5-2014

Robust Network Interdiction with Invisible Interdiction Assets

Nail Orkun Baycik

University of Arkansas, Fayetteville

Robust Network Interdiction with Invisible Interdiction Assets

Robust Network Interdiction with Invisible Interdiction Assets

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Industrial Engineering

By

Nail Orkun Baycik
Çankaya University
Bachelor of Science in Industrial Engineering, 2012
Çankaya University
Bachelor of Science in Computer Engineering, 2012

May 2014
University of Arkansas

This thesis is approved for recommendation to the Graduate Council.

Dr. Kelly Sullivan
Thesis Director

Dr. Edward Pohl
Committee Member

Dr. Chase Rainwater
Committee Member

Abstract

We study a version of the shortest path network interdiction problem in which the follower seeks to find the shortest path on a given network and the leader seeks to maximize the follower's shortest path length by interdicting arcs. We consider placement of interdictions that are not visible to the follower; however, we seek to locate interdictions in a manner that is robust against the possibility that some interdiction locations become known to the follower. We formulate the problem as a bi-level program, and use the simulated annealing heuristic to identify high quality solutions. We apply our approach to investigate the tradeoffs between conservative (i.e., the follower discovers all interdiction locations) and risky (i.e., the follower discovers no interdiction locations) assumptions regarding the leader's information advantage. A design of experiments study is performed in order to tune the parameters for the simulated annealing heuristic.

Table of Contents

1 Introduction **1**

2 Mathematical Models **4**

2.1 Follower's Problem 5

2.2 Informant's Problem 6

2.3 Leader-Informant-Follower Problem 9

3 Simulated Annealing Heuristic Approach **11**

4 Design of Experiments **21**

5 Analysis of Results for Example Instance **31**

6 Conclusion **35**

References **36**

List of Tables

1 Instances in the analysis set. 22

2 Minimum value, design center, and allowable change for each parameter. . . 22

3 Augmented matrix of coded variables used in the experiment set. 23

4 Summary of regression analysis performed for Instance 1. 24

5 Summary of regression analysis performed for Instance 2. 24

6 Summary of regression analysis performed for Instance 3. 24

7 Summary of regression analysis performed for Instance 4. 24

8 Step sizes for each parameter within instances. 25

9 Parameter settings in each step for Instance 1. 25

10 Parameter settings in each step for Instance 2. 26

11 Parameter settings in each step for Instance 3. 27

12 Parameter settings in each step for Instance 4. 28

13 Parameter settings resulting in the maximum objective values for each instance. 29

14 Final parameter values to solve the LIF problem. 29

15 Computational results performed with final parameter values. 30

16 Comparison of best known solution and solution with proposed parameter set. 30

17 Average and maximum LIF objective values for each instance. 31

List of Figures

1 Simulated annealing method pseudecode 14

2 Example network with 25 nodes and 60 arcs. 16

3 Example instance with $R = 0$ (Objective = 73). 17

4 Example instance with $R = 4$ (Objective = 45). 18

5 Example instance with $R = 6$ (Objective = 38). 19

6 Example instance with $R = 13$ (Objective = 28). 20

7 Change in the LIF objective with respect to different R-values. 32

8 SA Solution of the big instance with $R = 1$ (Objective = 417). 33

9 SA Solution of the big instance with $R = 5$ (Objective = 212). 33

10 SA Solution of the big instance with $R = 15$ (Objective = 198). 34

11 SA Solution of the big instance with $R = 66$ (Objective = 171). 34

1 Introduction

Network interdiction problems involve two players, a *leader*, and a *follower* who have conflicting objectives on a given network. The leader acts first (generally, interdicting the network components), and according to his/her actions, the follower determines a reaction to achieve his/her goal. Applications of network interdiction problems arise in drug trafficking [9, 18], critical infrastructure protection [2, 19], and nuclear material smuggling [13].

The objectives of the players depend on the type of the problem. For example, in a maximum flow network interdiction problem, the follower seeks to maximize the flow from the source to the sink, and the leader seeks to minimize the follower's maximum flow (see [3, 12, 14, 17, 18]). In the shortest path network interdiction problem, the follower aims at finding the shortest path on a given network whereas the leader seeks to maximize the follower's shortest path length (see [1, 5, 6, 10]).

The study of network interdiction problem began in 1960s. Wollmer [17] studied a problem in which the reduction in the maximum flow between the source and the sink is maximized by removing n arcs from the network. McMasters and Mustin [12] considered a maximum flow network interdiction problem where the interdiction resources are limited. Fulkerson and Harding [5] attempted to maximize the shortest path length in a graph subject to a limited budget. Malik et al. [10] considered a cardinality constrained version of the shortest path interdiction problem. In this study, our focus will be on the shortest path network interdiction problem.

Most shortest path interdiction studies assume that the interdictions are visible to the follower (see [1, 5, 7, 10]). However, this strict assumption may cause overly conservative solutions. We consider a case in which the interdictions are invisible to the follower. If interdictions are completely invisible to the follower, the problem becomes simple (at least when costs c result in a unique shortest path) because the follower's actions do not depend on the leader's actions. Thus, an optimal strategy for the leader is to interdict arcs with the largest d-values among arcs on the path of the shortest length with respect to distances c. However, this is a very risky strategy because if the follower learns about even a small number of interdiction locations, he/she can avoid all interdictions by choosing another path. A more conservative leader's strategy is to take a robust approach utilizing his/her information advantage; that is, the leader places hidden interdictions under the assumption that some will be discovered by the follower. To this end, we introduce the third player, the *informant*, who has the full knowledge of the network components but is willing to reveal only a limited amount of information to the follower so that he/she can follow the shortest path.

We now summarize the shortest path network interdiction problem (SPNIP) studies in the recent literature. Israeli and Wood [7] studied a basic version of SPNIP and developed two decomposition based algorithms to solve this problem. The first one introduces "supervalid inequalities" to improve the performance of the algorithm that works well when delays caused by the interdictions are not large, and the second one simplifies the master problem to a set covering problem and uses a greedy heuristic and an exact procedure to deal with the case where the delays are large.

Bayrak and Bailey [1] studied a version of SPNIP where there exists an information asymmetry between the leader and the follower about the arc lengths. That is, the leader knows the true arc lengths, but the follower can only estimate these lengths, which may be inaccurate. This type of problem arises in within the context of sensor placement problems in which the evader aims at evading the detection, and the interdictor seeks to maximize the probability of detecting the evader. Assuming that the leader knows the follower's perceived arc lengths, and the evader knows which arcs are interdicted, they reformulate an intuitive max-min formulation into a mixed integer linear program (MILP) by replacing the shortest path problem with a linear system of optimality conditions.

Salmerón [15] considered a related network interdiction problem with asymmetric information. In this problem, the attacker uses vehicle-born suicide attackers whereas the defender attempts to minimize the damages created by the attacker by allocating the limited interdiction assets. Three types of interdiction assets are available. The first type is visible to the attacker, the second one's location is only known by the defender, and the third one is transparent to the attacker but there is a biased perception (e.g. nonfunctioning surveillance cameras). They formulate this nontransparent defender-attacker problem as a MILP, and solve it by Benders Decomposition and a heuristic approach. Moreover, they provide a multi-objective model that takes several behaviors of the attacker into account simultaneously.

Morton et al. [13] posed two stochastic models for a maximum-reliability path interdiction problem that is closely related to SPNIP. In the first model, the follower has full knowledge of the interdictions but in the second model, the leader and the follower have different perceptions of each arc's parameters with and without interdiction. Like [1] (but unlike our model) the follower has full visibility of the interdiction locations in both of Morton's models.

In what follows, we use decision variables x to represent interdiction locations, y to represent information shared by the informant, and z to represent the follower's choice of path. In the following section, we introduce the Follower (F) problem that minimizes the length of the follower's path given fixed y-values, the Informant-Follower (IF) problem given fixed x-values that decides the amount of information to be shared with the follower as well as the follower's shortest path, and the Leader-Informant-Follower (LIF) problem that considers the objectives of the three players in effort to develop an optimal interdiction strategy. We then present a MILP formulation of IF problem. In Section 3, we introduce a heuristic solution approach for LIF. Section 4 details computational experiments, and Section 5 presents the implementation of the heuristic, and Section 6 concludes.

2 Mathematical Models

Consider a directed network $G = (N, A)$ where $N = \{1, ...n\}$ represents the set of nodes, and A represents the set of arcs. Each arc (i, j) has nominal length c_{ij}, which becomes $c_{ij} + d_{ij}$ $(d_{ij} \geq 0)$ if interdicted. The decision variable x_{ij} equals 1 if arc $(i, j) \in A$ is interdicted by the leader, and 0 otherwise. For fixed $x = \hat{x}$, the true length of an arc (i, j) is $c_{ij} + d_{ij}\hat{x}_{ij}$. We define the decision variable, z_{ij} which becomes 1 if arc $(i, j) \in A$ is traversed by the follower, and 0 otherwise. The decision for the informant is to decide on the amount of information to be shared with the follower, represented by y_{ij} where $0 \leq y_{ij} \leq d_{ij}\hat{x}_{ij}$. Thus, the perceived length of arc (i, j) is $c_{ij} + y_{ij}$.

