interest to facility planners, especially when they provide important goods and services. Salmeron et al. (2004) developed a model for identifying critical components of an electrical power grid and then discussed the problem of identifying which components to harden or protect. This concept was formalized by Church and Scaparra (2005) in a problem involving an existing supply/service system. It is important to recognize that the problem being constructed here involves the facilities planner and the antagonist or interdictor, each with opposing goals. The facilities planner wants to optimize security so that the system is less vulnerable to interdiction, and the interdictor will attempt to inflict the greatest harm. Thus, the value of any fortification plan will be computed on the basis of worst-case loss. We can calculate worst-case loss of r-facilities using the r-interdiction median (RIM) problem (Church et al. 2004) which can be defined formally as:

> Of the p *existing* locations of supply, find the subset of r facilities, which when removed, yields the highest level of weighted distance.

The RIM problem can be formulated as an integer-programming problem using the following additional notation:

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

$$x_{ij} = \begin{cases} 1, \text{ if demand } i \text{ assigns to a facility at } j \text{ after interdiction} \\ 0, \text{ otherwise} \end{cases}$$

$T_{ij} = \{k \in F \mid k \neq j \text{ and } d_{ik} > d_{ij}\}$, the set of existing sites (not including *j*) that are as far or farther than *j* is from demand *i*.

We can now formulate the r-interdiction median (RIM) problem as the following integer-programming problem (Church et al. 2004):

$$Max \quad Z = \sum_i \sum_{j \in F} a_i d_{ij} x_{ij} \tag{1}$$

Subject to:

$$\sum_{j \in F} x_{ij} = 1 \quad \text{for each demand } i \tag{2}$$

$$\sum_{j \in F} s_j = r \tag{3}$$

$$\sum_{k \in T_{ij}} x_{ik} \leq s_j \quad \text{for each } i \text{ and each } j \in F \tag{4}$$

$$\begin{aligned} x_{ij} &= 0,1 \quad \text{for each } i \text{ and each } j \in F \\ s_j &= 0,1 \quad \text{for each } j \in F \end{aligned} \tag{5}$$

The above model represents optimal interdiction when no security has been allocated. The objective involves maximizing the weighted distance or service cost after the removal of r-facilities. Constraint (2) specifies that each demand must assign to a facility after interdiction. Constraint (3) specifies that only r facilities are to be eliminated. Constraint (4) maintains that each demand must assign to their closest open facility after interdiction. This constraint basically allows an assignment to a further facility, only when that facility has been interdicted. This kind of closest assignment constraints was previously employed by Church and Cohon (1976) for siting energy facilities and by Hanjoul and Peeters (1987) in plant location models. Alternative constructs have been proposed in the literature to force closest assignment (CA). A more in-depth discussion of our particular choice will be provided in section 5.3. It is important to note that the RIM model can be used to analyze a given fortification plan by eliminating the s_j variables associated with any sites selected for fortification or setting the s_j variables to zero for facilities that have been selected for fortification.

The RIM model represents the interdictor. The interdictor attempts to do the greatest harm, while the systems planner attempts to thwart interdiction as best as possible through fortification. It is easy to observe that the RIMF problem can be described within a game theoretic framework as a leader-follower or Stackelberg game (Stackelberg, 1952). Such a game theoretic framework can be structured as a bilevel programming problem. In the next section we provide a bilevel formulation for the RIMF problem.

It is important to note that the RIMF was originally proposed as a problem by Church and Scaparra (2005). They assumed that interdiction resources were extremely limited, and this allowed them to structure a single level optimization model for RIMF. Unfortunately, only problems of very modest size could be solved through that formulation. In a subsequent work, Scaparra and Church (2005) developed an alternative single-level optimization model for RIMF, called MCPC. The new model was based upon a maximum covering type formulation and was solved to optimality after a specialized model reduction process. Although this model was significantly faster than the former one, it still presented the big limitation of requiring a complete enumeration of all possible ways of losing r of the p facilities. In this paper we propose an alternate solution methodology based upon a bilevel model that does not face such size restrictions. This is a major advancement for this type of security optimization problem.

