

BILEVEL MAXIMAL COVERING PROBLEM FOR LOCATION, RELOCATION
AND PROTECTION PLANNING OF PUBLIC SERVICE FACILITIES

by

Serhat Keçici

BS, Industrial Engineering, Middle East Technical University, 2008

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

2011

To my beloved family.

ACKNOWLEDGEMENTS

I would like to express my sincere gratitude to my thesis supervisor Associate Professor Necati Aras for his invaluable guidance and help during the preparation of this thesis. I attribute the level of my Masters degree to his encouragement and effort even in the hardest times. This thesis would not have been completed or written without him.

My special thanks go to my thesis co-supervisor Prof. Vedat Verter for his support and guidance. He hosted 3 months of this research at McGill University in a more caring way than I could ever expect.

I would also like to thank to Prof. Kuban Altınel and Assist. Prof. Deniz Aksen for their interest and sparing time in examining my thesis and taking part in my thesis jury.

I would like to show my gratitude to my friend Sema Şengül Akca whose precious contributions deserve the most appreciation from the initial to the final level and enabled me to complete this study.

ABSTRACT

BILEVEL MAXIMAL COVERING PROBLEM FOR LOCATION, RELOCATION AND PROTECTION PLANNING OF PUBLIC SERVICE FACILITIES

In this study, we consider the design problem of a public service facility network with existing facilities when there is a threat of possible terrorist attacks. The aim of the system planner, who is responsible for the operation of the network, is to open new facilities, relocate existing ones if necessary, and protect some of the facilities to ensure a maximum coverage of the demand that is assumed to be aggregated at customer zones. By doing so, the system planner anticipates that a number of unprotected facilities will be rendered out-of-service by terrorist attacks. It is assumed that the sum of the fixed costs of opening new facilities, the relocation costs, and the protection costs cannot exceed a predetermined budget level. Adopting the notion of gradual (partial) coverage, we formulate a bilevel programming model where the system planner is the leader and the attacker is the follower. The objective of the former is the maximization of the total service coverage, whereas the latter wants to minimize the same measure. We propose a heuristic solution procedure based on tabu search where the search space consists of the decisions of the system planner, and the corresponding objective value is computed by optimally solving the attacker's problem using CPLEX. To assess the quality of the solutions produced by the Tabu search heuristic, we also develop binary enumeration tree method, which explores all the possible combinations of opening new facilities, relocating existing ones, and protecting them. Since its time complexity is exponential, it can only be employed for relatively small instances.

ÖZET

KAMU HİZMETİ VEREN TESİSLERİN YERLEŞTİRME, YERDEĞİŞTİRME VE KORUNMA PLANLAMASI İÇİN ÇİFT DÜZEYLİ KAPSAMA PROBLEMİ

Bu çalışmada kamu hizmeti veren bir ağın muhtemel bir terorist saldırısı altında tasarlanması problemi üzerinde çalışılmıştır. Sistem planlayıcısı ağı işletilmesinden sorumludur ve amacı mevcut tesislerin yerini değiştirerek, yeni tesisler açarak ve tesislerin bazılarını koruyarak en yüksek toplam müşteri talebi kapsamını sağlamaktır. Bunu yaparken belli bir sayıda korunmamış tesisin terorist saldırısı sonucu servis dışı kalacağını hesaba katmaktadır. Yeni tesis açma maliyetleri, mevcut tesislerin yerlerinin değiştirilmesi maliyetleri ve tesislerin korunması maliyetlerinin toplamının belli bir bütçeyi aşmasına izin verilmemektedir. Kısmi kapsama kavramını kullanarak, bu problem çift düzeyli programlama modeli olarak formüle edilmiştir. Sistem planlayıcısı lider oyuncudur ve saldırı sonrası toplam kapsanan müşteri talebini enyükseklemek istemektedir. Terorist ise takipçi oyuncu olarak bunun tam tersini amaçlamaktadır. Bu problemin çözülmesi için Tabu Arama sezgiseli önerilmiştir. Arama uzayı sistem planlayıcısının kararlarını içermektedir ve nihayi toplam kapsama değeri teroristin probleminin CPLEX kullanılarak çözülmesi ile elde edilmektedir. Tabu Arama sezgiseli ile bulunan çözümlerin kalitesinin belirlenmesi amacıyla, problemi optimal olarak çözen bir ikili sayma ağacı metodu geliştirilmiştir. Yeni tesis açılması, mevcut tesislerin yer değiştirmesi ve tesislerin korunması alternatiflerinin bütün kombinasyonlarını değerlendiren bu yöntem, problem büyüklüğüne oranla üssel bir çözüm süresi gerektirmektedir ve dolayısıyla sadece küçük problemlerin çözümünde kullanılabilir.

TABLE OF CONTENTS

ACKNOWLEDGEMENTS	iv
ABSTRACT	v
ÖZET	vi
LIST OF FIGURES	x
LIST OF TABLES	xiii
LIST OF SYMBOLS/ABBREVIATIONS	xvi
1. INTRODUCTION	1
2. LITERATURE SURVEY	4
2.1. Covering Facility Location Problems	4
2.1.1. Extensions Relaxing Capacity Assumptions	6
2.1.2. Extensions Relaxing All or Nothing Coverage Assumption	6
2.1.3. Extensions Relaxing Discrete Potential Facility Locations Assumption	7
2.1.4. Extensions Relaxing Fixed Coverage Range Assumption	7
2.2. Interdiction Problems	8
3. PROBLEM FORMULATION	13
3.1. Bilevel Programming	13
3.2. Gradual Coverage	14
3.3. Bilevel r -Interdiction Covering Problem With Protection	15
4. SOLUTION PROCEDURES	21
4.1. Two Step Tabu Search Heuristic Algorithm	21
4.1.1. Parameters Definitions for TSH	24
4.1.2. Initial Solution and Neighborhood Generation	25
4.1.3. Evaluation of the Neighbors and Tabu Criterion	27
4.1.4. Termination of TSH	29
4.2. Binary Enumeration Tree Algorithm	29
4.2.1. Branching Procedure	30
4.2.2. Pruning due to Budget Infeasibility	32
4.2.3. Upper Bound Approach	32

4.2.4. Solutions with Budget Surplus	33
5. COMPUTATIONAL RESULTS	35
5.1. Problem Instances Generation	35
5.2. Computational Environment	37
5.3. Results of the BRICP Problem Instances	37
5.4. Analysis from Computational Perspective	40
5.4.1. Solving the ULP and LLP Separately	40
5.4.2. Contribution of Two Runs Approach in TSH	40
5.4.3. Contribution of UB Approach in BET	42
5.4.4. The Effect of the Number of Potential Sites on Solution Performance	43
5.4.5. Effect of Customer Configuration on Solution Performance	44
5.4.6. Effect of Budget in Solution Performance	45
5.4.7. Effect of r on the Solution Performance	46
5.5. Analysis from a Managerial Perspective	48
5.5.1. System Planner's Strategies	48
5.5.2. Two Types of Protection for the System Planner	49
5.5.3. Value of Accurate Information to the System Planner	51
5.5.4. The Impact of Customer Dispersion on the System Planner's Decisions	54
5.5.5. The value of Protection to the System Planner	55
5.5.6. The Value of Relocation to the System Planner	57
5.5.7. The Impact of Quantity of Potential Facility Sites	59
5.5.8. The Impact of Terrorist's Resources	59
6. CONCLUSION	62
APPENDIX A: REAL WORLD CUSTOMER CONFIGURATIONS	65
APPENDIX B: RANDOMLY GENERATED CUSTOMER CONFIGURATIONS	66
APPENDIX C: REAL WORLD CUSTOMER CONFIGURATIONS RESULTS	69
APPENDIX D: RANDOM CUSTOMER CONFIGURATIONS RESULTS	73
APPENDIX E: RESULTS OF THE TEST ABOUT K	85
APPENDIX F: RESULTS ACCORDING TO THE NUMBER OF CUSTOMERS	86
APPENDIX G: RESULTS OF THE TEST ABOUT BUDGET	87

APPENDIX H: RESULTS OF THE TEST FOR r	89
APPENDIX I: TEST RESULTS OF THE PROTECTION RESOURCES . . .	92
APPENDIX J: RESULTS OF INACCURATE KNOWLEDGE EFFECTS . .	96
APPENDIX K: ILLUSTRATIONS OF EXAMPLE SOLUTIONS	106
APPENDIX L: TEST RESULTS ON ALPHA	112
APPENDIX M: TEST RESULTS ON RELOCATION OPTION	129
REFERENCES	131

LIST OF FIGURES

Figure 4.1.	Examples of H vector	26
Figure 5.1.	CPU time as a function of the number of potential facility sites . .	44
Figure 5.2.	Gap as a function of the number of potential facility sites	44
Figure 5.3.	CPU time as a function of the number of customer sites	45
Figure 5.4.	Gap versus number of customer sites	45
Figure 5.5.	CPU time of TSH as a function of r	47
Figure 5.6.	CPU time of BET as a function of r	47
Figure 5.7.	Categorization of system planner's action in real world customer configurations	49
Figure 5.8.	Categorization of system planner's action in random customer configurations	49
Figure 5.9.	Marginal contribution of protection resources	50
Figure 5.10.	Average contribution per protection resource	51
Figure 5.11.	Coverage under accurate and inaccurate information to the system planner	53
Figure 5.12.	Marginal contribution of protection resources	54

Figure 5.13.	The effect of customer dispersion on final coverage value	55
Figure 5.14.	The effect of customer dispersion on the facility decisions	55
Figure 5.15.	Effect of α on choice of protection in uniform customer dispersion	56
Figure 5.16.	Effect of α on choice of protection in clustered customer dispersion	57
Figure 5.17.	Effect of relocation option to final coverage value	58
Figure 5.18.	Effect of relocation option to the system planner's strategy	58
Figure 5.19.	Effect of number of potential facility sites on final coverage	59
Figure 5.20.	Effect of terrorist resources on final coverage value	60
Figure 5.21.	Effect of terrorist's resources on the system planner's decision . . .	61
Figure A.1.	SJC 324 customer configuration	65
Figure A.2.	SJC 818 customer configuration	65
Figure B.1.	U200 customer configuration	66
Figure B.2.	U400 customer configuration	66
Figure B.3.	U800 customer configuration	67
Figure B.4.	C200 customer configuration	67
Figure B.5.	C400 customer configuration	68

Figure B.6.	C800 customer configuration	68
Figure K.1.	Solution of the problem instance U200(15–1)	106
Figure K.2.	Solution of the problem instance U400(20–3)	107
Figure K.3.	Solution of the problem instance U800(15–1)	108
Figure K.4.	Solution of the problem instance C200(15–7)	109
Figure K.5.	Solution of the problem instance C400(10–2)	110
Figure K.6.	Solution of the problem instance C800(20–9)	111

LIST OF TABLES

Table 4.1.	Parameters definition for TSH algorithm	25
Table 5.1.	Random customer configurations generation	36
Table 5.2.	Summary of the overall results	39
Table 5.3.	Test results of DSM on instances of SJC324 configuration	41
Table 5.4.	Test results of TSH FR and SR on instances of SJC324 instance	42
Table 5.5.	Test results of BET (with and without UB) on instances of U200 and C200	43
Table 5.6.	Results of BET and TSH for various budget amounts	46
Table C.1.	Solutions to problem instances based on SJC 324 customer config- uration	69
Table C.2.	Solutions to problem instances based on SJC 818 customer config- uration	71
Table D.1.	Solutions to problem instances based on U200 customer configura- tion	73
Table D.2.	Solutions to problem instances based on U400 customer configura- tion	75
Table D.3.	Solutions to problem instances based on U800 customer configura- tion	77
Table D.4.	Solutions to problem instances based on C200 customer configura- tion	79
Table D.5.	Solutions to problem instances based on C400 customer configura- tion	81

Table D.6.	Solutions to problem instances based on C800 customer configuration	83
Table E.1.	Summarized overall results according to the number of potential facility locations	85
Table F.1.	Summarized overall results according to the number of customers .	86
Table G.1.	Detailed results of the test about effects of budget on instances of SJC324 configuration	87
Table H.1.	Detailed test results for r on instances of SJC818 configuration . .	89
Table H.2.	Summarized results of test for r on instances of SJC818 configuration	91
Table H.3.	Average number of protections and attacks	91
Table I.1.	Detailed test results with protection limit where the locations are fixed on instances of SJC324	92
Table I.2.	Summarized results according to the marginal contribution of each protection	95
Table I.3.	Summarized results according to the average contribution per protection	95
Table J.1.	Detailed results on inaccurate knowledge on instances of SJC324 configuration	96
Table J.2.	Summarized results according to estimation error	105
Table L.1.	Detailed test results of alpha on instances of uniform customer configuration	112
Table L.2.	Detailed test results of alpha on instances of clustered customer configuration	119

Table M.1.	Detailed test results of relocation option on instances of SJC818	
	configuration	129

LIST OF SYMBOLS/ABBREVIATIONS

a_{ki}	Partial coverage of customer zone $k \in \mathcal{K}$ by the facility at site $i \in \mathcal{N}$
b	Available budget of the system planner
d	Distance between customer and cluster center
d_k	Demand of customer zone $k \in \mathcal{K}$
e_{ki}	Euclidian distance between customer k and site i
f_i	Fixed cost of opening a new facility at site $i \in \mathcal{N}_2$
g_{ij}	Cost of relocating an existing facility from site $i \in \mathcal{N}_1$ to site $j \in \mathcal{N}_2$
H	Vector representing the current solution
H'	Vector representing a neighboring solution H
h_i	Fixed cost of protecting a facility at site $i \in \mathcal{N}$
$\mathcal{K}$	Set of customer zones
$\mathcal{N}$	Set of existing and potential facility sites
$\mathcal{N}_1$	Set of existing facility sites
$\mathcal{N}_2$	Set of potential facility sites
P_i	Protection status of facility at site i
pw	Penalty weight
r	The maximum number of facilities the attacker can destroy
R_ℓ	Lower limit on coverage range
R_u	Upper limit on coverage range
S_i	Interdiction of facility at site i
U_{ij}	Relocation of facility at site i to site j
W_k	Fraction of customer zone k 's demand served after the attack
Y_i	New facility opening at site i
α	Relative cost of protection versus facility opening
θ	Angle between the customer location cluster center

BCRIMF-CE	Budget constrained r -interdiction median problem with capacity expansion
BET	Binary enumeration tree
BIP	Bilevel interdiction problem
BP	Bilevel programming
BRICP	Bilevel r interdiction problem with protection
C	Clustered
CMCLP	Capacitated maximal covering location problem
FR	First run
GDCP	Gradual coverage decay problem
GMCLP	Generalized maximum coverage location problem
IMF	Interdiction median problem with fortification
LP	Linear programming
LLP	Lower level problem
LSCP	Location set covering problem
MCLIP	Maximum covering location interdiction problem
MCLP	Maximal covering location problem
MCPC	Maximal covering problem with precedence constraints
MIP	Mixed integer programming
RIC	r -interdiction covering problem
RIM	r -interdiction median problem
RIMF	r -interdiction median problem with fortification
SJC	São José dos Campos
SR	Second run
SS	Search space
SSM	Separate solution methodology
TSH	Tabu search heuristic
U	Uniform
UB	Upper bound
ULP	Upper level problem

1. INTRODUCTION

Terrorist attacks are an increasing concern around the globe. Having existed throughout centuries, terrorist groups do not seem to disappear in the near future despite the harsh precautions taken against any illegal movement. On the contrary, they are increasing their effects in recent years through intelligently planned attacks. For instance, in 2003 two bombs exploded in İstanbul at two synagogues followed by two other bombs detonated by suicide bombers at the British Consulate and headquarters of a British bank. In 2004 10 bombs were simultaneously detonated in Madrid on 4 commuter trains causing 191 mortalities. The most appalling of all was probably the September 11 attacks with a more complex plan where four airplanes were hijacked and crashed intentionally by terrorists to the World Trade Center buildings and the Pentagon resulting in 3000 mortalities. All of these recent attacks seem to have been planned well in advance so as to cause the maximum harm to public. In a report to U.S. Congress, Perl [1] highlights increased sophistication in terrorists' plans of action as a major trend in terrorism. It is clear that the new generation of technology make the terrorists capable to collect the necessary information and do their operational planning prior to any attack with the intention of using their resources to cause maximum harm. Responding to this trend, governments have been strengthening the protection of critical facilities against such attacks. Strengthened protection of hard targets, however, heightened the attractiveness of soft targets for terrorist attacks [2]. The attack of Taliban to telecommunication towers in Afghanistan [3] is a significant example, where the terrorists' primary aim was disrupting service rather than killing people. As a consequence of that attack, public cell phone telecommunication was interrupted with no mortality. Another example is the assault on an ambulance station in Northern Ireland [4] where the station was shut down after the attack causing a major disruption in emergency services. The possibility of disruptive threats to an electric grid is highlighted in Motto et al. [5].