2.1 Follower's Problem

For fixed $y = \hat{y}$, the follower finds a shortest 1–n path with respect to perceived arc lengths $c_{ij} + \hat{y}_{ij}$. We model the follower's problem $\mathrm{F}(\hat{y})$ as follows:

$$[\mathrm{F}(\hat{y})] : \min_{z} \sum_{(i,j)\in A} [c_{ij} + \hat{y}_{ij}]\, z_{ij}, \tag{1a}$$

$$\text{s.t.} \sum_{j\in \mathrm{FS}(i)} z_{ij} - \sum_{j\in \mathrm{RS}(i)} z_{ji} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i = 2, \dots, n-1 \\ -1 & \text{if } i = n, \end{cases} \tag{1b}$$

$$z_{ij} \geq 0, \ \forall (i,j) \in A. \tag{1c}$$

In model (1), constraint (1b) is the flow-balance constraint, (1c) is the sign restriction on z-variables, and the objective function (1a) represents the perceived length of the path traversed by the follower.

In the following section, we will formulate the IF problem. To this end, we provide the dual of model (1). The dual model is given by

$$\max u_1 - u_n, \tag{2a}$$

$$\text{s.t. } u_i - u_j \leq c_{ij} + \hat{y}_{ij}, \ \forall (i,j) \in A, \tag{2b}$$

where $u_i, i \in N$, is the dual variable associated with the flow balance constraint (1b) for node i. In the next subsection, we give the formulation of the IF problem.

2.2 Informant's Problem

The informant's problem is to determine the proportion of each $d_{ij}x_{ij}$-value to be shared with the follower in order to minimize the actual length of the path chosen by the follower. Thus, for fixed $x = \hat{x}$ values, IF($\hat{x}$) is formulated as follows:

$$[\text{IF}(\hat{x})]\text{: } \min_{y,z} \sum_{(i,j)\in A} \left[c_{ij} + d_{ij}\hat{x}_{ij}\right] z_{ij}, \tag{3a}$$

$$\text{s.t. } \sum_{(i,j)\in A} y_{ij} \le R, \tag{3b}$$

$$0 \le y_{ij} \le d_{ij}\hat{x}_{ij} \ \forall (i,j) \in A, \tag{3c}$$

$$z \text{ is optimal for model (1) given } y. \tag{3d}$$

In model (3), constraint (3b) restricts the amount of information to be revealed to the follower to R, and constraint (3c) bounds the amount of information shared on arc (i, j) by d_{ij} when $x_{ij} = 1$ and requires that no information is shared when $x_{ij} = 0$. Constraint (3d) requires the follower to choose a shortest path according to perceived distances, and objective function (3a) represents the actual length of the path traversed by the follower.

In model (3), constraint (3d) requires the z-variables to be optimal for the follower's problem given y. Therefore, in order to construct the single-level IF formulation that includes the behavior of both the follower and the informant, we replace constraint (3d) with primal feasibility, dual feasibility, and strong duality conditions of model (1). Hence, the IF formulation for fixed $x = \hat{x}$ values is as follows:

$$
\begin{aligned}
[\mathrm{IF}(\hat{x})]\colon \min_{u,y,z} \quad & \sum_{(i,j)\in A} [c_{ij} + d_{ij}\hat{x}_{ij}]\, z_{ij}, && \text{(4a)}\\
\text{s.t.} \quad & \sum_{(i,j)\in A} y_{ij} \le R, && \text{(4b)}\\
& 0 \le y_{ij} \le d_{ij}\hat{x}_{ij},\ \forall (i,j) \in A, && \text{(4c)}\\
& \sum_{j\in \mathrm{FS}(i)} z_{ij} - \sum_{j\in \mathrm{RS}(i)} z_{ji} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i = 2,\dots,n-1 \\ -1 & \text{if } i = n, \end{cases} && \text{(4d)}\\
& u_i - u_j \le c_{ij} + y_{ij},\ \forall (i,j) \in A, && \text{(4e)}\\
& \sum_{(i,j)\in A} [c_{ij} + y_{ij}]\, z_{ij} = u_1 - u_n, && \text{(4f)}\\
& z_{ij} \in \{0,1\},\ \forall (i,j) \in A. && \text{(4g)}
\end{aligned}
$$

In model (4), constraint (4f) contains a product of two decision variables, y_{ij} and z_{ij}, which causes the formulation to become nonlinear. Hence, in order to obtain a linear formulation, we add the McCormick inequalities [11] to the model. That is, we impose binary restrictions on z and substitute variables w_{ij}, $(i,j) \in A$, to replace each occurrence of $y_{ij}z_{ij}$. Then, we linearize the model by adding the constraints $w_{ij} \le y_{ij}$, $w_{ij} \le d_{ij}\hat{x}_{ij}$, $w_{ij} \ge 0$, and $w_{ij} \ge y_{ij} - d_{ij}\hat{x}_{ij}(1 - z_{ij})$, for all $(i,j) \in A$. Hence, the single-level MILP formulation of IF for fixed $x = \hat{x}$ values is obtained as follows:

$$[\text{IF}(\hat{x})]\text{:} \min_{u,w,y,z} \sum_{(i,j)\in A} \left[c_{ij} + d_{ij}\hat{x}_{ij}\right] z_{ij}, \tag{5a}$$

$$\text{s.t.} \sum_{(i,j)\in A} y_{ij} \leq R, \tag{5b}$$

$$0 \leq y_{ij} \leq d_{ij}\hat{x}_{ij}, \ \forall (i,j) \in A, \tag{5c}$$

$$\sum_{j\in \text{FS}(i)} z_{ij} - \sum_{j\in \text{RS}(i)} z_{ji} = \begin{cases} 1 & \text{if } i = 1 \\ 0 & \text{if } i = 2, \ldots, n-1 \\ -1 & \text{if } i = n, \end{cases} \tag{5d}$$

$$u_i - u_j \leq c_{ij} + y_{ij}, \ \forall (i,j) \in A, \tag{5e}$$

$$\sum_{(i,j)\in A} \left[c_{ij}z_{ij} + w_{ij}\right] = u_1 - u_n, \tag{5f}$$

$$w_{ij} \leq y_{ij}, \ \forall (i,j) \in A, \tag{5g}$$

$$w_{ij} \leq d_{ij}\hat{x}_{ij}, \ \forall (i,j) \in A, \tag{5h}$$

$$w_{ij} \geq y_{ij} - d_{ij}\hat{x}_{ij}(1 - z_{ij}), \ \forall (i,j) \in A, \tag{5i}$$

$$w_{ij} \geq 0, \ \forall (i,j) \in A, \tag{5j}$$

$$z_{ij} \in \{0,1\}, \ \forall (i,j) \in A. \tag{5k}$$

We shall now explain why the linearization inequalities (5g) through (5j) are valid. Because model (5) includes all constraints in model (1) and its dual, weak duality of model (1) yields that $\sum_{(i,j)\in A}(c_{ij}z_{ij}+y_{ij}z_{ij}) \geq u_1-u_n$. From (5i) and (5j), we have $w_{ij} \geq y_{ij}z_{ij}$, $\forall (i,j) \in A$, since $z_{ij} \in \{0,1\}$. Therefore, $\sum_{(i,j)\in A}(c_{ij}z_{ij} + w_{ij}) \geq \sum_{(i,j)\in A}(c_{ij}z_{ij} + y_{ij}z_{ij}) \geq u_1 - u_n$. Thus, equality (5f) ensures that (4f) is satisfied. Moreover, by setting $w_{ij} = y_{ij}z_{ij}$ for all arcs (i, j), a feasible assignment of w is guaranteed to exist. Thus, by using the optimality conditions, and a linearization technique, we constructed a MILP formulation of the IF problem that can be solved optimally by using a MILP solver such as CPLEX.

2.3 Leader-Informant-Follower Problem

The overall problem includes the leader who decides which arcs to be interdicted that is represented by x_{ij} decision variables. That is, in LIF we need to solve $\max_{x \in X} f(x)$ where $f(x)$ is the optimal value of IF(x) and $X = \left\{x : \sum_{(i,j)\in A} x_{ij} \leq B; x_{ij} \in \{0, 1\}\,;\ \forall (i, j) \in A\right\}$. However, the inner problem (i.e., the IF formulation given in model (4)) in this bi-level structure is nonconvex because of the existence of binary z variables in model (5); thus it is not possible (or at least not obviously so) to take the dual of the inner problem and obtain a single-level formulation. We have considered multiple approaches to addressing this source of difficulty. We initially considered obtaining a relaxation of LIF by enumerating a subset of the network; 1–n paths, with the ultimate goal of generating paths dynamically within a branch-and-bound-like algorithm for solving LIF. As we shall explain, there are some issues with this approach that appear insurmountable. Thus, we propose to pursue heuristic solution approaches to obtain solutions for LIF. First, we overview the attempted path enumeration approach.

We now reformulate the IF problem by restricting the follower's choice of paths to a subset of all 1–n paths in the given network. Let $I = \{1, ..., m\}$ define a subset of all 1–n paths, and define a decision variable λ_i which becomes 1 if path $i \in I$ is chosen by the follower, and 0 otherwise. We denote the arcs in path i as P_i, and $\gamma(P_i) \equiv \sum_{(k,l)\in A}(c_{kl} + d_{kl}\hat{x}_{kl})$ represents the length of path i. Hence, the Path-Restricted Informant-Follower problem formulation is as follows:

$$\text{[IF-Res]:} \min_{\lambda,y} \sum_{i\in I} \gamma(P_i)\lambda_i, \tag{6a}$$

$$\text{s.t.} \sum_{i\in I} \lambda_i = 1, \tag{6b}$$

$$\sum_{(k,l)\in P_i} [c_{kl} + y_{kl}] - M(1-\lambda_i) - \sum_{(k,l)\in P_{i'}} [c_{kl} + y_{kl}] \le 0, \ \forall i, i' \in I, i \ne i' \tag{6c}$$

$$\sum_{(k,l)\in A} y_{kl} \le R, \tag{6d}$$

$$0 \le y_{kl} \le d_{kl}\hat{x}_{kl}, \ \forall (k,l) \in A \tag{6e}$$

$$\lambda_i \in \{0,1\}, \ \forall i \in I. \tag{6f}$$

In model (6), constraint (6b) enforces that the follower can choose exactly one path on the network, constraint (6c) guarantees that the path chosen by the follower is shorter than all other existing paths, constraint (6d) restricts the amount of information to be shared with the follower to R, constraint (6e) defines the lower and upper bounds on y, constraint (6f) defines the binary variables, and the objective (6a) calculates the length of the chosen path by the follower.

