OPTIMAL SENSOR DEPLOYMENT TO INCREASE THE SECURITY OF THE MINIMAL EXPOSURE PATH

by

Ezgi Karabulut

B.S., Industrial Engineering, Boğaziçi University, 2010

Submitted to the Institute for Graduate Studies in
Science and Engineering in partial fulfillment of
the requirements for the degree of
Master of Science

Graduate Program in Industrial Engineering

Boğaziçi University

2012

ACKNOWLEDGEMENTS

Everyone has that special person in their lives who, directly or indirectly, had a remarkable influence in shaping their future, and presented them a role model to look up to. I am lucky to have met that person at such an early age. I'd like to start by thanking Prof. Kuban Altınel for his endless support and guidance, for being an exemplary person in life, not merely in academia, for everything he has invested in me.

I'd further like to thank Prof. Necati Aras for his instructive comments and challenges. In these past two years he helped me build a perspective. I'd like to thank Prof. Cem Ersoy for taking part in my jury. I am also very grateful to Prof. Ümit Bilge especially for her warmth and kindness and all the precious faculty in our department for providing me the foundations of my future academic career as an industrial engineer. I thank TÜBİTAK for their financial support during my graduate study.

My dearest friends, with whom we shared life here, my comrade Mehmet, Gökalp, Burak and Turgut. We grew up together. All my fellow assistants, Oylum, Kaan, Nisa, Can, Onur, İsmail, Alper, Zeynep, Mustafa, Mert and Merve, for all the laughter and beautiful memories; Umut, Erinç and Emre for being the help needed and assistance as an elder; and especially Hande, who saw and supported the potential in me since I was a sophomore student. Deniz, who, above everything else, shares with me the same dream as being an academician one day. My life would be incomplete without her.

Last but definitely not the least, my dear family. My parents, who never hesitated a second when making sacrifices for the sake of my education. They deserve to take credit for all my accomplishments. My sister, ma chérie Dilara, the EQ supplier I need. I am trying very hard to set a good example to her, I hope I have succeeded. Finally, I dedicate my first piece of work in academic life to my beloved uncle, who passed away with the dream of his niece becoming a professor some day.

ABSTRACT

OPTIMAL SENSOR DEPLOYMENT TO INCREASE THE SECURITY OF THE MINIMAL EXPOSURE PATH

Wireless Sensor Networks (WSN) are based on collaborative work of a collection of sensors which are low-cost, multi-functional devices. There is a wide range of application areas of WSN including border surveillance and forest fire controls. Maximizing the security of maximal breach path, also known as the minimal exposure path, is an important approach in the design of WSN. This thesis focuses on this aspect, and develops a mixed integer bilevel programming formulation to determine the sensor locations that maximize the exposure on the minimal exposure path. The leader in the upper level problem intends to maximize the exposure on the maximal breach path by finding the optimal sensor locations. The follower in the lower level problem, on the other hand, selects the sensors to be destroyed to minimize the exposure on the maximal breach path. There are two basic heuristics proposed to solve this bilevel optimization problem. The first one employs tabu search on the upper level variables, and solves the lower level problem to optimality. The other heuristic breaks down the problem into three levels, performs tabu search on the first two levels, and solves the lowest level as the shortest path problem. The computational results indicate the success of the bilevel programming formulation and the solution methods.

ÖZET

EN KORUMASIZ GEÇİDİN SAVUNMASINI ENİYİLEYEN DUYGAÇ YERLERİNİN BULUNMASI

Kablosuz duygaç ağları, düşük güçlü, çok amaçlı ve ucuz, çok sayıda duygacın bir arada çalışmasını temel alır. Sınır denetiminden orman yangınlarının belirlenmesine kadar zengin bir uygulama alanı olan bu yapıların tasarımında önemli konulardan birisi korumasız geçidin güvenliğinin eniyilenmesidir. Bu konuya odaklanan çalışmada, duygaçların yerlerini, korumasız geçidin savunma güvenliğini enbüyükleyecek biçimde belirleyen bir çift düzeyli karışık tamsayı gösterimi geliştirilmektedir. Önder, korumasız geçidin güvenliğini enbüyükleyen eniyi duygaç yerlerini belirlemeyi amaçlarken, izleyici korumasız geçidin güvenliğini enküçükleyebilmek amacıyla işlevsizleştireceği duygaçlara karar vermektedir. Çift düzeyli gösterimin çözümü için önerilen yöntemlerden biri üst düzeyde Tabu arama gerçeklerken, alt düzeyde izleyicinin problemini kesin olarak çözmektedir. Diğeri ise problemi üç seviyeye ayırıp ilk iki seviyede Tabu arama gerçeklerken en alt seviyeyi en kısa yol problemi olarak çözmektetir. Deney verileriyle elde edilen bilgisayısal sonuçlar çift düzeyli programlama yaklaşımının başarılı olduğunu söylemektedir.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS

a_{jkr}	the coverage intensity realized at node j by a sensor of type r located at k
b	leader's budget for sensor deployment
$\bar{b}$	follower's budget for sensor destruction
c_r	unit cost of deploying a sensor of type r
$\bar{c}_r$	unit cost of destroying a sensor of type r
$d(j,k)$	Euclidean distance between node j of the intruder network and sensor location k
E	set of arcs in the intruder's network
F	leader's objective function
f	follower's objective function
G	leader's constraint set
g	follower's constraint set
N_i	set of nodes in the intruder's network
N_s	set of possible sensor locations
N_s^L	the left artificial copy of the set of possible sensor locations
N_s^R	the right artificial copy of the set of possible sensor locations
R	set of sensor types
u_{ijkr}	auxiliary variable denoting the multiplication $y_{kr}z_{ij}$
v_{ijkr}	auxiliary variable denoting the multiplication $x_{kr}z_{ij}$
w_j	binary variable that takes value 1 if follower visits node j in the shortest path
X	set of variables
x_{kr}	binary variable that takes value 1 if a sensor of type r is deployed at k and 0 otherwise
$(\hat{x}, \hat{y}, \hat{z})$	optimal solution of LP-DeNegre
y_{kr}	continuous variable denoting the destroyed fraction of the sensor of type r is located at k
(y^*, z^*)	optimal solution of the LLP

z_{ij}	binary variable that takes value 1 if follower uses the arc (i, j) in the shortest path and 0 otherwise
δ	minimum coverage level of nodes
ϵ	a very small number to guarantee that an inequality is satisfied strictly
λ	decrease in uncoverage of the network per unit cost
π_1	coefficients of DeNegre's valid inequality
π_0	right-hand side of DeNegre's valid inequality

LIST OF ACRONYMS/ABBREVIATIONS

CPU	Central Processing Unit
IP	Integer Programming
KKT	Karush-Kuhn-Tucker
LLP	Lower Level Problem
LP	Linear Programming
MECP	Minimal Exposure Coverage Problem
MILBP	Mixed Integer Linear Bilevel Programming Model
MILP	Mixed Integer Linear Programming
MINLBP	Mixed Integer Nonlinear Bilevel Programming Model
MIP	Mixed Integer Programming
SP	Shortest Path
T-IP	Tabu - Integer Programming Method
T-T-S	Tabu - Tabu - Shortest Path Method
TS	Tabu Search
ULP	Upper Level Problem
WSN	Wireless Sensor Networks

1. INTRODUCTION

Sensors are small, low-cost, low-power, multifunctional devices with three functional capabilities: sensing, communication and processing. Wireless Sensor Networks (WSNs) consist of large numbers of sensors, and have wide applications in remote environmental monitoring and target tracking.

Alongside problems such as localization, synchronization, and calibration in WSN [41], optimization researches focus mainly on routing, deployment and coverage problems. The most generic versions of these problems are associated with network flow, facility location-allocation, and set covering problems respectively. However, when they are considered in the WSN field, these problems need not preserve the generic structures. Coverage problems, for instance, include three main elements: coverage of the network, as the name indicates, connectivity of the network, and, as in most optimization problems, cost. Depending on the problem dealt with, the nature of these aspects may vary, i.e. some are treated as objectives, the others are kept as constraints. The coverage problem in WSN is not necessarily the minimization of total cost subject to the coverage and connectivity constraints; but can also emerge as the maximization of coverage subject to the connectivity and budget constraints. The decision variables in this case are related to the location of the sensors. Despite the unnecessity of the predetermination of sensor positions [1] due to their cheap and tiny nature; the location process may become advantageous in solving the coverage problem. This advantage also arises in solving problems in different contexts, which can also be modeled as a coverage problem in WSN. The most prominent example to those types of problems is the Art Gallery Problem [33], which is determination of the minimum number of sensors required in order to sense every point of the sensing field. The examples can be extended into scenarios where the WSN is associated and replaced with surveillance cameras or police stations.

There may be various approaches for the coverage problem [20], one of which is

the worst-case coverage, also known as the minimal exposure path, or maximal breach path. In this problem, given a set of sensor positions, the path with minimum coverage, i.e. least observability, is sought. When the problem is perceived as a security issue, the natural extension of this problem is the selection of sensor positions that maximize the observability of the minimal exposure path. Although a path and possible sensor locations are established in continuous domain, solution techniques are applied after discretization of the sensing field [12, 31].

Some problems in WSN, categorized under energy efficiency, security and pursuit-evasion games, are treated in a game-theoretic manner [27]. The notion of security originates the idea of attacker; analogous to the second player in Stackelberg games [35], also known as the follower. Depending on the structure of the problem, Stackelberg games can be formulated using a mathematical model, and these models are referred to in the literature as *bilevel programming problems* [3].

Bilevel programming is defined as "a mathematical program that contains an optimization problem in constraints" [9]. The existence of a leader and a follower are crucial in bilevel programming. Leader denotes the superior decision maker; and the optimization problem in the constraints belongs to the second decision maker, the follower. In general, the leader cannot determine the follower's action directly, however, has authority to manipulate the follower by controlling the environment necessary for follower's decision making. Technically speaking, the problems are not separable; the decision variables of the leader necessarily appear in the optimization problem of the follower.

The solutions of bilevel programming problems display structural differences from single level problems. The most trivial distinction is that in the case of integer programming problems, the optimal solution does not necessarily lie in the set of extreme points of the convex hull [32]. These differences make it compulsory to explore new solution techniques appropriate to the form of the bilevel problem dealt with. Some proposed solution techniques vary from the simple as reduction to single level, to the more

complicated as penalty functions suggested especially for nonlinear bilevel programming problems [13]. Although the fundamental tendency is towards optimal solution methods, the structure of the problem may compel derivation of heuristics [14].

This thesis proposes a new approach for the deployment of a WSN that is essentially a Stackelberg game, and can be modeled as a bilevel programming problem. The leader in our problem is in charge of determining the sensor locations in the network. The location of the sensors specifies the coverage intensity of each node in the network (which is a decreasing function of the distance between two nodes). The follower, after observing the locations of the sensors, destroys a certain amount of them (depending on his budget) and pursues the minimal exposure path in the remaining network. The sensors to be destroyed are chosen such that the length of the minimal exposure path is minimized. The objective of the leader, on the other hand, is to locate the sensors such that the length of the minimal exposure path in the undermined network is maximized. Similar to the existing works in the literature, a path is defined in a discretized network that coincides with the area to be sensed. This problem can be modeled as a mixed integer nonlinear bilevel problem, where nonlinearity occurs in the objective functions; and can be reduced to a mixed integer linear bilevel programming problem. Optimal solution techniques for problems in this form are not provided in the literature, and this research presents some heuristics that can be generalized to any problem that shares the same structure with ours.

2. LITERATURE SURVEY

2.1. Bilevel Programming

The solution methods for bilevel programming problems show vast variety depending on the structure of the problem. Variable types and nonlinearities are two important elements in determining proper methods. This variety leads to a prosperity in literature on solution techniques proposed for solving bilevel programming problems.

Colson *et al.* provide a comprehensive overview of bilevel optimization problems [13]. The study includes a survey of existing methods, categorized under four divisions: extreme point approaches for linear problems, Karush-Kuhn-Tucker (KKT) conditions and complementary pivoting for models with linear lower level problems, descent methods, and finally for nonlinear problems penalty function and trust region methods. Chinchuluun *et al.* give a brief introduction on these methods with extreme point algorithms and branch-and-bound algorithms [11].

The algorithm proposed by Bard and Moore in [4] is applicable to bilevel problems with binary variables. It is essentially an implicit enumeration scheme, based on binary trees. A second version is presented, differing in bilevel feasibility checks, which is more relaxed. They continue their research in [32], on mixed integer bilevel programming problems. The algorithm is again, based on branch-and-bound trees, and compared to the previous research, bounding is more dwelt on.

Wen and Hang [38] deal with bilevel problems whose upper level decision variables are binary, and lower level decision variables are continuous. Their solution methods are also based on binary search trees. They initially describe an exact algorithm; however it is inefficient due to the computation of bounds for nodes. They continue with a heuristic modification, that resembles the branch-and-bound procedure, and propose an index for prioritizing the variable to be branched.

Fanghnel and Dempe [19], on the other hand, consider the reversed case, where the upper level problem is continuous, and the lower level problem is discrete. They investigate the optimistic and pessimistic scenarios, and introduce a weak solution function. The study concludes with the conditions for local and global optimality.

The method of DeNegre and Ralphs reduces the bilevel problem to a single level [15]. Their initial assumption is that all variables are binary. The bilevel problem is reformulated by, first, discarding the lower level objective and combining the constraint sets, and second, relaxing the variables. The relaxed problem is solved, and valid inequalities are applied until and integer optimal solution is obtained; which then is checked for bilevel feasibility. The study suggests another valid inequality for cutting off bilevel infeasible integer extreme points of the relaxed problem. The procedure is repeated until the integer optimal solution of the relaxed problem satisfies the bilevel feasibility constraints. This solution is the global optimal solution of the bilevel problem.

Some interesting solution approaches are also encountered in studies not merely based on mathematical bilevel formulations, but are simply real world application problems. Küçükaydın *et al.* [26] for instance, solve their bilevel formulation of competitive facility location problem in two steps: first fixing the lower level variables and solving for the upper level problem; and then fixing the upper level variables at the values obtained in the first step, and solving the lower level problem for bilevel feasibility. Başdere *et al.* [5] use tabu search for their problem with binary upper level decision variables.

2.2. Network Interdiction Problems

Most network interdiction problems have the form of a Stackelberg game, with the notion of an intruder, a decision maker who intrudes, interdicts [39]. Hence, the problems whose mathematical models are provided make use of bilevel programming formulations. The decision variables, thus the structure of the problem, and even the

decision makers vary with different problem formulations.

Shortest path network interdiction problem is proposed by Israeli and Wood [23], and is basically interdicting the arcs of a network in order to maximize the shortest path. The leader determines the arcs to be interdicted, and the follower, given the interdicted arcs, selects the shortest path. Notice that once the interdiction variables are fixed, the lower level optimization problem is a simple shortest path problem. The bilevel problem is solved by decomposition algorithms, where the upper level generates all possible interdiction schemes.

A recent extension to shortest path network interdiction is presented by Bayrak and Bailey [6], in which the information shared by the leader and the follower is not identical. This problem is called shortest path network interdiction problem with asymmetric information. The upper level decision variables in the model are binary, and represent the interdiction of arcs, and the lower level optimization problem is a shortest path problem. Both decision makers want to modify the shortest path, the leader maximizing, and the follower minimizing. The asymmetry occurs in the coefficients of the objective functions, which are the arc weights. The challenge in this paper is the nonlinearity in some of the constraints and the objective function, and is resolved by a new variable definition.

One of the most prevailing areas of research in network interdiction problems is hazardous material (hazmat) transportation. Although the mathematical models in hazmat problems share a common structure, most of the time with the simple change of definition of parameters, the problem is carried to a different context. Often this change of context requires redefinition of variables, but in general it can be concluded that the distinction between models is not major.

One application of hazmat transportation, by Bianco *et al.* [8], focuses on the risk issue and introduces the concept of risks on arcs. The lower level optimization problem is the minimization of total risk on all arcs by determining the flow amounts on each

arc, given the arc capacities. The upper level decision maker sets the arc capacities to minimize the maximum risk on individual arcs. There is multicommodity flow, and all variables are defined to be continuous. This problem is solved by reduction to single level. What is discussed in this paper is the case of multiple optima in the lower level problem, and how the lower level solutions affect the upper level optimal solution.