In this thesis we study the impact of the possibility of terrorist attacks in the design of public service facility networks, which are considered as soft targets. We focus

on the interplay of the decisions of two players; the system planner who is mandated to develop and protect public service infrastructure to maximize total post attack service coverage, and the terrorists whose attacks are planned to cause the maximum service disruption. This problem lends itself to a bilevel programming formulation, where the government at the outer level and the attacker at the inner level are optimizing a service coverage objective. The system planner can relocate its existing facilities, open new ones and protect some of these facilities with the aim of maximizing the total post-attack coverage. The attacker, on the other hand, can simultaneously destroy a certain number of unprotected facilities with the intention of minimizing the resulting total coverage. We call our problem the bilevel r -interdiction covering problem with protection (BRICP).

We formulate our problem as a bilevel program where problems at both levels are mixed integer programming (MIP) problems. Bilevel programming (BP) problems are NP-hard even if the subproblems are linear programming (LP) problems [6]. Besides, general purpose MIP solvers are incapable of solving bilevel MIP models due the non-reducible structure including two interdependent problems. Therefore, we develop a tabu search heuristic (TSH) algorithm. The search procedure traces the solutions to the system planner's problem while for each solution the CPLEX solver returns the optimal reaction plan of the attacker. In an effort to assess the performance of our heuristic algorithm, a binary enumeration tree (BET) method is developed which provides optimal solutions to our BP problem for small instances. The comparison of the heuristic method with the optimal solutions generated by the binary enumeration tree gives the capability of the tabu search algorithm in finding good solutions within short reasonable time.

One of the differentiating features of this study is the integration of the facility location, relocation and protection decisions for maximizing the coverage of public service facility networks. In an effort to increase the applicability, the proposed model is made to explicitly represent the possibility of relocating some of the existing facilities. In other words, Unlike the previous studies in the literature, we assume that the system planner does not design a service facility network from scratch. We also relax the

popular assumption that demand at customer nodes are covered with “all or nothing” principle. Rather we adopt the gradual coverage concept in estimating the level of service provided by a network of facilities.

2. LITERATURE SURVEY

2.1. Covering Facility Location Problems

Facility location problems have received significant attraction in operations research since Weber introduced single facility location problem [7]. Since then, numerous types of problems are studied in location theory. The parallel advancement can be observed in methodologies for solving location problems (see Ghosh [8] and Daskin [9] for a detailed review on facility location literature). One main stream of research in location theory is the covering location problems, where customers within a certain distance (or travel time) to the facilities can be served. In this chapter we review the covering facility location literature by mentioning the major problem types. The BRICP problem we propose is relevant to that review in the sense that both the system planner's and the attacker's problems are special forms of covering problems with the objective functions maximizing and minimizing the total service coverage, respectively.

The covering concept was first introduced by Toregas [10]. As opposed to the classical location problems, he suggested the criterion of maximum travel time or distance between a facility and a customer for satisfaction of the customer demand. The criterion is well applicable in emergency services networks, where there is no regular travel between facilities and all customer sites in the network, but the facilities should be located so that the system is capable to respond to possible emergency situations in any customer site within a certain time. The location set covering problem (LSCP) locates the minimum number of facilities that satisfies a pre-specified upper limit on travel time to all customer sites. The second type of problem proposed in the related literature is the Maximal Covering Location Problem (MCLP) (Church and Revelle [11]). In MCLP, the mandatory coverage of customers in the LSCP is relaxed, whereas restriction on number of facilities is defined. A limited number of facilities are located so as to maximize the total population who can be served in a specified service distance or time. This approach widens the applicability of covering problems to the typical real life cases where there is a budget limit on locating facilities and non-perfect coverage

of the network can be tolerated.

Note that in these two initial covering location studies, there are certain assumptions:

- Unlimited capacity: The only requirement for a customer to get service from a facility is to be in the coverage range of that facility. Accordingly, solutions to these problems may involve facilities serving too few customers as well as facilities serving too many customers. In real life, however, facilities serving too many customers frequently fail in providing service due to congestion while facilities serving too few customers remain undesirably idle. In the problems mentioned above, all facilities are identical.
- All or nothing coverage: Customer demand is fully covered if there is a facility within a maximum covering distance or it is not covered at all. This definition of coverage is unrealistic due to the abrupt fall in coverage.
- Discrete potential facility locations: The optimal location configuration is sought within a finite set of potential facility locations. This approach is proper in optimization of emergency service facility locations. However there are real life problems such as locating communication tower or designing a sensor network where potential facility locations can realistically be considered as the entire plane. Discrete potential facility locations assumption is a limitation on applicability of the covering models in these areas.
- Fixed coverage range: The coverage range is a pre-specified constant for all facilities. Specifying the coverage range as a parameter, however, is problematic since there is no strict concrete range in real life. The range is also dependent on the facility specifications, larger facilities with longer ranges and smaller facilities with shorter ranges.

The recent research in covering location problems has mainly focused on relaxing these assumptions for the purpose of formulating more realistic problems. Accordingly we review the following literature relaxing each of these assumptions.

2.1.1. Extensions Relaxing Capacity Assumptions

The first study in the literature which takes the capacity limitation into consideration is an extension of earlier covering problems (Current and Storbeck [12]). They present capacitated versions of both MCLP and LSCP. In their formulations, each customer is assigned to a facility, except the ones which are not covered by any facility. The total demand of the customers assigned to a facility, is limited by a capacity constraint. This leads to a more realistic model where customer demand coverage is conditional on facility capacity limitations in addition to being within the maximum coverage distance to a facility.

In a following study, Pirkul and Schilling [13] include the assignment of uncovered customer demand to facilities. The authors suggest that in certain systems like public services, the responsibility of a given facility is not limited to providing service to customers within the coverage range, and accordingly, the uncovered demand should be counted as part of that facility's workload. In their formulation of capacitated MCLP (CMCLP) they impose the assignment of both covered and uncovered demand in capacity considerations while including only the covered demand in the maximization objective. To push the assignment of uncovered demand to closer facilities, the maximum service distance is included in the objective function with a constant weight.

2.1.2. Extensions Relaxing All or Nothing Coverage Assumption

Berman and Krass [14] address this issue by identifying a coverage rule represented by a non-increasing step function. In the Generalized Maximum Coverage Location Problem (GMCLP) they present, coverage diminishes stepwise in certain ranges instead of an abrupt fall from full to no coverage in MCLP. They develop several solution methodologies, of which a greedy heuristic performs the best in finding near-optimal solutions. Berman et al. [15] further extend the MCLP introducing the gradual covering concept. The authors define two coverage radii l and u , $l \leq u$. Customer demand is covered fully if there is a facility located within distance of l units to the demand point, and not covered if the closest facility is further than u units

away from the demand point. If the distance to the closest facility is between l and u , then the coverage of the demand point is determined by a coverage decay function which returns the partial coverage depending on the distance to the facility. They formulate an MILP model for their Gradual Coverage Decay Problem (GDCP). The authors suggest a greedy heuristic, LP-relaxation and branch and bound approaches for solution attempts to the GDCP while providing an alternative formulation with smaller number of variables. Karasakal et al. [16] investigate a similar problem but call it partial coverage.

2.1.3. Extensions Relaxing Discrete Potential Facility Locations Assumption

This extension to covering location problem represents a more commonly studied area of research, initiated by Mehrez and Stulman[17]. In this study both MCLP and LSCP are extended to planar facility location. The solution methodology they propose involves the reduction of the continuous solution space to dominant small finite set of points, among which the optimal solution is found via exhaustive enumeration or dynamic programming. The subsequent works in this area are many, which are excluded from our concern of the literature research with the aim of being focused towards the model we study.

2.1.4. Extensions Relaxing Fixed Coverage Range Assumption

The fixed coverage radius assumption is relaxed in a recent study by Berman et al. [18]. In this study, the coverage radius of each facility is controllable and can be increased in return for increased cost of opening that facility. The variable radius problem they develop minimizes the total cost while satisfying the coverage of all customer demand. They propose two formulations for the problem regarding the two cases: either there is a finite set of potential facility locations available (discrete case) or facilities can be located anywhere in the plane (planar case).

2.2. Interdiction Problems

Another issue in network design problems is the possibility of failure of a network component (i.e. arcs or facilities). The failure of a component in the network may cause a disorder in the normal functioning of the system resulting in the deterioration of the performance due to service disruptions or excessive transportation costs. The effect of such a disorder, on the other hand, can be minimized when the possibility of failure is taken into account in the design phase of the network. Network reliability and network interdiction models are both based on this motivation. The former considers the failures caused by natural catastrophic events, labor actions or other similar factors. Network reliability models mainly aim to design a network so that a combination of two cost components is minimized: the total regular operating costs and expected extra costs due to an unintentional failure. The typical examples of reliability models can be found in [19-22].

Unlike the reliability studies, interdiction models explore intentional causes of failures to the system such as terrorist attacks. Intentional attacks differ from natural disasters in that a terrorist intelligently attacks the system so as to cause the maximum damage. Therefore, interdiction models include an additional optimization for determining the terrorist's action. There are also reliability models where the worst case analysis is performed for the unintentional causes of failure, which requires incorporating an additional optimization model. Clearly, when the resources (i.e. number of facilities or arcs to be destroyed) are the same, a natural cause inducing a worst case loss is equivalent to a terrorist aiming the maximum harm. In this respect, the worst case reliability models are closely related to the interdiction literature. With the intention of reviewing the relevant literature to our BRICP problem, we will cite the worst case reliability models in addition to the interdiction models in this chapter. The reader is referred to Synder et al. [23] for a comprehensive survey on network reliability models.

In a typical interdiction model, there is an optimization problem from the attacker's perspective to maximize the deterioration in the system performance. The

result of this optimization gives the vulnerabilities of the system which is critical information for the system owner (or system planner) to design or modify the system towards robustness to the attacks. In later interdiction studies, attacker's interdiction problem is integrated into the problem of the system planner (e.g., designing a network, modifying an existing network) as a lower level problem. This leads to a bilevel structure, which we will call bilevel interdiction problem (BIP). BIPs provide an optimal solution to the system planner's problem, which ensures robustness to optimal interdiction scenarios.

The first interdiction problem was introduced by Wollmer [24] in which n arcs are removed from the network so that the maximum flow from the source to the sink was reduced most compared to the original flow. The motivation behind this study was to determine how sensitive a transportation network is to having its roads closed. In a more recent study [25], the bilevel version of this arc interdiction model is proposed. Here the network designer constructs a network to maximize the total profit from pre-interdiction and post interdiction flows (fixed network design costs deducted) while the attacker interdicts a set of constructed arcs in order to reduce the profit of the designer. The authors suggest three different attacker profiles following three different strategies; (1) attacking the arc with largest capacity, (2) attacking the arc with largest pre-interdiction flow and (3) attacking optimally to minimize the post interdiction profits. The total profits for the designer are similar against attacker profiles (1) and (2), both of which are dominated by the attacker profile (3) which results in a significant decrease in the designer's profit. The analysis on different attacker strategies shows that the enemy's ability to optimally solve the interdiction problem can be a crucial factor in effectively reducing the network designer's profit. A similar problem was studied by Berman et al. [26]. In their bilevel arc interdiction problem, the attacker destroys the most damaging link with the aim of maximizing the decrease in coverage, following the leader who locates p facilities for maximum total coverage of customer demand. Simulated annealing, tabu search and an ascent algorithm were proposed to solve the leader's problem while the follower's problem is solved through a special algorithm. In addition to models mentioned above, arc interdiction problems are studied also in [27] and [28].

A less commonly studied area in the early interdiction literature is the facility interdiction problems. As opposed to the arc interdiction models, in facility interdiction problems, facilities are potential targets for attackers. The first facility interdiction problem is studied in Church et al [29] where the authors propose two problems; r interdiction median problem (RIM) and r interdiction covering problem (RIC). The attacker aims to maximize the total demand weighted distance in RIM and to minimize the total coverage in RIC. Both problems find the subset of existing facilities which when removed serves the attacker's aim the most. In other words, RIM and RIC identify the most critical facilities in the network. These two single-level models are solved through a commercial MILP solver.

RIM was extended to include the ability of fortification of vulnerable facilities by Church et al. [30]. The interdiction median problem with fortification (IMF) identifies the best fortification plan on an existing p -median network so that the decrease in system efficiency upon the most disruptive attack to unprotected facilities is minimized. There are p existing facilities in the network, q of which are fortified by the leader and another r of which are attacked by the attacker. The defender's objective is to minimize the total post-attack demand weighted distance traveled to serve n demand nodes from the surviving $p - r$ facilities, while the attacker's objective is the reverse of that of the leader by interdicting r facilities. The attacker's optimization is incorporated into the problem as a constraint, which includes explicit enumeration of all interdiction patterns. A problem having 20 facilities is solved through a general-purpose MILP solver.

In order to improve the solution time of the IMF, Scaparra and Church proposes two alterations to the problem in [31] and [32]. The former includes a reformulation of IMF as a maximal covering problem with precedence constraints (MCPC). The authors devise an effective solution technique for the reformulated problem which produces near-optimal solutions and upper and lower bounds to the problem which reduces the problem to a size to be solved via a general-purpose MILP solver. The solution time to MCPC is sensitive to r , however, since the method requires enumeration of all interdiction patterns. The second alteration [32] is the bilevel version of the IMF

which is called r -interdiction median problem with fortification (RIMF). In RIMF, the fortification problem (defender's problem) is the upper level problem and the interdiction problem (the attacker's problem) is the lower level problem which is the RIC. The authors solve the bilevel problem through a tree search algorithm. Except for cases where r is small, the tree search algorithm outperforms the MCPC solution.

Aksen et al. [33] further extend the RIMF problem, replacing the limit on the number of facilities to be fortified with a budget constraint and incorporating capacity expansion to the model. Termed as budget constrained r -interdiction median problem with capacity expansion (BCRIMF-CE), this problem allows the defender to fortify facilities based on their predetermined fortification costs and a budget limit. Furthermore, in BCRIMF-CE, the capacity of facilities can be expanded with a unit cost to the defender to accommodate the post-attack demand of customers who were serviced by some other interdicted facility prior to the interdiction. The authors solve the BCRIMF-CE problem through an implicit enumeration algorithm which depends on a general purpose MILP solver to evaluate the attacker's response.

Extensions to the RIC model, being indeed more relevant to our work, are less widely studied in the literature. O'Hanley et al. [34] and O'Hanley et al. [35] utilize RIC formulations in studying biological conservation. In the first study, they develop a model to locate natural reserve sites with the aim of maximizing the minimum coverage level following the worst case loss of a subset of reserves. This extension to RIC is a bilevel problem, where the upper level problem is locating the reserve sites taking the lower level problem into consideration, which minimizes the coverage level through destroying reserve sites. As is the case in all worst case studies mentioned, the follower is a virtual player representing the worst case scenario rather than an actual attacker. O'Hanley et al. [35] focus on a variant of the problem where the location of the natural reserve sites are determined in the upper level problem so as to minimize loss of species under the worst case scenario which is established in the lower level problem with the maximization of loss of species. Both biological conservation problems are solved through bilevel decomposition method, where the upper level and the lower level problems become the master and subproblem, respectively. In this method, the

master problem and the subproblem are solved sequentially. For each matching feasible master and subproblem solutions, a number of inequalities are generated and appended to the master problem to move it towards better solutions.

A recent extension to RIC model (O’Hanley et al. [36]) is the maximum covering location interdiction problem (MCLIP) which is a coverage type service network design problem as is the case in our BRICP problem. In MCLIP, the defender locates p facilities so as to maximize the weighted combination of the demand weighted coverage prior to attack and the minimum post-interdiction demand weighted coverage where the interdiction refers to the loss of r facilities corresponding to the worst case scenario. Demand points within a distance of a predetermined fixed radius to a facility are covered while there is no capacity restriction on demand covered by a facility. Two formulations of the MCLIP are proposed in the paper. First, the authors formulate a MIP model, which can be solved through a general-purpose MILP solver. However, the model itself is exponential in size since it includes a set of variables and constraints for ever possible interdiction pattern. The second and more efficient formulation is a bilevel formulation of the problem with the defender and the attacker being the leader and the follower players, respectively. The objective function of the upper level problem is the maximization of the weighted combination of post attack and pre attack coverage while the second level problem is the minimization of the post-attack coverage. A bilevel decomposition algorithm is used for the solution.

3. PROBLEM FORMULATION

In this chapter we present our mathematical formulation to the BRICP problem. First, some concepts and methodologies are mentioned. Then, our mathematical model is thoroughly explained.

3.1. Bilevel Programming

A Bilevel programming (BP) problem corresponds to a Stackelberg game [37] where the problem at two levels represent two players, each with different interests (i.e. objective function) and limitations (i.e. constraints). As opposed to two-person zero sum games, Stackelberg games are sequential games. That is, in two players case, one of the players is the leader and acts according to a strategy so that including the effects of the second player's (the follower) reaction, the final outcome is the best for the leader. In a zero-sum game, however, the players act simultaneously neither being leader or follower where the solution is found via game theoretic solution procedures.