The general approach in the reformulation linearization technique of Sherali et al. [16] is to take advantage of the constraint structure and choose the specific constraints as the bound factors that are multiplied by the other constraints in the formulation. Hence, in our model (6), the corresponding bound factors are $y_{kl} \ge 0$ and $\sum_{i\in I} \lambda_i = 1$. Then, if we apply this technique by using these bound factors to our problem, and eliminate the redundant constraints, we obtain the linear formulation of the IF problem, which is actually the inner problem of the bi-level structure.

Unfortunately, the resulting model is not useful because, as we increase the number of paths, we add variables as well as constraints. Therefore, this means that we do not actually have a relaxation of the LIF. Due to the shortcomings of this approach, we have decided to pursue a heuristic method for LIF. We provide the considered heuristic approach in the next section.

3 Simulated Annealing Heuristic Approach

In order to find solutions to LIF, we utilize a simulated annealing (SA) heuristic that repeatedly solves IF(x) for different solutions $x \in X$ in attempt to identify a highly effective interdiction strategy. Problem LIF is given as $\max_{x \in X} f(x)$, where $f(x)$ is the optimal objective value to IF(x). Thus, our SA heuristic generates a sequence of candidate solutions x and, for each candidate solution, evaluates fitness by solving IF(x).

One of the important steps of the procedure is to determine the initial solution. In order to obtain a high-quality initial solution, we solve the shortest path network interdiction problem of Bayrak and Bailey [1] given in model (7) assuming all d-values are transparent to the follower.

$$\max \sum_{(i,j)\in A} \left(c_{ij}v_{ij} + (c_{ij}+d_{ij})w_{ij}\right), \tag{7a}$$

$$\text{s.t.} \sum_{(i,j)\in \mathrm{FS}(i)} (v_{ij}+w_{ij}) - \sum_{(i,j)\in \mathrm{RS}(i)} (v_{ij}+w_{ij}) = \begin{cases} 1 & \text{if } i=0 \\ 0 & \text{if } i=1,\dots,n-1 \\ -1 & \text{if } i=n, \end{cases} \tag{7b}$$

$$u_i - u_j - \bar{d}_{ij}x_{ij} \le \bar{c}_{ij},\ \forall (i,j)\in A, \tag{7c}$$

$$u_n - u_0 + \sum_{(i,j)\in A} \left(\bar{c}_{ij}v_{ij} + (\bar{c}_{ij}+\bar{d}_{ij})w_{ij}\right) = 0, \tag{7d}$$

$$v_{ij} + x_{ij} \le 1,\ \forall (i,j)\in A, \tag{7e}$$

$$w_{ij} - x_{ij} \le 0,\ \forall (i,j)\in A, \tag{7f}$$

$$v_{ij}, w_{ij} \ge 0,\ \forall (i,j)\in A, \tag{7g}$$

$$x \in X. \tag{7h}$$

In model (7), the follower's estimates on c_{ij} and d_{ij} are represented by $\bar{c}_{ij}$ and $\bar{d}_{ij}$, respectively. As in our models, x_{ij} indicates whether or not arc $(i,j)\in A$ is interdicted, and variables u_i, $i\in N$, correspond to dual variables for the associated flow balance constraints. Variables v_{ij} and w_{ij} are flow variables that indicate which arcs are traversed and, for a traversed arc, whether or not the arc was interdicted. To obtain an initial solution to LIF, we solve model (7) using $\bar{c}=c$ and $\bar{d}=d$ where c and d correspond to the data defined for our problem in Section 2. This information symmetric model is equivalent to our model if the R-value is large enough, and it is equivalent to the model in [7], as well.

This solution provides the initial solution represented by x_{Initial} for our SA approach. We set the current best solution at each iteration to x_{Current}. We perform "move" operations in each iteration to determine new candidate solutions. These new solutions are represented by x_{Moved}. In the subsequent iterations, the algorithm compares the current solution with the moved solution, and eventually output the best solution found so far.

Simulated annealing (SA) is a heuristic approach that was inspired of the physical annealing process, and proposed by Kirkpatrick et al. [8]. The algorithm starts at high temperature, T, and then by multiplying T by a cooling parameter, α, we go through lower temperatures. Our move operation swaps the values of two randomly selected bits from the binary vector x_{Current} in order to construct x_{Moved}, the new candidate solution. This process is performed n times at each iteration. Each time, the current solution may be updated, depending on the value of $f(x_{\text{Moved}})$, which is determined by solving IF(x_{Moved}). A move to an improved solution (i.e., $f(x_{\text{Moved}}) > f(x_{\text{Current}})$) is always accepted (i.e., x_{Current} is set equal to x_{Moved}) in this heuristic. Additionally, non-improving solutions are accepted with a probability $e^{\Delta f(x)/T}$, where $\Delta f(x) = f(x_{\text{Moved}}) - f(x_{\text{Current}})$ is the change in the objective function value. Note that non-improving solutions that are accepted in order to increase the chance of converging to a global (not local) maximum. Pseudocode of the SA method is provided in Figure 1.

We now illustrate LIF using the network given in Figure 2 that consists of 25 nodes and 60 arcs in which node 0 is the source node, and node 24 is the sink node. As this network is small, we can find an optimal solution by total enumeration. We illustrate the tradeoffs modeled by LIF by solving the instance sequentially for different R-values when $B = 3$. To this end, we first solve the instance with $R = 0$ and find the optimal objective value 73 (as

```
Solve model (7) to find the initial interdiction strategy, x_Initial.
x_Current ← x_Initial
x_Final ← x_Initial
for t = 1,..,m do
    for i = 1,..,n do
        Perform move operation to obtain x_Moved
        Solve the IF problem to find: f(x_Current), f(x_Moved)
        if f(x_Current) ≤ f(x_Moved) then
            x_Current ← x_Moved
        end if
        if U(0,1) ≤ e^((f(x_Moved) − f(x_Current))/T) then
            x_Current ← x_Moved
        else
            x_Current ← x_Current
        end if
        if f(x_Final) ≤ f(x_Current) then
            x_Final ← x_Current
        end if
    end for
    T := α × T
end for
```

Figure 1: Simulated annealing method pseudecode

Figure 3 illustrates). The dashed arcs represent the interdicted arcs and thick black arcs represent the follower's path, and the arcs having the form $(c_{ij}, d_{ij} : y_{ij})$ represent the shared arcs with the follower. Hence, in this particular instance, arcs $(0, 9), (9, 10)$, and $(10, 19)$ are interdicted, and the follower traverses $0 - 9 - 10 - 19 - 24$. In this case, the follower has no information about the effect of interdictions (as $R = 0$), which causes leader to apply a risky strategy by placing the interdictions on the path that appears shortest when no interdictions are present.

In Figure 4, we illustrate the instance with $R = 4$ whose optimal objective value is 45. In this instance, arcs $(0, 9), (2, 3)$, and $(10, 19)$ are interdicted, and the follower traverses $0 - 1 - 2 - 3 - 4 - 21 - 24$. The informant shares information about the true length of arcs

$(0, 9)$ and $(10, 19)$ with the follower. We observe that leader's interdiction strategy is more conservative than in the instance with $R = 0$ because some of the invisible interdictions are visible to follower. Thus, all of the interdictions are not placed on the original network's shortest path. Also, the follower does not follow the original network's shortest path as he/she knows that some of the arcs on this path are interdicted.

In Figure 5, we illustrate the instance with $R = 6$ whose optimal objective value is 38. In this instance, arcs $(2, 3), (9, 10)$ and $(10, 19)$ are interdicted, and the follower traverses $0-9-10-3-4-21-24$. The informant shares information about interdicted arcs $(2, 3)$ and $(10, 19)$ with the follower. Because the follower is more powerful (i.e., has access to more information) than in the previous instance, the leader's strategy is even more conservative and the LIF objective decreases again.

In Figure 6, we illustrate the instance with $R = 13$ whose optimal objective value is 28. In this instance, arcs $(0, 1), (0, 9)$, and $(0, 11)$ are interdicted, and the follower traverses $0-9-10-19-24$. In this instance, informant shares information about the true length of arcs $(0, 1)$ and $(0, 11)$ with the follower. In this case, the interdicted arcs form a source-sink cut, meaning the follower has no option but to traverse an interdicted arc. Again, allowing more information to be shared results in a more conservative interdiction strategy and a decreased LIF objective.

The LIF instances that we present show how parameter R facilitates selection of a robust strategy. By selecting the value of R appropriately, the leader's interdiction strategy is neither overly risky nor overly conservative. Note that as the R-value increases, the interdiction placements spread out vertically which is the result of the leader's robust strategy.