3. Formulating the RIMF as a bilevel programming problem

To construct the facilities fortification problem, RIMF, as a discrete bilevel programming problem we need to model the fortification decisions in addition to the interdiction decisions. Consider then the following additional type of decision variable:

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

We can then eliminate the possibility of interdicting a fortified site in the interdiction model by maintaining:

$$s_j \quad \leq \quad 1 - z_j$$

The bilevel model then comprises decisions of the systems planner, z_j, the interdictor, s_j, and resulting system performance based upon demand assignment variables, x_{ij}. We can formulate this model as:

$$\min H(\mathbf{z}) \tag{6}$$

subject to

$$\sum_{j \in F} z_j = q \tag{7}$$

$$z_j \in \{0,1\} \quad \text{for all } j \in F \tag{8}$$

where

$$H(\mathbf{z}) = \max \sum_{i \in N} \sum_{j \in F} a_i d_{ij} x_{ij} \tag{9}$$

subject to

$$\sum_{j \in F} x_{ij} = 1 \quad \text{for all } i \in N \tag{10}$$

$$\sum_{j \in F} s_j = r \tag{11}$$

$$\sum_{h \in T_{ij}} x_{ih} \leq s_j \quad \text{for all } i \in N \text{ and for all } j \in F \tag{12}$$

$$s_j \leq 1 - z_j \quad \text{for all } j \in F \tag{13}$$

$$s_j \in \{0,1\} \quad \text{for all } j \in F \tag{14}$$

$$x_{ij} \in \{0,1\} \quad \text{for all } i \in N \text{ and for all } j \in F \tag{15}$$

The lower level program (9)-(15) is simply the r-interdiction median problem (RIM) defined earlier, but with the additional constraints (13) which prevent the interdiction of any sites chosen to be fortified in the upper level problem. More specifically, in the lower level problem, the follower/interdictor decides the values of the interdiction variables with the objective (9) of maximizing the total weighted distance between customers and facilities after the attack. Note that in practice integer restrictions for the lower level are needed only for the interdiction variables, since, at optimality, the assignment variables will be binary integers simply as a consequence of the integrality of the s_j variables.

The optimal objective function value, H, of the lower level problem defines the leader's objective (6), who tries to minimize this worst-case weighted distance by allocating fortification resources. The constraints in the upper level problem simply state that only q facilities can be fortified (7) and that the fortification variables must be integer (8). Finally, constraints (13) link the upper and lower level problem. In the remainder of the paper, we will refer to the lower level problem which include constraints (13) as conditional RIM (CRIM), due to the conditional nature of what has been fortified.

It is interesting to note that bilevel programming has been used quite recently as a construct for location problems involving competition (see Eiselt and Laporte, 1996, and Bhadury et al., 2003). The fortification problem represents another instance of where this construct is useful. Details on bilevel programming can be found in Bard (1998) and Dempe (2002). All general cases of bilevel programming models fall into the class of NP-hard (Hansen et al., 1992). 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. The difficulties encountered with the presence of integer restrictions in a bilevel format are explained in Moore and Bard (1990) and Vicente et al (1996). Overall, the difficulty in solving such a problem depends on: 1) the class of discrete bilevel programs, and 2) parameter position in the lower level problem. The RIMF problem can be classified as especially complex as integer restrictions appear at both levels of the problem. Research on solving discrete bilevel programming problems is very limited (see Moore and Bard (1990) and Karlof and Wang (1996) for examples) and thus the process proposed here for RIMF becomes one of the few techniques developed for such problems.

4. Solving RIMF as a bilevel problem

To solve the bilevel formulation (6)-(15) of the r-interdiction median problem with fortification we propose an implicit enumeration algorithm. The entire approach is built on a simple observation made by Church and Scaparra (2005) 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 (9)-(15) 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 6 operating facilities (numbered from 1 to 6), 2 interdictions and 2 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 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 can not 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). In the picture, 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.

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.

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 conditional RIM 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 implicit enumeration algorithm solves at most* $\frac{r^{q+1}-1}{r-1}$ *conditional RIM problems, where r is the number of interdiction 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 Ahuja et al., 1993). 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 conditional RIM 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, an 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 conditional RIM.

5. Solving RIM and CRIM efficiently

The computational effort of the proposed implicit enumeration procedure (IE) 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 introduced by Church et al. (2004) and given in section 2. The extension to the conditional RIM is straightforward, since constraints (13) 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 closest assignment 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 for the CRIM model as presented in the lower level program (9)-(15) and RIM in (1)-(5) 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 will occur if the r-closest facilities to that demand have been interdicted. This means that the worst case for a given demand i will occur when that demand assigns to its $r+1^{\text{st}}$ closest facility. This observation can be used to reduce the size of RIM by defining the following additional sets.