In a BP formulation, second player's problem appears in the set of constraints of leader's problem called the lower level problem (LLP) and the upper level problem (ULP), respectively. The decision variables corresponding the leader's action are ULP variables. To obtain the reaction of the attacker, ULP variables are set constant to represent a feasible action of the leader and LLP is solved. Given the ULP variables as constant, the solution to LLP determines LLP variables representing the optimal reaction of the attacker. The overall solution to the bilevel problem is the one giving the best ULP objective function value among the set of all feasible pairs of leader's action and attacker's optimal reaction. Both ULP and LLP can be linear, integer, mixed integer or nonlinear programming problems depending on the type of ULP and LLP variables.

We formulate our BRICP problem as a BP model which best fits into the nature of the interplay between the system planner and the attacker.

3.2. Gradual Coverage

In classical threshold coverage notion, introduced in LSCP [10] and MCLP [11] problems, the demand points within a predetermined critical distance to a facility are fully covered whereas beyond that critical distance the demand is not covered at all. This leads to inconsistent cases such as the following. Suppose the critical distance is 10 kilometers in a service facility location problem and there are two demand points A and B, which are close to each other and 9.9 and 10.1 kilometers away to the closest facility, respectively. Unrealistically, the demand at point A is fully covered while the demand at point B is not covered at all under critical distance coverage assumption. Furthermore, the choice of the critical distance is deceptive since indeed there is no such single distance definition in real life and the solution to the coverage location problem is likely to be sensitive to the critical distance.

In our BRICP problem, as opposed to the threshold coverage distance assumption, we define service coverage according to the gradual coverage concept introduced by Berman et al. [15]. Under gradual coverage, the coverage of a demand point is full when the distance between the demand point and a facility is smaller than a lower limit R_ℓ , and it decreases gradually to zero until the distance becomes larger than an upper limit $R_u > R_\ell$. The gradual coverage idea eliminates the assumption of “all or nothing” property of the classical threshold coverage notion. We adopt a linear decay function between R_ℓ and R_u . That is, when the distance between customer zone k and the facility at site i is denoted by e_{ki} , the level of partial service coverage a_{ki} is given as

$$a_{ki} = \begin{cases} 1 & \text{if } e_{ki} \leq R_\ell \\ \frac{R_u - e_{ki}}{R_u - R_\ell} & \text{if } R_\ell < e_{ki} \leq R_u \\ 0 & \text{if } e_{ki} > R_u \end{cases}$$

Clearly, $a_{ki} \in [0, 1]$. Note that e_{ki} refers to the Euclidian distance between customer zone k and facility i .

We assume that when a customer zone is gradually (or partially) covered by multiple facilities, the final coverage can be obtained by taking the maximum of the partial coverage value of each facility. This implies that the resulting coverage is achieved by the closest facility to a customer zone when R_ℓ and R_u are the same for each facility. It is in contrast with another recently developed cooperative coverage model, where all facilities contribute to the coverage of each customer zone [14]. The cooperative coverage notion is applicable in locating facilities like warning sirens and communication towers where the signals received at a customer node is the sum of the signals transmitted from several sources. Our individual coverage approach, on the other hand, is more suitable in locating public service facilities, in which the service provided does not follow an additive nature.

3.3. Bilevel r -Interdiction Covering Problem With Protection

In our problem setting we assume that there exist $N_1 = |\mathcal{N}_1|$ operational existing facilities that provide service to customers residing at $\mathcal{K}$ different zones with demand d_k . Due to the existence of an attacker, whose aim is to minimize the service coverage by destroying r unprotected facilities, the system planner wants to maximize the post-attack service coverage by relocating some of the existing facilities to any of the $N_2 = |\mathcal{N}_2|$ potential facility sites, opening new facilities at some of these potential sites, and protecting some of the existing and new facilities so that they become invulnerable to any attack. There is a fixed cost associated with each of these decisions. It is required that the total cost cannot exceed a prespecified amount of budget b . We assume that the attacker cannot attack a protected facility and an unprotected facility is fully out-of-service when attacked. The coverage of the customer nodes are determined by means of the individual gradual coverage rule. We assume that there is no capacity limitation for the facilities; the only condition for a facility to full (partial) service to a customer is being within the full (partial) coverage distance to this customer.

Here we relax the the assumption of the approach “locate p facilities, protect q of them”. Instead, we adopt another approach by explicitly considering the tradeoff between protection and location for the system planner so that the number of facilities

to be located and the number of facilities to be protected themselves become a decision for the system planner. As a result, the system planner may follow the extreme strategies of opening more facilities without protecting any of them or opening fewer facilities all of which are protected against attack. In our problem setting, facility relocation is also integrated to this tradeoff. Note that, there is no such tradeoff in the attacker's solution space and accordingly we do not adopt the budget limit approach in the attacker's problem, where we assume maximum of r attacks.

In our BP formulation to the problem, ULP corresponds to the system planner who is the leader of the game, while the LLP belongs to the attacker who is the follower. It is assumed that the system planner has complete information about the attacker's LLP, i.e., the objective function and the constraints of the LLP. Similarly, the attacker is assumed to be fully knowledgeable about the locations of customers together with the demands, locations of the facilities and protection status of each facility so that he minimizes the total customer demand covered without wasting his resources by attacking protected facilities. Prior to choosing which action to take, the system planner takes into account the attacker's best reaction. In fact, given the solution to ULP, the LLP problem is the foreknown RIC problem with two modifications: the adoption of gradual coverage concept instead of the threshold coverage and the presence of protected facilities which cannot be interdicted.

Before presenting our BP model with binary variables at both levels, we give the notation used in the formulation.

Index Sets:

$$\begin{aligned}
\mathcal{K} &= \text{set of customer zones} \\
\mathcal{N}_1 &= \text{set of existing facility sites} \\
\mathcal{N}_2 &= \text{set of potential facility sites} \\
\mathcal{N} = \mathcal{N}_1 \cup \mathcal{N}_2 &= \text{set of existing and potential facility sites}
\end{aligned}$$

Parameters:

$$\begin{aligned}
d_k &= \text{demand of customer zone } k \in \mathcal{K} \\
f_i &= \text{fixed cost of opening a new facility at site } i \in \mathcal{N}_2 \\
g_{ij} &= \text{cost of relocating an existing facility from site } i \in \mathcal{N}_1 \text{ to site } j \in \mathcal{N}_2 \\
h_i &= \text{fixed cost of protecting a facility at site } i \in \mathcal{N} \\
b &= \text{available budget of the system planner} \\
r &= \text{the maximum number of facilities the attacker can destroy} \\
a_{ki} &= \text{partial coverage of customer zone } k \in \mathcal{K} \text{ by the facility at site } i \in \mathcal{N}
\end{aligned}$$

Decision Variables:

$$\begin{aligned}
Y_i &= \begin{cases} 1 & \text{if there is a facility at site } i \in \mathcal{N} \\ 0 & \text{otherwise} \end{cases} \\
U_{ij} &= \begin{cases} 1 & \text{if an existing facility at site } i \in \mathcal{N}_1 \text{ is relocated to site } j \in \mathcal{N}_2 \\ 0 & \text{otherwise} \end{cases} \\
S_i &= \begin{cases} 1 & \text{if facility at site } i \in \mathcal{N} \text{ is destroyed by the attacker} \\ 0 & \text{otherwise} \end{cases} \\
P_i &= \begin{cases} 1 & \text{if facility at site } i \in \mathcal{N} \text{ is protected} \\ 0 & \text{otherwise} \end{cases} \\
W_k &= \text{fraction of customer zone } k \text{'s demand served after the attack}
\end{aligned}$$

Note that if there is an existing facility at site $i \in \mathcal{N}_1$, or a new facility is opened at site $i \in \mathcal{N}_2$ or an existing facility is relocated to site $i \in \mathcal{N}_2$, then the corresponding $Y_i = 1$. Consequently, the BP model is given as follows:

$$\max_{\mathbf{Y}, \mathbf{U}, \mathbf{P}} z = \sum_{k \in \mathcal{K}} d_k W_k \quad (3.1)$$

s.t.

$$(1 - Y_i) = \sum_{j \in \mathcal{N}_2} U_{ij} \quad i \in \mathcal{N}_1 \quad (3.2)$$

$$\sum_{i \in \mathcal{N}_1} U_{ij} \leq Y_j \quad j \in \mathcal{N}_2 \quad (3.3)$$

$$\sum_{i \in \mathcal{N}_2} f_i(Y_i - \sum_{j \in \mathcal{N}_1} U_{ji}) + \sum_{i \in \mathcal{N}_1} \sum_{j \in \mathcal{N}_2} g_{ij} U_{ij} + \sum_{i \in \mathcal{N}} h_i P_i \leq b \quad (3.4)$$

$$P_i \leq Y_i \quad i \in \mathcal{N} \quad (3.5)$$

$$Y_i, P_i \in \{0, 1\} \quad i \in \mathcal{N} \quad (3.6)$$

$$U_{ij} \in \{0, 1\} \quad i \in \mathcal{N}_1, j \in \mathcal{N}_2 \quad (3.7)$$

$$\min_{\mathbf{S}, \mathbf{W}} \sum_{k \in \mathcal{K}} d_k W_k \quad (3.8)$$

s.t.

$$a_{ki}(Y_i - S_i) \leq W_k \quad i \in \mathcal{N}, k \in \mathcal{K} \quad (3.9)$$

$$S_i \leq Y_i - P_i \quad i \in \mathcal{N} \quad (3.10)$$

$$\sum_{i \in \mathcal{N}} S_i \leq r \quad (3.11)$$

$$0 \leq W_k \leq 1 \quad k \in \mathcal{K} \quad (3.12)$$

$$S_i \in \{0, 1\} \quad i \in \mathcal{N} \quad (3.13)$$

In the above formulation (3.1–3.7) represent the system planner's ULP, while (3.8–3.13) constitute the attackers LLP. In the ULP, the objective function (3.1) is the maximization of the total post-attack service coverage. Constraints (3.2) and (3.3) establish the relationship between location variables Y_i and relocation variables U_{ij} . In

particular, the first set of constraints guarantee that if a facility is relocated from site $i \in \mathcal{N}_1$ to any other site $j \in \mathcal{N}_2$, then the location variable Y_i must be equal to zero (i.e. no facility left at location i). The second set of constraints state that if a facility is relocated to site $j \in \mathcal{N}_2$, then $Y_j = 1$ must hold true. Note that, in addition to the relocation case, these constraints are valid also in the following cases:

- i. When no facility is relocated to site $j \in \mathcal{N}_2$ and no new facility is opened at site $j \in \mathcal{N}_2$. Since there is no relocation to site j , the left-hand side of (3.3) is zero while the right-hand side is zero due to no facility opening at site j .
- ii. When no facility is relocated to site $j \in \mathcal{N}_2$ and a new facility is opened at site $j \in \mathcal{N}_2$. Since there is no relocation to site j , the left-hand side of (3.3) is zero while the right-hand side is one due to opening of a new facility at site j .
- iii. When the existing facility at site $i \in \mathcal{N}_1$ is not relocated to any potential site. Since there is no relocation from site i , the right-hand side of (3.2) is zero while the left-hand side is also zero as there is an existing facility at site i .

Clearly, the relocation of multiple existing facilities to a certain potential facility site is prohibited via constraints (3.3). Constraints (3.4) guarantees that only an opened facility can be protected. Constraint (3.5) makes sure that the total expenditure of the system planner does not exceed the available budget. The first term on the left-hand side of this inequality represents the cost of new facility openings, while the second and third terms constitute the cost of the relocations and protections, respectively. Binary restrictions are imposed on location variables Y_i and protection variables P_i in constraints (3.6), and on relocation variables U_{ij} in constraints (3.7).

The variables Y_i , U_{ij} and P_i are decision variables of ULP which represent the action taken by the system planner. They act as input parameters for the LLP. In the LLP, the objective function (3.8) is the same as that of the system planner with the exception of the sense of optimization. Namely, the attacker aims at reducing the total post-attack service coverage as much as possible. Constraints (3.9) ensure the coverage of each customer zone by the closest open facility that is not interdicted by the attacker. Note that if customer zone k can be served from different facilities, the final

coverage represented by variable W_k is set to the maximum of the partial coverage due to each facility. In other words, this customer zone is served from the closest facility. Constraints (3.10) are logical conditions, which prevent the attacker from interdicting facilities either not existing at all or opened but also protected by the system planner. Constraint (3.11) states that the attacker can interdict at most r facilities. Constraints (3.12) ensure that coverage variables W_k are between zero and one, and constraints (3.13) are binary restrictions on interdiction variables S_i .

4. SOLUTION PROCEDURES

As shown by Moore and Bard [38] mixed-integer BP problems are NP-hard. Since there are binary variables in both the ULP and LLP problems in our model, it is clear that we need efficient heuristic methods to produce solutions of high quality. To this end, we devise a tabu search heuristic (TSH), which consists of the search for the best facility location, relocation, and protection decisions of the system planner represented by the $\mathbf{Y}$, $\mathbf{U}$ and $\mathbf{P}$ variables in the ULP. The solutions to the LLP, on the other hand, obtained through a general purpose MIP solver like CPLEX. To the solve BRICP problem, we apply TSH twofold. The TSH algorithm, with the help of two independent runs, searches a larger portion of the entire solution space resulting in better solution. To assess the quality of the solutions produced by the TSH, we develop a binary enumeration tree (BET) algorithm, which is guaranteed to find an optimal solution to any given instance. As is the case in the heuristic approach, CPLEX is employed to find the attacker's response and compute the corresponding service coverage in BET. In this chapter we first present the details of the TSH which is followed by the explanation of the BET.

4.1. Two Step Tabu Search Heuristic Algorithm

In a typical tabu search metaheuristic algorithm, the search starts from an *initial solution* and progresses iteratively. In each *iteration*, a set of neighboring solutions, called *neighborhood*, is generated around the *current solution* through some move operators. Among the neighboring solutions, the one with the best objective value is selected as the current solution for the next iteration. The exception to this selection rule is imposed by a *tabu list*. The tabu list consists of the moves which have generated the recent visited solutions. The neighboring solutions generated by the *tabu moves* present in the tabu list are ignored in the selection of the next current solution with the aim of avoiding the revisiting the recent solutions which results in *cycling*. However, a neighbor which satisfies the *aspiration criterion* is selected as the current solution regardless of the tabu criteria. *Tabu tenure* corresponds to the number of iterations

a tabu move will reside in the tabu list. The overall best feasible solution is stored as the incumbent solution and it is updated whenever the objective value for a new feasible current solution is higher. Upon the satisfaction of the *termination criterion*, the incumbent solution represents the best solution to the problem.

Our application of the TSH consists of two steps: first run (FR) and second run (SR). Each of the FR and SR in our TSH algorithm follows typical tabu search procedures including some minor exceptions. Each run searches for the best facility location, relocation and protection decisions of the system planner represented by the $\mathbf{Y}$, $\mathbf{U}$ and $\mathbf{P}$ variables in the ULP formulation. Once the values of these variables are fixed, the attacker's LLP becomes a small size MIP problem which can be solved to optimality by employing a general-purpose solver like CPLEX. Consequently, for each combination of the $\mathbf{Y}$, $\mathbf{U}$, and $\mathbf{P}$ variables set by the TSH, the attacker's optimal interdiction scheme is found along with the corresponding objective value of both the system planner and the attacker, i.e., the post-attack service coverage.

It should be noted that, due to our problem definition, the following must hold for any solution during the search. An existing facility can only be relocated to a potential facility site; it cannot be relocated to another existing facility site, even though this site is empty due to another relocation. An existing facility is never closed. New facilities can only be opened at potential facility sites but not at existing facility sites. Any existing or newly opened facility can be protected regardless of their locations. However, we also allow the infeasible solutions, where the budget of the system planner is overrun, to be considered within the search, in an effort to increase the search diversification. Consequently, we define search space (SS) as the set of all combinations of the binary alternatives of the $\mathbf{Y}$, $\mathbf{U}$, and $\mathbf{P}$ variables which constitute a feasible solution subject to the constraints of the ULP with the exception that a certain amount of budget overrun is tolerated. Clearly, an optimal solution to the BRICP problem instance is always within the corresponding SS.