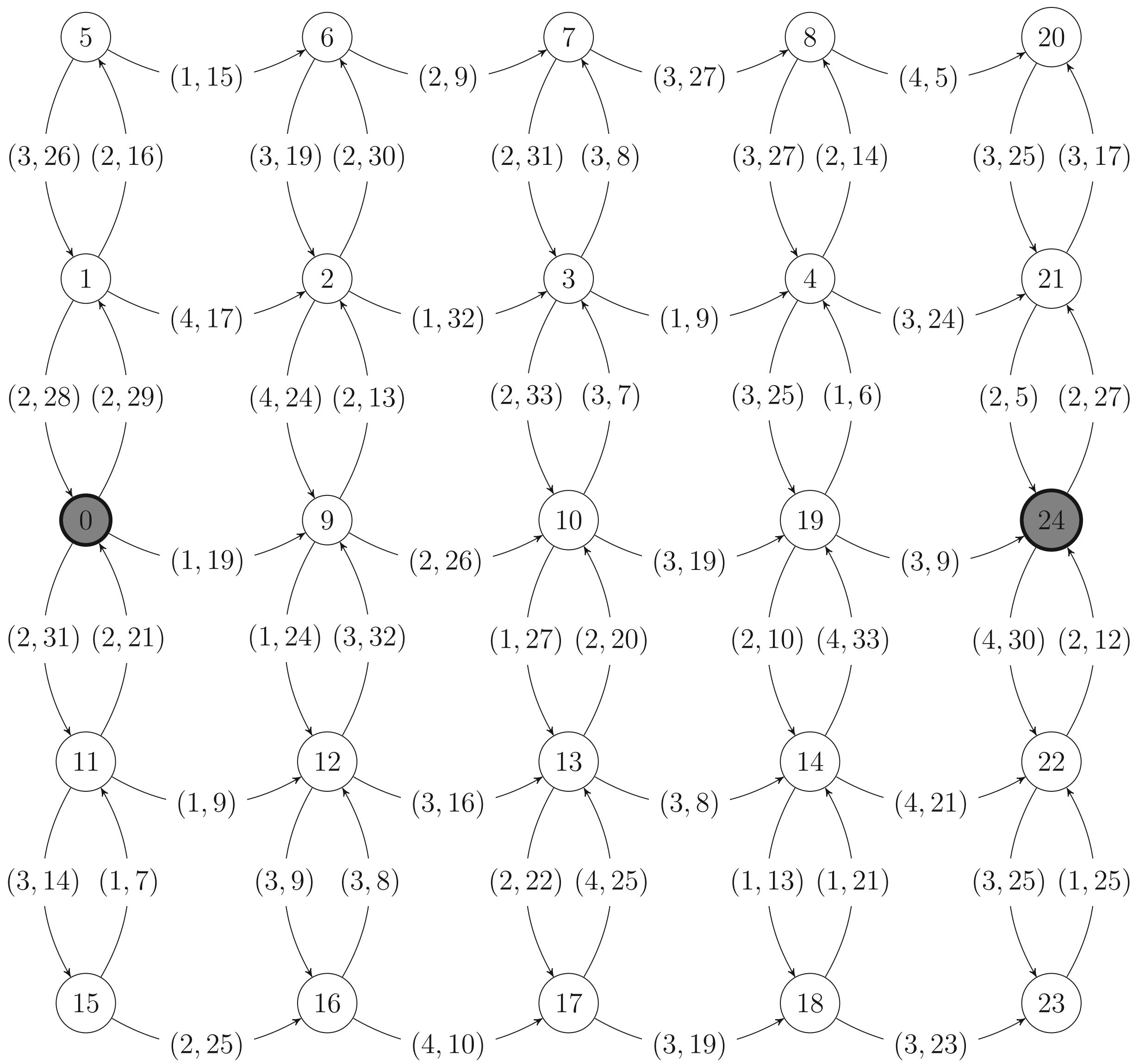

Figure 2: Example network with 25 nodes and 60 arcs. Labels on arc (i, j) are of the form (c_{ij}, d_{ij}).

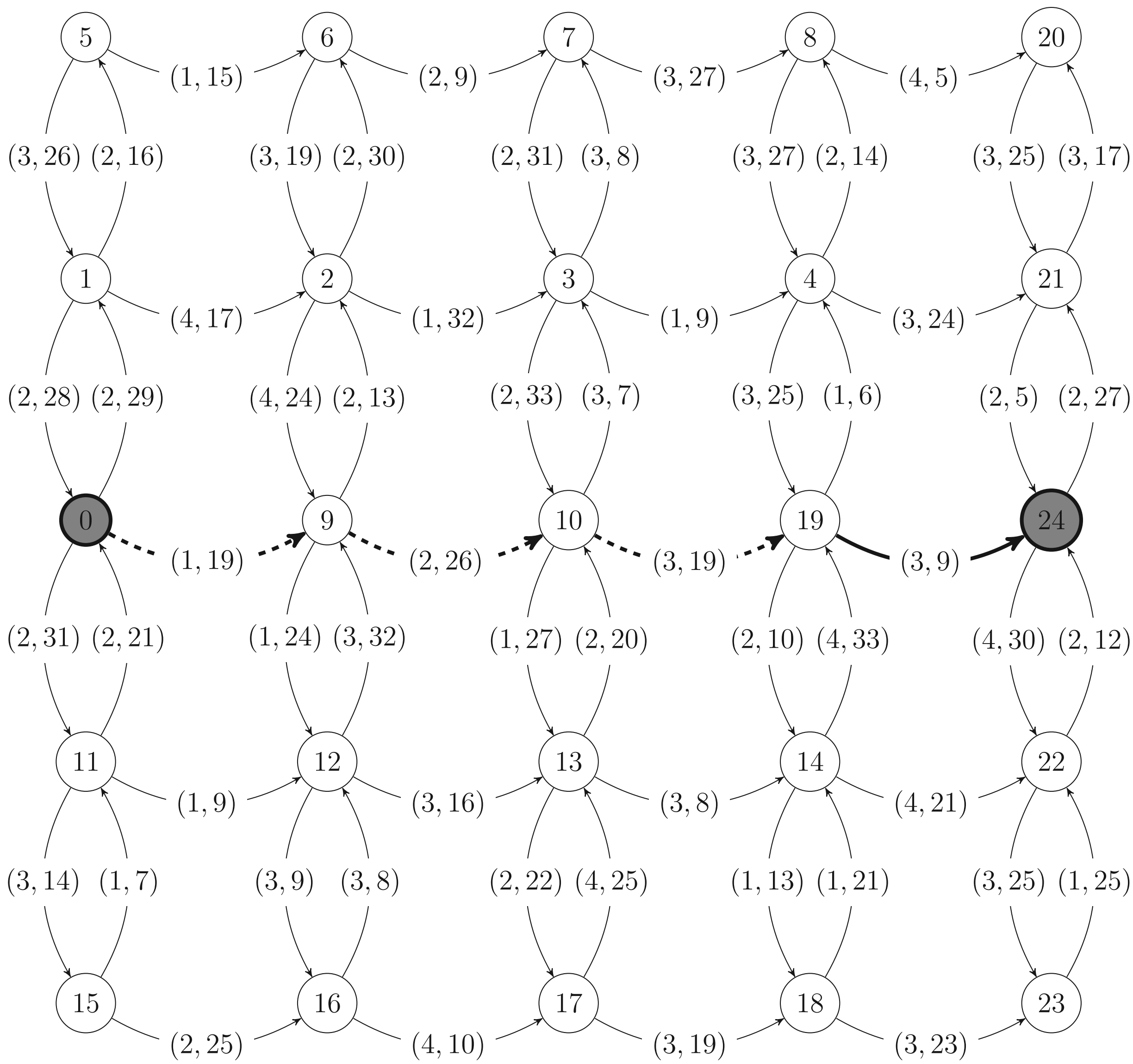

Figure 3: Example instance with $R = 0$ (Objective = 73). Labels on arc (i, j) are of the form (c_{ij}, d_{ij}).