Verter and Kara, in [25], provide another aspect of hazmat transportation. The upper level optimization problem minimizes total exposition to hazmat, and the lower level optimization problem minimizes the total distance traversed. The decision variables for the upper and lower level problems are arc availabilities and arc selections respectively. All variables are defined to be binary, however the lower level variables can be relaxed due to the total unimodularity. With this relaxation the bilevel problem is reduced to single level and solved accordingly.

Hazmat transportation network design problems need not necessarily have bilevel models. Verter and Kara [37] propose a path based single level model, where the leader determines the paths to be used by the follower. This may, however, be criticized by being unrealistic due to its over-regulated nature. Furthermore, in order to attain global optimality, all paths need to be generated, causing exponential number of variables.

A different approach to network design comes from Erkut and Alp [18], where they solve for the optimum communication Steiner tree, which is renamed as Minimum Risk Hazmat Tree, instead of the shortest path. Berman *et al.* [7] model the problem of designing emergency response networks by referring to maximal cover problem.

Besides hazmat transportation, electric networks are also considered under network interdiction problems. Salmeron *et al.* analyze the electric grid security [34]. Unlike other problems encountered, the intruder is the leader in this problem and controls the interdiction variables. The lower level problem is a general power flow model, and the bilevel problem is the maximization of minimum total cost. The upper and lower level decision variables are binary and continuous respectively. The proposed

method for the solution of this problem is generating interdiction plans as upper level decision variables and using a Benders decomposition-like iterative algorithm.

An interesting piece of research in literature is by Holmgren *et al.* [22], where the problem is modeled as a Nash game, rather than a Stackelberg game. The problem is a two player, zero-sum, strictly competitive, simultaneous game and no mathematical formulation is provided. The experimentation of the research is based on 12 different attack scenarios and 6 different defense strategies.

2.3. Coverage Problem

The basic structure of minimal exposure path related coverage problem in WSN is exemplified by Yates *et al.* in [40]. The problem is the minimization of maximum probability of an undetected path, where exposure is equivalent to the probability of detection. The upper level decision maker chooses the sensor locations, which determines the arc coverage values, and given this network, the lower level decision maker selects the shortest path. All variables in this problem are binary. The challenge is that the objective function is the multiplication of nonlinear exponential terms. To overcome this challenge, the definition of variables and parameters are played with. All possible paths are generated, this discards the nonlinearity, and the coefficients are updated by taking logarithms, which eliminates both the multiplication and exponentiality. This new formulation has a downside of exponential number of constraints, and to resolve that a special form of Benders decomposition is applied.

Megerian *et al.* bring a novel perspective to the computation of maximal breach path [30]. In order to find the maximal breach path in a WSN given the locations of sensors, unlike the general approach of discretizing the area, they work in the continuous domain. They claim that the maximal breach path lies on the borders of the Voronoi diagram. The problem does not involve locating the sensors, it is merely an analysis of the best and worst-case coverages in the given network using Voronoi diagrams.

Jain *et al.*'s approach in [24] is parallel to generating all possible sensor location schemes, referred to as strategies, and treat those strategies as decision variables. The aim is the protection of targets, and there is a payoff associated with the protection and harm of each target. They solve this problem by a branch-and-price algorithm.

The coverage problem investigated in [17] by Dhillon *et al.* is the minimization of the number of sensors such that every point in the field is sensed with at least a minimum confidence level, which is defined as the probability of detection. The probability of detection is converted to the probability of miss, followed by the application of logarithm, for linearization purposes. The suggested solution algorithm iteratively selects the best candidate point for sensor placement, and the candidates are prioritized according to their effect on the individual miss probabilities of the points in the network. The algorithm is further developed by Dhillon and Chakbarty in [16] to consider the candidates by different priorities; maximizing average coverage and minimum coverage values.

Başdere *et al.* deal with another coverage problem in WSN that has a bilevel model formulation [5]. The problem is the maximization of the minimum total number of covered nodes. The upper level decision maker determines the sensor placements, and the lower level decision maker selects the sensors to be destroyed that minimize the total number of covered nodes. The final sensor locations determine the node coverage levels, and whether or not the coverage level exceeds a certain threshold is denoted by another variable. This formulation has product terms in the constraints, which are removed by introducing new variables. Initially, all variables are defined to be binary, however, in an extension of the model, sensor destruction variables are considered to be continuous as well.

3. PROBLEM FORMULATION

The problem studied in this thesis is the determination of sensor locations that maximize the length of the maximal breach path in a given network. Maximal breach path (also known as the minimal exposure path) through a sensing field is defined such that the distance from any point on the path to the closest sensor is maximum [20], the path that has the least observability in the field. As the leader, we are in the position of the defender, and the maximal breach path constitutes a threat to the security of the network by being the least observable path. When an intruder attempts to trespass the network, he prefers to follow the path which is furthest from the sensors, i.e. the maximal breach path. Thus, the length of the maximal breach path is equivalent to the observability of the path that the intruder takes. The defender aims to maximize the security of this path by maximizing its length, and the intruder wants to minimize the security by minimizing the length.

In general, a path is described in continuous domain, however, since the breach path cannot be computed in continuous domain using mixed integer programming formulations, we define a network, consisting of nodes and arcs, that coincides with the WSN field to search for the path. The pursuit of the shortest path is performed over this network. The length of a path designates its observability, its exposure to the sensors, and is defined as the double sum of intensities of all sensors on all nodes of the path. Therefore, the location of the sensors is the key element in modifying the shortest path. This relation between the nodes and sensor locations leads to the concept of weights of nodes (designating the coverage intensity on the node), whereas the shortest path problem applies to problems with weights on arcs. To preserve the notion of the shortest path, the weights are assigned to arcs, namely the weight of all the outgoing arcs from a node have the weight associated with that node.

3.1. Bilevel Programming Problems

There are two decision makers in this problem: one that deploys the sensors and the other that destroys the sensors and follows the shortest path in the remaining network. These two decision makers can be interpreted in various ways, as the defender and intruder, the protagonist and antagonist, or the leader and follower. The important point here is that they have conflicting objectives. In this model, the leader is prioritized, and is given the right for the first move. The problem is viewed from the leader's perspective, however this is not sufficient to anticipate the follower's decisions. One cannot simply set the follower's decisions variables under the leader's objective, considering the fact that their objectives disagree, and the follower's decisions need not comply with the leader's expectations. For that reason, the actions of the follower must be determined by another optimization problem embedded in the leader's optimization problem. These types of problems are named as bilevel programming problems in the literature and have the following structure, where x and y are the decision variables for the leader and the follower respectively:

$$\begin{aligned} &\max_{x} F(x,y) \\ &\text{s.t.} \qquad G(x,y) \leq 0 \\ &\qquad \max_{y} f(x,y) \\ &\qquad \text{s.t.} \qquad g(x,y) \leq 0 \end{aligned}$$

3.2. Mixed Integer Nonlinear Bilevel Programming Formulation

As previously stated, the follower selects the set of sensors to be destroyed that minimizes the shortest path. The leader, foreseeing the action of the follower, searches for the sensor locations that yield the longest value for the shortest path after the destruction of sensors. In other words, the leader wants to maximize the length of the shortest path that the follower chooses.

The following are the definition of the sets, parameters and decision variables used in the formulation:

Table 3.1. Declaration of sets, parameters, and decision variables.

Sets:	
N_s	set of possible sensor locations
R	set of sensor types
N_i	set of nodes in the intruder's network
E	set of arcs in the intruder's network
Parameters:	
a_{jkr}	the coverage intensity realized at node j of the intruder's network by a sensor of type r located at point k
c_r	unit cost of deploying a sensor of type r
$\bar{c}_r$	unit cost of destroying a sensor of type r
b	leader's budget for sensor deployment
$\bar{b}$	follower's budget for sensor destruction
Decision variables:	
x_{kr}	binary variable that takes value 1 if a sensor of type r is deployed at point k
y_{kr}	continuous variable denoting the destroyed fraction of type r sensor located at point k
z_{ij}	binary variable that takes value 1 if follower uses the arc (i, j) in the shortest path
u_{ijkr}	auxiliary variable representing the multiplication $y_{kr}z_{ij}$
v_{ijkr}	auxiliary variable representing the multiplication $x_{kr}z_{ij}$

Then the Minimal Exposure Coverage Problem (MECP) can be formulated as the following Mixed Integer Nonlinear Bilevel Problem (MINLBP).

$$\max_{x} \sum_{(i,j)\in E} \sum_{k\in N_s} \sum_{r\in R} a_{jkr} x_{kr} (1 - y_{kr}) z_{ij} \tag{3.1}$$

s.t.

$$\sum_{k\in N_s} \sum_{r\in R} c_r x_{kr} \leq b \tag{3.2}$$

$$x_{kr} \in \{0, 1\} \qquad k \in N_s, r \in R \tag{3.3}$$

$$\min_{y,z} \sum_{(i,j)\in E} \sum_{k\in N_s} \sum_{r\in R} a_{jkr} x_{kr} (1 - y_{kr}) z_{ij} \tag{3.4}$$

s.t.

$$\sum_{k\in N_s} \sum_{r\in R} \bar{c}_r y_{kr} \leq \bar{b} \tag{3.5}$$

$$y_{kr} \leq x_{kr} \qquad k \in N_s, r \in R \tag{3.6}$$

$$\sum_{j:(i,j)\in E} z_{ij} - \sum_{j:(j,i)\in E} z_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \qquad i \in N_i \tag{3.7}$$

$$z_{ij} \in \{0, 1\} \qquad (i, j) \in E \tag{3.8}$$

$$y_{kr} \geq 0 \qquad k \in N_s, r \in R \tag{3.9}$$

In this formulation, (3.1) to (3.3) represent the upper level problem (ULP), and (3.4) to (3.9) the lower level problem (LLP). The only external constraints regarding sensor locations are cost related budget constraints, (3.2) and (3.5). Constraints (3.7) are generic path constraints, and constraints (3.6) prevent any attempt to destroy a sensor that has not been deployed. As of the objective function, a type r sensor at point k is active if it has been deployed and in the amount that has not been destroyed, $x_{kr}(1 - y_{kr})$. Then the observability of a node j becomes the sum of $a_{jkr} x_{kr}(1 - y_{kr})$ values, for all locations $k \in N_s$ and all types $r \in R$. The drawback of this formulation originates from the nonlinearity in the objective function. To begin with, notice that

$x_{kr}(1 - y_{kr})$ is equal to $x_{kr} - y_{kr}$ due to constraint (3.6). When $x_{kr} = 0$, y_{kr} is forced to 0 and the value is 0 and when $x_{kr} = 1$, $x_{kr}(1 - y_{kr}) = 1 - y_{kr} = x_{kr} - y_{kr}$ holds trivially. Hence, the objective functions (3.1) and (3.4) become

$$\sum_{(i,j)\in E} \sum_{k\in N_s} \sum_{r\in R} a_{jkr}(x_{kr} - y_{kr})z_{ij}. \tag{3.10}$$

3.3. Mixed Integer Linear Bilevel Programming Formulation

Despite this linearization attempt, the nonlinearity in the objective function persists. Considering the fact that the nonlinearity is caused by the multiplication of two variables, at least one of which is binary, the proposed solution is to introduce a new variable to replace the product of the decision variables y_{kr} and z_{ij} as well as x_{kr} and z_{ij}. It is important to underline that the term $x_{kr}z_{ij}$ in the objective function of the follower is not nonlinear, because the lower level optimization problem treats x_{kr} as a parameter coming from the ULP. If we let u_{ijkr} replace the product $y_{kr}z_{ij}$, and v_{ijkr} replace the product $x_{kr}z_{ij}$ for the ULP, we obtain the following mixed integer linear bilevel problem (MILBP):

$$\max_{x,v} \sum_{(i,j)\in E} \sum_{k\in N_s} \sum_{r\in R} a_{jkr}(v_{ijkr} - u_{ijkr}) \tag{3.11}$$

s.t.

$$\sum_{k\in N_s} \sum_{r\in R} c_r x_{kr} \leq b \tag{3.12}$$

$$v_{ijkr} \leq x_{kr} \qquad (i,j) \in E, k \in N_s, r \in R \tag{3.13}$$

$$v_{ijkr} \leq z_{ij} \qquad (i,j) \in E, k \in N_s, r \in R \tag{3.14}$$

$$v_{ijkr} \geq x_{kr} + z_{ij} - 1 \qquad (i,j) \in E, k \in N_s, r \in R \quad (3.15)$$

$$x_{kr} \in \{0,1\} \qquad k \in N_s, r \in R \quad (3.16)$$

$$v_{ijkr} \geq 0 \qquad (i,j) \in E, k \in N_s, r \in R \quad (3.17)$$

$$\min_{y,z,u} \sum_{i \in N_i} \sum_{j \in N_i} \sum_{k \in N_s} \sum_{r \in R} a_{jkr}(x_{kr} z_{ij} - u_{ijkr}) \quad (3.18)$$

s.t.

$$\sum_{k \in N_s} \sum_{r \in R} \bar{c}_r y_{kr} \leq \bar{b} \quad (3.19)$$

$$y_{kr} \leq x_{kr} \qquad k \in N_s, r \in R \quad (3.20)$$

$$\sum_{j \in N_i} z_{ij} - \sum_{j \in i} z_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \qquad i \in N_i \quad (3.21)$$

$$u_{ijkr} \leq y_{kr} \qquad (i,j) \in E, k \in N_s, r \in R \quad (3.22)$$

$$u_{ijkr} \leq z_{ij} \qquad (i,j) \in E, k \in N_s, r \in R \quad (3.23)$$

$$u_{ijkr} \geq y_{kr} + z_{ij} - 1 \qquad (i,j) \in E, k \in N_s, r \in R \quad (3.24)$$

$$z_{ij} \in \{0,1\} \qquad (i,j) \in E \quad (3.25)$$

$$y_{kr} \geq 0 \qquad k \in N_s, r \in R \quad (3.26)$$

$$u_{ijkr} \geq 0 \qquad (i,j) \in E, k \in N_s, r \in R \quad (3.27)$$

The new formulation MILBP is now a linear bilevel mixed integer program. The constraints (3.13), (3.14), and (3.15) are required for the definition of v_{ijkr}, and (3.22), (3.23), and (3.24) are required for the definition of u_{ijkr}. The other constraints are as explained in the MINLBP formulation (3.1) – (3.9). Since x_{kr} and z_{ij} are binary variables, it is sufficient to relax u_{ijkr} and v_{ijkr} to be continuous variables.

4. SOLUTION METHODS

4.1. Exact Solution Methods

Alongside all the heuristics generated for solving bilevel programming problems, DeNegre and Ralphs [15] focus their research on the optimal solution of bilevel programming problems with pure integer upper and lower level problems. They initially propose their method as an improvement for the branch-and-cut search method suggested by Bard and Moore [4]. However the implementation of branch-and-cut is not necessary to utilize DeNegre's method. In fact, we adapt to our problem the suggestion of DeNegre for processing a single node of the branch-and-cut tree. It is true that our problem formulation includes non-integer variables, the most prominent of which is the set of destruction variables. However, with a small modification from partial destruction of sensors to complete destruction, this problem is overcome, and all the remaining variables can be used as binary.