The motivation behind the proposed TSH is to approximate the optimal solution faster in the SS using two independent search procedures. Based on the optimal results

to the BRICP problem obtained by the BET algorithm, two extreme patterns are observed in which the optimal solution for the system planner exists. For a given budget, the system planner either opens fewer new facilities and protects the majority (usually all) of the new and existing facilities, or opens more facilities and uses protection option for only the minority (usually none) of the new or existing facilities. The optimal solution to the problem is within one of these two extreme patterns in the SS. Hence, the search by TSH should cover both of these patterns, in order to reach solutions of high quality. Our observations, show that TSH requires a considerable amount of iterations to reach the other extreme once it reaches one of the two extremes because of the low objective values of the solutions in the region lying between these two extreme patterns of the SS. A meaningful move operator, on the other hand, cannot be defined to provide a jump between these two regions (e.g., drop all protections and add facilities with an increased budget, or close some of the open facilities so that the budget is high enough for protecting all the remaining ones). Instead, we propose two sequential tabu runs, FR and SR, each manipulated towards one of the two extreme patterns in SS. This helps to cover both extremes in the SS and the best solution among the solutions of these two searches is a final solution of higher quality. The FR of TSH, without any additional restrictions on the move operators, tends towards the extreme where the majority of the facilities are protected. To manipulate the SR of TSH towards the other extreme with minor protection, we prohibit the moves which add protection to the facilities during the initial iterations of the search. Both FR and SR start from the same initial solution. Once both runs are terminated, the solution which gives the largest coverage becomes the final solution to the TSH.

We find it appropriate to suggest a metaphor in emphasizing the motivation behind the use of TSH. Suppose, a certain region represents the SS where each point in that region refers to a solution to the system planner's problem and the altitude of that point is the objective value (i.e. total coverage). The search performed in this region, is a mechanism which moves iteratively from a current point to a new current point by analyzing a certain neighborhood so as to explore high altitude points in the region. In this context, the two extreme patterns can be represented by two hills in that region, one of which consisting of the highest point in the region (i.e. the optimal solution).

These hills are separated by a deep valley which symbolizes the low objective values for the solutions lying between these two extremes. Then, our proposal can be explained as follows within this illustration. Rather than expecting one mechanism to search both these extremes which requires passing across this valley several times during the search, we propose two search mechanisms each with a task of exploring one of these extreme patterns. This eliminates the excessive amount of time (i.e. iterations) spent in passing across the valley.

In this chapter the TSH is explained in detail. Since the two runs of the TSH are not different except the prohibition during the initial iterations, a single explanation of the TSH procedure is provided and the distinctions are mentioned when needed.

4.1.1. Parameters Definitions for TSH

We define the parameters for our tabu search heuristic in Table 4.1.

The solution to the system planner's problem consists of relocation and protection decisions in addition to location decisions. Hence, in contrast to the traditional tabu search heuristics, the search in our TSH consists of simultaneous location, relocation and protection variables. Accordingly, current solution (H), neighbor solution (H') and incumbent solution (inc_sol) in the TSH need to hold the corresponding information. We develop the following encoding technique in recording this information.

The vectors H , H' and Inc_sol have $nex + nfac$ entries. The first nex entries of these vectors represent the existing facility sites while the rest represent the potential facility sites. A new facility is coded as "1". That is, any entry of "1" represents an open facility at that site. Each existing facility is coded separately as $2, 3, \dots, nex+1$. Note that in case of multiple existing facilities, separate coding is critical in terms of identifying the existing facilities. When an existing facility is relocated to a potential facility site, then the corresponding entry is set to one of the values $2, 3, \dots, nex+1$. In addition, the protection properties of the existing/new facilities are kept in the signs of the entries of H . Any entry being equal to "-1" refers to a protected newly opened

Table 4.1. Parameters definition for TSH algorithm

Parameters	Definition
<i>MaxIter</i>	Maximum number of iterations
<i>MaxNI</i>	Maximum number of non-improving iterations
<i>nex</i>	Number of existing facilities
<i>npot</i>	Number of potential facilities
<i>H</i>	Vector representing the current solution It holds the location, relocation and protection information.
<i>H'</i>	Vector representing a neighboring solution to the current solution <i>H</i> .
<i>Inc_sol</i>	Vector in which the incumbent solution is recorded together with the objective value
<i>TabuList</i>	Array holding all the visited solutions
<i>iter</i>	Order of the current iteration (i.e. number of previous iterations + 1)
<i>NIter</i>	Number of consecutive non-improving iterations
<i>NeighList</i>	Array holding neighbor solutions generated in current iteration
<i>pw</i>	Penalty weight

facility where “-2” refers to an existing facility being protected. Some examples are provided in Figure 4.1.1, where $nex=2$ and $npot=10$. The first vector refers to the initial solution, where all existing facilities are located in their original location and there is no newly opened facility. In the next two vectors one of the existing facilities is relocated to a potential facility site and there are two newly opened facilities. The latter of these two involves the protection of one of the existing and newly open facilities.

4.1.2. Initial Solution and Neighborhood Generation

Our TSH starts from a solution, where all existing facilities are at their original sites and there are no new facilities. The initial cost to the system planner is zero while a certain initial coverage is provided by the existing facilities in the network. This initial solution is used in both of the FR and SR of TSH, which allows the manipulation of the

2	3	0	0	0	0	0	0	0	0	0	0
2	0	1	0	0	1	0	0	0	3	0	0
-2	0	1	0	0	-1	0	0	0	3	0	0

Figure 4.1. Examples of H vector

search direction towards the two extremes of SS. Neighborhood generation is performed with the help of the following types of moves:

- 1-Add: A new unprotected facility is opened at one empty potential facility site. This move generates as many neighbors as the number of empty potential facility locations.
- 1-Drop: A newly opened protected/unprotected facility is closed. This move generates as many neighbors as the number of newly opened facilities. This move is not applied to existing facilities as they cannot be closed.
- 1-Swap: A new or existing protected/unprotected facility is relocated. Note that an existing facility can be relocated to its original location or to a potential site while a newly opened facility can be relocated to a potential site only. The number of neighbors generated is approximately equal to the multiplication of the total number of open facilities and the number of facility sites.
- 1-Flip: The protection status of a new/existing facility is switched on or off. The number of neighbors generated is equal to the number of open and existing facilities.
- 1-Exchange: The site of an existing facility is exchanged with the site of an existing/new facility. Clearly, neither of the existing facilities should be at its original site for this exchange to be feasible. The approximate number of neighbors generated is equal to the product of the number of existing facilities and the total number of existing and new facilities.
- Swap-Prot: Two facilities with opposite protection modes will swap their modes. Swap-Prot generates every possible swap within existing/new facilities regardless

of their locations. The number of neighbors can be approximated by the product of the number of unprotected open facilities and the number of protected open facilities.

For the FR of TSH, there is no limitation on the moves. During the initial iterations, neighbors generated through the 1-Add and the 1-Flip moves are selected as the best neighbor sequentially, which results in opening of protected facilities until the budget limit is reached. Clearly, the FR searches the extreme pattern in SS where the majority of the facilities are protected. On the other hand, the SR of TSH is directed towards the other extreme pattern in SS simply by prohibiting the neighbor generation through the 1-Flip move in initial iterations. This results in a search in the extreme pattern where protection is rarely used. Note that once the limitation is removed after the initial iterations, the search is free to direct to solutions out of the extreme regions. Hence, when such extremes are not rigid or the best solution to the ULP is an intermediate solution between the two extreme patterns, the tabu search is free to tend towards solutions out of these extremes.

4.1.3. Evaluation of the Neighbors and Tabu Criterion

Our TSH utilizes the strategic oscillation concept, i.e., we admit also those solutions in which the total cost violates the budget constraint. We refer to these as budget infeasible solutions. In the calculation of such a solution's objective value, we multiply the budget overrun (the amount by which the total cost exceeds the available budget) with a dynamically changing penalty weight (pw), and subtract the resulting infeasibility from the service coverage. (pw) is updated in each iteration based on the percentage of the infeasible solutions among the last 10 best neighboring solutions. In fact, the main rationale in choosing the value of the penalty parameter mentioned above is to ensure that feasible and infeasible solutions have approximately the same likelihood to be the best neighboring solution and to qualify as the next current solution. Hence, at each iteration, when the fraction of infeasible solutions among recently visited solutions is higher than 50%, the pw is increased, which causes the penalty deduction for infeasibility to be higher and consequently increases the possibility of choosing feasible

solutions as the best neighboring solution during the following iterations. Similarly, pw is decreased when the fraction of infeasible solutions is lower than 50%, so as to encourage the selection of infeasible solutions. This strategic oscillation approach helps to the diversification of the TSH in the exploration of the solution space.

A neighboring solution (H') to the current solution (H) generated through the moves defined before represents a solution to the ULP. To obtain the final coverage value corresponding this solution, CPLEX is called to solve the LLP. This provides the objective function value for the LLP and ULP. Each neighboring solution is inserted to the *NeighList* in form of the vector H' together with the penalty deducted from the objective function value.

To avoid cycling we use a *TabuList* which contains all the current solutions that are visited at the previous iterations of the heuristic. This approach is in contrast with the traditional application of tabu list consisting of recently admitted move operations. The application of the traditional tabu list does not guarantee avoidance of cycling. As was the case in the application of traditional tabu list in solving our BRICP problem, previously visited solutions may be reached through a certain sequence of non-tabu moves. This results in the selection of a previously visited solution as the current solution. Since the neighborhood generation is deterministic in our problem (i.e. we generate all possible neighbors), once a previous solution is selected as the current solution, the exact same neighbor generation and selection progression occurs as in the former iterations following the previous visit. Clearly, in that case, cycling lasts indefinitely. In our tabu list employment, however, a neighboring solution is ignored when it is found to be equivalent to a solution in the *TabuList*. Since all of the previously visited solutions are included in the *TabuList* (i.e. indefinite tabu tenure), the cycling avoidance is guaranteed. Note, however, that in our definition of tabu list there is an extra computational burden in recording new tabu entries and confirming that a neighbor solution is not a tabu solution.

Consequently, at the end of each iteration, the best solution is chosen as the current solution among the solutions recorded in the *NeighList* according to the cor-

responding penalty deducted objective function values. In this selection rule, we avoid the solutions which are included in the *TabuList*. Since selection of a solution in *TabuList* as the current solution means certain cycling, we do not apply an exception to the tabu criteria like aspiration criteria.

4.1.4. Termination of TSH

We utilize two conditions for the termination of the TSH: the maximum number of iterations performed and the maximum number of consecutive nonimproving iterations. The former is satisfied when $iter = MaxIter$ while the latter is satisfied when $NIter = MaxNI$.

If either one of these two conditions is satisfied, then the iterations are stopped and the best feasible solution encountered throughout the iterations (i.e. *Inc_sol*) is declared as the solution of the Tabu search algorithm.

First the FR of TSH is performed and the corresponding best found solution is recorded. Then, another potential solution to the problem is obtained through the SR of TSH. The final solution of the TSH is the chosen among these two based on the final coverage values.

4.2. Binary Enumeration Tree Algorithm

We employ a Binary Enumeration Tree (BET) algorithm in an effort to solve the BRICP problem to optimality. In this algorithm, we enumerate all feasible combinations of location (**Y**), relocation (**U**) and protection (**P**) variables of the ULP by means of BET so that in each leaf node of the tree, there is a solution for the system planner's problem. The response of the attacker to each ULP solution is obtained through CPLEX by solving the LLP problem. The CPLEX returns values of the corresponding attack variable **S** as well as the final total coverage upon the interdiction. Note that BET is a methodology with firm exponential complexity. In solution of our ULP of BRICP, number of binary variables is not limited to number of potential facility sites,

since we consider also the protection and relocation decisions in our problem. Hence we need to develop an efficient branching procedure and some shortcuts to keep the problem solution time of BET in a reasonable range for small problem sizes. We first explain the branching procedure and then present several shortcuts to the algorithm in the following parts.

4.2.1. Branching Procedure

The branching should provide binary values to the location, relocation and protection variables. Each solution should be feasible subject to the ULP constraints. Our branching for the ULP consists of three stages.

First stage of branching is performed on the binary alternatives to location of new facilities to potential sites. In this stage starting from the root node, we branch for all potential sites, setting one of which to “1” and “0” alternatives in each level of branching. Thus, each node corresponds to a potential facility site together with the binary value standing for new facility location at that site. The first stage branching is over when opening and not opening a new facility alternatives are considered for all the potential sites. Clearly, the depth of this stage is equal to the number of potential sites. Child nodes in the final level of this stage are termed as leaf_1 nodes. Each leaf_1 node stands for a partial solution to the ULP which includes new facility opening alternatives.

The second stage branching starts from final nodes of the upper stage. In other words this stage is composed of subtrees of leaf_1 nodes. Here, given a pattern for location of new facilities, branching is performed over protection alternatives of newly open facilities. Namely, each level of branching generates alternatives of protecting or not protecting a certain open facility. The depth of a subtree is equal to the number of open facilities given in the leaf_1 node. Similar to the upper stage branching, leaf_2 nodes stands for a partial solution to the ULP, alternatives of new facility openings together with the protection alternatives of those.

The third stage branching considers the relocation and protection alternatives of the existing facilities. Recall that, any existing facility can be kept in its original site without any cost or can be relocated to a potential site incurring a relocation cost whereas each existing facility may or may not be protected regardless of their location. Third stage branching is performed for all leaf_2 nodes. Any subtree to a leaf_2 node is composed of levels of branching each representing an existing facility. For any existing facility, numerous child nodes are created which includes both the protection/no protection versions to alternatives (no relocation and relocation to each empty potential facility site). While the depth is constant and equal to the number of existing facilities, number of branches in any level depends on the number of empty sites which is determined by the location variables set in the first stage. The branching terminated at the end of this stage. We call the last child nodes, final leaf nodes, each of which represents a complete solution to the ULP.

Once a feasible solution alternative to ULP is obtained in a final leaf node, as was the case in the heuristic procedure, we utilize CPLEX embedded in the algorithm to solve the attacker's MIP problem. Given the location, relocation and protection pattern of the system planner, the solution of the LLP provides interdiction strategy of the attacker as well as the total final coverage value. The branching follows a depth first strategy. As soon as a final node is reached, the objective function value for that solution is obtained through CPLEX. The best solution is updated whenever a solution in a final leaf provides higher final coverage than the current best. Upon completion of creating all possible final leaf nodes, the best solution found is the optimal solution to the BRICP problem.

Note that, the BET algorithm enumerates every feasible solution to the system planner's problem. In this sense, it is equivalent to an exhaustive enumeration method. However, we emphasize the tree structure of the methodology since we develop some shortcuts thanks to the branching procedure. In following parts we present those shortcuts.

4.2.2. Pruning due to Budget Infeasibility

Complexity of a BET algorithm is obviously exponential, since the tree size increases exponentially with the size of the problem (i.e. number of variables to be branched on). In an effort to decrease the tree size, we develop a pruning rule which eliminates branching for infeasible solutions.

Note that in any node in the enumeration tree we have a partial solution. There are some decisions of system planner that are already made in the upper nodes and some decisions are to be made in the lower nodes. For any node we calculate cost already incurred based on the decisions already made in the upper nodes. The tree structure is such that, the total cost to system planner in any child node is either the same as or is greater than the total cost in the parent node. In other words, cost to the system planner is non-decreasing in the course of branching. Once budget is overrun in a node, any final solution reached through the subtree of that node will definitely be budget infeasible. We stop branching over a node whenever budget constraint violation is encountered with an intention to eliminate the computational effort wasted for branching on infeasible solutions.

4.2.3. Upper Bound Approach

Unlike the pruning approach, the upper bound (UB) approach is applied in the final leaf nodes of the tree. The pruning provides that the corresponding solutions in final leaf nodes are feasible solutions to the ULP. Here, the effort is to decrease the computational time spent for solving the LLP corresponding to some unpromising ULP solutions.

In a final leaf node, the locations of the facilities and their protection statuses are already determined. Since the optimal attacker response to such a location and protection pattern is unknown, the total final coverage value can be computed upon the CPLEX call. We can devise an UB on the final total coverage, however, with an aim to eliminate unnecessary CPLEX calls. Suppose, a final leaf node corresponds

to a solution composed of total p open facilities including the existing and new ones, q of which are protected. The terrorist on the other hand has resources to attack r facilities. If $p - q > r$, then the terrorist attacks r of the $(p - q)$ unprotected facilities optimally to deteriorate the final coverage. In other words, the terrorist chooses the best strategy among the alternative attack strategies which results in the least final coverage (i.e. optimal objective value to LLP problem). An arbitrary attack strategy results in a final coverage greater than or equal to the coverage due to optimization of the attacker. Hence, final coverage due to an arbitrary attack is an UB to the LLP solution.

In every final leaf node, the corresponding UB is calculated provided that $p - q > r$. When this UB is less than or equal to the coverage value of the current best solution, then the attacker's problem is not solved. If the UB is higher however, then the attacker's problem is solved through CPLEX. Note that, UB being higher than the current best solution does not guarantee a better solution. Current best solution is updated only if the final coverage returned from CPLEX is higher. In case where $0 < p - q \leq r$, the attacker attacks $(p - q)$ facilities among the $(p - q)$ unprotected facility. In other words, there exists only one reasonable attack for the attacker; attacking all unprotected facilities. Dropping the unprotected facilities from the solution, the final coverage value can be reached without calling CPLEX. Clearly, when $p - q = 0$, which means that all facilities are protected, the trivial optimal solution to the attacker is not to attack at all and the final total coverage value can be evaluated via the solution of the system planner.