G_i $=$ the set of $r+1^{\text{st}}$ closest facilities to demand i before interdiction.

U_{ij} $=\left\{k \in G_i \mid k \neq j \text{ and } d_{ik} > d_{ij}\right\}$, the set of existing sites (not including j) that are as far or farther than j is from demand i, but not further than the $r+1^{\text{st}}$ closest site from i.

F_i $=$ the set of r closest sites to demand i before interdiction

RIM can then be reformulated as:

$$\max \sum_{i \in N} \sum_{j \in G_i} a_i d_{ij} x_{ij} \tag{16}$$

Subject to:

$$\sum_{j \in F} s_j = r \tag{17}$$

$$\sum_{j \in G_i} x_{ij} = 1 \quad \text{for all } i \in N \tag{18}$$

$$\sum_{h \in U_{ij}} x_{ih} \leq s_j \quad \text{for all } i \in N \text{ and for all } j \in F_i \tag{19}$$

$$s_j \in \{0,1\} \quad \text{for all } j \in F \tag{20}$$

$$x_{ij} \in \{0,1\} \quad \text{for all } i \in N \text{ and for all } j \in G_i \tag{21}$$

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 (19) 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 (2003) 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 (2003) demonstrated that two demands may assign 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 (2003).

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 sites closeness for any pair of demands. Based upon this equivalency, the original assignment variables can be replaced by a smaller set of variables, A. 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 parameter, 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:

$$\max \sum_{i\in N}\sum_{j\in G_i}\sum_{v\in A}\alpha_{ijv}a_i d_{ij}x_v \tag{22}$$

Subject to:

$$\sum_{j\in F} s_j = r \tag{23}$$

$$\sum_{j\in G_i}\sum_{v\in A}\alpha_{ijv}x_v = 1 \quad \text{for all } i\in N \tag{24}$$

$$\sum_{h\in U_{ij}}\sum_{v\in A}\alpha_{ihv}x_v \le s_j \quad \text{for all } i\in N \text{ and for all } j\in F_i \tag{25}$$

$$s_j \in \{0,1\} \quad \text{for all } j\in F \tag{26}$$

$$x_v \in \{0,1\} \quad \text{for all } v\in A \tag{27}$$

The computational benefits of the new condensed formulation have been tested on two different data sets: the 150 node London, Ontario data set (Goodchild and Noronha, 1983) and the 316 node Alberta data set (2003). We solved both problems with the mixed integer programming optimizer CPLEX 9.0, using different values of the parameters p and r. More specifically, we let p vary between 20 and 50 and r between 1 and 15. Also, we assume that the p facilities are initially located at the optimal p-median sites. 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 closest assignment constraints.

5.3. Closest Assignment Constraints

It was mentioned in section 2 that there are several ways of enforcing closest assignment (CA). A comprehensive discussion of the structural properties of closest assignment constraints in location problems is provided in Gerrard and Church (1996). 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 (1970) in the context of the budget constrained median problem. The Rojeski and ReVelle constraints are among the most widely cited closest assignment constraints and, as the Church and Cohon constraints (4), (12), (19) and (25) used in our RIM formulations, have the nice property of inherently yielding integral assignment variables. Other CA constructs (see Church and Gerrard, 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 expresses in a RR type form as follows:

$$x_{ij} \geq (1\text{-}s_j)\text{-} \sum_{h \in U_{ij}} (1\text{-}s_h) \text{ for all } i \in N \text{ and for all } j \in F_i, \tag{28}$$

where the set T_{ij} has the usual meaning of representing the set of all the facilities which are closer to demand i than facility j, but not further than the $r + 1$ closest site to i. Constraints (28) 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 ($\sum_{h \in U_{ij}} (1\text{-}s_h) = 0$), then demand i must assign to j. However, if any of the closer facilities is operational, the right hand side of (28) is always less or equal to zero and, hence, relation (28) has no effect on the assignment. Constraints (4) and (28) can be used interchangeably in the RIM formulation (1)-(5) to enforce closest assignment. When the COBRA version of RIM is considered, the CC constraints (25) can be replaced by the following modified version of the RR constraints (28), which take into account the variable substitutions:

$$\sum_{v \in A} \alpha_{ijv} x_v \geq (1\text{-}s_j)\text{-} \sum_{h \in T_{ij}} (1\text{-}s_h) \text{ for all } i \in N \text{ and for all } j \in F_i. \tag{29}$$