The underlying idea in this iterative method is ignoring the bilevel optimality condition during the search of candidate solutions, and performing a check of bilevel feasibility only after a candidate is obtained. The lower level problem (LLP) is merged into the upper level problem (ULP) by discarding the lower level objective function and including the constraint set of the LLP in the ULP. This procedure yields a single level binary linear programming problem. The variables are further relaxed to be continuous in the $[0, 1]$ interval and what remains is a linear programming problem, LP-DeNegre. LP-DeNegre is defined to be a dynamic formulation, at every iteration of the method new cuts are included in the model. The following, (4.1) – (4.14), is the initial LP which forms the base model of the method:

$$\max \sum_{(i,j)\in E} \sum_{k\in N_s} \sum_{r\in R} a_{jkr}(v_{ijkr} - u_{ijkr}) \tag{4.1}$$

s.t.

$$\sum_{k\in N_s} \sum_{r\in R} c_r x_{kr} \leq b \tag{4.2}$$

$$\sum_{k\in N_s} \sum_{r\in R} \bar{c}_r y_{kr} \leq \bar{b} \tag{4.3}$$

$$y_{kr} \leq x_{kr} \qquad k \in N_s, r \in R \tag{4.4}$$

$$\sum_{j:(i,j)\in E} z_{ij} - \sum_{j:(j,i)\in E} z_{ji} = \begin{cases} 1 & i = s \\ -1 & i = t \\ 0 & \text{otherwise} \end{cases} \qquad i \in N_i \tag{4.5}$$

$$v_{ijkr} \leq x_{kr} \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.6}$$

$$v_{ijkr} \leq z_{ij} \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.7}$$

$$v_{ijkr} \geq x_{kr} + z_{ij} - 1 \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.8}$$

$$u_{ijkr} \leq y_{kr} \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.9}$$

$$u_{ijkr} \leq z_{ij} \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.10}$$

$$u_{ijkr} \geq y_{kr} + z_{ij} - 1 \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.11}$$

$$0 \leq z_{ij} \leq 1 \qquad (i,j) \in E \tag{4.12}$$

$$y_{kr}, x_{kr} \geq 0 \qquad k \in N_s, r \in R \tag{4.13}$$

$$u_{ijkr}, v_{ijkr} \geq 0 \qquad (i,j) \in E, k \in N_s, r \in R \tag{4.14}$$

Each iteration of DeNegre's method starts by solving LP-DeNegre. If the optimal solution contains fractional values, standard cutting plane procedures are applied to remove the current solution from the set of feasible solutions. Otherwise, if the solution is binary, this solution, say $(\hat{x}, \hat{y}, \hat{z})$, is treated as a candidate solution, and bilevel feasibility check is performed. The bilevel feasibility check is essentially fixing the upper level variables, $\hat{x}$, and solving the original LLP of MILBP. Denote the optimal solution of the LLP by (y^*, z^*). If the objective function value associated with $(\hat{x}, \hat{y}, \hat{z})$ is equal

to the objective function value associated with $(\hat{x}, y^*, z^*)$, then it can be concluded that the solution $(\hat{x}, \hat{y}, \hat{z})$ is bilevel feasible, and since it also satisfies the integrality conditions, it is an optimal solution to our bilevel problem. Notice that (y^*, z^*) may not necessarily be equal to $(\hat{y}, \hat{z})$ due to multiple optima case. As long as the objective function values are equal, $(\hat{y}, \hat{z})$ is also an optimal solution of the LLP.

However, if $(\hat{x}, \hat{y}, \hat{z})$ fails the bilevel feasibility check, then it must be removed from the feasible solution set. DeNegre and Ralphs propose a valid inequality for this purpose; it has the following generic form

$$\pi_1 X \leq \pi_0 - 1. \tag{4.15}$$

Let the constraint set of LP-DeNegre (including the cuts applied in previous iterations) be denoted by $AX \leq b$, or equivalently $a_j X \leq b_j$, for $j = 1 \dots m$, where $X = (x, y, z, u, v)$ is the vector of variables and m is the number of constraints; and let J be the set of tight constraints at $(\hat{x}, \hat{y}, \hat{z})$. Then π_1 and π_0 in (4.15) are defined to be $\pi_1 = \sum_{j \in J} a_j$, and $\pi_0 = \sum_{j \in J} b_j$. This new inequality (4.15) is included in LP-DeNegre, and the next iteration begins. The iterations continue until the optimal solution of LP-DeNegre is an integer solution satisfying the bilevel feasibility check.

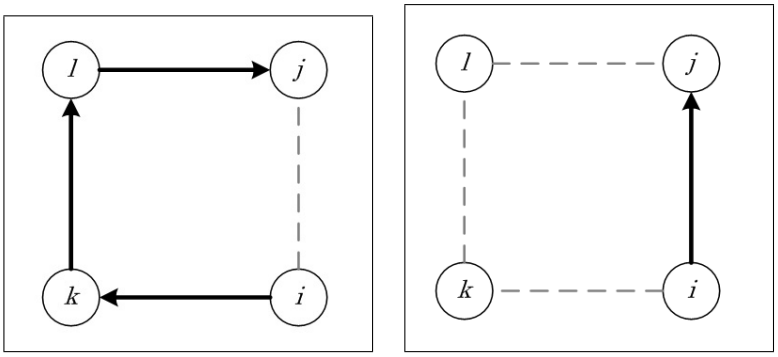
The main drawback of this method when applied to our problem is that the objective functions of LLP and ULP are equal, with opposite sense of optimization. This method is more appropriate for problems where the upper and lower level objectives do not necessarily comply with (in which case there would be no need for bilevel programming), however do not oppose each others'. Because of this opposition in our problem, LP-DeNegre tends to generate candidate solutions that do not satisfy bilevel feasibility conditions. In order to prevent that, one can embed some additional constraints to the base model (4.1) to (4.14), which the optimal solution of the LLP will certainly satisfy.

The additional constraints we propose can be categorized into three groups:

Cycle prevention: The follower is searching for the shortest path, and it is trivial that in a network where all arc weights are nonnegative, there is a shortest path which does not include cycles. Due to the definition of edges in our network (N_i, E), only cycles of length 2 are possible, between neighboring nodes, and the following constraints (4.16) are sufficient to prevent these 2-cycles:

$$z_{j,j+1} + z_{j+1,j} \leq 1 \tag{4.16}$$

Shortcut: It is worthy to remind the weight assignment to arcs once again. The weights are assigned to arcs such that the length of a path is equal to the sum of the weights of nodes on the path. Consider the paths in Figure 4.1a and Figure 4.1b.



(a) Example Path 1 (b) Example Path 2

Figure 4.1. Example Paths Demonstrating *U-shapes*.

Regardless of the incoming path to node i and the outgoing path from node j, the demonstrated path in Figure 4.1a adds $weight_i + weight_k + weight_l$ to the length of the path, whereas the path in Figure 4.1b has only the component $weight_i$, where $weight_i$ is, as the name suggests, the weight of node i, or equivalently, the weight of all outgoing arcs from node i. It is evident that the length of a path having an induced

path as in Figure 4.1a is at least as large as the path obtained by replacing the structure in Figure 4.1a by a single arc as in Figure 4.1b. We name the structures in Figure 4.1a as *U-shape* and claim that the shortest path in the given network does not contain any *U-shape*s. In order to prevent the *U-shape*s, we introduce another set of binary variables, w_j, that denotes whether or not node j is visited along the path. Then, for all permissible arcs (i, j), if both nodes i and j are visited, then the arc (i, j) must be included in the path.

$$\sum_{i \in N_i:(i,j) \in E} z_{ij} = w_j \tag{4.17}$$

$$w_i + w_j \leq z_{ij} + 1 \tag{4.18}$$

Constraint (4.17) forces w_j to be 1 if node j has been visited; and constraint (4.18) sets $z_{ij} = 1$ if both nodes i and j have been visited. Notice that constraints (4.18) are only defined for $(i, j) \in E$.

Budget: It is typical for the budget constraints to be given in less than or equal to form, since enforcement of precise use of the budget might yield suboptimality, or even infeasibility in some cases. However, in this problem, it is trivial that destroying a sensor improves the follower's objective; and if the budget allows the destruction of an additional sensor, the follower will choose to destroy it. In other words, at optimality the amount of unused budget of the follower should be strictly less than the minimum cost of destroying a sensor. Let $c_{min} = \min_{r \in R} c_r$, and ϵ a small number which guarantees that the original inequality is satisfied strictly. Then, the budget lower bound constraint becomes

$$\sum_{k \in N_s} \sum_{r \in R} \bar{c}_r y_{kr} \geq \bar{b} - \bar{c}_{min} + \epsilon. \tag{4.19}$$

It is important to include constraint (4.19), because when the sensor destruction decisions are made under the leader's initiatives, the solution tends to minimize the number of destroyed sensors, which conflicts with the follower's objective.

4.2. Heuristic Solution Methods

Current solvers are designed to handle a single level optimization problems, hence they are not suitable for bilevel programming problems. This creates the motive to separate the search over the upper and lower level variables, define two independent problems and find a means of information transfer between them. Our heuristics display this structure and investigate the solution space in separate levels.

The upper level problem is common in all proposed heuristics and is simply a combinatorial problem of selecting a subset from the set of possible sensor locations for sensor deployment. In our heuristics, tabu search algorithm (TS) is implemented, a widespread method for combinatorial problems, the details of which are provided in [21]. Tabu search algorithm can be summarized as follows: Given a solution, referred to as the *current solution*, a list of neighboring solutions is generated. The most promising solution from this list is selected to be the next current solution, provided that it satisfies some criteria. Cycling during the search is prevented by defining a list of prohibited moves, called *tabu moves*, however these can be overruled if some *aspiration criteria* are satisfied. The search continues until the *termination criteria* are met. During the search, the objective function values of the current solutions need not be nondecreasing, and in order to compensate for any possible disadvantages of missing out a promising solution, the solution with the highest objective function value is stored under the name *incumbent solution*.

4.2.1. T-IP : Tabu - Integer Programming

The first heuristic that we propose is named T-IP. T-IP divides the problem into two levels, and as stated previously, performs tabu search over the upper level variables, i.e. the sensor deployment decisions. In TS, when evaluating the neighboring solutions, the objective function value corresponding to that solution is required. In this case, the objective function value corresponding to a set of sensor locations for deployment in the upper level search should also include the sensor destruction decisions in the lower level. These decisions are produced by solving the lower level problem to optimality. Namely, the mixed integer linear programming (MILP) problem obtained by fixing the upper level decision variables in the LLP is solved to optimality with the use of MILP solvers.

4.2.2. T-T-S: Tabu - Tabu - Shortest Path

T-T-S methods extend the original problem to three levels, the first two levels are combinatorial sensor deployment and destruction problems respectively, and the third level is a shortest path (SP) problem. In T-T-S methods the upper level decision space is also searched via TS; however, unlike T-IP, the LLPs are not solved to optimality, but rather solved by tabu search over the destruction variables. Once the deployments and the destructions are fixed, the remaining problem is an SP problem and is easily solved to optimality. However, notice that in order to apply tabu search for the LLP, the decision variables should be defined binary. This problem is overcome by, without altering the problem structure, prohibiting partial destruction and allowing only complete destruction of sensors.

We should point out that the solutions of the LLP by heuristics may be suboptimal. The LLP is a minimization problem and suboptimal solutions given by TS most commonly have higher objective function values than the optimal results. The upper level problem, on the other hand, is a maximization problem and during the neighborhood search TS favors neighboring solutions with higher objective function

values. What should not be overlooked is that a candidate solution with a higher objective function value selected by the upper level tabu search might indeed have a lower optimal objective function value than provided by the lower level tabu search, and might not be as promising as expected. Because of that reason, the need for an auto-correction mechanism arises. Namely, one needs to check whether the sensor destruction decisions yielded by the lower level tabu search is actually the true response of the follower given the sensor locations; or if not, how far it is from the optimal response. The answer to these questions is given by solving the LLP at a given set of sensor locations to optimality. This is what is meant by auto-correction, and, as expected, is more time consuming than the tabu search heuristic for the LLP. The tradeoff at this point is between calling the auto-correction function as frequently as possible to increase accuracy, and as rarely as possible to preserve low time complexity. Two different versions of T-T-S are developed, which are dissimilar in this regard.

Recall once again the procedure of selecting the new current solution in tabu search. The list of neighbor solutions are ranked according to their objective function values and the solution with the highest objective function value is considered as the new candidate. If the objective function value of the candidate solution is higher than the incumbent objective value, aspiration criteria apply. Otherwise, whether or not this candidate disobeys the tabu moves is checked. If the candidate is found appropriate, the current solution is updated as the candidate solution, if not, the candidate is discarded and the procedure is repeated for the next solution in the list as the candidate.

Two versions only modify the aspiration criteria step. The general structure of the procedure is preserved.

4.2.2.1. T-T-S1. When encountered with a candidate solution whose objective function value is larger than the incumbent value, the LLP corresponding to that candidate solution is solved to optimality. If the new obtained objective function value is equal to the previous value, then the candidate solution is the new current solution, and the incumbent solution is updated accordingly. Otherwise, the objective function value of

the candidate solution is updated as the new obtained objective function value, and the list of neighbor solutions are reranked.

4.2.2.1. T-T-S2. For a candidate solution with a higher objective function value than the incumbent value, the corresponding LLP is solved to optimality. However, this time if the previous objective function value is within 10% of the new obtained optimal objective function value the candidate solution is the new current solution, and the incumbent solution is updated accordingly. This 10% may be interpreted as the suboptimality gap, the tolerance level. If the value does not fall in this interval, the objective function value of the candidate solution is updated as the new obtained objective function value, and the list of neighbor solutions are reranked.

4.2.3. Tabu Search Criteria

4.2.3.1. Initialization Rules. The selection of initial solution may be a crucial element in the performance of some heuristics. In order to dismiss possible negative effects arising from fixing the initial solution, we employed three methods for initialization, each having different approaches.

- *TC*: TC uses a greedy heuristic, proposed in [2], having the objective of maximizing total coverage. At each iteration of this heuristic, the sensor to be deployed is determined by a pair of sensor locations, $k \in N_s$, and sensor types, $r \in R$, that satisfies, $(k^*, r^*) = \text{argmax}_{k,r} \{\lambda_{kr}\}$, where λ_{kr} is defined as

$$\lambda_{kr} = \frac{\sum_{j \in N_i} a_{jkr} \max(0; \delta - cov_j)}{c_{kr}} \tag{4.20}$$

  In (4.20), cov_j denotes the coverage intensity realized at node $j \in N_i$ by the existing sensors, and node j is assumed to be covered if its coverage is greater than or equal to δ, for some precision level δ. Then, $\min(0; cov_j - \delta)$ can be interpreted as the undercoverage amount of node j, and the numerator of expression (4.20)

as the total improvement on undercoverage amount of uncovered sensors by deploying a type r sensor on point k. When divided by the cost of deployment, λ_{kr} becomes the decrease in the uncoverage of the network per unit cost. The iterations continue until the remaining budget is less than the cost of sensors.

- M: This initialization method is applicable when the set N_s of possible sensor locations exhibits a grid-like structure. The method places the sensors in the middle columns of the grid, starts from the center and continues placing the sensors up and down the column as long as the budget allows. The scheme is demonstrated in Figure 4.2, where the numbers associated with nodes denote the order of placement. The intuition behind the method is that the sensors towards the center of the sensing field provide a higher coverage than the ones close to the borders.

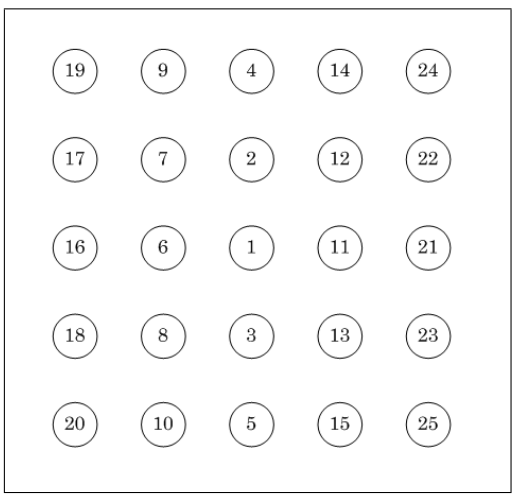


Figure 4.2. Sensor Placement Scheme of M.

Although this method is only applied for N_s having grid structure, it can also be applied for random sensor locations by preprocessing the data, computing the distance of each location from the center of gravity of the sensing field, and deploying the sensors in increasing order of the distances.

- R: It is observed in various areas of research that random search may yield higher performance than expected. Thus, one of the initialization methods experimented is random selection of sensor deployments satisfying budget constraints.

4.2.3.2. Neighborhood Selections. Generally there are three main operations for generating neighbors: *Add*, *Drop* and *Swap*. As the names imply, given a set of locations of deployed sensors, *Add* deploys a non-deployed sensor, *Drop* removes a deployed sensor, and *Swap* removes a deployed sensor and deploys a non-existing sensor. For the purposes of having a broader view of the neighborhood, *2-Add*, *2-Drop* and *2-Swap* operations are also suggested. They simply repeate the same procedure for a second sensor. Variations of *Swap* operation are possible depending on the input parameters, i.e. by swapping one expensive sensor with multiple cheaper sensors.