4.2.4. Solutions with Budget Surplus

We call the solutions to the ULP solutions with budget surplus, where the budget surplus is higher than or equal to the minimum facility opening cost among the empty potential facility sites. Suppose H is a solution with budget surplus. Then there exists another feasible solution G , which has the exact same location, relocation and protection pattern as H with the exception of an additional facility opened in one of the empty sites. Note that, presence of such a solution is due to the budget surplus

in H . Then, given these assumptions, G is at least as good solution to the ULP as H . This is not violated when the attacker's response is considered. Compared to H , in G attacker has one more facility to attack. Even if the additional open facility is attacked, then G is an equivalent solution to H . Otherwise, when the attacker does not change its attack plan, then the coverage is at least the same as in H , with the possible additional coverage provided by the additional facility.

Suppose there are two final leaf nodes representing the solutions G and H . Then we can eliminate optimization of the attacker's problem for the leaf corresponding to H , since there is another solution G is at least as good as H . Note that, since our approach enumerates all feasible solutions to the ULP, G will be included in the enumeration definitely. If H is an optimal solution, then G is definitely an alternative optimal solution to the problem. Accordingly, BET will reach G as the optimal solution unless presence of other alternative optimal solutions. In general, we ignore any final leaf node which corresponds to a solution with budget surplus.

5. COMPUTATIONAL RESULTS

5.1. Problem Instances Generation

We test the performance of the proposed solution methods on 240 problem instances. We generate these instances using eight customer configurations. Two of those are based on real-world data which include the two-dimensional coordinates and the demand values of the blocks in the city of São José dos Campos (SJC), Brazil [9]. Each block represents a customer zone in the city, and the number of houses, apartments, and commercial and government buildings in each block provides the demand of that zone. The first data set consists of 324 zones, while the second one has 818 zones. We refer to these as SJC324 and SJC818, respectively. In addition to these real world data sets, we generate six random customer configurations. In these configurations we consider two customer distribution patterns: uniform distribution of customers (U) and clustered customers (C). The latter is a more realistic distribution of customers where each customer belongs to one of the five clusters. In clustered customer configurations, the distances between customers of the same cluster are generated using normal distribution with mean zero and standard deviation 100. From each customer configuration we generate three types of instances in terms of the number of customers: 200, 400 and 800 customers. The details of the six instance types U200, U400, U800, C200, C400, and C800 are given in Table 5.1, whereas the visualizations of the real-world and random configurations on xy -plane can be seen in Appendices A and B, respectively.

As summarized in Table 5.1, a two-dimensional cartesian space with side length 1000 is used in all random customer configurations. Customer locations and the corresponding demand values are determined independent of each other. The customer demand in both configuration types is determined randomly from the set $\{5, 10, 15, 20, 25, 30, 35, 40, 45, 50\}$. Customer locations in uniform configurations are determined from a uniform distribution for vertical and horizontal coordinates of each customer. In clustered distribution, on the other hand, each customer is located with

Table 5.1. Random customer configurations generation

Customer configuraitons	Parameters	Values
Uniform; U200, U400, U800	Number of customers	200, 400, 800
	Coordinate range	[0-1000,0-1000]
	x -coordinate	Uniform(0,1000)
	y -coordinate	Uniform(0,1000)
	Customer demand	DiscreteUniform{5,10,...,50}
Clustered; C200, C400, C800	Number of customers	200, 400, 800
	Coordinate range	[0-1000,0-1000]
	Number of clusters	5
	Cluster center c_x	Uniform(200,800)
	Cluster center c_y	Uniform(200,800)
	Customer belongs to cluster	DiscreteUniform{1,2,...,5}
	Distance to cluster (d)	Normal(0,100)
	Angle to cluster (θ)	Uniform(0,2 π)
	x -coordinate	$c_x + \cos(\theta).d$
	y -coordinate	$c_y + \sin(\theta).d$
	Customer demand	DiscreteUniform5,10,,50

angle θ and distanced d to one of the cluster centers. While θ is generated uniformly, d is generated by absolute value of the normal distribution with mean zero and standard deviation 100. The corresponding vertical and horizontal coordinates on the plane are calculated through the functions given in the table.

From each of the six customer configurations we generate 30 instances by randomly selecting $N_1 = 2$ of the customer points as existing facility sites and $N_2 = 10, 15, 20$ of the customer points as potential sites. This implies that there are 10 instances with the same number of potential facility sites, i.e, 10, 15 or 20, where the number of existing facility sites is always two. Given the coordinates of the existing and potential facility sites as well as the customer sites, the coverage parameter (a_{ki}) is

determined. The fixed cost of opening a new facility (f_i) and cost of relocating an existing facility (g_{ij}) are integer values drawn randomly from the sets $\{4000, 4500, \dots, 6500\}$ and $\{1500, 1600, \dots, 2500\}$, respectively. The fixed cost of protecting a facility (h_i) is taken as a constant for each problem instance, which is equal to α times the average fixed cost of opening a facility. Here α denotes the relative cost of protection versus facility opening. The available budget of the system planner (b) is set to 30,000 and the maximum number of attacks (r) is equal to two.

Note that in the following sections we create some additional variations of these 240 problem instances by modifying some of the parameters like the number of attacks (r), budget (b) and relative cost of protection (α).

5.2. Computational Environment

The source codes of both the TSH and BET algorithms are written in C# language and compiled through Microsoft Visual Studio 2005. CPLEX Callable Library 11.2 is called via C# environment for the solution of the attacker's LLP. CPU times have been measured on a workstation equipped with two Intel Xeon X5460 3.16 GHz Quad-Core processors and 16 GB RAM. One core of the processor in the workstation is reserved for each test.

We present our experiments in the following order. First, the overall results of the 240 problem instances using the TSH and BET algorithms are provided. Following that, we do some experiments on the computational time. Finally we present the managerial insights generated from the overall results and further experiments on BRICP problem.

5.3. Results of the BRICP Problem Instances

Here we test the solution algorithms on the problem instances. BET algorithm can solve instances with $N_2=10, 15$ to optimality. The size of the instances with $N_2=20$ are too large to be solved by BET since the number of combinations grows exponen-

tially.

The detailed results for the instances based on real-world customer configurations (SJC324, SJC818) and randomly generated configurations (U200, U400, U800, C200, C400, C800) can be found in Appendix C and D, respectively. The coverage values obtained by TSH and BET for each problem instance and the corresponding CPU times are provided. In these tables, “% Gap” refers to the gap between the service coverage values attained by BET and TSH, which is computed by the formula $100 \times (z_{BET} - z_{TSH})/z_{BET}$. Also, “# OF”, “# PF”, and “# IF” denote, respectively, the number of open facilities (the sum of the newly opened facilities and relocated facilities), the number of protected facilities, and the number of interdicted facilities in the optimal solution found by BET.

The results are summarized in Table 5.2. An instance group refers to 10 random instances of the given customer configuration and the number of potential facility sites (N_2). The average of coverage values and CPU times are provided for each instance group as well as the average gap between the optimal solution and the heuristic solution. Based on 180 test instances for which BET can provide optimal solutions, we can conclude that the accuracy of TSH is 0.24% on the average. The maximum gap is recorded as 5.21% whereas in 167 out of 180 instances the gap is lower than 1%. Furthermore, the TSH algorithm is capable of finding the optimal solution in 137 instances. The average solution times for BET and TSH are 361 and 144 respectively seconds for instance group with $N_2=10$ and 4078 and 331 seconds for instance group with $N_2=15$. The experiments show that TSH is an efficient heuristic which can generate good near optimal solutions for BRICP problem instances within a reasonable amount of time. BET, however, needs a prohibitively large amount of time to solve the problem to optimality depending on the size of the problem instance (i.e. number of potential facility sites).

Table 5.2. Summary of the overall results

Instance Group	BET		TSH		% Gap
	Coverage	CPU	Coverage	CPU	
SJC324-10	4217.08	158.9	4217.08	73	0.00
SJC818-10	4906.97	931.9	4906.97	338	0.00
U200-10	1454.24	108.1	1450.71	28	0.24
U400-10	2634.36	193.4	2621.12	71	0.50
U800-10	5332.39	548.1	5321.02	174	0.21
C200-10	4282.49	148.2	4279.74	60	0.06
C400-10	7144.90	201	7144.90	115	0.00
C800-10	12663.78	599.1	12663.78	290	0.00
SJC324-15	4514.36	3047.8	4449.03	200	1.45
SJC818-15	5456.12	9676.6	5440.34	726	0.29
U200-15	1627.62	1375.8	1621.84	70	0.36
U400-15	3028.74	2682.4	3013.62	145	0.50
U800-15	6016.80	5910.5	6003.27	375	0.22
C200-15	4468.02	1518	4464.47	139	0.08
C400-15	7772.18	2508.4	7772.18	275	0.00
C800-15	13291.59	5905.8	13291.59	714	0.00
SJC324-20	—	—	4686.03	609	—
SJC818-20	—	—	5315.43	1,386	—
U200-20	—	—	1648.77	133	—
U400-20	—	—	3091.36	294	—
U800-20	—	—	6102.63	839	—
C200-20	—	—	4566.15	241	—
C400-20	—	—	7896.86	567	—
C800-20	—	—	13801.20	1,541	—

5.4. Analysis from Computational Perspective

5.4.1. Solving the ULP and LLP Separately

Bilevel problems are not reducible to two independent problems occurring at different levels. To highlight that, we refer to the following experiment. The ULP and LLP of BRICP are solved in sequence as if they are independent problems. First the ULP is solved independent from the LLP so that the total coverage is maximized through locating new facilities and relocating existing ones ignoring the attacks. Then, given the optimal solution to the ULP, LLP is solved which gives the attacker's best attack plan. The resulting coverage is taken as the final solution of the decomposed version of the BRICP to two subproblems. We utilize our TSH algorithm to solve the ULP by setting $r = 0$. Note that, when $r = 0$, then the LLP becomes ineffective and the TSH gives solution to the ULP independent of the attacker's problem. Once the best found solution for ULP is obtained, the LLP is solved via CPLEX which gives the optimal attack plan of the attacker. This solution procedure is referred to as decomposed solution methodology (DSM). We perform the experiment using 15 instances based on SJC324 customer configuration. In Table 5.3 the final coverage values of DSM, TSH and BET are shown. “% Gap” refers to the percentage of gap between coverage values of DSM and the optimal results (i.e., BET results).

The results prove that the subproblems in a bilevel problem cannot be solved independent from each other. DSM solutions may deviate up to 15% from the optimal solution while the average gap is 6.75%.

5.4.2. Contribution of Two Runs Approach in TSH

Recall that we apply two runs of TSH. FR of TSH is the tabu run which is expected to search the extreme region in SS where the majority of the facilities are protected. On the other hand, SR of TSH is manipulated towards the extreme region in SS where the minority of the facilities are protected.

Table 5.3. Test results of DSM on instances of SJC324 configuration

Instance	Coverage Values			% Gap
	DSM	TSH	BET	
10-1	3583.35	3727.80	3727.80	3.87
10-2	3814.13	4134.84	4134.84	7.76
10-3	3816.16	4510.15	4510.15	15.39
10-4	4064.27	4429.47	4429.47	8.24
10-5	3925.83	4342.87	4342.87	9.60
15-1	3902.11	4328.42	4364.03	10.58
15-2	4263.72	4558.63	4558.63	6.47
15-3	4369.50	4560.37	4560.37	4.19
15-4	4394.05	4397.98	4397.98	0.09
15-5	4330.78	4316.16	4387.86	1.30
20-1	4876.61	4979.11	—	
10-2	3829.85	4041.34	—	
20-3	5003.01	5067.96	—	
20-4	5148.72	5178.47	—	
20-5	3979.18	4771.09	—	

To emphasize the improvement achieved by applying the two runs, we compare the performance of BET with the performance of single tabu runs of FR and SR. We perform our analysis on 20 instances of the SJC324 configuration with $N_2=10, 15$. Table 5.4 shows the results of this experiment while the performance of TSH for these instances can be found in Appendix C. It can be seen that, when applied individually, neither FR nor SR provides good solutions to BRICP problem. The maximum gap is found to be 17% and 15% for FR and SR, respectively whereas it is 5% in TSH. Also, in terms of accuracy the TSH (0.47%) outperforms FR (3%) and SR (4%) of TSH.

Table 5.4. Test results of TSH FR and SR on instances of SJC324 instance

Instance	BET	FR of TSH				SR of TSH			
	Coverage	Coverage	% Gap	#OF	#PF	Coverage	% Gap	#OF	#PF
10-1	3727.80	3594.15	3.59	4	4	3727.80	0.00	7	0
10-2	4134.84	4134.84	0.00	4	4	4105.33	0.71	8	0
10-3	4510.15	4510.15	0.00	4	4	3816.16	15.39	7	0
10-4	4429.47	4429.47	0.00	4	4	4095.62	7.54	7	0
10-5	4342.87	4342.87	0.00	4	4	3925.83	9.60	7	0
10-6	4507.40	4507.40	0.00	4	4	3952.00	12.32	7	0
10-7	3587.93	2964.64	17.37	3	3	3587.93	0.00	7	0
10-8	3838.17	3663.26	4.56	4	4	3838.17	0.00	8	1
10-9	4863.98	4863.98	0.00	4	4	4614.83	5.12	7	1
10-10	4228.17	4228.17	0.00	4	4	3813.34	9.81	7	1
15-1	4364.03	4325.36	0.89	4	4	4328.42	0.82	7	0
15-2	4558.63	4369.13	4.16	4	4	4558.63	0.00	7	0
15-3	4560.37	4560.37	0.00	4	4	4533.95	0.58	8	0
15-4	4397.98	4345.47	1.19	4	4	4397.98	0.00	7	0
15-5	4387.86	4316.16	1.63	4	4	4272.24	2.63	8	0
15-6	3824.93	3824.93	0.00	4	4	3768.95	1.46	7	1
15-7	3911.72	3556.06	9.09	4	4	3707.99	5.21	7	1
15-8	5087.84	4776.24	6.12	4	4	4844.75	4.78	7	0
15-9	4880.58	4796.61	1.72	4	4	4596.69	5.82	8	0
15-10	5169.68	4550.63	11.97	4	4	5169.68	0.00	8	0

5.4.3. Contribution of UB Approach in BET

In order to highlight the contribution of the upper bound (UB) approach, incorporated in our BET algorithm we perform an experiment on eight small instances ($N_2=10$), the results of which are provided in Table 5.5. CPU_1 refers to the solution time of BET without the UB improvement where CPU_2 is the solution time of the original BET algorithm. The ratio of the solution times provides concrete evidence that

Table 5.5. Test results of BET (with and without UB) on instances of U200 and C200

Instance	BET (without UB)		BET		CPU ₁ /CPU ₂
	Coverage	CPU ₁	Coverage	CPU ₂	
U200					
10-1	1427.09	8707	1427.09	150	58.05
10-2	1575.21	10656	1575.21	219	48.66
10-3	1496.00	7874	1496.00	105	74.99
10-4	1340.27	9320	1340.27	89	104.72
C200					
10-1	4109.95	9806	4109.95	80	122.58
10-2	4405.27	12827	4405.27	239	53.67
10-3	4235.89	11034	4235.89	166	66.47
10-4	3925.42	7405	3925.42	29	255.34

UB approach incorporated in BET algorithm is a key improvement, without which the BET is 72 times slower on average in solving the BRICP problem instances.

5.4.4. The Effect of the Number of Potential Sites on Solution Performance

The number of potential facility sites is one of the most critical factors in determining the problem size and therefore is expected to affect the solution performance. We test the sensitivity of this factor on our solution procedures TSH and BET. The overall results of uniform and clustered configuration instances are presented in Appendix C and D while the summary of the results according to the number of potential facility sites can be found in Appendix E. Figure 5.1 illustrates the solution time performance of TSH and BET. As can be seen, solution times of both are dependent on the problem size, whereas the BET algorithm's solution time increases exponentially as expected. The accuracy of TSH, as shown in Figure 5.2, is slightly dependent on the problem size. The accuracy is recorded as 0.17 and 0.195 for $N_2 = 10$ and 15, respectively.