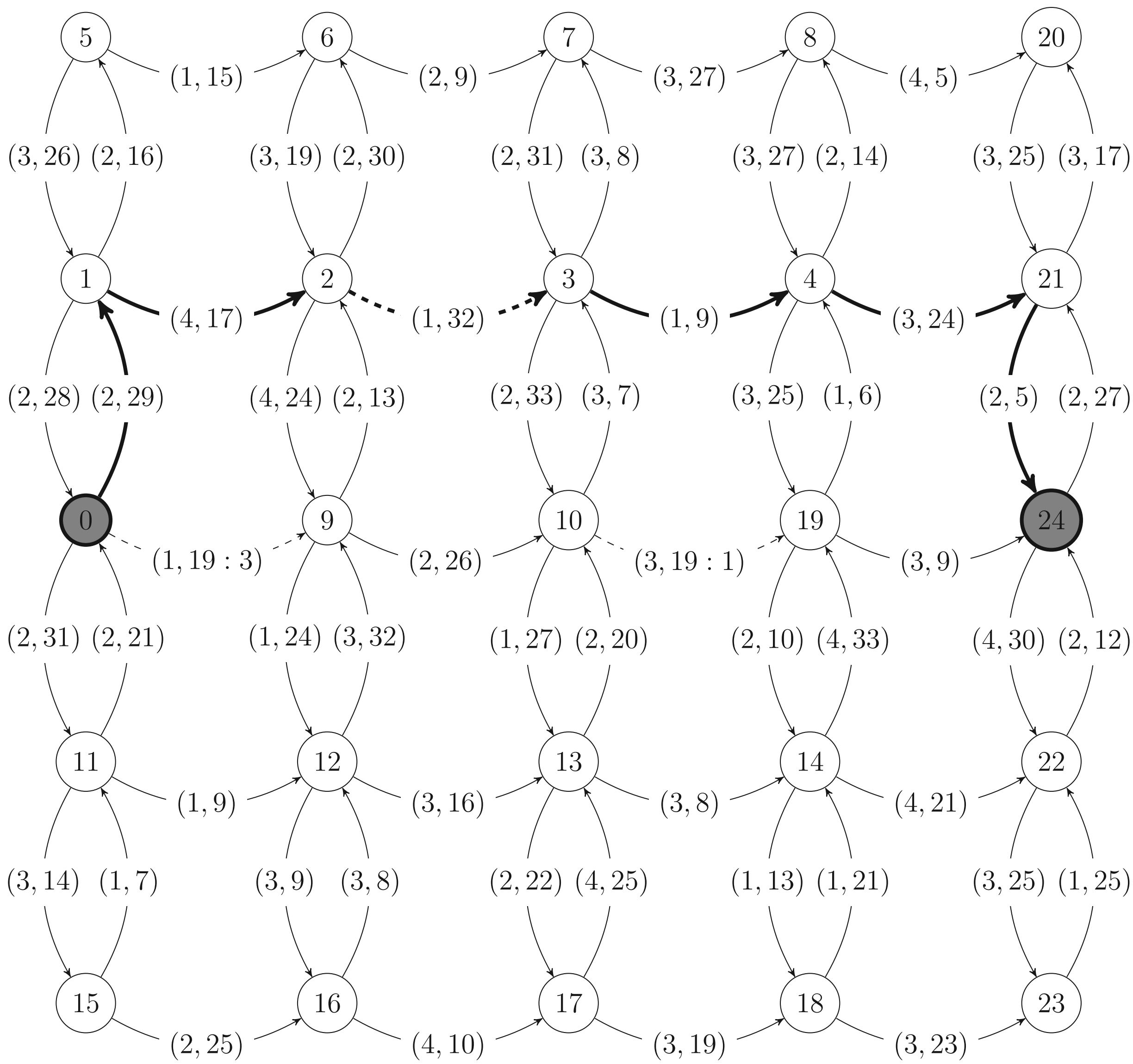

Figure 4: Example instance with $R = 4$ (Objective = 45). Labels on arc (i, j) are of the form (c_{ij}, d_{ij}).

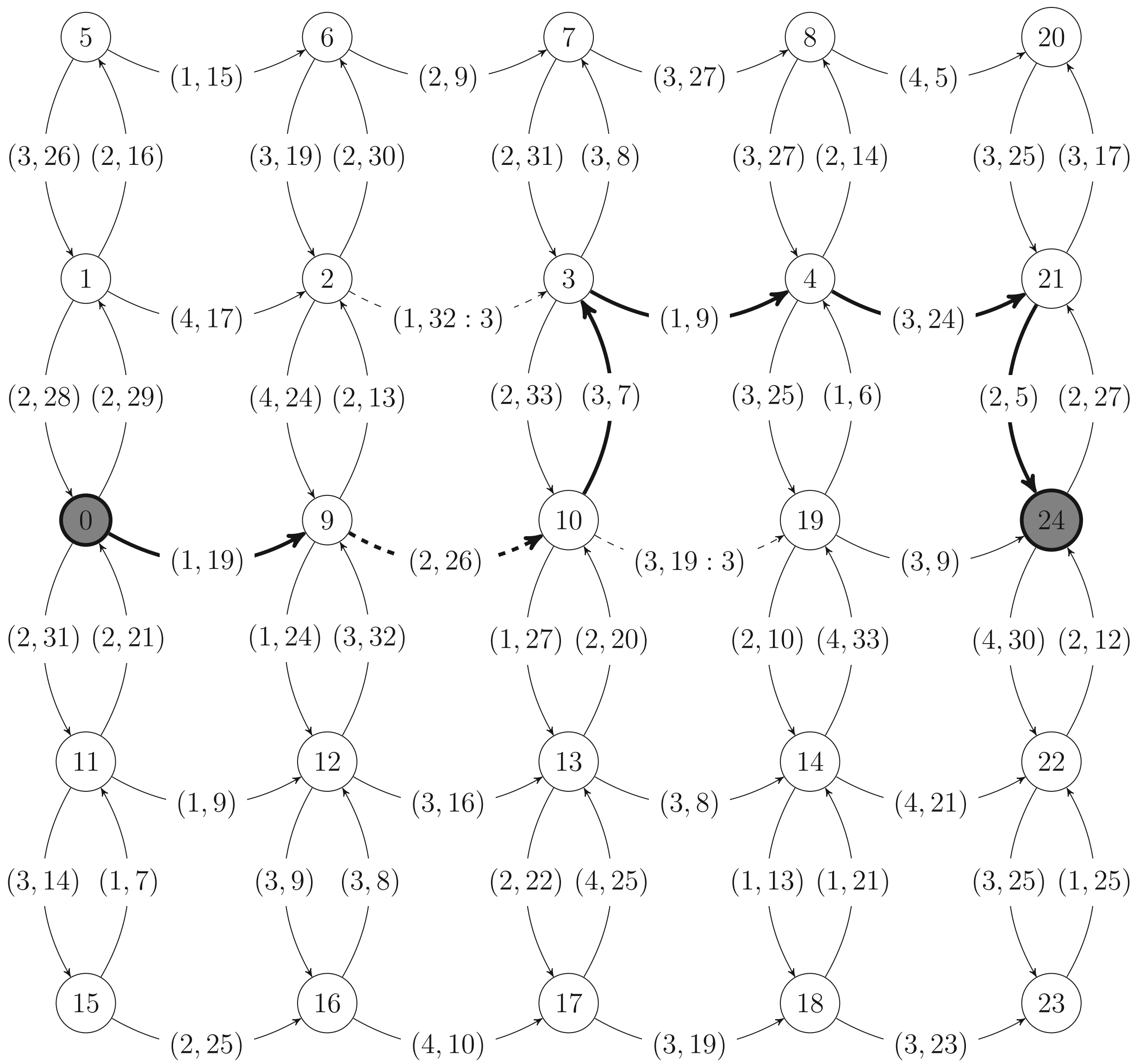

Figure 5: Example instance with $R = 6$ (Objective = 38). Labels on arc (i, j) are of the form (c_{ij}, d_{ij}).

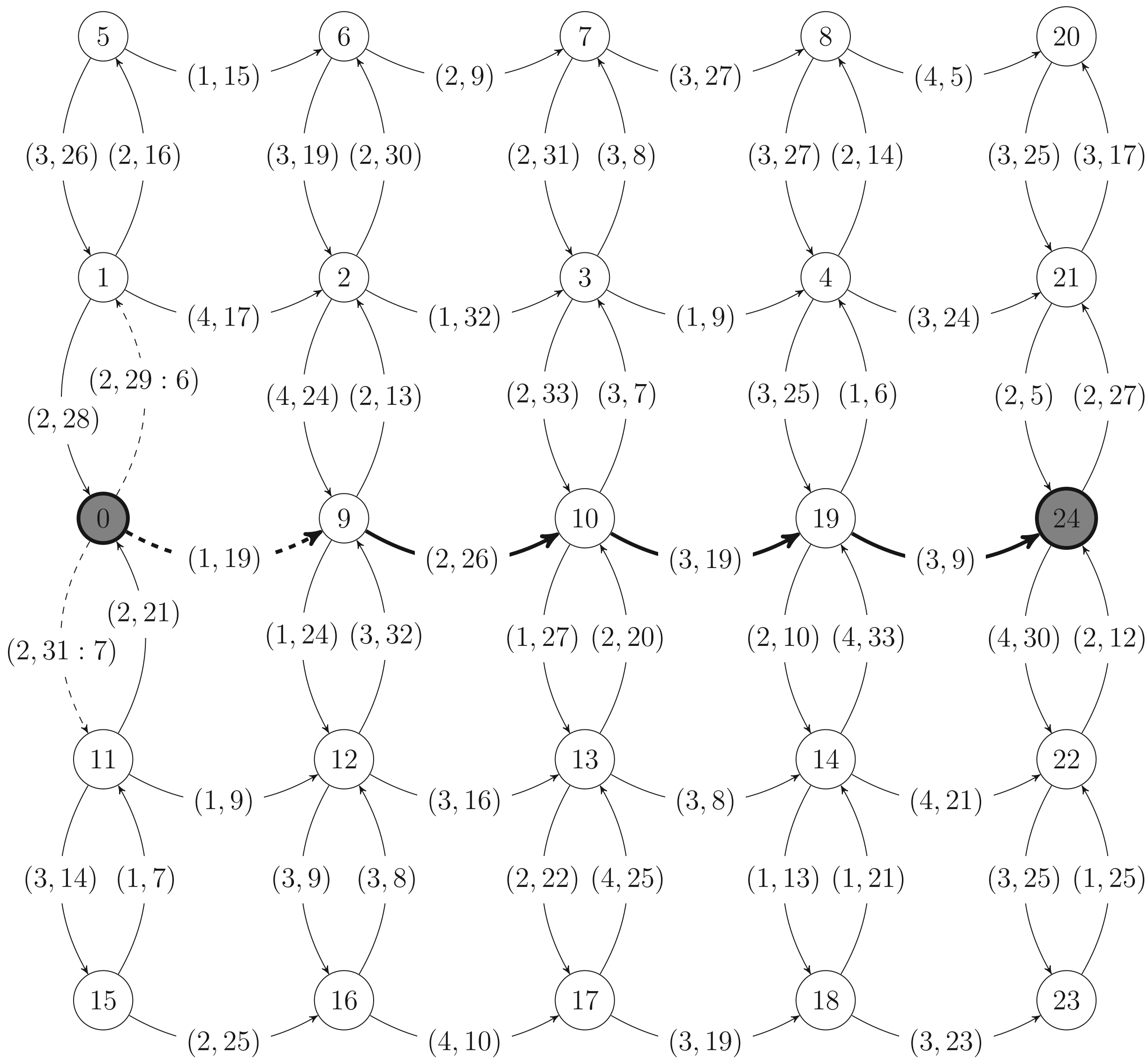

Figure 6: Example instance with $R = 13$ (Objective = 28). Labels on arc (i, j) are of the form (c_{ij}, d_{ij}).