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 closest assignment constraints used. We also compare the relative efficiency of the two forms of CA constraints (RR and CC). All results reported in this and the next were run on a PC with a Pentium 4, 2.8Ghtz processor and 1GB of RAM. Each RIM was solved with the branch and bound based solver CPLEX 9.0, supplied with specific directives to improve performance. The results are summarized in Fig. 3 and Fig. 4, which illustrate the impact of the different formulation options on the computing time needed to solve the London 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. 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 implicit enumeration 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 implicit enumeration

The proposed tree search approach was coded in C++ and associated RIM and CRIM problems were solved with the branch and bound based solver CPLEX 9.0. The computational experience was 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 (2005) 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 closest assignment 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 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 $q = 5$). 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 is 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 (IE) and the maximum covering with precedence constraints approach (MCPC) proposed in Scaparra and Church (2005). The experiments were conducted on the larger set of problem instances used in Scaparra and Church (2005), with the only difference that we allow up to 8 interdictions instead of 7. The experiments include tests on the London and Alberta data sets with 25 and 30 existing facilities, and up to 7 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 to 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 5 fortification resources, MCPC required almost 17 hours to solve the London problem, whereas IE solved it in around 10 minutes. With 7 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 (2005), 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 Scaparra and Church, 2005, for an in-depth discussion of this point).

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 2 displays results for the London data set with 40, 50 and 60 operating facilities. More specifically, Table 2 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 hours for the largest parameter combination). 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 7 interdictions required between less than a minute for small values of q to around 6 hours 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.

6.2. Impact of protective resources

We now discuss the effect of adding additional protective resources on the total efficiency. To this end, Fig. 4 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 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).

6.3. 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, 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 6 of them. Table 3 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. The frequency with which each facility occurs in the 6 fortification patterns is depicted in Fig. 5. There are three facilities (83, 92 and 149) which appear in every optimal fortification set and which, consequently, must 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 2 or 3 facilities (rows $r = 2$ and $r = 3$ in Table 3). The optimal protection against the worst case interdiction of 4 or 5 facilities also requires the fortification of the same set of facilities (rows $r = 4$ and $r = 5$), 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 4 and Fig. 6, which provide the same information as Table 3 and Fig. 4 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 6 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. Developing this kind of models will be the subject of future research.

7. Summary and conclusions

Vulnerability to sudden service disruptions due to deliberate sabotage and terrorist attacks is an issue of growing importance. There are two principal ways in which to design for higher levels of system reliability in the event of an attack. The first is to design and build a system from scratch that is as resilient as possible, in anticipation of certain attack scenarios. The second approach is to harden an existing system, by optimizing the allocation of security resources. This paper addresses the second approach and involves a service/delivery system based upon a p-median framework. In this paper we assume that the facility system already exists and that limited resources are available to protect some of the facilities from harm by an antagonist. We assume that the antagonist/interdictor is intelligent and will inflict the greatest harm, given a level of attack resources. The model that addresses this problem is called the r-interdiction median problem with fortification (RIMF)(Church and Scaparra, 2005). This paper has presented a new approach for solving the RIMF problem, utilizing a bilevel programming model formulation solved by an implicit enumeration process. This efficiency of this process rests in part on being able to solve the "interdictor" problem efficiently. This paper addresses how the "interdictor" problem, called CRIM, can be solved efficiently by a specially condensed problem formulation. Details of this model, variations of two forms of closest assignments constraints, and the tree search process are presented and numerous computational examples are used to show the efficacy of different possible model formats in the tree search algorithm. Overall, this 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.

This modeling work focuses on securing as best as possible an existing system. Practically speaking, this is the type of problem faced by system managers, who need to focus on keeping a system in operation and who can devote some resources to increasing security. Even though interdiction can significantly degrade service, results indicate that optimal protection strategies, even for a few key facilities, can increase the resilience when attacked. Future research should be directed towards expanding the model framework to include capacitated facilities, supply chains, and expected losses in addition to worst-case losses.

Acknowledgments

This work was partially supported by the Research and Special Projects Agency of U.S.D.O.T. We wish to thank Mike Goodchild of UCSB, Val Noronha of DGRC, Inc., and K.T. Thurmali of U.S.D.O.T. for their comments and encouragement.

References

Ahuja, R.K., T.L. Magnanti, J.B. Orlin. 1993. *Networks Flows,* Prentice-Hall. Englewood Cliffs, New Jersey.