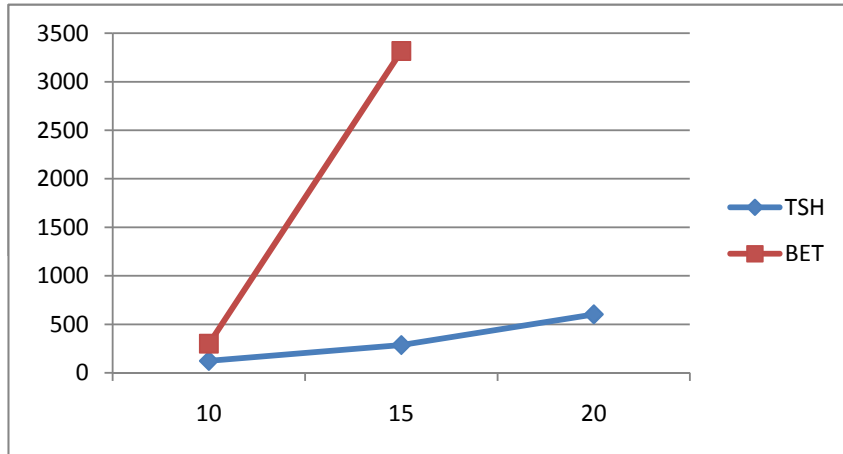


Figure 5.1. CPU time as a function of the number of potential facility sites

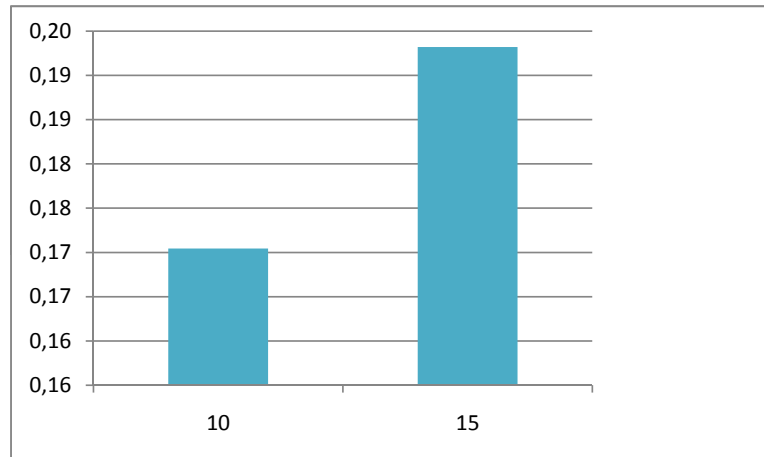


Figure 5.2. Gap as a function of the number of potential facility sites

5.4.5. Effect of Customer Configuration on Solution Performance

In our test bed, we have two distinct customer configuration types: clustered and uniform dispersion of customers. In addition, we have three different sizes of each type involving 200, 400 and 800 customers. We test the performance of the solution algorithms for these customer configurations. The overall results (see Appendix C and D) are summarized in Appendix F according to customer configurations. The results are illustrated in Figure 5.3 and Figure 5.4. The former shows the computation times of TSH and BET based on customer configurations. The computation time increases in both TSH and BET while BET is more sensitive. It is observed that the customer dispersion affects the solution time of TSH. In clustered dispersion case, the heuristic solution time is longer while the gap between the optimal and the heuristic solution is

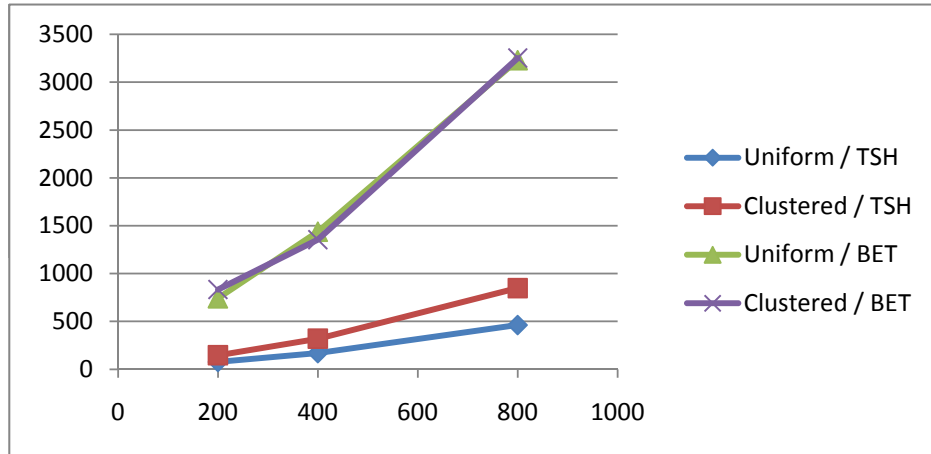


Figure 5.3. CPU time as a function of the number of customer sites

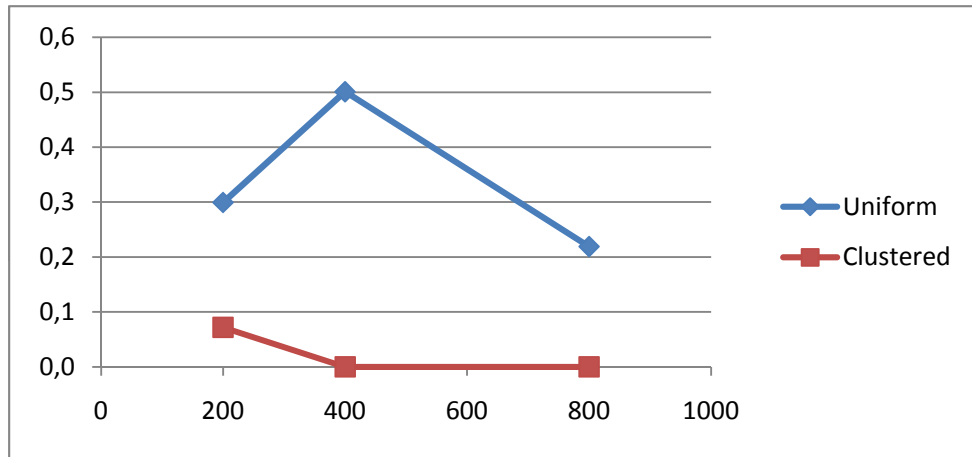


Figure 5.4. Gap versus number of customer sites

higher in the uniform case as shown in the latter figure.

5.4.6. Effect of Budget in Solution Performance

Budget restriction in our model has a key role on the system planner's action. When the budget is high, the system planner can locate more facilities, protect more facilities or use the relocation option more often compared to the case where the budget is lower. In other words, the solution space of the ULP becomes larger. We run the following test in order to observe the effect of the budget increase on the performance of the solution procedures TSH and BET.

Four instances of each size of the SJC324 customer configuration are used in

Table 5.6. Results of BET and TSH for various budget amounts

Instance Group	Average CPU time		Average gap
	BET	TSH	
$b=25,000$	3795.63	209.25	0.17
$b=30,000$	4446.01	237.92	1.46
$b=35,000$	4891.18	249.50	1.84

this test. Three modifications of each instance with budget values 25,000, 30,000 and 35,000, are solved via BET and TSH. The results are summarized in Table 5.6 while the detailed results can be found in Appendix G.

The average CPU time increases both for the TSH and BET. When the budget increases, it directly increases the number of solutions included in the search of BET. The increase in the solution time of TSH, however, depends on the increase in the number of neighbors generated in each iteration. Recall that the size of the neighborhood is dependent on the number of open facilities. Also, the quality of the solutions reached by TSH decreases with an increase in the budget. We conclude here that the solution algorithms are budget sensitive, that is, when the budget increases one should expect an increase in the solution time. To overcome the decrease in the solution quality of the TSH, additional iterations should be performed by increasing *MaxIter* and *MaxNI*.

5.4.7. Effect of r on the Solution Performance

We use 15 problem instances of SJC818 configuration equally divided into 3 groups according to the number of potential sites (i.e. $N_2 = 10, 15, 20$). Recall that the instances have $r = 2$ in their original setting. We modify these instances by setting r to one and three. The detailed results of the test can be found in Appendix H. We illustrate the results as follows. Figure 5.5 shows the average CPU times for TSH, each series represents one of the three problem sizes. Similarly Figure 5.6 shows the average CPU times for BET. Problem instances with a potential number of facilities equal to 20 cannot be solved by BET and therefore they are not included in the results.

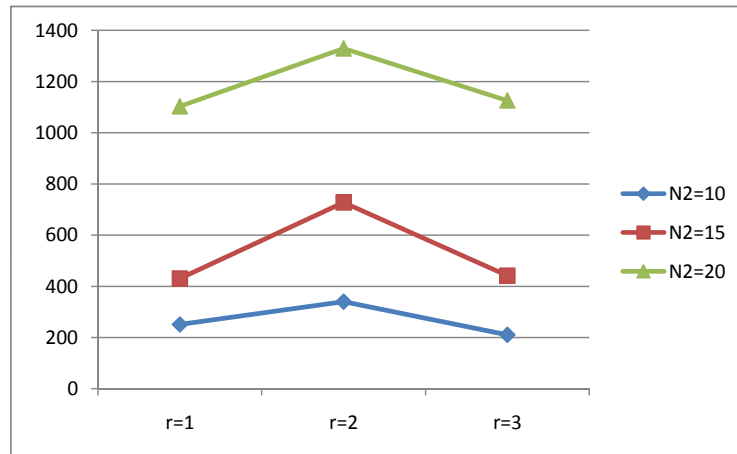


Figure 5.5. CPU time of TSH as a function of r

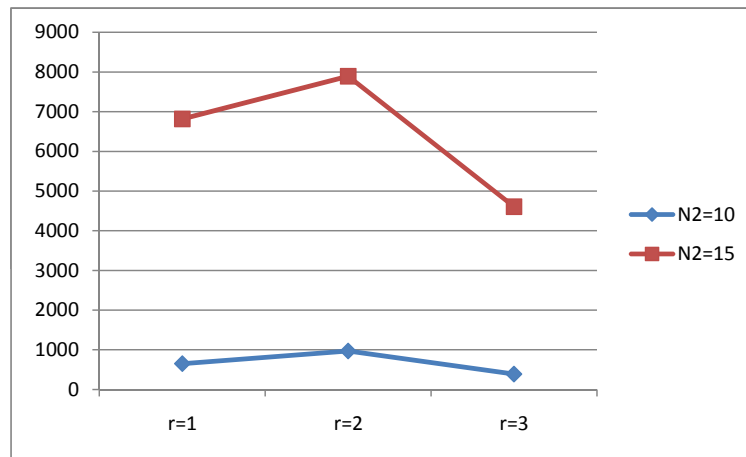


Figure 5.6. CPU time of BET as a function of r

The CPU time in both solution procedures is longest when r is equal to two. When $r = 1$ and $r = 3$, however the solution time is lower. In the problems where $r = 1$, the optimal solutions do not involve protection. When $r = 3$, on the other hand, the system planner always protects all of the open facilities. These choices are due to obvious low risk and obvious high risk environments in these cases respectively. However, in cases where $r = 2$ the system planner does not seem to have a trivial choice of protection strategy. Namely, there are solutions involving intensive protection as well as solutions where protection is not used at all. We can conclude that in cases where there is obvious choice of protection strategy, both BET and TSH perform better in terms of solution time.

Note that we fail to test the interdependence between the quality of the heuristic

solutions and increasing terrorist resources, since there is only one instance out of total 45 for which we can calculate a positive gap. TSH found the optimal solutions in all of the other instances.

5.5. Analysis from a Managerial Perspective

In this section we carry out tests on the BRICP problem with the aim of understanding the problem from a managerial perspective. We use the TSH algorithm for finding solutions to the problem instances with the following analyses we want to identify the strategies of the system planner and the attacker and infer some managerial insights.

5.5.1. System Planner's Strategies

In the results to the problem instances of uniform, clustered and real-world customer configurations, we can observe that one of the following two extreme strategies prevail for the system planner. In the first one, the system planner opens a relatively small number of facilities and protects all of them. For example, when the number of open facilities is four, then all of the facilities are protected, which implies that the attacker cannot hit any facility. The other extreme pattern is to open a large number of facilities and protect only very few or none of them. As an example, we can mention the instances where the system planner has seven or eight open facilities. In these cases, at most one facility is protected and the attacker interdicts two unprotected facilities. To illustrate this behavior of the system planner we categorize the choices of the system planner as protecting all of the open facilities and protecting none of the open facilities. The exceptions to these patterns are included in the category named "other". Figure 5.7 shows the categorization of the system planner's best actions in the real-world data based problem instances SJC324 and SJC818. The similar categorization for the solutions to the randomly generated problem instances (U200, U400, U800, C200, C400, C800) is provided in Figure 5.8. As shown in these figures, the strategy of the system planner is either protecting all or protecting none of the open facilities, with the exception of 7% of the real-world instances and 2% of randomly

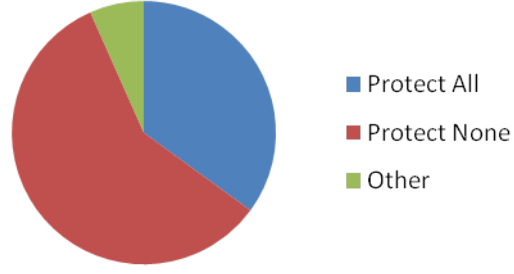


Figure 5.7. Categorization of system planner's action in real world customer configurations

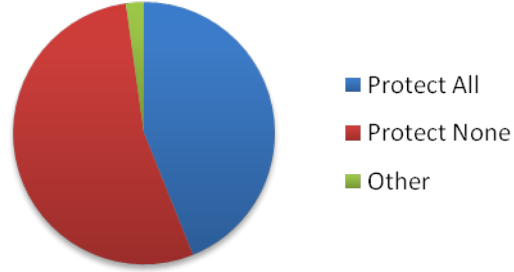


Figure 5.8. Categorization of system planner's action in random customer configurations

generated instances.

5.5.2. Two Types of Protection for the System Planner

With an effort to understand the two strategies of the system planner we perform the following test. We take 16 problem instances of the SJC324 configuration, in which the system planner chooses to open four facilities with protection. Then, we fix the locations of the newly open and existing facilities (we assign very high costs to other sites), and set the budget value and unit protection cost so that the system planner can protect a certain number of facilities. The attacker has the capability to attack two facilities ($r = 2$). Using this setting, we solve each instance five times, where the protection limit varies from zero to four facilities. Note that, when the protection is limited to four facilities, trivially, the best solution for the system planner is protecting all of the four facilities which is the same with the best solution to the original instance. Clearly, the total final coverage increases as the limit on the number

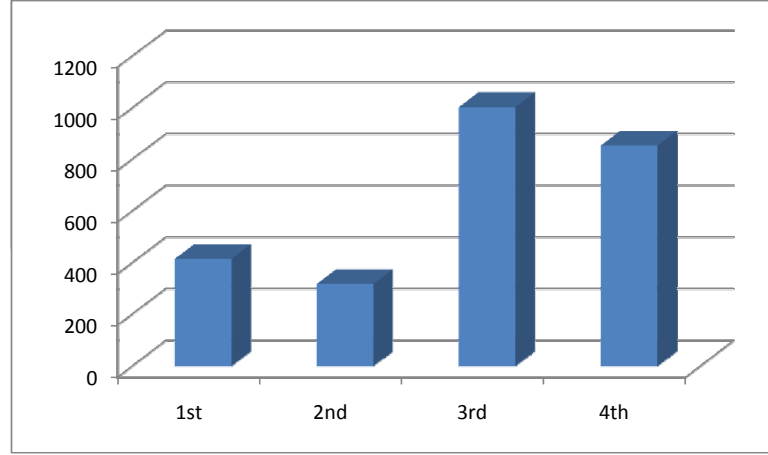


Figure 5.9. Marginal contribution of protection resources

of protections increases. We call the increase in the coverage value due to addition of each protection resource as the marginal contribution of that protection resource. Namely, the marginal contribution of the first protection resource refers to the average difference between the final coverage values of the cases where one and zero facilities are allowed to be protected. The marginal contributions of the second, third and fourth protection resources are defined accordingly.

The results of this analysis for each problem instance are shown in Appendix I. In Figure 5.9 the average marginal contributions of the protection resources are shown. It can be seen that, the contribution of the last two protection resources are fairly higher. Accordingly, we specify two types of protections of the system planner: first two protections with lower average marginal contributions are “redirecting protections” while the last two are “preventive protections”. The former does not avoid the terrorist attacks as there are at least two other unprotected facilities that the terrorist can attack using its resources. We call them as “redirecting” since protections cause the terrorist to redirect its attacks to other facilities. The latter, on the other hand, prevents utilization of some portion of the attacker’s resource (i.e. the attacker can attack less than r facilities). In our case, the terrorist has resources to attack at most two facilities. Therefore the preventive protections are the last two protections which decrease the number of attacks to either zero or one. Clearly, the system planner should have utilized the redirecting protections in order to be able to prevent the terrorist attacks through preventive protections.