As stated previously, we solved these instances via enumeration technique, which may be acceptable for small problems. However, since we will consider bigger problem instances, and cannot solve them to optimality, we apply the SA heuristic. In order to tune the SA heuristic's parameters, we perform a design of experiments (DOE) study. In the next section, the details of the DOE study are presented.

4 Design of Experiments

In order to solve small instances, we could enumerate all possible interdiction scenarios and find the optimal solution. However, for bigger instances, it is not possible to follow the same approach, so we propose applying the SA heuristic to find solutions to our LIF problem. This heuristic requires some predefined parameters that affect the performance of the method, so we use designed experiments to determine effective parameter settings. We follow an approach that is closely related to the study of Coy et al. [4]. A step-by-step explanation of our approach follows:

Step 1. Choose an initial set of problem instances differing in size to perform a pilot study.

Step 2. Select the design center, Δ-values (i.e., amount of change allowed for any parameter), and the minimum and maximum values for each parameter.

Step 3. Select good parameter settings for the problems based on DOE.

Step 4. Determine the most effective parameter settings.

The first step in the overall approach is to select a subset of instances to analyze, which is called the *analysis set*. This set covers one small, two mid-size, and one big instance. We define the size of the network in terms of the number of arcs and number of nodes. The values for B, R, c and d are generated randomly. The properties of the instances in the analysis set are illustrated in Table 1:

Table 1: Instances in the analysis set.

Network	B	R	Number of arcs	Number of nodes	c	d
1	3	8	60	25	$U \sim (1, 25)$	$U \sim (1, 25)$
2	30	6	268	100	$U \sim (1, 20)$	$U \sim (0, 50)$
2	30	8	268	100	$U \sim (1, 15)$	$U \sim (1, 50)$
3	5	1	823	225	$U \sim (1, 10)$	$U \sim (1, 50)$

After choosing the problem instances in our analysis set, we perform a set of runs to determine the design center and Δ-values for each parameter. The design center is identified by performing sufficient number of runs, and then choosing the parameter values yielding the best known solutions. In order to set the Δ-values, we increase or decrease the values of the design center of each parameter by a small amount, and stop when the solutions do not improve anymore. Therefore, after performing the required computational effort, the minimum value, design center, and Δ-values for each parameter are given in Table 2.

Table 2: Minimum value, design center, and allowable change for each parameter.

Parameter	Minimum value	Design center	Δ
m	50	100	100
T	2000	5000	2000
α	0.2	0.5	0.2

Note that we only consider m to represent the total number of iterations in our parameter settings and keep the total number of iterations (i.e., $m \times n$) constant, which is equal to

1000. Also, if a parameter's minimum value is greater than its "design center $-\Delta$" value, we set that parameter's value to its minimum value in our runs. For example, parameter m's minimum value is 50, and $100 - 100 = 0$. In this case, we set $m = 50$ for the corresponding run since $50 > 0$.

The next step is to generate the factorial experimental design. We choose a full factorial design, and convert this design to a matrix of coded variables. Table 3 illustrates the augmented matrix of coded variables for the experiment set:

Table 3: Augmented matrix of coded variables used in the experiment set.

Run	m	T	α
1	-1	-1	-1
2	1	-1	-1
3	-1	1	-1
4	1	1	-1
5	-1	-1	1
6	1	-1	1
7	-1	1	1
8	1	1	1

The matrix provided in Table 3 can be used to calculate the parameter values for each run. That is, we add or subtract the Δ-value to the design center to obtain the parameter combination for a run. We perform five replications for each combination using a different seed to initialize the heuristic's random number generator in each case. After performing the required runs, we fit a linear regression model to the averaged data of the results (as done by Coy et al. [4]). The results of the regression analyses for the four instances are listed in Tables 4, 5, 6, and 7, respectively:

Based on the results of the regression analyses, we conclude that all linear models are significant at 0.10 level. The minimum adjusted R^2 value is 0.23 which is of instance 1, and

Table 4: Summary of regression analysis performed for Instance 1.

Variable	Coefficient	P–value	df	SS	MS	F	Statistics	
Intercept	89.75	2.92×10^{-7}	3	4.50	1.50	1.71	R^2	0.56
m	0.01	0.09	4	3.50	0.88		AdjustedR^2	0.23
α	7.9×10^{-16}	1.00	7	8.00				
T	-9.3×10^{-20}	1.00						

Table 5: Summary of regression analysis performed for Instance 2.

Variable	Coefficient	P–value	df	SS	MS	F	Statistics	
Intercept	383.88	1.5×10^{-7}	3	647.90	216.00	18.56	R^2	0.93
m	0.05	0.03	4	46.54	11.64		AdjustedR^2	0.88
α	-37.00	3.6×10^{-3}	7	694.54				
T	-16×10^{-3}	0.06						

Table 6: Summary of regression analysis performed for Instance 3.

Variable	Coefficient	P–value	df	SS	MS	F	Statistics	
Intercept	354.70	8.4×10^{-6}	3	1707	568.99	7.72	R^2	0.85
m	0.13	0.04	4	294.78	73.70		AdjustedR^2	0.74
α	-51.13	0.03	7	2001.76				
T	-2×10^{-3}	0.21						

Table 7: Summary of regression analysis performed for Instance 4.

Variable	Coefficient	P–value	df	SS	MS	F	Statistics	
Intercept	219.58	2.4×10^{-6}	3	547.86	182.62	12.15	R^2	0.90
m	0.05	0.04	4	60.14	15.04		AdjustedR^2	0.82
α	-35.13	6.9×10^{-3}	7	608.00				
T	-9.1×10^{-4}	0.25						

the maximum value found is 0.88. Also, each parameter is statistically significant in at least one regression model.

As we have the coefficients for each parameter, we then seek to find the step size for each parameter within each instance. Toward this end, we find the maximum of the absolute values of each parameter coefficient for a particular instance, and then divide the regression coefficient of each parameter by this maximum value and multiply by Δ in order to normalize the parameter values. Table 8 shows the step sizes of each parameter for the corresponding instances:

Table 8: Step sizes for each parameter within instances.

Instance	m	α	T
1	100.00	0.00	0.00
2	0.14	-0.20	-0.09
3	0.21	-0.20	0.00
4	0.15	-0.20	0.00

To illustrate the way we obtained Table 8, consider Instance 1. The absolute value of the maximum regression coefficient is 0.01. To calculate the step size of m, $(0.01/0.01) \times 100 = 100$, and for instance 2, the step size of m is $(0.05/37) \times 100 = 0.14$. Step sizes of the parameters that are not statistically significant are 0.

The next step is to evaluate parameter settings along the regression equation's steepest ascent direction from the design center. The procedure starts at design center, and in each step, we add one-sixth of the step size of each parameter to its previous level. In case of violating the upper or lower levels of any parameter, we leave that parameter at its extreme value as we continue to modify the other parameter values. Thus, the parameter settings for Instances $1, 2, 3$, and 4 are given in Tables 9, 10, 11, and 12 respectively:

Table 9: Parameter settings in each step for Instance 1.

Step	m	n	α	T	Average LIF Objective
0	100	10	0.5	5000	91
1	117	8	0.5	5000	92
2	133	7	0.5	5000	91
3	150	6	0.5	5000	90
4	167	5	0.5	5000	90
5	183	5	0.5	5000	90
6	200	5	0.5	5000	90

Table 10: Parameter settings in each step for Instance 2.

Step	m	n	α	T	Average LIF Objective
0	100	10	0.50	5000.00	372.2
1	100	10	0.47	4999.99	372.6
2	100	10	0.44	4999.98	361.0
3	100	10	0.40	4999.96	361.2
4	100	10	0.37	4999.95	367.2
5	100	10	0.34	4999.93	367.0
6	100	10	0.30	4999.92	375.4
7	100	10	0.27	4999.90	375.4
8	100	10	0.24	4999.89	375.4
9	100	10	0.20	4999.88	373.2
10	110	9	0.20	4994.00	366.8
11	120	8	0.20	4988.00	366.4
12	130	7	0.20	4982.00	373.2
13	140	7	0.20	4976.00	373.2
14	150	6	0.20	4970.00	378.2
15	160	6	0.20	4964.00	378.2
16	170	5	0.20	4958.00	378.6
17	180	5	0.20	4952.00	379.6
18	190	5	0.20	4946.00	379.6
19	200	5	0.20	4940.00	379.6

Table 11: Parameter settings in each step for Instance 3.

Step	m	n	α	T	Average LIF Objective
0	100	10	0.50	5000	324.2
1	100	10	0.47	5000	350.6
2	100	10	0.44	5000	352.2
3	100	10	0.40	5000	353.4
4	100	10	0.37	5000	358.2
5	100	10	0.34	5000	355.6
6	100	10	0.30	5000	354.2
7	100	10	0.27	5000	355.8
8	100	10	0.24	5000	353.8
9	100	10	0.20	5000	361.4
10	110	9	0.20	5000	351.2
11	120	8	0.20	5000	355.4
12	130	7	0.20	5000	339.8
13	140	7	0.20	5000	346.4
14	150	6	0.20	5000	344.4
15	160	6	0.20	5000	344.4
16	170	5	0.20	5000	339.8
17	180	5	0.20	5000	343.4
18	190	5	0.20	5000	347.6
19	200	5	0.20	5000	347.6

Table 12: Parameter settings in each step for Instance 4.