Alp, O., Z. Drezner, and E. Erkut, An Efficient Genetic Algorithm for the *p*-Median Problem, Annals of Operations Research 122 (2003), 21-42.

Bard, J. F. 1998. *Practical Bilevel Optimization*. Kluwer Academic Publishers, The Netherlands.

Bhadury, J., Eiselt, H.A., Jaramillo, J.H. 2003. An alternating heuristic for medianoid and centroid problems in the plane. *Computers and Operations Research* 30 553-565.

Bundschuh, M., D. Klabjan, D.L. Thurston. 2003. Modelling robust and reliable supply chains. Working paper, University of Illinois at Urbana-Champaign.

Church, R.L. 2003. COBRA: A new formulation of the classic *p*-median location problem. *Annals of Operations Research* 122 103-120.

Church, R.L., J.L, Cohon. 1976. Multiobjective location analysis of regional energy facility siting problems. Report prepared for the U.S. Energy Research and Development Administration (BNL 50567).

Church, R.L., M.P., Scaparra. 2005. Protecting Critical Assets: The *r*-interdiction median problem with fortification. Working Paper No. 79. Kent Business School, University of Kent, UK.

Church, R.L., M.P. Scaparra, R.S., Middleton. 2004. Identifying critical infrastructure: The median and covering facility interdiction problems. *Annals of the Association of American Geographers* 94(3) 491-502.

Dempe, S. 2002. *Foundations of Bilevel Programming.* Kluwer Academic Publishers, The Netherlands.

Eiselt, H.A., G. Laporte. 1996. Sequential location problems. *European Journal of Operational Research* 96 217-231.

Gerrard, R.A., R.L. Church. 1996. Closest assignment constraints and location models: Properties and structure. *Location Science* 4(4) 251-270.

Goodchild, M., V. Noronha, Location-allocation for small computers, Monograph No. 8., Department of Geography, The University of Iowa, Iowa City, 1983.

Hanjoul, P., D. Peeters. 1987. A facility location problem with clients' preference orderings. *Regional Science and Urban Economics* 17 451-473.

Hansen, B., B. Jaumard, G. Savard 1992. New branch and bound rules for linear bilevel programming, SIAM Journal on Scientific and Statistical Computing 13 1194-1217.

Israeli, E. 1999. System interdiction and defense, Ph.D. dissertation, Operations Research Department, Naval Post Graduate School, Monterey, CA.

Jüttner, U., H. Peck, M. Christopher. 2003. Supply chain risk management: Outlining an agenda for future research. *International Journal of Logistics: Research and Applications* 6(4) 197-210

Karlof, J.K., W. Wang. 1996. Bilevel Programming applied to the flow shop scheduling problem. *Computers and Operations Research* 23 443-451.

Moore, J.T., J.F. Bard. 1990. The mixed integer linear bilevel programming problem. *Operations Research* 38 911-921.

O'Hanley, J.R. 2005. Optimizing reserve selection planning in the face of uncertain site loss. Ph.D. thesis, Department of Environmental Science, Policy and Management, University of California, Berkeley.

Rice, J.B, F. Caniato. 2003. Building a secure and resilient supply network. *Supply Chain Management Review* 7(5) 22-30.

Rojeski, P., C.S. ReVelle. 1970. Central facilities location under an investment constraint. *Geographical Analysis* 2 343-360.

Salmeron, J., K. Wood, R. Baldick. 2004. Analysis of Electric Grid Security Under Terrorist Threat. *IEEE Transactions on Power Systems* 19(2) 905-912.

Scaparra, M.P., R.L. Church. 2005. An optimal approach for the interdiction median problem with fortification. Working Paper No. 78. Kent Business School, University of Kent, UK.

Sheffi, Y. 2001. Supply chain management under the threat of international terrorism. *International Journal of Logistics Management* 12(2) 1-11.

Snyder, L. V., M. S. Daskin. 2005. Reliability models for facility location: The expected failure cost case. Forthcoming in *Transportation Science*.

Stackelberg, H. 1952. *The Theory of Market Economy*. Oxford University Press.

Vicente, L., G. Savard, J. Judice. 1996. Discrete Linear Bilevel Programming Problem. *Journal of Optimization Theory and Applications* 89 597-614.