An observation for this problem is that neighboring solutions obtained by *Drop* operations will never be as promising as the ones obtained by the other operations, since having one less sensor will definitely cause less coverage, and will not be selected as the next current solution.

As a result of this, our heuristics start with a maximal set of sensor deployments, and if there is a single type of sensor, use only *Swap* and *2-Swap*, otherwise also use along with variations of *Swap* and *2-Swap*, *Add* and *2-Add* for available budget obtained by swapping an expensive sensor with a cheaper sensor.

4.2.3.3. Tabu Restrictions. In our heuristics, we apply two different types of tabu moves, and designate the heuristics as of *mode 1* and *mode 2*. In classical tabu search algorithms, every solution that is selected as the current solution has a certain move associated with it, determined by the difference between two consecutive solutions, similar to Hamming distance. Throughout the search, this move cannot be reversed for a certain number of tabu iterations. The reverse moves that are prohibited are classified as *tabu moves*.

The classical tabu approach is preserved in mode 2, and the duration of a move to remain in the tabu moves list is determined randomly, discrete uniform distribution in the interval $(3, 10)$.

In mode 1, on the other hand, we redefine the notion of a tabu move, discard the definition of a move as the difference between two consecutive solutions, and merely prohibit the repetition of a solution that has already been treated as the current solution before. Similar to the tabu moves list in mode 2, mode 1 holds a tabu solution list, a hash table as a matter of fact, where all solutions corresponding to a combination of sensor locations have an associated hash value that is stored in the list.

4.2.3.4. Aspiration. The aim of tabu restrictions is to prevent any solutions not complying with these restrictions from becoming the next current solution. However, if a solution obtained by a tabu move has an objective function value larger than the incumbent objective function value, this rule no longer applies and it is allowed to be selected as the next current solution.

4.2.3.5. Termination. The algorithm terminates when no improvement is attained on the incumbent objective function value for 20 iterations.

4.2.3.6. Avoiding Recalculation. We further propose another enhancement of our search algorithm, which does not deteriorate tabu search structure. For all visited solutions, the objective function of which has been computed, we store an entry in a dictionary, which keeps the hash value of the solution and the corresponding objective function value. Whenever the algorithm generates a random neighbor which has been solved for before, instead of resolving it, the objective function value is retrieved from the dictionary. This modification does not alter the search direction; and, despite the extra memory usage, saves significant computation time.

5. COMPUTATIONAL RESULTS

5.1. Experimental Setting

The methods have been tested on fabricated sample instances. This construction process requires the definition of sets and parameters in the problem. Prior to setting the parameters, the sets must be defined. Three main groups of sets in our model are described below:

- N_s: The selection of sensor points is described in detail in Section 5.1.2.
- R: The experiments are conducted for a single type of sensor.
- N_i, E: The details of the intruder network are provided in Section 5.1.1.

With these definition of sets, the parameter values are determined as follows:

- a_{jkr}: The intensity level is inversely proportional with the distance between two points. For this purpose we define a_{jkr} as:

$$a_{jkr} = \frac{1}{d(j,k)} \tag{5.1}$$

  where $d(j,k)$ is the Euclidean distance between node j of the intruder network, and sensor point k.
- c_r, $\bar{c_r}$: The cost of deploying and destroying a single sensor, c_r and $\bar{c_r}$, respectively, is taken as 1. Since there is a single type of sensor, it can be interpreted as if the budget constraints have been normalized, and the budget directly represents the number of sensors to be deployed and destroyed.
- b, $\bar{b}$: For each instance, the experiments are repeated for three levels of sensor deployment budget, and for each sensor deployment budget three levels for sensor destruction budget is selected. Sensor deployment budget, b, is chosen as 15%, 20% and 25% of the size of the sensor network, the total number of possible sensor

locations; and sensor destruction budget, $\bar{b}$, is computed as 20%, 30% and 40% of b.

5.1.1. Choice of Intruder Network

There are two critical elements in the definition of the intruder network: the number of nodes and the set of edges. To begin with, the sensing field spans the square area with vertices (1,1), (1,10), (10,10) and (10,1). The nodes of the intruder network are placed in this region and also share the square form, each row and column of nodes being equidistant.

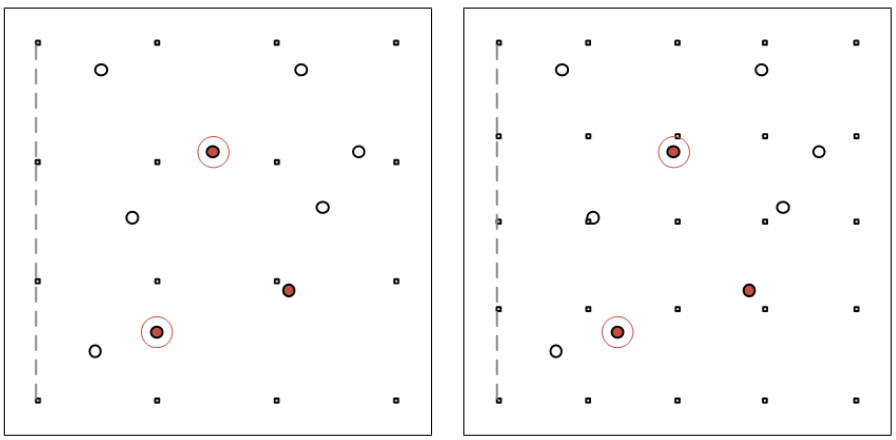
The movement of the follower in this network is a forward path from one side of the network to the other side. Without loss of generality, we assume the path starts from the bottom of the network, and ends at the top. The path may start from an arbitrary node from the bottommost row, and end at an arbitrary node from the topmost row. A simple shortest path problem is defined between two given vertices, a single starting point and a single ending point. To attain this structure, artificial nodes s and t are added below the lowest row and above the highest row respectively. The aim of the inclusion of these artificial nodes is to remove the diversity in the starting and ending points of the path, and be able to state that the follower searches for the shortest $s-t$ path. Node s is linked to all possible candidate nodes for starting the path, and node t is linked to all possible candidate nodes for ending the path. It is important to note that backward moves in the network are not allowed. Thus, the edge set E can be partitioned into three groups:

(i) Artificial arcs linking s and t to the original network, as mentioned previously.

(ii) $(j, j+1)$ and $(j+1, j)$ if nodes j and $j+1$ are in the same row. These arcs allow the change of direction in the path.

(iii) $(j, j+n)$, where n is the number of nodes in any row of the network. These arcs enable forward move within the network.

Another critical issue is the determination of the network size. It is clear that as the number of nodes increases, the discretization of the sensing field becomes more accurate, and the path more precise. The drawback, on the other hand, is that the number of variables also increases in the order $O(n^2)$, and the solution of the problem requires a longer time. In order to eliminate this tradeoff, we repeated the optimal solution procedure for different levels of $|N_i|$ values. We denote the size of a network by $n \times n$, where n is the number of nodes in a row or a column of the network. Below in Figures 5.1 and 5.2 are some sample results of optimal sensor locations and paths. In each figure, small squares are the nodes of the intruders network, circles are possible sensor locations, and the dashed line is the path yielded by the method. A filled circle indicates a sensor deployment in that location, and surrounded circles indicate a destroyed sensor.



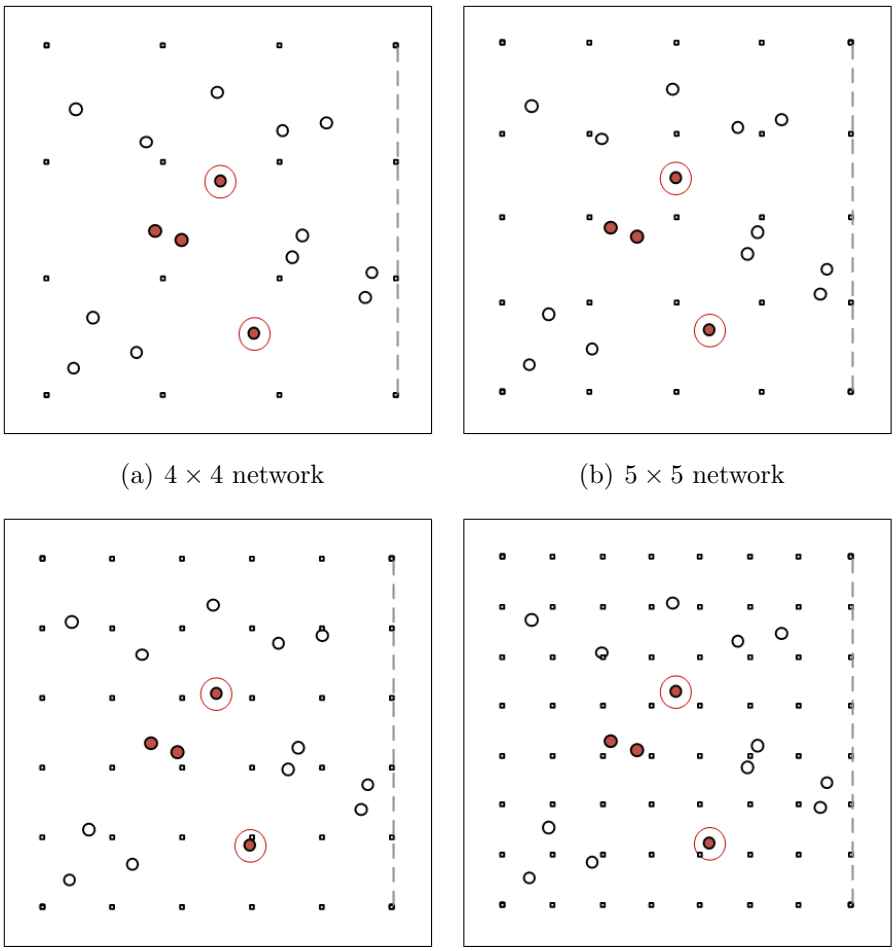
(a) 4×4 network

(b) 5×5 network

(c) 6×6 network

(d) 8×8 network

Figure 5.1. Optimal solutions of sample instances for 9 sensor locations.



(a) 4×4 network (b) 5×5 network

(c) 6×6 network (d) 8×8 network

Figure 5.2. Optimal solutions of sample instances for 16 sensor locations.

It has been observed that when the intruder network size is 6×6 and 8×8, the optimal path barely changes; whereas the computation time increases drastically. In the light of these results, our experiments are conducted for two levels of intruder network size: 4×4 and 5×5.

5.1.2. Choice of Sensor Locations

The sensing field is defined to be the square region with vertices $(1.5, 1.5)$, $(1.5, 9.5)$, $(9.5, 9.5)$ and $(9.5, 1.5)$, embedded in the intruders network; and the sensors are placed within this region.

The first type of sensor network is designed in grid structure. Given n^2 potential sensor locations, a grid of n rows and n columns is built, with equidistant rows and columns. For the second type of sensor networks, the sensor locations are determined randomly within the sensing field, preferably displaying a uniform distribution as much as possible.

The experiments are conducted on both very sparse and very dense networks. The number of sensors tested for both grid and random networks are 25, 36, 49, 81, 100, 144 and 225.

5.1.3. Technical Details

T-IP method requires an MIP solver for the optimal solution of the LLP. Some sample instances have been solved using Gurobi 5.0 and Cplex 12.2. The results indicated a higher performance for Gurobi, in terms of running time. Hence, the remaining instances utilize Gurobi 5.0 for solving the LLP to optimality.

A significant observation on the instances with larger intruder and sensor networks is the longer computation times. The purpose of a heuristic method is solving a problem in a reasonable amount of time with high accuracy. However, with the increase in the instance size, the running time falls out of this reasonable time range. In order to control extreme deviations on durations, we imposed a time upper bound in our algorithms. The iterations stop once the time limit of 5 hours is reached.

Finally, the algorithms are coded in C# environment of Microsoft Visual Studio 2005. The experiments are conducted on a workstation having Intel Xenon CPU X5460 3.16 GHz processor and 32 GB RAM, and working within Microsoft Windows Server 2003 Enterprise x86 Edition environment.

5.2. Result of Test Instances

5.2.1. Evaluating Performances

The primary criterion for evaluating the performance of a method uses the deviation of the solution from the optimal solution. Unfortunately, the exact solution procedure in hand, DeNegre's Method, is very inefficient, hence is not used. The only optimal solution method remaining is enumeration; however, it cannot be carried out for instances other than the smallest ones. This procedure basically enumerates all possible sensor deployment schemes that fully use the budget, and solve for the corresponding objective function value in the same way that T-IP does. In this manner, we could only obtain the optimal objective function values for the instances where $|N_s| = 25$. For the remaining instances, the objective function value given by a method is compared to the best value obtained in different experiments on the same instant, varying by methods (i.e. T-IP, T-T-S1 and T-T-S2) and method modifications (i.e. modes and initialization rules).

5.2.2. Numerical Results

The detailed results of the experiments are provided in the appendix in Tables A.1 through A.26. Below we present in Tables 5.1 through 5.4 the percent deviations of the solutions from the best known objective value, averaged over each N_s type. The best known value is the optimal objective value for only $|N_s| = 25$ cases, the remaining ones are the best objective function value obtained by all methods. It can be extracted from the results that the methods achieved to obtain the optimal result more frequently for grid N_s, and this is because of the additional initialization method M. As mentioned previously, M is only employed for grid N_s due to the nature of the algorithm.

Of all these three initialization methods, M, R and TC, M portrays the highest performance in general. Although the optimality is not guaranteed, the intuition behind this initialization method can be justified. The performance of R and TC, on the

other hand, vary depending on the sensor network type. In grid sensor networks, TC follows M in yielding paths with higher visibility. However in random sensor networks, in the absence of M, the performance of TC is dominated by R.

The comparison of solution modes is unfortunately inconclusive. It is true that changing the definition of illegal moves alters the search direction; this can be deducted from the fact that the obtained solutions differ. Nevertheless, this difference does not bring along any superiorities; and on the average the exposure on the path are very close for different methods.

Table 5.1. Average % deviations from the best result for grid N_s and 4×4 N_i.

$\|N_s\|$	T-IP	T-T-S1	T-T-S2
25	-5.405	-3.926	-3.496
36	-6.689	-4.221	-3.652
49	-12.741	-9.708	-8.980
81	-8.393	-8.978	-7.552
100	-7.064	-7.589	-8.843
144	-7.048	-8.730	-8.875
225	-6.156	-10.305	-9.973
Average	-7.745	-7.987	-7.716

Table 5.2. Average % deviations from the best result for grid N_s and 5×5 N_i.

$\|N_s\|$	T-IP	T-T-S1	T-T-S2
25	-2.079	-4.641	-4.681
36	-8.092	-3.879	-3.731
49	-7.976	-6.319	-5.648
81	-8.564	-8.665	-7.971
100	-7.519	-9.516	-7.801
144	-5.727	-8.617	-8.642
225	-5.938	-9.231	-9.690
Average	-6.796	-7.592	-7.172

As to the performance of the algorithms, notice that the values in these tables

Table 5.3. Average % deviations from the best result for random N_s and 4×4 N_i.

$\|N_s\|$	T-IP	T-T-S1	T-T-S2
25	-4.936	-2.383	-2.294
36	-4.332	-2.354	-2.137
49	-2.631	-0.393	-0.347
81	-2.745	-5.090	-3.511
100	-2.491	-5.515	-3.735
144	-3.451	-7.724	-7.417
225	-1.730	-6.092	-6.017
Average	-3.032	-4.488	-3.845

Table 5.4. Average % deviations from the best result for random N_s and 5×5 N_i.

$\|N_s\|$	T-IP	T-T-S1	T-T-S2
25	-4.589	-2.586	-2.419
36	-4.941	-3.343	-2.825
49	-1.959	-0.488	-0.652
81	-1.949	-4.723	-3.386
100	-2.583	-5.660	-3.846
144	-2.761	-6.598	-5.737
225	-1.513	-4.733	-5.011
Average	-2.722	-4.208	-3.552

show the percent deviation from the best value, which is the highest value. Thus, a high absolute deviation means a bad performance. Tables 5.1 through 5.4 indicate that T-IP method overperforms T-T-S methods in large N_s instances, however falls behind them in smaller instances. T-IP is designed to be a more accurate algorithm than T-T-S methods since it solves the LLP to optimality, unlike T-T-S methods which use approximations of the heuristics. Thus the higher performance in larger instances is not unexpected. On the other hand, the performance in smaller instances is disappointing. The reason for that can be argued as follows: during the neighborhood search, T-T-S methods have a higher probability of moving to poorer results, and this

direction of search leads the method to unsuccessful neighborhoods in larger instances; but in smaller instances, it is yet another mechanism to prevent being stuck in a local minimum, and has a higher probability of jumping to a better solution region since the solution space is smaller.