Step	m	n	α	T	Average LIF Objective
0	100	10	0.50	5000	216.4
1	100	10	0.47	5000	186.8
2	100	10	0.44	5000	201.6
3	100	10	0.40	5000	179.4
4	100	10	0.37	5000	186.2
5	100	10	0.34	5000	185.4
6	100	10	0.30	5000	191.8
7	100	10	0.27	5000	193.0
8	100	10	0.24	5000	207.2
9	100	10	0.20	5000	207.2
10	110	9	0.20	5000	204.2
11	120	8	0.20	5000	204.2
12	130	7	0.20	5000	211.8
13	140	7	0.20	5000	212.4
14	150	6	0.20	5000	211.8
15	160	6	0.20	5000	212.4
16	170	5	0.20	5000	209.8
17	180	5	0.20	5000	211.8
18	190	5	0.20	5000	212.4
19	200	5	0.20	5000	215.4

Note that after Step 9 for Instances $2, 3$, and 4, we hit the minimum value of α, and therefore stop decreasing its value. However, parameters T and m do not hit their extreme values, but because their minimum/maximum values are too low/high to be reached within small steps, we increase the values of m by a specific amount, and adjust T accordingly. When m has a value that makes it impossible to calculate n value, which is equal to $1000/m$, we select an n value that results in the product of m and n close to 1000.

For each step, we perform five replications, and take the averages of the resulting objective values. Since the objective of the LIF problem is to maximize the length of the follower's path, we look for the step in which the maximum average length is found. Hence, Table 13 shows the parameter combinations that yield the best available solutions for each instance:

Table 13: Parameter settings resulting in the maximum objective values for each instance.

Instance	m	n	α	T
1	117	8	0.50	5000.00
2	200	5	0.20	4940.00
3	100	10	0.20	5000.00
4	100	10	0.50	5000.00

The last step is to average the parameter values given in Table 13 in order to conclude the search for the most effective parameter settings. Table 14 lists the final values for each parameter:

Table 14: Final parameter values to solve the LIF problem.

m	n	α	T
130	8	0.35	4985

After having the proposed parameter values for the LIF problem, we can now solve the instances in the analysis set to compare the results found in the preceding steps. The summary of the results for each instance are given in Table 15:

Table 15: Computational results performed with final parameter values.

Instance	m	n	α	T	Average LIF Objective
1	130	8	0.35	4985	90.0
2	130	8	0.35	4985	375.4
3	130	8	0.35	4985	334.8
4	130	8	0.35	4985	207.2

During the step size calculations, the average maximum value for Instance 1 is 92, and we could find 90 with our proposed parameter set. For Instance 2, the average maximum value is 379.6 within the step size computations, but the maximum averaged value is 375.4 with our proposed parameter set. For Instance 3, we could find the average maximum value 361.4, whereas our proposed parameter set could find the maximum averaged value 334.8. For Instance 4, the average maximum value found for step size computations is 216.4, and the proposed parameter set results in maximum averaged value of 207.2. The comparison between solutions found in the step size computations and found with the proposed parameter set is given in Table 16:

Table 16: Comparison of best known solution and solution with proposed parameter set.

Instance	Best value found previously	Best value found with proposed values	GAP
1	92.0	90.0	-0.022
2	379.6	375.4	-0.011
3	361.4	334.8	-0.074
4	216.4	207.2	-0.043

Note that in Table 16, a $(-)$ sign indicates the proposed parameter set gives worse solution. We can conclude that the set of parameters found at the end of this DOE study can provide reasonably good solutions. We observe that better solutions can be found by using different parameter values, but each network topologies require different combinations, so the largest gap of 0.074 is still reasonable. Although one could certainly perform the response surface procedure of this section to re-tune the heuristic for each instance considered (and obtain better results by doing so), this would require a significant number of computational runs for, e.g., the type of analysis that is performed in the following section. In the next section, we therefore utilize the averaged parameter values to analyze a larger network.

5 Analysis of Results for Example Instance

Using the SA parameters from the previous section, we now consider a larger network having 361 nodes and 1362 arcs. For this particular instance, computations of the c-vector are uniformly and independently distributed between 1 and 10, and d is uniformly distributed between 1 and 50. The total number of interdictions is restricted to 20. We solve this problem for different R-values, and perform 5 replications in each instance. Table 17 provides the average and maximum LIF objective values for each instance having different R-values:

Table 17: Average and maximum LIF objective values for each instance.

Instance	R	Average LIF Objective	Maximum LIF Objective
1	0	362.6	417
2	1	325.4	381
3	2	223.6	243
4	5	212.0	212
5	7	207.0	207
6	10	206.0	206
7	15	198.0	198
8	20	190.0	190
9	66	171.0	171
10	∞	171.0	171

Based on the results given in Table 17, we can state that the LIF objective decreases as R-value increases. That is, the length of the follower's path is shorter if he/she has more information about the true lengths of the arcs. The shortest path network interdiction problem, which provides the initial solution to our heuristic approach, given in model (7) is equal to our problem as R becomes large enough, and the objective values are both equal to 171. In other words, $R = 66$ units of information is large enough for this particular instance. Figure 7 illustrates the change in the best known LIF objective value with respect to the change in R-values:

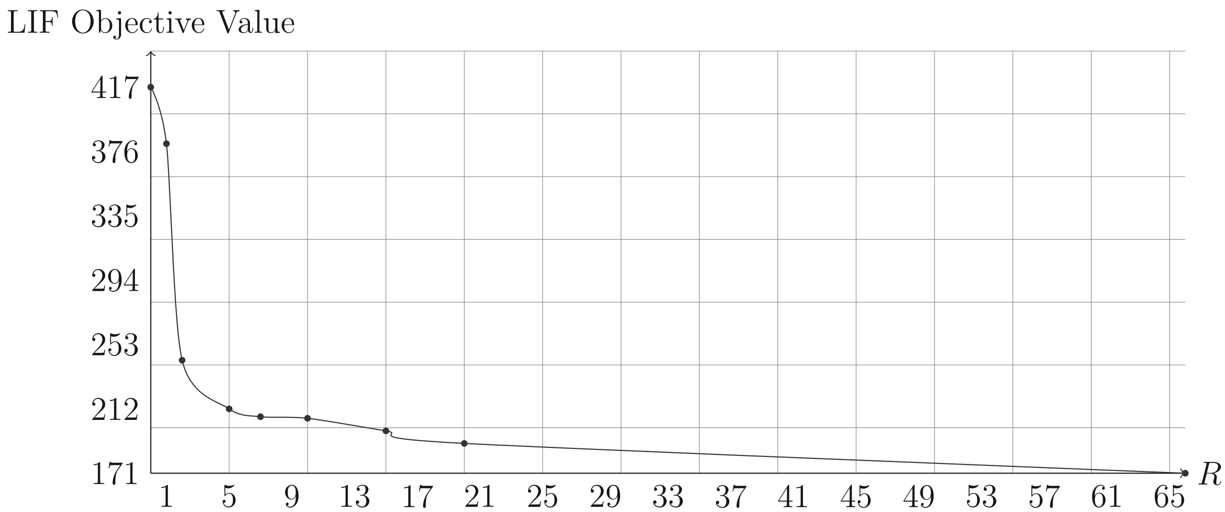

Figure 7: Change in the LIF objective with respect to different R-values.

The solution of the instances with SA are given in Figures 8, 9, 10, 11. The dots on the network represent nodes, where the leftmost circled node is the source and rightmost circled node is the sink. The lines connecting the nodes represent the interdicted arcs, and the dark highlighted path is the follower's path. We observe that the interdictions spread out in the middle part of the network as R increases. In addition, the leader prefers a more robust strategy in placing the interdictions, that is, he/she does not interdict only the arcs on the follower's shortest path.

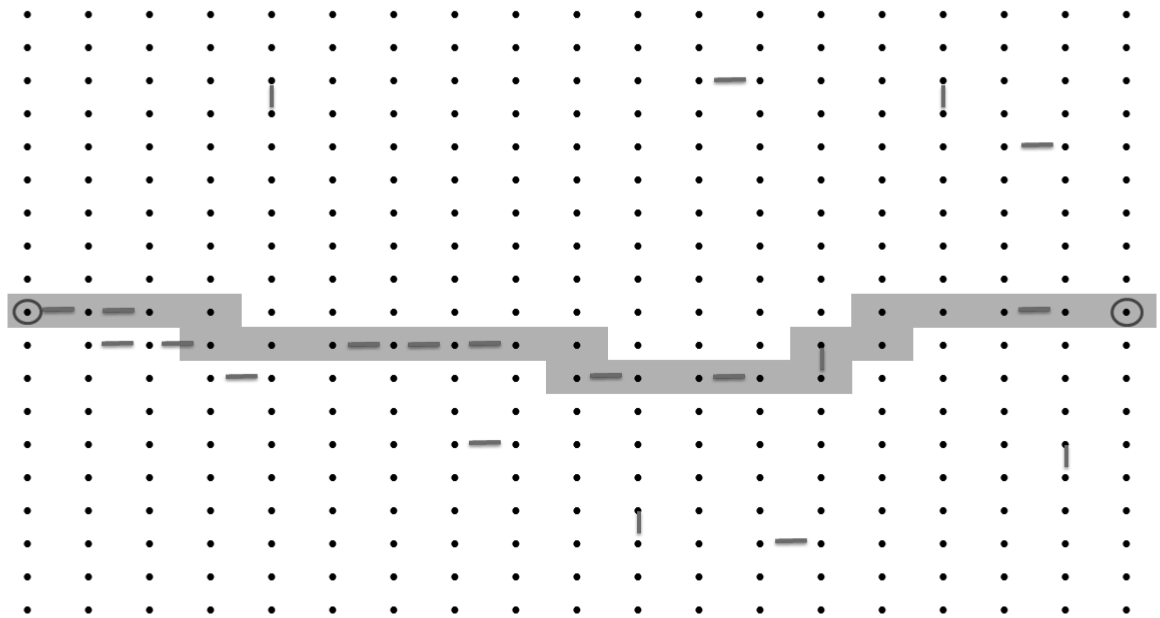

Figure 8: SA Solution of the big instance with $R = 1$ (Objective = 417).