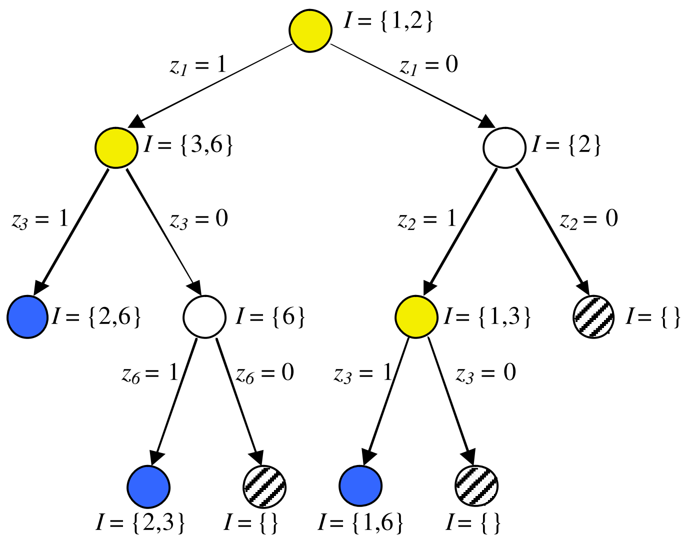

Fig. 1. Binary Tree Search. Example with $p = 6$, $q = 2$ and $r = 2$.

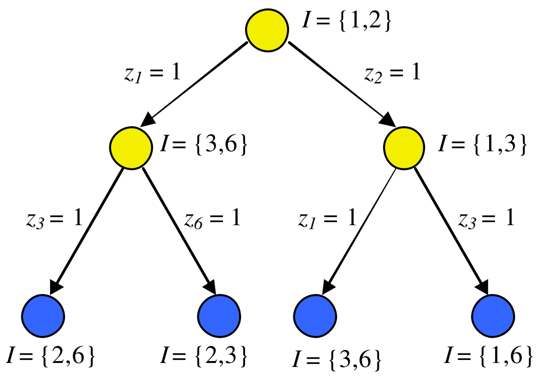

Fig. 2. Tree Search. Example with $p = 6$, $q = 2$ and $r = 2$.

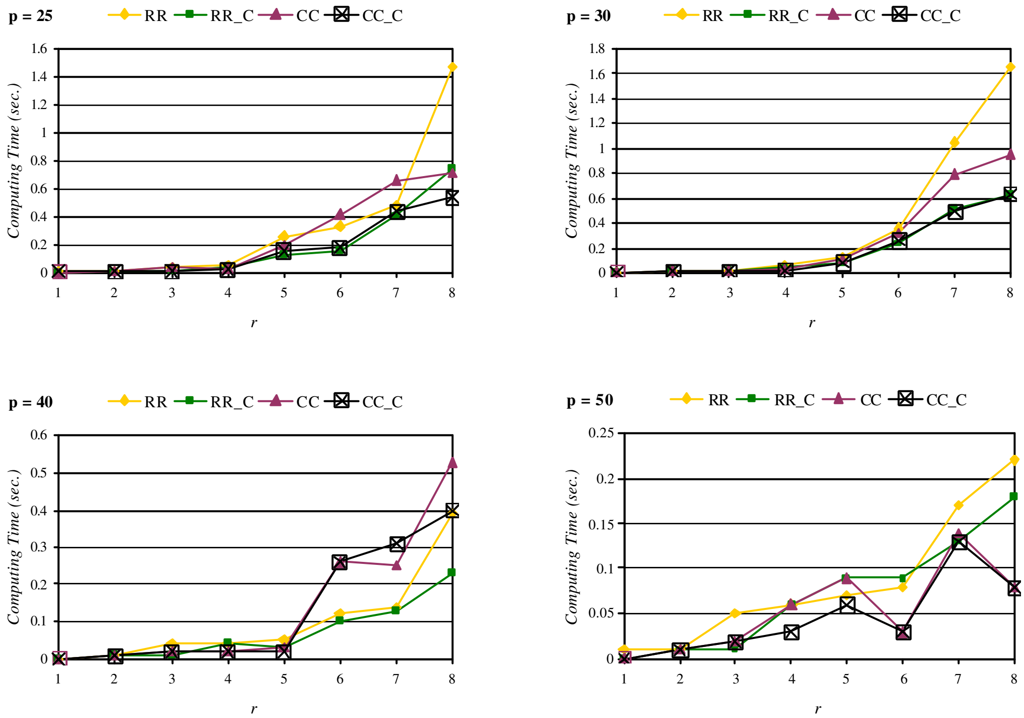

Fig.3. Impact of COBRA reduction and CA constraints on computing time for the London data set.