Another important point is that the variance of the results. The deviations are in general higher when the sensor network has grid structure. This is a reasonable result, considering the closeness of some sensors in the random network. In grid network, the neighbors of a sensor are equidistant, and when two solutions differ, for instance, in only one sensor location, the gap between the objective function values is much higher compared to the case in random network, where we might have very close sensors, and the objective function values change only slightly under the same scenario.

Table 5.5. Average CPU requirements (min) for 4×4 N_i.

$\|N_s\|$	T-IP	T-T-S1	T-T-S2
25	2.1	0.3	0.3
36	4.0	0.7	0.7
49	6.0	2.8	5.7
81	14.1	18.3	39.2
100	20.6	30.0	41.9
144	39.3	31.0	41.2
225	100.8	34.8	38.1

The amount of increase in the computation times is noteworthy, as can be seen in Tables 5.5 and 5.6. The T-IP algorithm hits the five hour time limit for $|N_s| = 225$ instances when using 5×5 N_i, however can easily solve 4×4 N_i instances. This should give an insight on the solvability of 6×6 and 8×8 N_i. 6×6 N_i terminates due to time limitation even in $|N_s| = 100$ instances.

A noteworthy remark on the paths is that, in general, the solutions provided by our heuristics have their paths leaning on one sides of the network. This generalization has exceptions for very sparse sensor deployments, i.e. $|N_s| = 9$ or 16, where it is a

Table 5.6. Average CPU requirements (min) for 5×5 N_i.

$\lvert N_s \rvert$	T-IP	T-T-S1	T-T-S2
25	8.0	0.6	0.5
36	12.1	1.4	1.1
49	21.3	4.7	5.2
81	62.0	23.6	44.2
100	87.9	35.9	59.0
144	188.4	49.0	56.4
225	278.6	67.2	50.5

better decision for the defender to place the sensors on the sides and force the intruder to take an path from the middle of the network. However, as the middle region gets denser in terms of sensor deployments, the follower tends to spend his budget cleaning one of the sides and move along that side. This is intuitively a reasonable (though not necessarily optimal) choice, when we make the analogy of a burglar in a museum who makes his movements along the walls.

5.3. Dealing with Border Effect

We name the selection of paths on the side of the intruder's network as the *border effect*. Despite the previous justification of the border effect when the WSN is considered to be a closed field with boundaries (i.e. insides of a building), it fails to explain an important application area of WSN which is border surveillance. The form of a border is a long thin line, and the problem is solved for only a section of it. The solution obtained for the investigated section is repeated over the remaining region, which diminishes the notion of left and right boundaries. With the repetition assumption, the border effect can be overcome by a simple modification of intensity parameters.

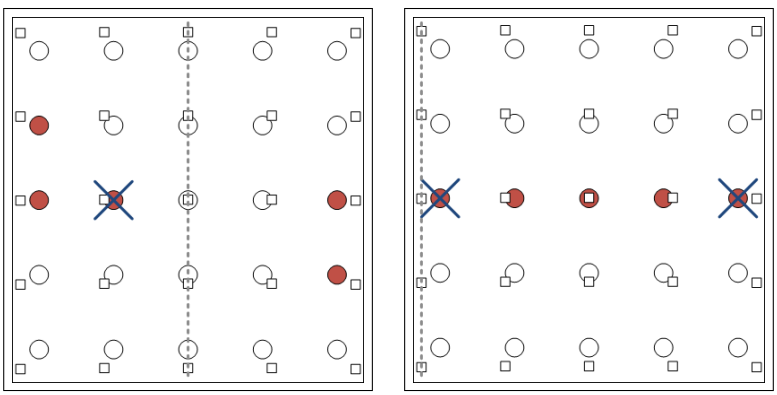
We start by creating two artificial sets of sensor network, placed on both sides of the original sensor network. Denote these networks as N_s^L and N_s^R. For each sensor

point $k \in N_s$, there is a copy $k^L \in N_s^L$ and $k^R \in N_s^R$. The location of sensors in the artificial networks can be calculated by only changing the x-coordinates of the original sensor points. The width of the sensing field is subtracted from the original x-coordinates of k to obtain the location of k^L in the left artificial network, and it is added to the original x-coordinates of k to obtain the sensor location of k^R in the right artificial network.

The key point here is that the deployment of a sensor in point k also implies the deployment of sensors in points k^L and k^R. Therefore, the effect of deploying a sensor in point k should also take into account the sensors in points k^L and k^R. This effect is represented in the problem by the intensity parameters a, so the following modification in the definition of the parameter is sufficient to deal with the border effect.

$$a_{jkr} = \frac{1}{d(j,k)} + \frac{1}{d(j,k^L)} + \frac{1}{d(j,k^R)} \tag{5.2}$$

The following in Figure 5.3 represents the change in optimal solutions with the new definition of intensity parameters. Figure 5.3a is the optimal solution for the parameter described in Section 5.1, and Figure 5.3b is the optimal solution when the border effect is discarded by (5.2).



(a) With border effect (b) Without border effect

Figure 5.3. Optimal solutions of sample instances with and without border effect.

6. CONCLUSION

In this thesis, we develop a bilevel mixed integer linear programming formulation to model a coverage problem in WSN. This model structure does not have an efficient suggested solution method in the literature; hence we propose three heuristics for the solution of this problem. The methods are tested on two different intruder networks, and 14 different sensor networks in total; and the tests are repeated for several budget values. Unfortunately, there is no benchmark value to compare the results of our methods with; therefore our evaluations are within the range of our heuristics.

With the problem formulation we have in hand, the future work focuses on solving it. There are various directions that can be followed in this pursuit: improving what we have in hand, and finding other methods.

One thing we have but could not successfully implement is DeNegres method. Even only for branching purposes, it wouldnt be a waste of time dwelling on that method. One of the biggest obstacles is chopping down the feasible region to obtain a candidate integer extreme point; and the other is forcing this candidate solution to satisfy bilevel feasibility constraints. We already proposed three additional sets of constraints to overcome the second obstacle, and as for the first one IP literature has various cutting plane suggestions.

Lets not forget that in order to satisfy real world expectations in WSN, our methods should be able to solve large instances. Although the effect of accuracy in the search increases with larger sensor networks, as shown in T-IP, the methods require larger solution times.

Our heuristics all enable TS when dealing with combinatorial problems. We made that preference based on the performance of TS in general; however, as future work, other heuristics can also be experimented to solve the ULP.

Finally, although this research tackles the issue in WSN perspective, it can also be generalized to the solution of MILBPs with pure integer ULPs and mixed integer LLPs. This generalization may allow the experimentation of our methods on different data sets having different structures.

APPENDIX A: Experiment Results

A.1. Grid Sensor Network Result

Table A.1. T-IP experiment results for grid N_s of size 25 – 49 and 4×4 N_i.

				Mode 1			Mode 2		
$\|N_s\|$	b	$\bar{b}$	Optimal Value	M	R	TC	M	R	TC
25	3	1	1.434	1.434	1.270	1.277	1.434	1.434	1.258
25	5	1	3.337	3.337	3.278	3.296	3.296	2.783	3.315
25	5	2	2.037	2.037	1.975	1.885	2.037	1.975	1.830
25	6	1	4.561	4.423	4.228	4.561	4.228	4.228	4.228
25	6	2	2.918	2.635	2.821	2.821	2.635	2.578	2.557
36	5	1	–	3.408	3.235	3.369	2.766	3.206	3.369
36	5	2	–	1.925	1.925	1.866	1.925	1.925	1.925
36	7	1	–	4.196	5.176	5.073	4.206	5.196	5.073
36	7	2	–	3.355	3.742	3.879	3.898	3.879	3.753
36	9	1	–	5.636	6.504	7.269	5.636	7.185	7.306
36	9	2	–	4.795	4.795	5.470	4.795	4.841	5.675
36	9	3	–	3.954	4.352	3.787	3.954	3.789	3.841
49	7	1	–	4.805	4.397	4.225	4.805	4.526	4.225
49	7	2	–	4.069	3.708	3.300	4.069	3.414	3.361
49	10	2	–	5.380	6.395	5.341	5.380	6.344	5.499
49	10	3	–	4.720	4.523	4.416	4.651	4.609	4.574
49	10	4	–	3.993	3.831	3.686	3.923	3.881	3.559
49	12	2	–	6.689	6.966	6.896	6.816	6.860	7.173
49	12	3	–	6.135	6.022	5.962	5.959	6.052	6.029
49	12	4	–	5.145	4.710	4.943	5.226	5.035	5.104
Average				4.104	4.193	4.166	4.082	4.187	4.183

Table A.2. T-IP experiment results for grid N_s of size 81 – 100 and 4×4 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
81	12	2	6.846	8.013	7.700	6.891	6.949	7.700
81	12	3	6.110	5.983	5.943	6.155	5.808	6.030
81	12	4	5.379	5.090	5.230	5.424	4.754	5.177
81	16	3	8.804	8.929	9.636	8.961	8.905	9.631
81	16	4	8.015	7.933	7.875	7.977	7.836	7.882
81	16	6	6.529	6.500	6.061	6.575	6.073	6.268
81	20	4	10.784	11.140	11.395	10.846	11.866	11.395
81	20	6	9.282	8.904	9.041	9.292	8.941	8.337
81	20	8	7.820	7.425	7.321	7.829	7.183	6.595
100	15	3	8.275	7.958	8.700	8.275	8.166	8.757
100	15	4	7.491	6.952	6.995	7.491	6.896	7.053
100	15	6	5.941	5.598	5.324	5.977	5.569	5.773
100	20	4	11.016	11.486	12.435	11.016	11.088	12.197
100	20	6	9.483	9.101	8.666	9.483	8.465	8.765
100	20	8	8.000	7.752	6.817	8.000	7.668	7.032
100	25	5	13.603	13.688	14.490	13.587	13.686	15.592
100	25	7	12.053	11.327	12.144	12.037	11.579	11.329
100	25	10	9.803	9.058	8.908	9.787	9.035	8.471
Average			8.624	8.491	8.593	8.645	8.359	8.555

Table A.3. T-IP experiment results for grid N_s of size 144 – 225 and 4×4 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
144	21	4	11.871	11.733	12.878	11.871	11.239	13.184
144	21	6	10.341	9.414	9.644	10.341	9.093	9.724
144	21	8	8.836	7.966	7.417	8.836	8.005	7.346
144	28	5	15.898	15.840	16.836	15.867	16.436	16.836
144	28	8	13.592	12.634	12.399	13.551	12.927	12.361
144	28	11	11.345	10.304	9.456	11.270	10.611	9.647
144	36	7	19.651	20.455	21.951	19.847	19.942	20.549
144	36	10	17.301	16.822	16.583	17.358	15.992	16.797
144	36	14	14.303	12.937	12.533	14.360	13.296	12.937
225	33	6	18.567	18.382	22.109	18.737	19.918	19.767
225	33	9	16.352	15.082	15.358	16.387	15.224	15.499
225	33	13	13.361	11.719	11.701	13.405	12.490	12.012
225	45	9	24.794	24.255	26.572	24.794	23.413	26.075
225	45	13	21.771	20.200	20.710	21.771	19.885	20.206
225	45	18	18.106	16.663	16.259	18.106	16.571	15.848
225	56	11	30.747	30.379	32.399	30.844	30.411	32.009
225	56	16	26.821	25.312	25.668	26.862	24.422	25.644
225	56	22	22.335	20.224	19.464	22.314	20.239	19.706
Average			17.555	16.684	17.219	17.584	16.673	17.008

Table A.4. T-T-S1 experiment results for grid N_s of size 25 – 81 and 4×4 N_i.

				Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	M	R	TC	M	R	TC
25	3	1	1.434	1.434	1.434	1.434	1.434	1.434	1.277
25	5	1	3.337	3.289	3.289	3.337	3.254	3.296	3.296
25	5	2	2.037	2.037	1.942	1.942	2.037	1.918	1.942
25	6	1	4.561	4.130	4.205	4.504	4.130	4.130	4.107
25	6	2	2.918	2.762	2.578	2.777	2.821	2.821	2.821
36	5	1	–	3.268	3.369	3.369	3.350	3.279	3.287
36	5	2	–	1.925	1.866	1.925	1.925	1.925	1.905
36	7	1	–	5.187	4.943	5.058	5.091	4.848	5.033
36	7	2	–	3.762	3.768	3.607	3.779	3.861	3.822
36	9	1	–	6.900	6.558	7.146	6.772	7.020	7.187
36	9	2	–	5.442	5.527	5.512	5.605	5.627	5.512
36	9	3	–	3.954	3.711	3.794	3.899	3.889	3.707
49	7	1	–	6.187	4.569	5.513	6.545	5.121	4.825
49	7	2	–	4.123	3.703	3.701	4.597	3.618	3.335
49	10	2	–	5.351	6.256	6.551	5.351	6.359	5.311
49	10	3	–	4.616	4.474	4.387	4.616	4.524	4.387
49	10	4	–	3.888	3.548	3.559	3.888	3.773	3.559
49	12	2	–	7.500	7.361	8.310	6.916	7.473	8.078
49	12	3	–	5.959	6.178	5.748	5.921	6.289	5.748
49	12	4	–	5.232	4.927	5.014	5.186	4.858	4.947
81	12	2	–	6.894	7.419	8.376	6.875	7.452	8.213
81	12	3	–	6.101	6.449	6.202	6.101	5.952	5.908
81	12	4	–	5.369	4.952	5.045	5.369	5.171	5.061
81	16	3	–	8.766	10.103	9.611	8.837	9.255	9.611
81	16	4	–	7.986	7.620	7.828	7.986	7.751	8.159
81	16	6	–	6.519	5.322	5.682	6.496	5.322	5.682
81	20	4	–	10.812	11.336	11.826	10.746	11.530	11.894
81	20	6	–	9.138	7.835	8.959	9.152	9.168	8.757
81	20	8	–	7.602	7.409	6.595	7.643	7.082	6.595
Average				5.384	5.264	5.425	5.390	5.336	5.309

Table A.5. T-T-S1 experiment results for grid N_s of size 100 – 225 and 4×4 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
100	15	3	8.318	9.582	9.288	8.302	7.968	9.146
100	15	4	7.534	7.229	7.024	7.518	7.314	7.188
100	15	6	5.901	5.768	5.568	5.901	5.285	5.731
100	20	4	11.016	11.800	11.366	11.016	12.574	11.366
100	20	6	9.483	8.286	8.739	9.483	8.286	9.207
100	20	8	8.000	6.588	6.881	8.000	6.588	7.392
100	25	5	13.616	14.208	14.192	13.604	13.687	14.399
100	25	7	12.049	10.096	12.022	12.014	11.764	11.541
100	25	10	9.759	7.820	9.128	9.787	7.820	8.525
144	21	4	11.814	12.030	13.499	11.794	13.046	13.989
144	21	6	10.253	8.391	8.918	10.253	9.722	8.918
144	21	8	8.740	6.875	6.989	8.740	6.875	7.432
144	28	5	15.845	17.281	16.836	15.823	17.181	16.836
144	28	8	13.468	12.509	13.111	13.491	12.605	12.295
144	28	11	11.268	8.812	9.352	11.268	10.537	9.352
144	36	7	19.548	19.105	20.542	19.524	19.105	20.542
144	36	10	17.343	15.889	16.092	17.252	15.889	16.092
144	36	14	13.841	12.344	12.121	13.841	12.344	12.121
225	33	6	18.414	16.759	19.459	18.491	16.759	19.459
225	33	9	16.191	13.757	14.599	16.305	13.757	14.599
225	33	13	13.287	10.493	10.823	13.333	10.493	10.823
225	45	9	24.794	23.396	25.049	24.794	23.396	25.049
225	45	13	21.771	19.285	19.334	21.771	19.285	19.334
225	45	18	18.106	14.992	14.425	18.106	14.992	14.425
225	56	11	30.636	28.300	31.851	30.592	28.300	31.851
225	56	16	26.727	24.629	24.754	26.771	23.529	24.754
225	56	22	22.311	18.640	18.712	22.255	18.640	18.712
Average			14.816	13.514	14.099	14.816	13.620	14.114

Table A.6. T-T-S2 experiment results for grid N_s of size 25 – 81 and 4×4 N_i.

				Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	M	R	TC	M	R	TC
25	3	1	1.434	1.434	1.434	1.434	1.434	1.434	1.277
25	5	1	3.337	3.289	3.337	3.337	3.254	3.296	3.296
25	5	2	2.037	2.037	1.975	1.942	2.037	1.918	1.942
25	6	1	4.561	4.130	4.205	4.504	4.130	4.130	4.107
25	6	2	2.918	2.762	2.821	2.821	2.821	2.821	2.821
36	5	1	–	3.268	3.231	3.369	3.350	3.279	3.339
36	5	2	–	1.925	1.925	1.925	1.925	1.886	1.905
36	7	1	–	5.117	5.064	5.058	5.091	5.188	5.033
36	7	2	–	3.762	3.629	3.779	3.602	3.861	3.822
36	9	1	–	6.900	6.558	7.146	6.772	7.020	7.187
36	9	2	–	5.442	5.527	5.512	5.605	5.627	5.512
36	9	3	–	3.954	3.889	4.171	3.916	4.346	3.615
49	7	1	–	6.187	4.569	5.154	6.545	5.121	5.326
49	7	2	–	4.123	3.703	3.628	4.597	3.614	3.695
49	10	2	–	5.351	6.256	6.551	5.351	6.382	5.311
49	10	3	–	4.616	4.602	4.387	4.616	4.539	4.426
49	10	4	–	3.888	3.632	3.604	3.888	3.711	3.649
49	12	2	–	6.816	7.361	8.310	6.916	7.473	8.078
49	12	3	–	5.959	6.607	5.901	5.921	6.063	6.117
49	12	4	–	5.233	5.175	5.199	5.186	5.252	4.947
81	12	2	–	6.832	7.419	7.700	6.883	7.452	7.700
81	12	3	–	6.101	6.024	6.526	6.101	5.766	6.790
81	12	4	–	5.461	5.047	5.099	5.369	5.060	4.981
81	16	3	–	10.280	9.500	10.443	10.656	9.652	10.260
81	16	4	–	7.944	8.957	8.699	7.953	8.538	8.206
81	16	6	–	6.496	5.322	5.682	6.496	5.322	5.682
81	20	4	–	12.129	12.160	11.395	11.984	11.562	11.395
81	20	6	–	8.988	7.835	8.841	9.240	8.499	9.035
81	20	8	–	7.658	6.772	6.595	7.689	6.728	6.848
Average				5.451	5.329	5.473	5.494	5.363	5.390

Table A.7. T-T-S2 experiment results for grid N_s of size 100 – 225 and 4×4 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
100	15	3	8.318	9.136	7.497	8.302	8.726	9.756
100	15	4	7.534	7.625	7.464	7.518	7.441	7.600
100	15	6	5.967	5.241	4.945	5.967	5.172	5.073
100	20	4	11.016	12.270	11.366	11.016	12.033	11.820
100	20	6	9.483	8.286	8.378	9.483	8.286	8.378
100	20	8	8.000	6.588	6.636	8.000	6.588	6.529
100	25	5	14.208	14.384	13.766	13.499	14.488	13.766
100	25	7	12.032	10.096	11.968	12.015	11.289	11.641
100	25	10	9.573	7.820	7.923	9.573	7.820	7.923
144	21	4	11.794	12.931	14.005	11.794	13.478	12.732
144	21	6	10.233	8.391	9.048	10.233	9.671	9.048
144	21	8	8.736	6.875	7.471	8.736	6.875	7.443
144	28	5	15.746	16.616	16.934	15.746	16.916	16.934
144	28	8	13.468	12.332	12.181	13.468	12.442	12.661
144	28	11	11.195	8.812	9.589	11.195	9.550	9.337
144	36	7	19.337	21.140	20.542	19.268	19.104	20.542
144	36	10	16.625	15.889	16.092	16.725	15.889	16.092
144	36	14	13.841	12.344	12.121	14.008	12.344	12.121
225	33	6	18.311	16.759	20.188	18.311	16.759	20.856
225	33	9	16.113	13.757	14.721	16.113	13.757	15.423
225	33	13	13.227	10.493	10.834	13.227	10.493	11.212
225	45	9	24.794	23.396	25.049	24.794	23.396	25.049
225	45	13	21.771	19.285	19.334	21.771	19.420	19.334
225	45	18	18.106	14.992	14.425	18.106	14.992	14.425
225	56	11	30.255	28.300	31.851	30.255	30.786	31.851
225	56	16	26.496	24.888	24.754	26.496	23.529	24.754
225	56	22	22.124	18.640	18.712	22.124	18.640	18.712
Average			14.752	13.603	13.992	14.731	13.699	14.111

Table A.8. T–IP experiment results for grid N_s of size 25 – 81 and 5×5 N_i.

				Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	M	R	TC	M	R	TC
25	3	1	1.815	1.815	1.605	1.616	1.815	1.735	1.632
25	5	1	4.011	4.011	4.011	4.011	4.011	4.011	3.858
25	5	2	2.563	2.563	2.512	2.512	2.563	2.563	2.563
25	6	1	5.956	5.956	5.581	5.939	5.956	5.956	5.956
25	6	2	3.765	3.581	3.765	3.765	3.765	3.765	3.533
36	5	1	–	4.013	3.974	4.138	4.013	4.068	4.080
36	5	2	–	2.433	2.433	2.433	2.433	2.433	2.284
36	7	1	–	5.313	6.543	6.500	6.523	6.593	6.051
36	7	2	–	4.243	4.608	4.282	4.114	4.148	4.200
36	9	1	–	7.133	9.306	9.074	7.133	9.451	8.666
36	9	2	–	6.063	6.046	6.314	6.063	6.138	6.190
36	9	3	–	4.993	4.993	4.854	4.993	4.940	4.785
49	7	1	–	5.273	5.586	6.289	5.273	6.689	6.687
49	7	2	–	4.351	4.872	4.227	4.351	4.236	4.804
49	10	2	–	6.790	8.068	7.666	6.790	8.102	6.987
49	10	3	–	5.962	5.719	5.546	5.865	5.657	5.758
49	10	4	–	5.040	4.811	4.624	4.943	4.836	4.870
49	12	2	–	8.446	9.150	9.935	8.615	8.645	9.993
49	12	3	–	7.649	7.454	8.593	7.520	7.753	7.410
49	12	4	–	6.684	6.193	6.507	6.598	6.562	6.439
81	12	2	–	8.644	9.245	9.152	8.707	9.374	9.014
81	12	3	–	7.710	7.822	7.781	7.773	7.784	7.674
81	12	4	–	6.783	6.442	6.441	6.845	6.522	6.385
81	16	3	–	11.124	11.271	11.342	11.342	11.220	11.377
81	16	4	–	10.127	10.247	10.040	10.068	10.551	10.006
81	16	6	–	8.233	7.319	7.908	8.298	7.659	8.094
81	20	4	–	13.768	15.080	14.082	13.722	14.390	14.037
81	20	6	–	11.755	11.042	11.036	11.588	11.270	11.341
81	20	8	–	9.893	9.258	9.364	9.895	9.592	9.026
Average				6.564	6.723	6.758	6.606	6.781	6.679

Table A.9. T–IP experiment results for grid N_s of size 100 – 225 and 5×5 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
100	15	3	10.460	10.274	10.758	10.460	10.590	10.655
100	15	4	9.466	8.963	8.995	9.466	8.880	9.217
100	15	6	7.500	7.066	6.448	7.500	7.228	6.824
100	20	4	13.898	14.051	14.261	13.898	13.790	14.214
100	20	6	11.956	11.378	11.420	11.956	11.081	11.211
100	20	8	10.087	8.803	8.661	10.087	9.383	8.687
100	25	5	17.171	17.836	17.680	17.149	17.615	17.928
100	25	7	15.205	14.283	14.857	15.183	14.274	14.488
100	25	10	12.365	11.278	11.453	12.343	11.210	11.785
144	21	4	14.996	14.950	16.618	14.996	14.729	15.894
144	21	6	13.057	12.098	12.431	13.057	11.423	12.654
144	21	8	11.157	10.247	9.396	11.157	10.047	9.497
144	28	5	20.032	20.148	21.008	20.024	20.595	22.082
144	28	8	17.092	16.513	16.211	17.085	16.128	16.265
144	28	11	14.328	12.135	13.670	14.258	13.569	13.530
144	36	7	24.816	24.353	26.771	25.082	24.491	25.723
144	36	10	21.830	20.926	20.769	21.908	20.468	21.334
144	36	14	18.040	16.495	17.636	18.106	17.506	17.534
225	33	6	23.468	23.049	25.882	23.693	23.428	25.830
225	33	9	20.651	19.031	19.593	20.841	19.429	19.931
225	33	13	16.916	15.253	14.222	16.916	16.203	14.184
225	45	9	31.296	29.638	32.736	31.296	30.162	32.520
225	45	13	27.483	25.316	26.078	27.483	25.152	25.458
225	45	18	22.835	20.938	19.670	22.835	20.159	19.794
225	56	11	38.831	37.294	41.510	38.966	37.305	40.501
225	56	16	33.862	31.469	32.009	33.926	30.741	32.532
225	56	22	28.204	25.722	24.404	28.165	25.968	24.167
Average			18.778	17.760	18.339	18.809	17.835	18.313

Table A.10. T–T–S1 experiment results for grid N_s of size 25 – 81 and 5×5 N_i.

				Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	M	R	TC	M	R	TC
25	3	1	1.815	1.815	1.815	1.735	1.815	1.735	1.632
25	5	1	4.011	3.980	3.980	4.006	4.006	4.011	4.011
25	5	2	2.563	2.563	2.293	2.433	2.563	2.417	2.433
25	6	1	5.956	5.956	5.443	5.505	5.441	5.380	5.533
25	6	2	3.765	3.516	3.516	3.533	3.340	3.581	3.194
36	5	1	–	4.068	4.068	4.034	3.974	4.051	4.125
36	5	2	–	2.433	2.352	2.433	2.433	2.433	2.314
36	7	1	–	6.507	6.363	6.684	6.511	6.558	6.684
36	7	2	–	4.633	4.869	4.586	4.740	4.368	4.740
36	9	1	–	8.924	9.231	9.286	9.412	9.198	9.227
36	9	2	–	7.038	6.998	6.981	6.946	7.023	6.256
36	9	3	–	4.993	4.733	4.739	4.942	4.915	4.888
49	7	1	–	6.523	6.552	6.523	6.273	6.365	6.520
49	7	2	–	4.763	4.872	4.653	4.576	4.690	4.349
49	10	2	–	6.756	7.928	8.269	6.756	7.949	8.047
49	10	3	–	5.822	5.673	5.761	5.822	5.709	5.697
49	10	4	–	4.900	4.469	4.839	4.900	4.942	4.775
49	12	2	–	8.615	9.372	10.551	8.748	10.496	10.461
49	12	3	–	7.520	7.995	7.210	7.473	7.992	6.592
49	12	4	–	6.598	6.221	6.397	6.539	6.235	6.325
81	12	2	–	8.694	10.554	8.956	8.665	10.577	10.652
81	12	3	–	7.824	8.157	7.617	7.707	7.495	7.624
81	12	4	–	6.897	6.468	5.991	6.780	6.470	5.991
81	16	3	–	11.076	13.088	11.342	11.170	11.515	13.410
81	16	4	–	10.081	9.588	9.958	10.158	9.772	10.190
81	16	6	–	8.221	6.688	7.408	8.188	6.688	8.191
81	20	4	–	13.657	16.462	14.529	13.561	14.669	14.762
81	20	6	–	11.536	9.850	10.990	11.545	11.578	11.277
81	20	8	–	9.580	9.344	8.300	9.641	8.916	8.713
Average				6.741	6.860	6.733	6.711	6.818	6.849

Table A.11. T–T–S1 experiment results for grid N_s of size 100 – 225 and 5×5 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
100	15	3	10.510	12.258	11.398	10.496	10.135	10.655
100	15	4	9.515	9.726	9.305	9.502	8.751	9.198
100	15	6	7.444	7.103	6.134	7.444	6.651	6.134
100	20	4	13.898	14.996	13.949	13.898	13.017	13.949
100	20	6	11.956	10.416	11.088	11.956	10.416	11.088
100	20	8	10.087	8.278	8.426	10.087	8.278	8.426
100	25	5	17.198	15.051	16.942	17.195	17.343	16.942
100	25	7	15.209	14.457	14.081	15.194	14.886	14.081
100	25	10	12.305	9.832	10.160	12.343	9.832	10.160
144	21	4	14.929	15.208	15.238	14.907	15.229	16.830
144	21	6	12.952	10.539	12.271	12.952	11.745	12.142
144	21	8	11.037	8.633	9.264	11.037	8.633	9.264
144	28	5	19.884	21.097	21.587	19.962	21.839	20.753
144	28	8	16.999	15.505	15.944	17.032	16.056	16.070
144	28	11	14.223	11.069	11.698	14.223	13.295	11.698
144	36	7	24.772	24.064	25.349	24.650	24.064	25.349
144	36	10	21.874	19.993	20.666	21.762	19.993	20.666
144	36	14	17.455	15.519	15.273	18.048	15.519	15.273
225	33	6	23.260	21.061	24.876	23.354	21.061	24.876
225	33	9	20.441	17.278	19.386	20.609	17.278	19.386
225	33	13	16.759	15.175	13.777	16.823	13.177	13.777
225	45	9	31.296	29.460	31.450	31.296	29.460	31.450
225	45	13	27.483	24.252	24.748	27.483	24.252	24.748
225	45	18	22.835	18.846	18.224	22.835	18.846	18.224
225	56	11	38.686	35.636	40.851	38.625	35.636	40.851
225	56	16	33.732	31.014	31.219	33.798	29.617	31.219
225	56	22	28.142	23.440	22.832	28.083	23.440	22.832
Average			18.699	17.034	17.635	18.726	16.980	17.631

Table A.12. T–T–S2 experiment results for grid N_s of size 25 – 81 and 5×5 N_i.

				Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	M	R	TC	M	R	TC
25	3	1	1.815	1.815	1.815	1.616	1.815	1.735	1.616
25	5	1	4.011	3.980	3.919	4.011	4.006	4.011	4.011
25	5	2	2.563	2.563	2.496	2.452	2.563	2.417	2.452
25	6	1	5.956	5.956	5.443	5.380	5.440	5.380	5.956
25	6	2	3.765	3.516	3.263	3.533	3.340	3.581	3.194
36	5	1	–	4.068	4.068	4.138	4.013	4.068	4.125
36	5	2	–	2.433	2.433	2.433	2.433	2.376	2.314
36	7	1	–	6.507	6.684	6.416	6.701	6.343	6.430
36	7	2	–	4.254	4.620	4.954	4.830	4.619	4.211
36	9	1	–	8.924	9.231	9.286	9.412	9.177	9.227
36	9	2	–	7.037	6.998	6.981	6.946	6.777	6.695
36	9	3	–	4.993	4.732	5.449	4.942	4.820	4.888
49	7	1	–	6.523	6.552	6.523	6.728	6.365	6.520
49	7	2	–	4.763	4.872	4.653	4.576	4.657	4.805
49	10	2	–	6.756	7.954	7.032	6.756	8.099	8.128
49	10	3	–	5.822	5.673	5.929	5.822	5.721	5.542
49	10	4	–	4.900	4.469	5.007	5.015	4.583	4.727
49	12	2	–	8.615	9.372	10.551	8.748	10.516	10.461
49	12	3	–	7.520	7.758	7.929	7.473	8.116	8.617
49	12	4	–	6.598	6.341	6.226	6.539	6.370	6.235
81	12	2	–	8.634	10.554	9.075	8.702	10.577	8.955
81	12	3	–	7.707	7.762	7.705	7.707	7.592	7.984
81	12	4	–	6.779	6.745	6.183	6.779	6.437	6.380
81	16	3	–	11.177	13.225	13.096	13.680	12.994	13.539
81	16	4	–	10.022	11.139	10.320	10.022	10.642	9.393
81	16	6	–	8.221	6.688	7.458	8.188	6.688	7.442
81	20	4	–	14.305	14.695	15.495	13.511	16.128	15.709
81	20	6	–	11.434	9.850	10.810	11.414	10.492	11.210
81	20	8	–	9.580	8.499	8.726	9.580	8.475	8.597
Average				6.738	6.822	6.875	6.817	6.888	6.874

Table A.13. T–T–S2 experiment results for grid N_s of size 100 – 225 and 5×5 N_i.