Figure 9: SA Solution of the big instance with $R = 5$ (Objective = 212).

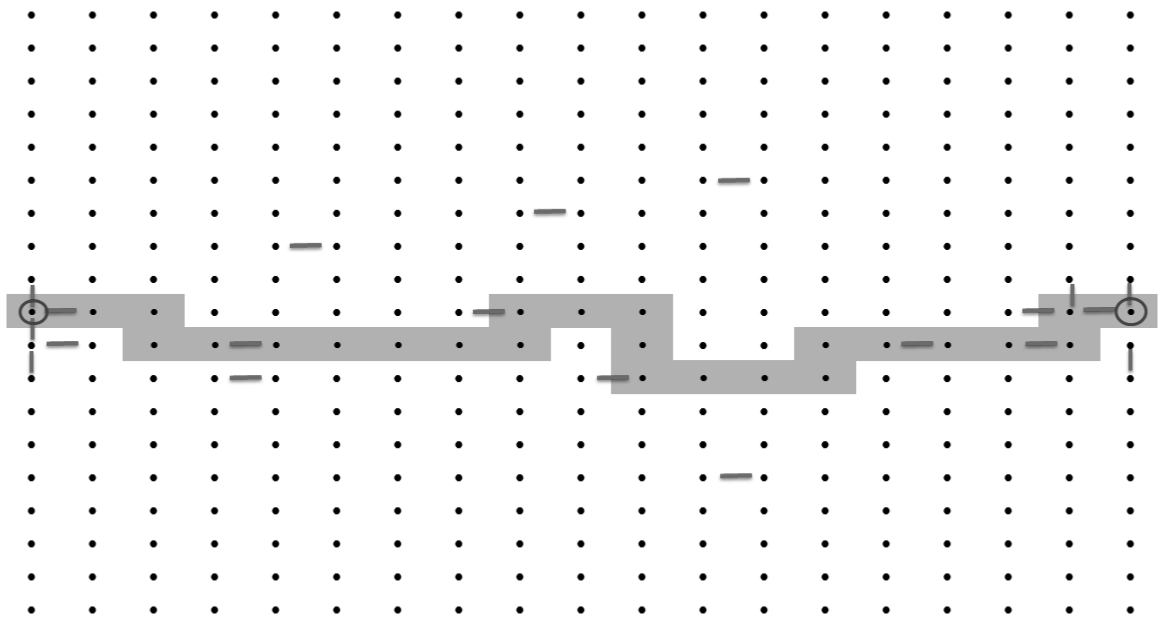

Figure 10: SA Solution of the big instance with $R = 15$ (Objective = 198).

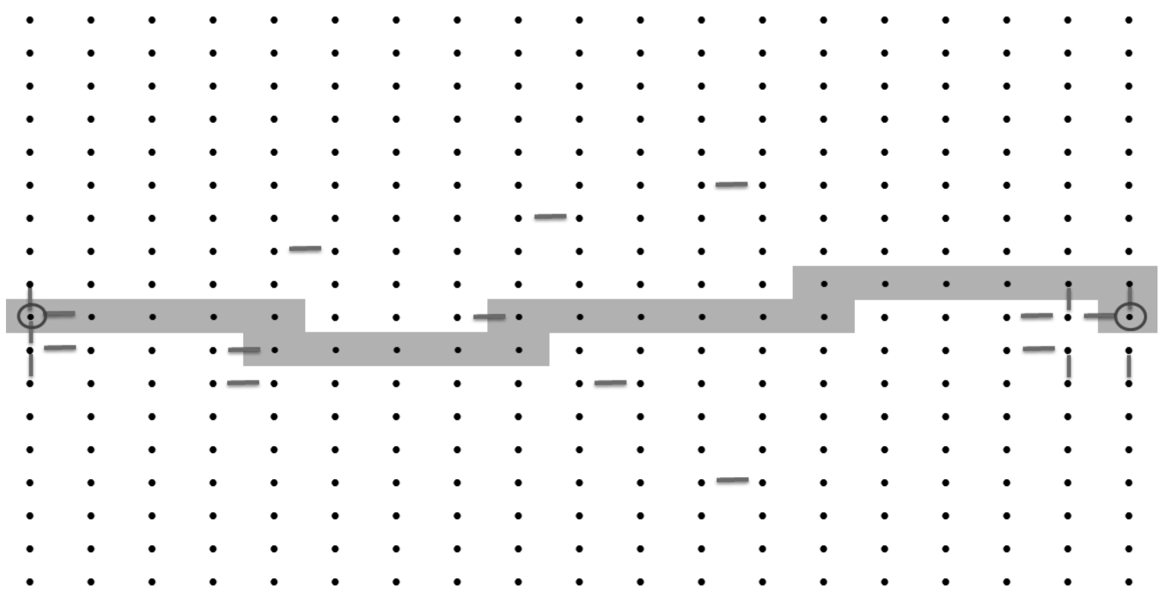

Figure 11: SA Solution of the big instance with $R = 66$ (Objective = 171).

6 Conclusion

Within this study, we consider a shortest path network interdiction problem including three players, a *leader, follower*, and *informant*. The follower and informant aim at finding the shortest path on a given network, whereas the leader seeks to maximize the follower's path length by interdicting the network components. Unlike existing studies, we include a case where the interdictions placed by the leader may become visible through the informant's activities. We present the formulations of Follower (F) problem, the Informant-Follower (IF) problem, and the Leader-Informant-Follower (LIF) problem that considers the objectives of the three players. Because the inner problem, (IF) is nonconvex, we are unable to obtain a single level formulation of the LIF problem. In order to deal with this, we apply the simulated annealing heuristic to find solutions to our problem. As in all heuristic methods, the parameter values have significant effect on the performance of the overall heuristic. With the purpose of finding the most effective parameter settings, we perform a DOE study, and propose a set of parameter values that can be implemented on any instance. We then apply these tuned parameters to a large-scale network having 361 nodes and 1362 arcs. We observe that the interdictions are placed with a conservative strategy as some of the information that is thought to be invisible to the follower may become visible. The LIF objective value decreases as R increases because the follower has more information about the true lengths of the arcs, and determine his/her path accordingly.

References

[1] H. Bayrak and M. D. Bailey. Shortest path network interdiction with asymmetric information. *Networks*, 52:133–140, 2008.

[2] G. Brown, M. Carlyle, J. Salmerón, and K. Wood. Defending critical infrastructure. *INFORMS*, 36:530–544, 2006.

[3] H. W. Corley and H. Chang. Finding the n most vital nodes in a flow network. *Management Science*, 21:362–364, 1974.

[4] S. P. Coy, B. L. Golden, G. C. Runger, and E. A. Wasil. Using experimental design to find effective parameter settings for heuristics. *Journal of Heuristics*, 7:77–97, 2001.

[5] D. R. Fulkerson and G. C. Harding. Maximizing the minimum source-sink path subject to a budget constraint. *Mathematical Programming*, 13:116–118, 1977.

[6] B. Golden. A problem in network interdiction. *Naval Research Logistics Quarterly*, 25:711–713, 1978.

[7] E. Israeli and R. K. Wood. Shortest-path network interdiction. *Networks*, 40:97–111, 2002.

[8] S. Kirkpatrick, C. Gelatt Jr, and M. Vecchi. Optimization by simulated annealing. *J. Stat. Phys*, 34:975, 1983.

[9] A. Malaviya, C. Rainwater, and T. Sharkey. Multi-period network interdiction problems with applications to city-level drug enforcement. *IIE Transactions*, 44:368–380, 2012.

[10] K. Malik, A. K. Mittal, and S. K. Gupta. The k most vital arcs in the shortest path problem. *Operations Research Letters*, 8:223–227, 1989.

[11] G. P. McCormick. Computability of global solutions to factorable nonconvex programs: Part I–Convex underestimating problems. *Mathematical programming*, 10:147–175, 1976.

[12] A. W. McMasters and T. M. Mustin. Optimal interdiction of a supply network. *Naval Research Logistics*, 17:261–268, 1970.

[13] D. P. Morton, F. Pan, and K. J. Saeger. Models for nuclear smuggling interdiction. *IIE Transactions*, 39:3–14, 2007.

[14] H. D. Ratliff, G. T. Sicilla, and S. H. Lubore. Finding the n most vital links in flow networks. *Management Science*, 21:531–539, 1975.

[15] J. Salmerón. Deception tactics for network interdiction: A multiobjective approach. *Networks*, 60:45–58, 2012.

[16] H. D. Sherali, W. P. Adams, and P. J. Driscoll. Exploiting special structures in constructing a hirearchy of relaxations for 0-1 mixed integer problems. *Operations Research*, 46:396–405, 1998.

[17] R. Wollmer. Removing arcs from a network. *Journal of the Operations Research Society of America*, 12:934–940, 1964.

[18] R. K. Wood. Deterministic network interdiction. *Mathematical and Computer Modelling*, 17:1–18, 1993.

[19] J. Yates, R. Batta, and M. Karwan. Optimal placement of sensors and interception resource assessment for the perception of regional infrastructure from covert attack. *Journal of Transportation Security*, 4:145–169, 2011.