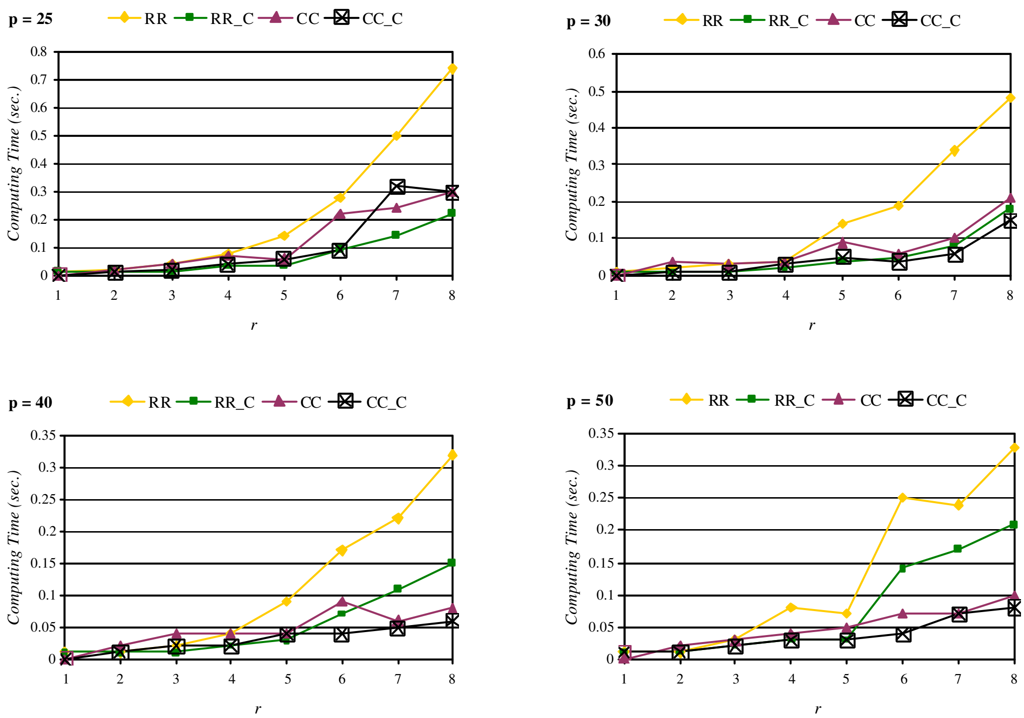

Fig. 4. Impact of COBRA reduction and CA constraints on computing time for the Alberta data set.

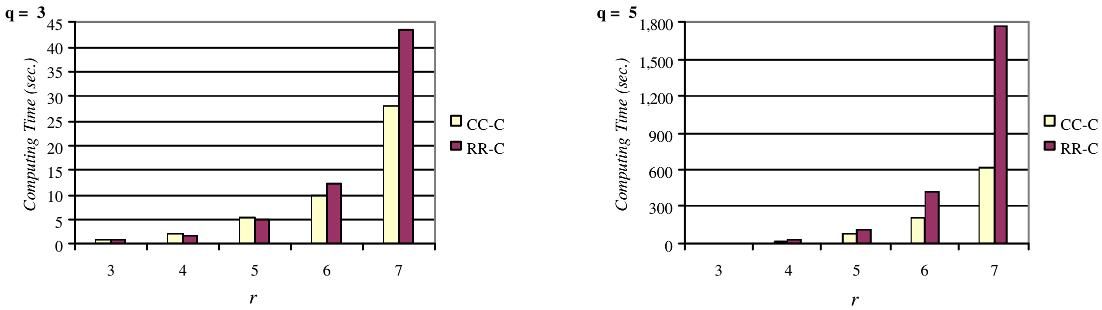

Fig. 5. Time comparison between CC and RR constraints for the London data set with p=25.