			Mode 1			Mode 2		
$\lvert N_s \rvert$	b	$\bar{b}$	M	R	TC	M	R	TC
100	15	3	10.510	11.932	11.130	10.496	11.278	11.557
100	15	4	9.515	9.069	9.130	9.502	9.364	9.625
100	15	6	7.444	6.925	6.675	7.444	6.702	6.621
100	20	4	13.898	15.945	13.949	13.898	15.139	15.868
100	20	6	11.956	11.010	11.088	11.956	10.416	11.088
100	20	8	10.087	8.278	8.426	10.087	8.711	8.426
100	25	5	17.198	18.583	16.718	17.039	19.571	16.718
100	25	7	15.209	14.606	13.857	15.051	14.232	13.857
100	25	10	12.132	10.665	9.937	12.210	10.605	9.937
144	21	4	14.929	17.419	16.838	14.858	17.360	16.783
144	21	6	12.969	10.539	12.142	12.919	12.045	12.250
144	21	8	11.028	8.633	9.264	11.028	8.633	9.264
144	28	5	19.884	21.892	20.627	19.954	21.808	20.627
144	28	8	16.998	15.517	15.944	16.838	15.408	15.944
144	28	11	14.129	11.069	11.698	14.129	11.773	11.698
144	36	7	24.240	24.064	24.991	23.902	24.064	24.991
144	36	10	21.114	19.993	20.386	21.095	19.993	20.386
144	36	14	17.455	15.519	15.357	17.665	15.519	15.357
225	33	6	23.124	21.061	24.876	23.124	21.061	24.876
225	33	9	20.335	17.278	19.386	20.335	17.278	19.386
225	33	13	16.682	13.177	13.777	16.682	13.177	13.777
225	45	9	31.296	30.321	31.449	31.296	29.460	31.449
225	45	13	27.483	24.252	24.748	27.483	24.252	24.748
225	45	18	22.835	18.846	18.224	22.835	18.846	18.224
225	56	11	38.194	35.636	39.540	38.194	39.762	39.540
225	56	16	33.447	29.617	30.624	33.447	29.617	30.624
225	56	22	27.904	23.440	22.617	27.904	23.440	22.617
Average			18.592	17.233	17.533	18.569	17.389	17.639

A.2. Random Sensor Network Result

Table A.14. T-IP experiment results for random N_s of size 25 – 49 and 4×4 N_i.

				Mode 1		Mode 2	
$\|N_s\|$	b	$\bar{b}$	Optimal Value	R	TC	R	TC
25	3	1	1.337	1.337	1.337	1.335	1.335
25	5	1	3.122	2.833	3.060	2.810	2.726
25	5	2	1.967	1.900	1.810	1.967	1.810
25	6	1	4.129	3.650	3.840	3.483	3.893
25	6	2	2.709	2.696	2.709	2.627	2.663
36	5	1	–	3.091	2.827	3.237	3.008
36	5	2	–	1.867	1.937	1.902	1.937
36	7	1	–	5.039	4.977	4.371	5.125
36	7	2	–	3.464	3.623	3.556	3.320
36	9	1	–	6.442	6.746	6.063	6.591
36	9	2	–	4.914	5.111	4.809	5.553
36	9	3	–	3.801	3.751	3.722	3.861
49	7	1	–	4.502	4.364	4.710	4.214
49	7	2	–	3.259	3.246	3.391	3.484
49	10	2	–	5.792	5.951	5.786	5.613
49	10	3	–	4.531	4.484	4.440	4.538
49	10	4	–	3.760	3.539	3.870	3.733
49	12	2	–	7.329	7.338	7.284	7.418
49	12	3	–	6.050	6.044	6.163	6.116
49	12	4	–	4.987	5.084	4.861	5.006
Average				4.062	4.089	4.019	4.097

Table A.15. T-IP experiment results for random N_s of size 81 – 100 and 4×4 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
81	12	2	7.180	6.983	7.325	7.274
81	12	3	5.998	5.953	6.155	5.998
81	12	4	5.090	5.002	5.197	5.111
81	16	3	9.228	9.199	8.943	10.285
81	16	4	8.169	7.976	8.044	8.146
81	16	6	6.266	6.062	6.004	6.147
81	20	4	10.988	11.417	11.022	11.092
81	20	6	9.112	8.900	9.107	9.129
81	20	8	7.553	7.031	7.323	7.153
100	15	3	8.145	8.865	8.321	8.535
100	15	4	7.199	7.164	7.249	7.270
100	15	6	5.594	5.566	5.826	5.521
100	20	4	10.886	11.818	10.800	11.916
100	20	6	9.082	9.167	8.986	8.973
100	20	8	7.498	7.059	7.519	7.260
100	25	5	13.695	14.245	13.665	14.263
100	25	7	11.644	11.815	11.696	11.684
100	25	10	9.277	9.007	9.396	8.988
Average			8.478	8.513	8.476	8.597

Table A.16. T-IP experiment results for random N_s of size 144 – 225 and 4×4 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
144	21	4	11.759	12.393	12.077	12.009
144	21	6	9.852	9.662	9.732	9.946
144	21	8	8.664	7.585	8.097	7.649
144	28	5	15.763	17.029	15.779	16.907
144	28	8	13.123	12.900	12.886	12.673
144	28	11	10.958	9.639	10.893	10.173
144	36	7	19.712	20.929	19.754	20.637
144	36	10	17.004	16.945	16.891	16.624
144	36	14	14.036	13.260	13.471	12.776
225	33	6	19.721	20.113	19.432	20.649
225	33	9	15.876	15.571	15.914	15.427
225	33	13	11.892	11.804	12.118	11.832
225	45	9	25.778	25.086	25.197	25.399
225	45	13	20.505	20.249	20.611	20.348
225	45	18	16.230	15.647	16.012	15.614
225	56	11	31.113	31.790	31.244	32.122
225	56	16	25.513	25.870	25.450	25.925
225	56	22	20.037	19.873	20.517	20.044
Average			17.085	17.019	17.004	17.042

Table A.17. T-T-S1 experiment results for random N_s of size 25 – 81 and 4×4 N_i.

				Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	R	TC	R	TC
25	3	1	1.337	1.335	1.335	1.337	1.296
25	5	1	3.122	3.081	3.088	3.122	3.081
25	5	2	1.967	1.884	1.928	1.864	1.840
25	6	1	4.129	4.091	3.996	4.000	3.772
25	6	2	2.709	2.655	2.684	2.607	2.709
36	5	1	–	3.239	3.320	3.239	3.136
36	5	2	–	1.921	1.914	1.853	1.889
36	7	1	–	5.128	5.112	5.047	5.174
36	7	2	–	3.434	3.415	3.629	3.439
36	9	1	–	6.801	6.527	6.656	7.033
36	9	2	–	4.902	4.901	5.312	5.115
36	9	3	–	3.752	3.886	3.797	3.822
49	7	1	–	4.763	4.990	5.072	4.987
49	7	2	–	3.468	3.246	3.468	3.494
49	10	2	–	5.577	5.639	5.577	5.575
49	10	3	–	4.526	4.540	4.498	4.576
49	10	4	–	3.576	3.656	3.866	3.894
49	12	2	–	7.725	7.702	7.368	7.968
49	12	3	–	6.347	5.974	6.267	6.108
49	12	4	–	5.036	4.996	5.103	4.986
81	12	2	–	7.126	7.584	7.216	7.339
81	12	3	–	6.088	6.165	6.044	5.958
81	12	4	–	4.769	5.113	4.769	4.888
81	16	3	–	9.379	8.832	9.033	9.560
81	16	4	–	8.078	7.759	8.013	7.759
81	16	6	–	6.084	5.794	6.034	5.794
81	20	4	–	11.042	10.680	10.897	10.680
81	20	6	–	7.815	8.864	9.051	8.428
81	20	8	–	6.406	7.068	7.366	7.152
Average				5.173	5.197	5.245	5.223

Table A.18. T-T-S1 experiment results for random N_s of size 100 – 225 and 4×4 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
100	15	3	8.305	8.719	8.070	7.690
100	15	4	7.188	7.290	7.147	7.210
100	15	6	5.762	4.941	5.712	4.941
100	20	4	11.026	11.323	11.089	11.323
100	20	6	8.202	9.331	9.040	8.427
100	20	8	7.256	6.506	7.166	6.506
100	25	5	14.884	13.812	12.926	13.812
100	25	7	11.880	11.174	11.822	11.174
100	25	10	8.525	8.319	9.389	8.319
144	21	4	11.625	11.812	11.352	11.812
144	21	6	9.664	9.235	9.925	9.235
144	21	8	7.795	7.370	7.795	7.370
144	28	5	15.673	15.885	15.673	15.885
144	28	8	12.625	12.306	12.625	12.306
144	28	11	10.146	9.328	10.146	9.328
144	36	7	20.583	19.237	19.192	19.237
144	36	10	16.448	15.890	16.448	15.890
144	36	14	12.867	11.903	12.867	11.903
225	33	6	19.157	18.808	19.470	18.808
225	33	9	15.735	14.791	15.623	14.791
225	33	13	11.784	10.886	11.166	10.886
225	45	9	23.798	24.550	23.798	24.550
225	45	13	19.661	19.732	19.661	19.732
225	45	18	15.767	14.937	15.212	14.937
225	56	11	30.960	30.008	30.960	30.008
225	56	16	25.242	23.834	25.242	23.834
225	56	22	19.682	18.169	19.682	18.169
Average			14.157	13.707	14.044	13.633

Table A.19. T-T-S2 experiment results for random N_s of size 25 – 81 and 4×4 N_i.

				Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	R	TC	R	TC
25	3	1	1.337	1.335	1.335	1.337	1.296
25	5	1	3.122	3.081	3.088	3.122	3.081
25	5	2	1.967	1.967	1.928	1.956	1.840
25	6	1	4.129	4.091	3.812	4.000	3.772
25	6	2	2.709	2.602	2.663	2.607	2.709
36	5	1	–	3.239	3.320	3.239	3.136
36	5	2	–	1.921	1.914	1.874	1.889
36	7	1	–	5.128	5.112	5.047	5.174
36	7	2	–	3.600	3.415	3.522	3.439
36	9	1	–	6.801	6.527	6.656	7.033
36	9	2	–	4.902	4.901	5.312	5.115
36	9	3	–	3.752	3.886	3.954	3.797
49	7	1	–	4.763	4.990	5.072	4.987
49	7	2	–	3.468	3.246	3.468	3.401
49	10	2	–	5.577	5.845	5.577	5.592
49	10	3	–	4.526	4.635	4.606	4.519
49	10	4	–	3.793	3.795	3.788	3.907
49	12	2	–	7.725	7.702	7.368	7.968
49	12	3	–	6.047	5.974	5.993	6.131
49	12	4	–	4.967	4.996	5.022	5.067
81	12	2	–	7.126	7.584	7.377	7.339
81	12	3	–	6.091	6.165	6.241	6.045
81	12	4	–	4.769	5.176	4.769	5.204
81	16	3	–	9.557	9.271	9.128	9.815
81	16	4	–	8.501	7.759	8.313	7.759
81	16	6	–	6.091	5.794	6.063	5.994
81	20	4	–	11.358	11.845	11.149	10.680
81	20	6	–	8.758	8.896	8.747	8.920
81	20	8	–	6.406	6.978	7.113	6.949
Average				5.239	2.260	5.256	5.261

Table A.20. T-T-S2 experiment results for random N_s of size 100 – 225 and 4×4 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
100	15	3	9.289	8.968	8.959	9.499
100	15	4	7.835	7.890	7.423	7.650
100	15	6	5.349	4.941	5.261	4.941
100	20	4	11.753	11.323	12.366	11.323
100	20	6	8.202	9.185	8.828	8.427
100	20	8	7.023	6.506	6.840	6.506
100	25	5	15.209	13.812	14.186	13.812
100	25	7	11.842	11.174	11.080	11.174
100	25	10	8.525	8.319	8.798	8.682
144	21	4	12.900	11.812	11.352	12.575
144	21	6	9.546	9.235	9.623	9.235
144	21	8	7.795	7.370	8.042	7.370
144	28	5	15.673	15.885	15.673	15.885
144	28	8	12.625	12.305	12.625	12.305
144	28	11	10.146	9.328	10.146	9.328
144	36	7	19.742	19.237	19.192	19.237
144	36	10	16.448	15.890	16.448	15.890
144	36	14	12.867	11.903	12.867	11.903
225	33	6	20.165	18.808	20.040	18.808
225	33	9	15.394	14.791	15.569	14.791
225	33	13	11.166	10.886	11.166	10.886
225	45	9	23.798	24.550	23.798	24.550
225	45	13	20.316	19.732	19.661	19.732
225	45	18	15.678	14.937	15.212	14.937
225	56	11	30.960	30.008	30.960	30.008
225	56	16	25.242	23.834	25.242	23.834
225	56	22	19.682	18.169	19.682	18.169
Average			14.266	13.733	14.113	13.758

Table A.21. T–IP experiment results for random N_s of size 25 - 81 and 5×5 N_i.

				Mode 1		Mode 2	
$\|N_s\|$	b	$\bar{b}$	Optimal Value	R	TC	R	TC
25	3	1	1.694	1.694	1.687	1.687	1.653
25	5	1	3.945	3.789	3.864	3.562	3.904
25	5	2	2.486	2.397	2.486	2.439	2.377
25	6	1	5.243	4.386	4.898	4.412	4.701
25	6	2	3.414	3.414	3.342	3.324	3.140
36	5	1	–	4.081	4.203	4.186	4.118
36	5	2	–	2.486	2.239	2.295	2.445
36	7	1	–	6.473	5.645	6.530	6.238
36	7	2	–	4.418	4.211	4.238	4.099
36	9	1	–	7.698	8.072	8.309	8.820
36	9	2	–	6.262	7.053	6.143	6.076
36	9	3	–	5.090	4.892	5.037	4.636
49	7	1	–	6.039	5.600	6.088	5.658
49	7	2	–	4.125	4.269	4.213	4.298
49	10	2	–	7.319	7.061	7.394	7.218
49	10	3	–	5.729	5.669	5.616	5.715
49	10	4	–	4.750	4.921	4.906	4.951
49	12	2	–	9.304	9.636	9.254	9.309
49	12	3	–	7.663	7.662	7.813	7.614
49	12	4	–	6.301	6.272	6.144	6.369
81	12	2	–	9.249	9.635	9.113	9.682
81	12	3	–	7.732	7.836	7.742	8.006
81	12	4	–	6.430	6.572	6.500	6.361
81	16	3	–	11.323	11.959	11.966	12.215
81	16	4	–	10.309	10.251	10.140	10.427
81	16	6	–	7.917	7.443	7.990	7.554
81	20	4	–	14.137	14.001	14.087	14.196
81	20	6	–	11.535	11.321	11.539	11.335
81	20	8	–	9.345	9.529	9.245	9.282
Average				6.600	6.629	6.618	6.634

Table A.22. T–IP experiment results for random N_s of size 100 - 225 and 5×5 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
100	15	3	10.322	11.620	10.306	10.641
100	15	4	9.102	9.233	9.161	9.007
100	15	6	7.343	6.777	7.299	7.289
100	20	4	13.786	14.630	13.669	14.220
100	20	6	11.538	11.413	11.356	11.139
100	20	8	9.644	9.645	9.503	9.500
100	25	5	17.351	18.059	17.323	18.391
100	25	7	15.047	14.504	14.813	14.771
100	25	10	11.705	11.868	11.699	11.741
144	21	4	14.887	14.764	14.867	14.654
144	21	6	12.375	12.218	12.267	12.267
144	21	8	10.496	10.853	10.259	9.394
144	28	5	19.919	21.302	19.940	20.997
144	28	8	16.224	16.304	16.275	16.056
144	28	11	13.765	13.034	13.497	13.950
144	36	7	25.142	25.468	25.159	26.163
144	36	10	21.069	21.694	21.402	21.033
144	36	14	17.318	15.749	17.758	15.530
225	33	6	24.584	24.340	25.440	24.410
225	33	9	19.765	19.977	20.184	19.653
225	33	13	14.914	15.691	15.019	14.998
225	45	9	31.811	32.061	31.850	32.175
225	45	13	25.995	25.286	26.116	26.245
225	45	18	20.496	20.104	20.801	20.262
225	56	11	39.447	39.621	39.346	39.622
225	56	16	32.259	32.036	32.357	32.089
225	56	22	25.276	24.506	25.313	24.872
Average			18.207	18.250	18.258	18.188

Table A.23. T–T–S1 experiment results for random N_s of size 25 - 81 and 5×5 N_i.

				Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	R	TC	R	TC
25	3	1	1.694	1.694	1.640	1.694	1.687
25	5	1	3.945	3.892	3.900	3.945	3.820
2525	5	2	2.486	2.411	2.307	2.350	2.416
25	6	1	5.243	5.164	5.222	4.889	5.214
25	6	2	3.414	3.358	3.307	3.295	3.192
36	5	1	–	4.181	4.095	4.181	4.126
36	5	2	–	2.426	2.281	2.336	2.386
36	7	1	–	6.406	6.268	6.551	6.256
36	7	2	–	4.343	4.599	4.652	4.168
36	9	1	–	8.927	8.998	8.582	8.779
36	9	2	–	6.422	6.021	6.664	6.028
36	9	3	–	4.741	4.882	4.809	5.108
49	7	1	–	6.062	5.905	6.532	5.911
49	7	2	–	4.482	4.449	4.575	4.540
49	10	2	–	7.061	7.030	7.061	7.030
49	10	3	–	5.723	5.669	5.696	5.669
49	10	4	–	4.759	4.690	4.978	4.843
49	12	2	–	9.913	9.185	9.674	9.026
49	12	3	–	8.051	7.997	7.975	7.763
49	12	4	–	6.457	6.363	6.392	6.074
81	12	2	–	9.041	9.346	9.776	9.691
81	12	3	–	7.710	7.847	7.651	8.301
81	12	4	–	6.014	6.381	6.014	6.443
81	16	3	–	11.904	12.065	11.572	11.608
81	16	4	–	10.012	9.745	10.136	10.134
81	16	6	–	7.675	6.924	7.614	6.924
81	20	4	–	13.791	14.597	13.770	13.437
81	20	6	–	9.846	10.434	11.439	10.434
81	20	8	–	8.059	9.196	9.267	9.287
Average				6.570	6.598	6.692	6.562

Table A.24. T–T–S1 experiment results for random N_s of size 100 - 225 and 5×5 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
100	15	3	10.525	11.271	10.204	10.371
100	15	4	9.206	8.863	9.026	8.863
100	15	6	7.281	6.631	7.095	6.998
100	20	4	15.113	13.957	14.038	13.957
100	20	6	10.343	10.913	11.423	10.913
100	20	8	9.143	8.197	9.467	8.197
100	25	5	18.379	17.455	16.340	17.455
100	25	7	15.030	14.334	14.955	14.334
100	25	10	10.746	10.179	11.810	10.179
144	21	4	14.333	14.654	14.333	15.295
144	21	6	12.214	12.158	12.556	11.884
144	21	8	9.832	9.220	9.832	9.220
144	28	5	19.827	19.408	19.827	19.408
144	28	8	15.931	15.956	15.931	16.433
144	28	11	12.797	11.515	12.797	11.515
144	36	7	26.034	24.653	24.266	24.653
144	36	10	20.779	20.270	20.779	20.270
144	36	14	16.229	15.116	16.229	15.116
225	33	6	24.465	23.531	25.244	23.531
225	33	9	19.781	19.040	19.703	19.040
225	33	13	14.862	13.899	14.060	13.899
225	45	9	30.102	31.346	32.439	31.346
225	45	13	24.841	25.053	24.841	25.053
225	45	18	20.310	18.712	19.179	18.712
225	56	11	39.165	38.930	39.165	38.930
225	56	16	31.884	31.038	31.884	31.038
225	56	22	24.819	23.422	24.819	23.422
Average			17.925	17.397	17.861	17.409

Table A.25. T–T–S2 experiment results for random N_s of size 25 - 81 and 5×5 N_i.

				Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	Optimal Value	R	TC	R	TC
25	3	1	1.694	1.687	1.694	1.694	1.640
25	5	1	3.945	3.892	3.900	3.945	3.903
2525	5	2	2.486	2.486	2.307	2.466	2.486
25	6	1	5.243	5.164	5.222	4.889	4.842
25	6	2	3.414	3.284	3.307	3.295	3.192
36	5	1	–	4.181	4.095	4.181	4.126
36	5	2	–	2.426	2.446	2.426	2.386
36	7	1	–	6.406	6.268	6.551	6.256
36	7	2	–	4.343	4.599	4.452	4.168
36	9	1	–	8.927	8.998	8.582	8.797
36	9	2	–	6.422	6.021	6.664	6.933
36	9	3	–	4.741	4.754	4.809	5.021
49	7	1	–	6.062	5.905	6.532	5.911
49	7	2	–	4.482	4.353	4.575	4.540
49	10	2	–	7.061	7.258	7.061	7.030
49	10	3	–	5.723	5.669	5.696	5.669
49	10	4	–	4.792	4.690	4.736	4.597
49	12	2	–	9.913	9.185	9.674	9.026
49	12	3	–	8.051	7.801	8.038	8.156
49	12	4	–	6.358	6.498	6.340	6.074
81	12	2	–	9.041	9.346	9.776	9.691
81	12	3	–	7.710	7.847	7.815	7.780
81	12	4	–	6.014	6.441	6.014	6.404
81	16	3	–	11.832	11.760	12.313	12.843
81	16	4	–	10.264	9.745	10.303	10.412
81	16	6	–	7.756	7.585	7.532	6.924
81	20	4	–	14.536	14.877	14.884	14.277
81	20	6	–	11.207	11.320	10.988	10.434
81	20	8	–	8.059	8.716	9.038	8.826
Average				6.649	6.642	6.733	6.633

Table A.26. T–T–S2 experiment results for random N_s of size 100 - 225 and 5×5 N_i.

			Mode 1		Mode 2	
$\lvert N_s \rvert$	b	$\bar{b}$	R	TC	R	TC
100	15	3	11.622	11.254	11.511	10.371
100	15	4	9.658	9.847	9.688	9.471
100	15	6	6.866	6.668	6.792	6.533
100	20	4	15.174	15.604	15.814	13.956
100	20	6	11.128	10.913	11.067	10.913
100	20	8	8.758	8.197	8.386	8.197
100	25	5	17.687	18.731	18.979	19.465
100	25	7	13.986	14.334	14.988	14.583
100	25	10	10.746	10.179	11.258	10.179
144	21	4	15.406	16.821	14.333	15.665
144	21	6	12.098	12.103	12.331	11.884
144	21	8	10.242	9.220	9.832	9.220
144	28	5	20.778	19.408	21.312	19.408
144	28	8	15.931	15.789	15.931	16.411
144	28	11	12.797	11.515	12.797	11.515
144	36	7	24.957	24.653	24.266	24.653
144	36	10	20.779	20.270	20.779	20.270
144	36	14	16.229	15.116	16.229	15.116
225	33	6	24.388	25.030	24.393	23.531
225	33	9	19.747	19.040	19.233	19.040
225	33	13	14.632	13.899	14.060	13.899
225	45	9	30.102	31.346	31.530	31.346
225	45	13	24.841	25.053	24.841	25.053
225	45	18	19.178	18.712	19.178	18.712
225	56	11	39.165	38.930	39.165	38.930
225	56	16	31.884	31.038	31.884	31.038
225	56	22	24.819	23.422	24.819	23.422
Average			17.911	17.670	17.978	17.510

REFERENCES

1. Akyildiz, I.F., W.Su, Y.Sankarasubramaniam, and E.Cayirci, "A Survey on Sensor Networks", *IEEE Communications Magazine*, Vol. 8, pp. 102-114, 2002.

2. Altınel, İ.K., N.Aras, E.Güney, and C.Ersoy, "Binary integer programming formulation andheuristics for differentiated coverage in heterogeneous sensor networks", *Computer Networks*, Vol. 52, pp. 2419-2431, 2008.

3. Bard, J.F., *Practical Bilevel Optimization: Algorithms and Applications*, Kluwer Academic Publishers, Boston, 1998.

4. Bard, J.F, and J.T.Moore, "An Algorithm for the Discrete Bilevel Programming Problem", *Naval Research Logistics*, Vol. 39, pp. 419-435, 1992.

5. Başdere, M., N.Aras, İ.K.Altınel, and S.Afşar, "A Leader-Follower Game for the Point Coverage Problem in Wireless Sensor Networks", *European Journal of Industrial Engineering*, Forthcoming 2012.

6. Bayrak, H., and M.D.Bailey, "Shortest path network interdiction with asymmetric information", *NETWORKS*, Vol. 52, No.3, pp. 133-140, 2008.

7. Berman, O., V.Verter, and B.Y.Kara, "Designing emergency response networks for hazardous materials transportation", *Computers & Operations Research*, Vol. 34, pp. 1374-1388, 2007.

8. Bianco, L., M.Caramia, and S.Giordani, "A bilevel flow model for hazmat transportation network design", *Transportation Research Part C*, Vol. 17, 175-196, 2009.

9. Bracken, J., and J.T.McGill, "Mathematical Programs with Optimization Problems in the Constraints", *Operations Research*, Vol. 21, No. 1, pp. 37-44, 1973.

10. Cappanera, P., and M.P.Scappara, "Optimal Allocation of Protective Resources in Shortest-Path Networks", *Transportation Science*, Vol. 45, No.1, pp. 64-80, 2011.

11. Chinchuluun, A., P.M.Pardalos, and H.X.Huang, "Multilevel (Hierarchical) Optimization: Complexity Issues, Optimality Conditions, Algorithms", *Advances in Applied Mathematics and Global Optimization*, Vol. 17, pp. 197-221, 2009.

12. Clouqueur, T., V.Phipatanasuphorn, P.Ramanathan, and K.K.Saluja, "Sensor deployment strategy for target detection", *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications*, pp. 42-48, 2002.

13. Colson, B., P.Marcotte, and G.Savard, "An Overview of Bilevel Optimization" *Annals of Operations Research*, Vol. 153, pp. 235-256, 2007.

14. Dempe, S., "Annotated Bibliography on Bilevel Programming and Mathematical Programs with Equilibrium Constraints", *Optimization*, Vol. 52, No.3, pp. 333-359, 2003.

15. DeNegre, S.T., and T.K.Ralphs, "A Branch-and-Cut Algorithm for Integer Bilevel Linear Programs", *Operations Research and Cyber-Infrastructure*, Vol. 47, pp. 65-78, 2009.

16. Dhillon, S.S., and K.Chakrabarty, "Sensor Placement for Effective Coverage and Surveillance in Distributed Sensor Networks", *IEEE Wireless Communications and Networking Conference*, Vol. 3, pp. 1609-1614, 2003.

17. Dhillon, S.S., K.Chakrabarty, and S.S.Iyengar, "Sensor Placement for Grid Coverage under Imprecise Detections", *Proceedings of the Fifth International Conference on Information Fusion*, Vol. 2, pp. 1581-1587, 2002.

18. Erkut, E., and O.Alp, "Designing a road network for hazardous materials shipments", *Computers & Operations Research*, Vol. 34, pp. 1389-1405, 2007.

19. Fanghanel, D., and S.Dempe, "Bilevel Programming with Discrete Lower Level Problems", *Optimization*, Vol.58, No. 8, pp. 1029-1047, 2009.

20. Ghosh, A., and S.K.Das, "Coverage and connectivity issues in wireless sensor networks", *Pervasive and Mobile Computing*, Vol. 4, pp. 303-334, 2008.

21. Glover, F., and M.Laguna, *Tabu Search*, Kluwer Academic Publishers, Dordrecht, Netherlands, 1997.

22. Holmgren, A.J., E.Jenelius, and J.Westin, "Evaluating Strategies for Defending Electric Power Networks Against Antagonistic Attacks", *IEEE TRANSACTIONS ON POWER SYSTEMS*, Vol. 22, No.1, pp. 76-84, 2007.

23. Israeli, E., and R.K.Wood, "Shortest-Path Network Interdiction", *NETWORKS*, Vol. 40, No.2, pp. 97-111, 2002.

24. Jain, M., E.Kardes, C.Kiekintveld, F.Ordonez, and M.Tambe, "Optimal Defender Allocation for Massive Security Games: A Branch and Price Approach", *AAMAS 2010 Workshop on Optimisation in Multi-Agent Systems (OptMas)*, 2010.

25. Kara, B.Y., and V.Verter, "Designing a road network for hazardous material transportation", *Transportation Science*, Vol. 38, No.2, pp. 188-196, 2004.

26. Küçükaydın, H., N.Aras, and İ.K.Altınel, "A Leader-Follower Game in Competitive Facility Location", *Computers & Operations Research*, Vol. 39, pp. 437-448, 2012.

27. Machado, R., and S.Tekinay, "A Survey of Game Theoretic Approaches in Wireless Sensor Networks", *Computer Networks*, Vol. 52, pp. 3047-3061, 2008.

28. Megerian, S., F.Koushanfar, G.Qu, G.Veltri, and M.Potkonjak, "Exposure in wireless sensor networks: Theory and practical solutions", *Wireless Networks*, Vol. 8, No.5, pp. 443-454, 2002.

29. Meguerdichian, S., F.Koushanfar, M.Potkonjak, and M.B.Srivastava, "Coverage problems in wireless ad-hoc sensor networks", *Proceedings of IEEE Infocom*, pp. 115-121, 2001.

30. Meguerdichian, S., F.Koushanfar, M.Potkonjak, and M.B.Srivastava, "Worst and Best-Case Coverage in Sensor Networks", *IEEE TRANSACTIONS ON MOBILE COMPUTING*, Vol. 4, No.1, pp. 84-92, 2005.

31. Meguerdichian, S., F.Koushanfar, G.Qu, and M.Potkonjak, "Exposure in wireless ad-hoc sensor networks", *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking*, pp. 139-150, 2001.

32. Moore, J.T., and J.F.Bard, "The Mixed Integer Linear Bilevel Programming Problem", *Operations Research*, Vol. 38, No. 5, pp. 911-921, 1990.

33. O'Rourke, J., *Art Gallery Theorems and Algorithms*, Oxford University Press, Inc., Oxford, 1987.

34. Salmeron, J., K.Wood, and R.Baldick, "Analysis of Electric Grid Security Under Terrorist Threat", *IEEE TRANSACTIONS ON POWER SYSTEMS*, Vol. 19, No.2, pp. 905-912, 2004.

35. Stackelberg,H. von, *Market Structure and Equilibrium*, Springer, New York, 2011.

36. Veltri, G., Q.Huang, G.Qu, and M.Potkonjak, "Minimal and maximal exposure path algorithms for wireless embedded sensor networks", *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems*, pp. 40-50, 2003.

37. Verter, V., and B.Y.Kara, "A path-based approach for hazmat transport network design", *Management Science*, Vol. 54, No.1, pp. 29-40, 2008.

38. Wen, U.P., and Y.H.Yang, "Algorithms for Solving the Mixed Integer Two-Level Linear Programming Problem", *Computers & Operations Research*, Vol. 17, No. 2,

pp.133-142, 1990.

39. Wood, R.K., "Deterministic Network Interdiction", *Mathematical and Computer Modeling*, Vol. 17, No. 2, pp. 1-18, 1993.

40. Yates, J., R.Batta, and M.Karwan, "Optimal Placement of Sensors and Interception Resource Assessment for the Protection of Regional Infrastructure from Covert Attack", *Journal of Transportation Security*, Vol.4, No.2, pp. 145-169, 2011.

41. Yick, J., B.Mukherjee, and D.Ghosal, "Wireless sensor network survey", *Computer Networks* Vol. 52, pp. 2292-2330, 2008.