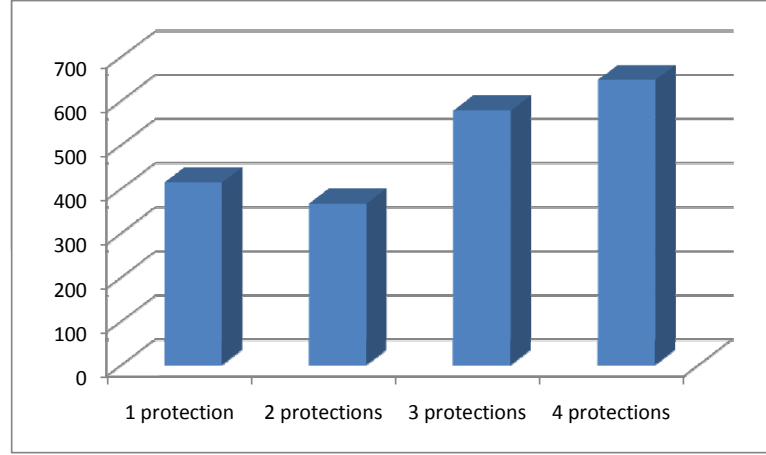


Figure 5.10. Average contribution per protection resource

This situation explains the extreme patterns in the following manner. Once the system planner is ready to protect two facilities, it can be expected that the system planner will be ready to pay for the protection of the third and fourth facilities as well, since the contributions are quite higher. In other words, the system planner does not prefer protecting one or two facilities because the contribution is low for these.

To illustrate this choice further, we use the average contribution concept. Figure 5.10, in contrast to marginal contribution, shows the average contribution per protection resource in cases where one, two, three and four facilities are protected. As can be seen, the average contribution per protection of a facility is higher in cases where three and four protections are made. Since the cost for protection of any facility is the same regardless of its location, we can infer that the average contribution per unit protection expenditure is higher in cases of protecting the majority of the facilities. Therefore, the strategy of protecting few facilities is mostly dominated by the strategy of protecting all or none of the facilities.

5.5.3. Value of Accurate Information to the System Planner

Recall that our problem setting assumes that the system planner has complete accurate information about the attacker's resources. Namely, the system planner designs a robust service network knowing that there will be exactly r intelligent attacks to the facilities in the network. In the real world, however, perfectly accurate information

is not possible especially in cases where the stakeholders are enemies to each other. In view of that, we study the impact of the misestimation of the terrorist's capability by the system planner.

First we define two notions. "Presumed r " (r_p) is the number of attacks to the facilities, which is considered by the system planner when designing the network. "Actual r " (r_a), on the other hand, is the actual number of attacks performed by the attacker. In our test setting, we distinguish the notions of "presumed r " and "actual r " which are considered to be the same in the original case. Accordingly, we solve the BRICP for $r = r_p$ in order to obtain the best location, relocation and protection plan of the system planner who presumes r_p potential attacks. Given that the plan of the system planner corresponding to r_p , we solve the attacker's problem with $r = r_a$, for various r_a values including $r_a \neq r_p$. The solution to the attacker's problem where $r_a = r_p$ gives us the final coverage under perfectly accurate information while solutions to cases where $r_a \neq r_p$ refers to the final coverage values subsequent to the misestimation of terrorist resources by the system planner. We utilize the TSH algorithm to obtain the leader's plan under a presumed number of attacks of r_p , while CPLEX is employed afterwards to evaluate the consequences of r_a actual number of attacks. We apply this test on five problem instances of each size based on SJC324 customer configuration. We consider all combinations of r_p and r_a from among the values $\{0, 1, 2, 3\}$.

The detailed results of this test are provided in Appendix J. The summarized illustration is presented in Figure 5.11. Each data series refers to a presumption of the system planner, while the horizontal axis includes the actual attack pattern and the vertical axis shows the average final coverage for each situation. If the actual number of attacks is unknown to the system planner, the safest choice is to presume the highest number of attacks ($r_p=3$) where the total coverage is independent of terrorist resources since all the facilities are protected. When the actual resources are low ($r_a=0$), however, this choice causes a significant amount of coverage loss, compared to the case where $r_p = r_a$. Clearly, this loss in coverage value is due to overestimation of terrorist resources. Indeed, the underestimation of the terrorist resources results in a loss in

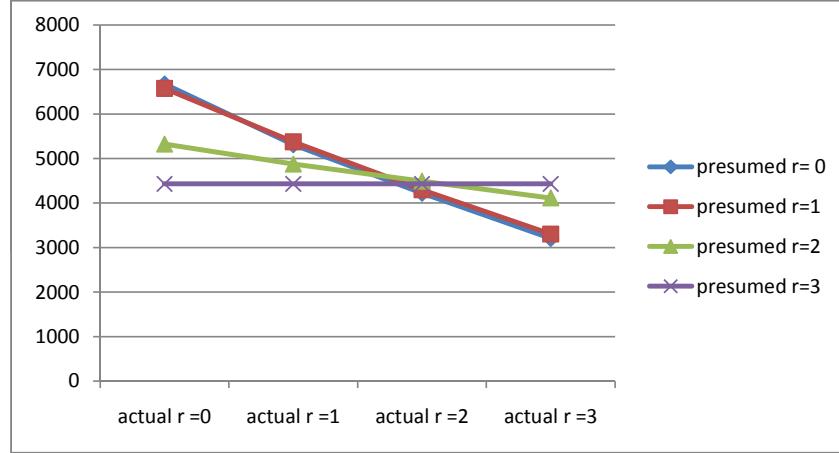


Figure 5.11. Coverage under accurate and inaccurate information to the system planner

service level. For instance, in case $r_a=3$ and $r_p=0$, the final coverage value is lower than the case with $r_p=3$. This is due to system planner being unaware of the terrorist and consequently being unprepared for such an attack to the network.

To reflect the impact of overestimation and underestimation of terrorist's resources prior to the design of the network, we present the following findings in Figure 5.12. The instances solved in this test are grouped into classes according to the deviation of presumed r from the actual r . The average deterioration in the final coverage value compared to the case of accurate estimation is shown in the vertical axis while the horizontal axis represents the deviation. For example, “+1” refers to the cases where $r_p = r_a + 1$ and “-1” refers to cases where $r_p = r_a - 1$. The results imply that accurate information is valuable for the system planner. It is observed that deviations of one result in a loss of covered demand of 200 units on average, where the misestimation may lead up to loss of 2242 units of demand in extreme cases. Furthermore, an interesting finding is that the overestimation of terrorist resources is more critical than the underestimation in terms of the resulting loss in the service level. The underestimation of terrorist's resources by three units results in a loss of 1200 units of demand while the overestimation by the same amount causes a loss of 2200 units of demand. On average, the loss is 866 and 532 units of demand due to the underestimation and overestimation, respectively. The results of this test are beneficial in understanding the importance of accurate information as well as the criticality of overestimation of

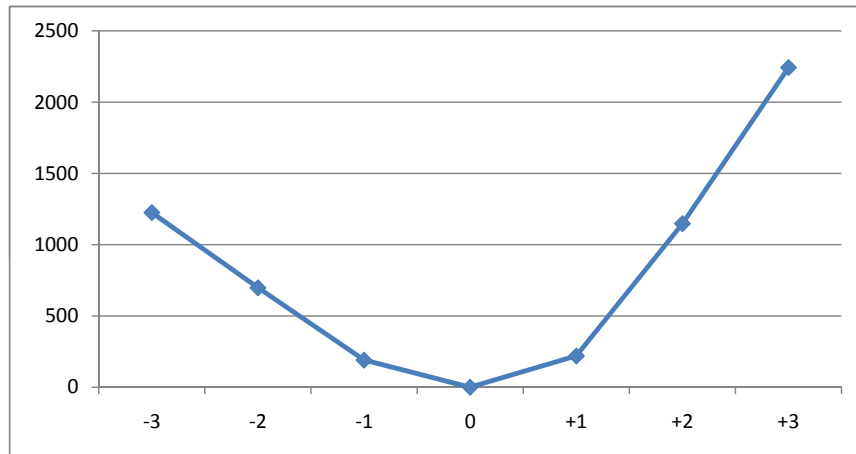


Figure 5.12. Marginal contribution of protection resources

the terrorist resources.

5.5.4. The Impact of Customer Dispersion on the System Planner's Decisions

The customer dispersion is another potential factor influencing the system planner's decision. To test this, we compare the overall results of the instances of uniform and clustered problem configurations (see Appendix C and D). Recall that, in uniform customer configurations, customers are uniformly distributed on the plane where in clustered configurations customers are prone to be located closer to some cluster centers randomly located on the plane. Given the same budget and the same number of customers with identical uniform distribution of demand, the final coverage value is higher in case of clustered location of customers. This finding is presented in Figure 5.13.

Figure 5.14 illustrates, on the other hand, the impact of the customer dispersion on the strategy of the system planner. Average number of open, protected and interdicted facilities is provided for two cases of customer dispersion. In case of uniformly dispersed customers, the system planner chooses to protect few of the open facilities (mostly none), whereas the all-protection strategy is mostly the best choice when the customers are located in clusters. This behavior can be explained via the observation of the results plotted on an xy-plane. As can be seen in some example plots (see Ap-

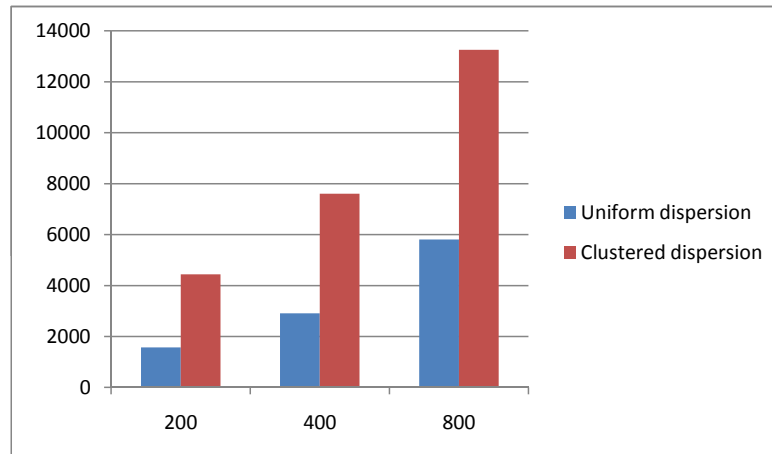


Figure 5.13. The effect of customer dispersion on final coverage value

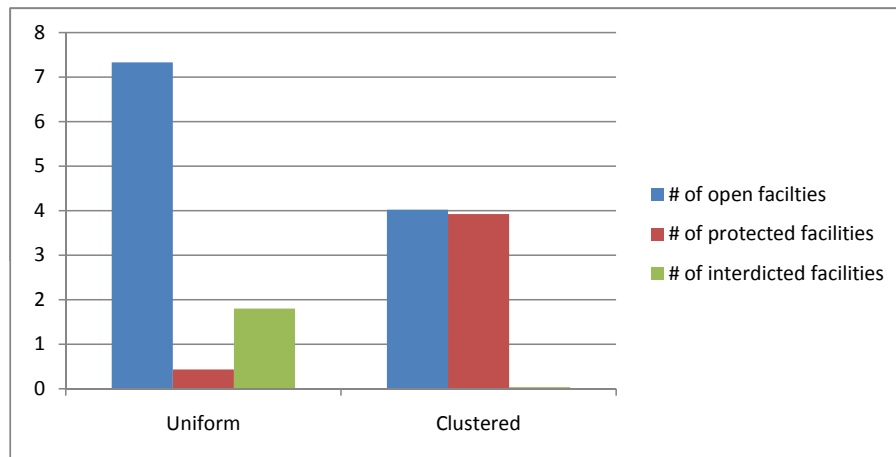


Figure 5.14. The effect of customer dispersion on the facility decisions

pendix K), the system planner chooses to open facilities near cluster centers and does not tolerate them to be attacked. In uniform dispersion case, however, the facility locations are indifferent in terms of the amount of the nearby demand. Hence, the system planner tolerates the attacks to most critical facilities, in order to utilize its budget on opening more facilities. The value of protection for the system planner in both cases will be analyzed separately in the following section.

5.5.5. The value of Protection to the System Planner

We perform the following test in order to analyze the value of protection option to the system planner. This value can be represented by the alpha (α) parameter, which is the coefficient used to generate protection cost data. Namely, the fixed cost

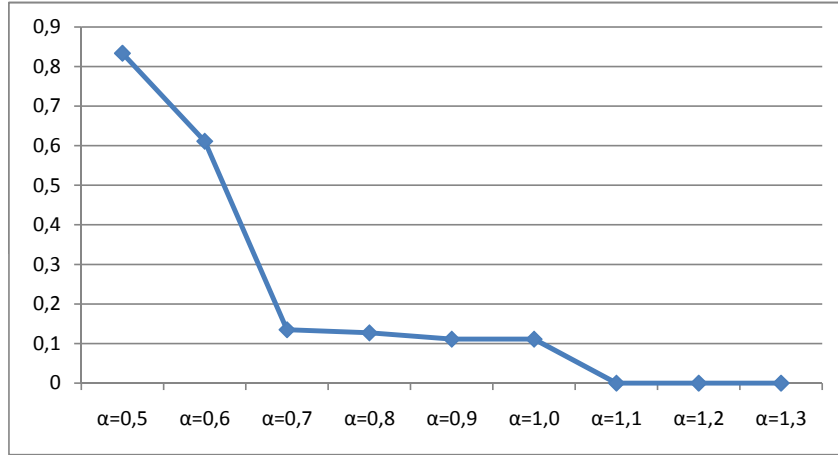


Figure 5.15. Effect of α on choice of protection in uniform customer dispersion

of protection in each instance equals to the average of fixed facility opening costs multiplied by α . In other words, α is the relative cost of protecting a facility against opening a facility.

We use a total of 36 instances; two instances of each size among the uniform and the clustered customer configurations. We modify the cost of protection by changing α , which is set to values from 0.5 to 2.1. The detailed results for instances of uniform and clustered customer configurations are provided in Appendix L. As can be seen, protection choices of the system planner are obtained for different α values. The results for the instances of uniform customer configurations are illustrated in Figure 5.15. The horizontal axis shows the average fraction of the total number of protected facilities among all open facilities. $\alpha = 0.5$ is a relative value which the system planner is ready to pay to protect nine out of 10 facilities. Expectedly, as α increases, the fraction of protection decreases. The fraction of protection being zero when $\alpha = 1.1$ indicates that the relative value of any protection is not as high as 1.1 for the system planner. The highest relative price that the system planner is willing to pay for a protection is 1.0, whereas one out of 10 facilities only is critical.

In the clustered customer dispersion case, however, the value of protection is higher for the system planner since the facility locations are more critical due to variable customer density. The results are illustrated in Figure 5.16. In contrast to the uniform customer dispersion cases, here the system planner is willing to pay as much relative

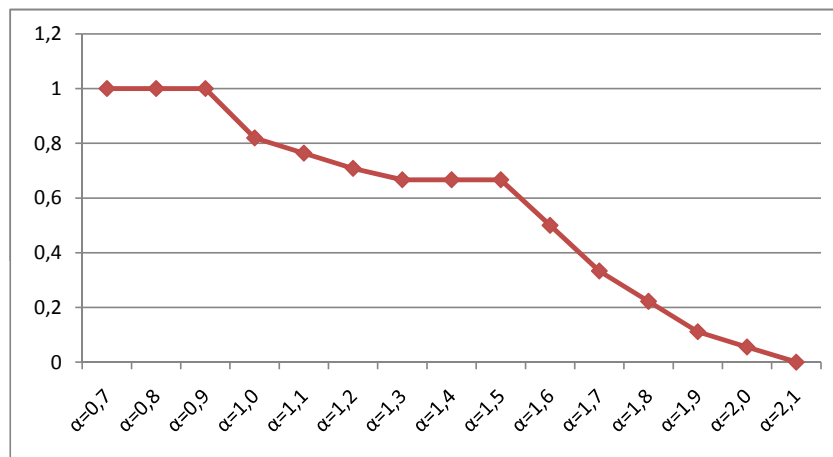


Figure 5.16. Effect of α on choice of protection in clustered customer dispersion

price as 0.9 to have all the open facilities protected. The highest relative price, on the other hand, is 2.0. Note that relative price of 2.0 indicates a very high price which means that the protection cost is equal to twice the opening cost of a new facility. One out of 20 facilities is critical to the system planner.

Two conclusions are drawn from this test. First, the value of protection for the system planner depends highly on the customer dispersion. The value of protection is higher when the customers are located non-uniformly on the plane compared to the uniform location. Secondly, the protection of a critical facility, in case of clustered customer dispersion, can be exceptionally valuable to the system planner.

5.5.6. The Value of Relocation to the System Planner

To analyze the contribution of relocating the existing facilities we compare the solutions of the instances of SJC818 configuration with the modification that relocation costs are set to a very high value so that no relocations are feasible. The system planner is allowed to protect existing facilities, however. The detailed results are provided in Appendix M. We illustrate the average coverage values for both cases in Figure 5.17. The contribution of the relocation option to the total coverage in case of 10 potential sites is not significant. In contrast, when the number of potential sites is higher, the contribution of relocation option to the total coverage is 10% on average.