			London			Alberta		
p	*q*	*r*	Obj. Val.	MCPC Time	IE Time	Obj. Val.	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	1,819.89	260.14
25	7	8	172,623.60	1,428.89	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	1,862.97	132.12
30	7	7	136,730.79	1,529.27	2,382.11	34,767,116	--	425.46
30	7	8	144,073.46	--	4,255.58	36,853,397	--	1,333.44

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

p	q*	Objective Value				Running Time (sec.)			
		$r = 2$	$r = 3$	$r = 4$	$r = 5$	$r = 2$	$r = 3$	$r = 4$	$r = 5$
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	3,467.25
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	1,665.66
	20%	58,553.08	62,261.17	67,026.53	71,140.40	2.11	77.23	664.77	7,441.07
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	2,229.49
	20%	45,310.07	48,814.47	52,011.79	55,469.28	2.55	254.38	1,568.08	13,088.10

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

Table 2. Computational results for the London data set solved for larger values of p.

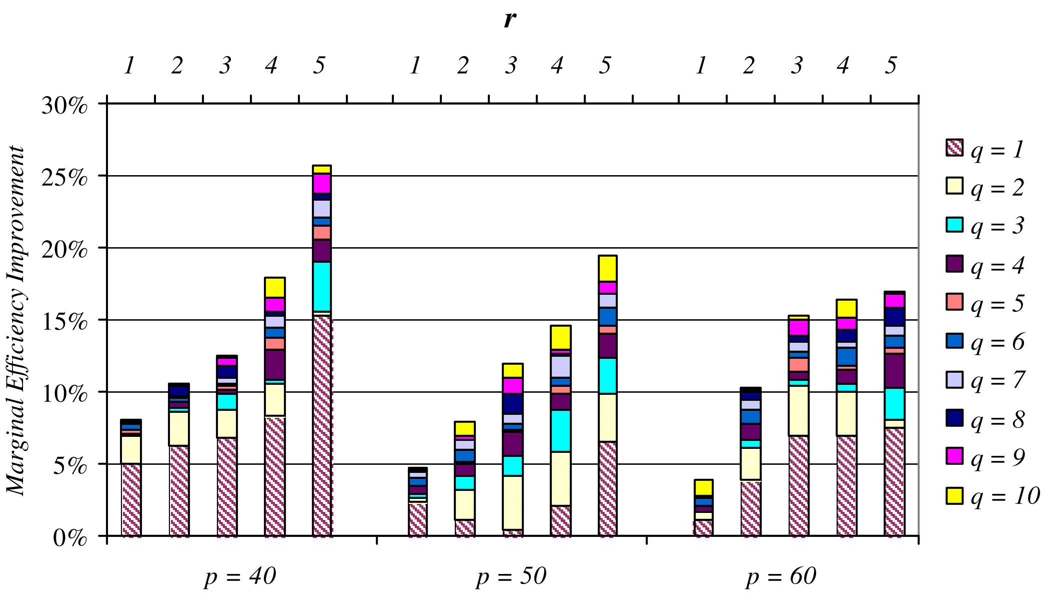

Fig. 6. Marginal percentage improvement in efficiency due to any additional fortificatio n.

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

Table 3. Optimal set of 6 fortifications for the London data set with 30 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

Table 4. Optimal set of 6 fortifications for the London data set with 60 facilities and various numbers of interdictions.

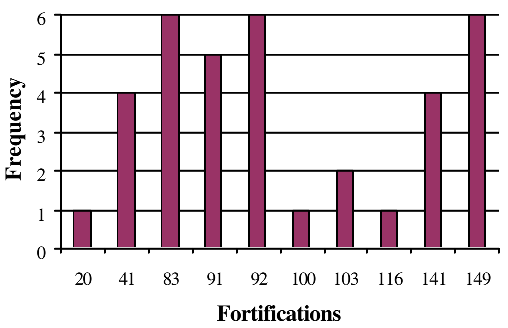

Fig. 5. Frequency with which each facility appears in a fortification set in Table 3.

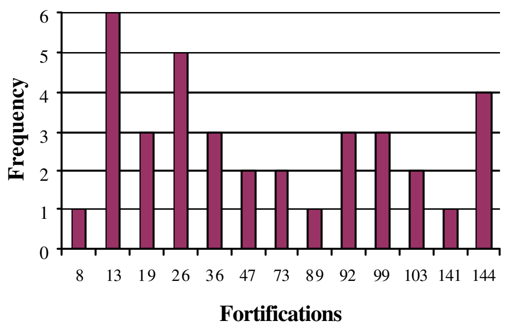

Fig. 6. Frequency with which each facility appears in a fortification set in Table 4.

UNIVERSITY OF KENT

http://www.kent.ac.uk/kbs/research-information/index.htm