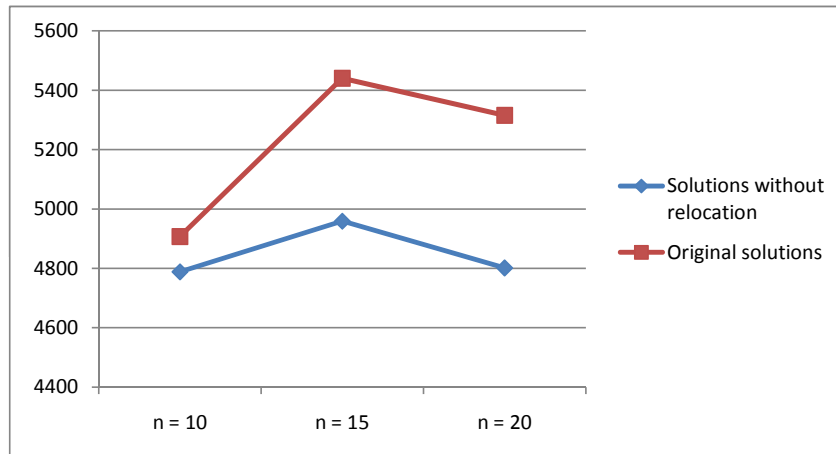


Figure 5.17. Effect of relocation option to final coverage value

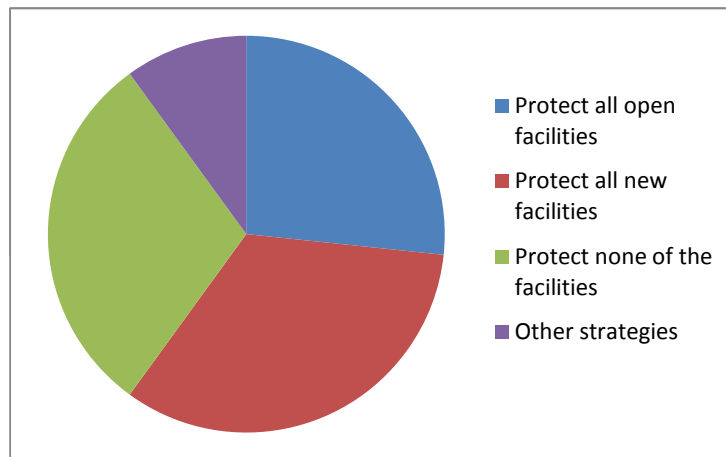


Figure 5.18. Effect of relocation option to the system planner's strategy

Furthermore, when relocation is not allowed, the best solutions for the system planner follow an additional extreme pattern. Recall that in the original case the system planner follows two extreme patterns, protecting all of the open facilities and protecting none of the open facilities. In addition to these, when relocation is not allowed, the system planner chooses to protect all open facilities except the existing facilities. The choices of the system planner are summarized in Figure 5.18. In 33% of the problem instances, the system planner chooses this new strategy. This behavior shows that when the locations of the existing facilities are fixed, these facilities become less critical to the system planner and the interdictions of these facilities are tolerated.

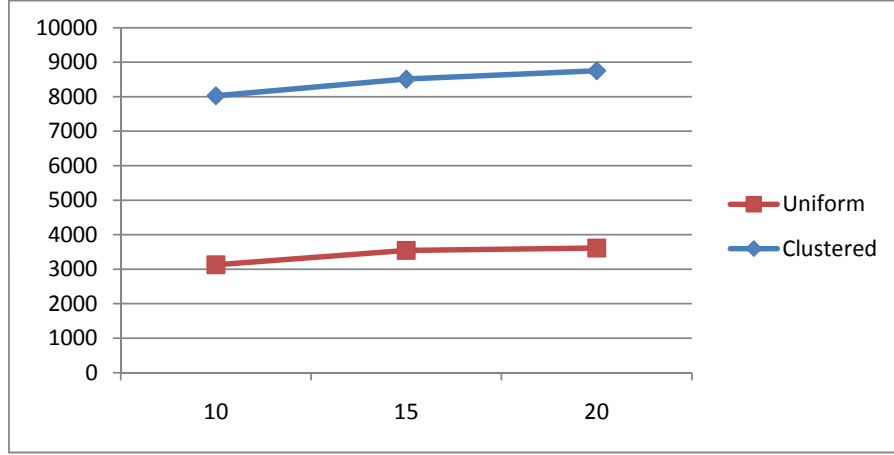


Figure 5.19. Effect of number of potential facility sites on final coverage

5.5.7. The Impact of Quantity of Potential Facility Sites

A potential factor in service level is the number of potential sites. Therefore we expect higher coverage values on average when there are more potential sites available. We analyze the degree of increase in coverage via comparing the solutions of the instances with $N_2 = 10, 15, 20$. The results are summarized in Figure 5.19.

An increase in potential facility sites from 10 to 15 and from 15 to 20 results in an average improvement in the coverage value by 13% and 1%, respectively when customers are uniformly distributed. When the customers are located in clusters, the corresponding average improvements in service level are 6% and 3%, respectively. Since the improvement is obtained using the same amount of budget, these improvements are appealing for the system planner. Consequently, prior to designing the service network, efforts on increasing the potential facility sites is worthwhile.

5.5.8. The Impact of Terrorist's Resources

We also analyze the relation between the amount of terrorist resources and the system planner's strategy. The results of instances of SJC818 configuration for different r values are presented in Appendix C. Figure 5.20 shows the change in total coverage value upon increase in terrorist's resources. As can be seen, the total post-attack coverage value that the system planner can provide decreases with increasing r when

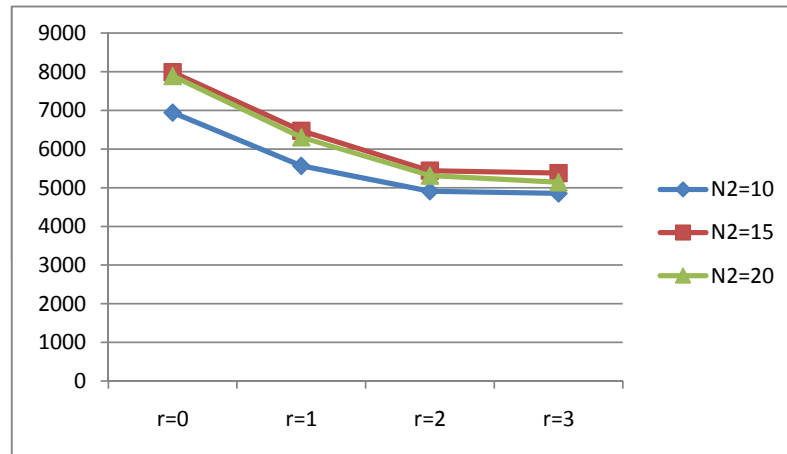


Figure 5.20. Effect of terrorist resources on final coverage value

the number of potential facilities is 10, 15 and 20. It is observed, however, that there is a certain value in terms of terrorist resources's, like $r = 2$ in this study, beyond which the system planner is alarmed and chooses to protect the entire network. Clearly, a further increase in terrorist's resources beyond the alarm value is non-effective, i.e. it does not decrease the final coverage level.

In Figure 5.21 we can see the system planner's choice when alarmed. The average number of protected facilities is maximum at $r=3$, when all open facilities are protected. Accordingly the average number of attacks becomes zero.

We can observe that the terrorist reaches its goal of reducing the service level even in the cases where no attack cannot be performed. As is the case when $r = 3$, although being passive, terrorist reaches its goal of service deterioration.

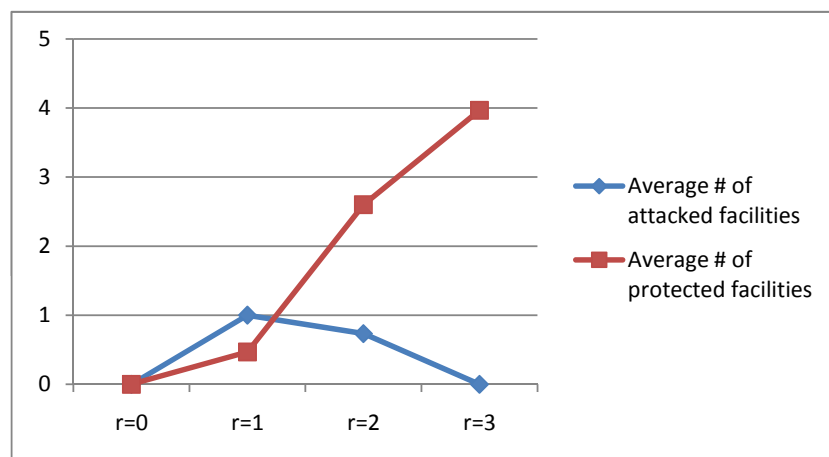


Figure 5.21. Effect of terrorist's resources on the system planner's decision

6. CONCLUSION

In this thesis, we address the problem of designing a service network in the presence of an intelligent attacker. The system planner locates new facilities, relocates existing facilities and protects new or existing facilities. Once the network design is determined, the attacker attacks a set of unprotected facilities. The action space of the system planner is constrained by a pre-specified budget limit, while the attacker is limited to a certain number of attacks. Accordingly, the system planner aims to maximize the total post-attack coverage of customer demand. This problem setting is novel due to the simultaneous consideration of facility location and protection decisions of the system planner as well as the interdiction decision of the attacker within the context of a covering problem. Moreover, we relax the common assumption made in maximal covering problems that a customer zone is either covered fully or not at all. Instead, we adopt the more realistic gradual coverage concept in estimating the level of service provided by a network of facilities. Also, incorporating the existing facilities to the network design problem is notable as a contribution to the literature of interdiction and covering location problems.

This problem is formulated as a bilevel programming model, where the leader is the system planner and the attacker is the follower. We propose two solution methods for the problem. First, we develop a tabu search based heuristic method (TSH), which includes the application of two runs of tabu search sequentially. Each run searches one of the two critical patterns within the solution space corresponding to the system planner, while CPLEX is employed in order to optimize the attacker's subproblem. Secondly, we develop an exact algorithm based on binary enumeration tree (BET). BET algorithm branches over the solutions of the system planner and calls CPLEX to solve the attacker's subproblem. The tree size is reduced through an UB approach and consequently problems of small and medium sizes are solved to optimality in a reasonable amount of time by BET.

We test the performance of the TSH and BET procedures on 240 random problem instances derived from three different customer dispersions; real-world customer dispersion, uniform dispersion and clustered dispersion. Instances with $N_2=10$ and 15 are solved by both methods. On the basis of the comparisons, we conclude that the TSH is capable of producing high quality solutions to the BRICP problem. The maximum and average gap between the optimal and heuristic solutions are found to be 5.21% and 0.24%, respectively. Moreover, in 167 out of 180 instances the gap was lower than 1% while the TSH was capable of finding the optimal solutions for 137 problem instances. We conduct further computational experiments and present the strengths as well as the sensitivities of the algorithms in terms of computational performance. The solution time of BET increases exponentially with the number of potential sites. On the other hand, the performance of TSH is most sensitive to an increase in budget value where a higher number of iterations are required for good approximations to the optimal solutions.

A key finding is that the system planner's optimal strategy follows two patterns. The first one is to open a relatively small number of facilities and protect all of them, while the second one is to open a large number of facilities and protect only very few or none of them. We explain this behavior by the increasing marginal contribution of protection resources. Two types of protections are specified accordingly: "redirecting protections" and "preventive protections". The former represents the protection choices which aims to redirect the attacks to other facilities. The latter, on the other hand, follows the redirecting protections and limits the utilization of the attack resources by decreasing the number of unprotected facilities. As proven by empirical results, the preventive protections have higher marginal contribution to the system planner's objective than the redirecting protections.

As is the case in our problem, in bilevel interdiction problems complete and accurate information is assumed between the defender and the attacker. The cases where the information is inaccurate, on the other hand, are rarely addressed in the literature. Using our problem setting, we analyzed the value of accurate information to the system planner. We adapt the BRICP problem into the situation where the

system planner's presumption about the terrorist's resources for attacks is different than the actual resources of the terrorist. Consequently, it is found that the accuracy of the information about the attacker is critical to the system planner as inaccurate information causes a drastic deterioration in the final coverage. A further interesting finding is that the overestimation of terrorist resources is found to be more detrimental to the final coverage than their underestimation

The presence of relocation and protection options in our problem framework enables two additional remarkable analysis. First, relocation option is ignored. The solutions to this version show that the relocation of existing facilities contributes to the final coverage by 10% on average. Besides, system planner's best strategy involves another pattern, where all facilities are protected but the existing ones. Clearly, when locations are fixed, the existing facilities become strategically less important and their interdiction is tolerated by the system planner. The second analysis aims to approximate the relative value of protection against opening a new facility. It is found that the relative value of protection is highly dependent on customer dispersion where the protection of a critical facility, in case of clustered customer dispersion, is exceptionally valuable to the system planner.

We believe that bilevel interdiction covering problems deserve further endeavor of research to infer additional insights about defensive efforts in network design. In this respect, we suggest as further research the relaxation of discrete interdiction assumption in BRICP where facilities will not be destroyed completely after an attack, but will persist to provide service at a lower capacity level which will depend on the extent of the interdiction. Similarly, partial protection of facilities can be allowed. That is, the system planner can utilize the protection resources so that only a critical capacity of a facility is guaranteed to survive in case of an attack.

APPENDIX A: REAL WORLD CUSTOMER CONFIGURATIONS

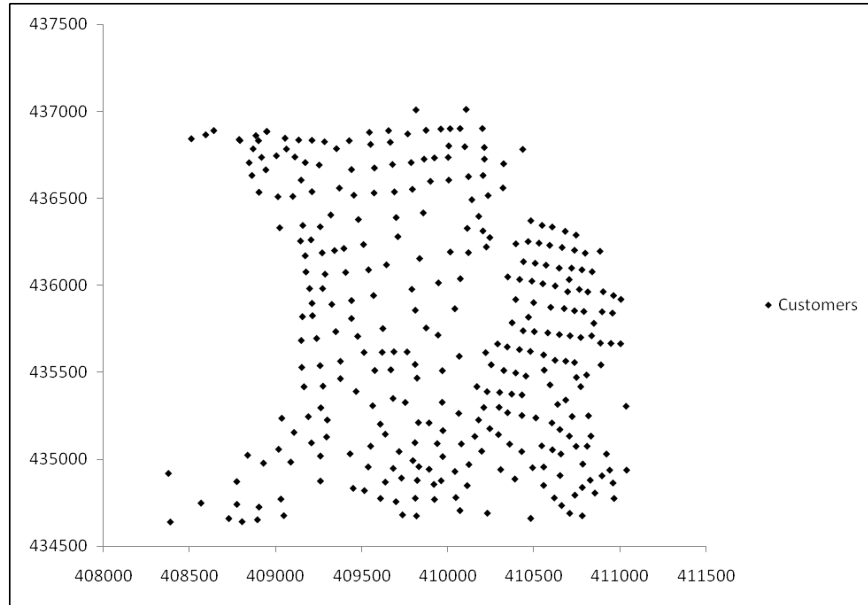


Figure A.1. SJC 324 customer configuration



Figure A.2. SJC 818 customer configuration

APPENDIX B: RANDOMLY GENERATED CUSTOMER CONFIGURATIONS

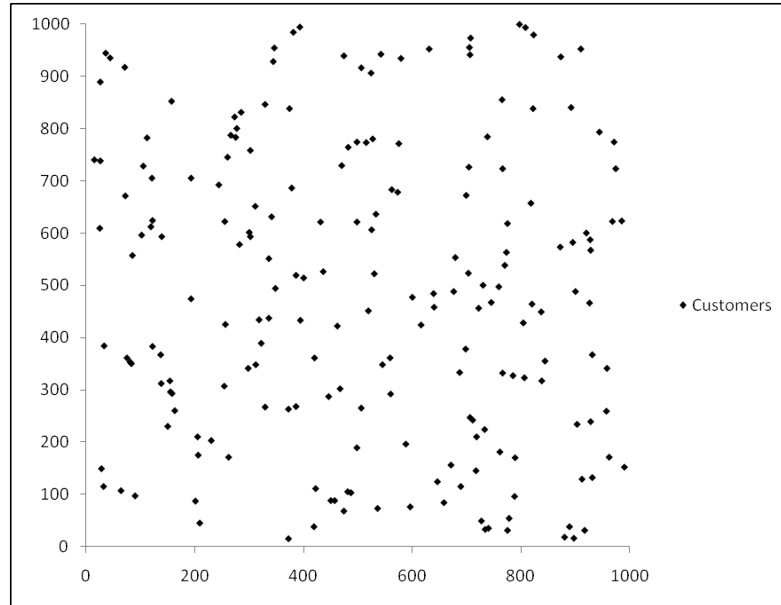


Figure B.1. U200 customer configuration

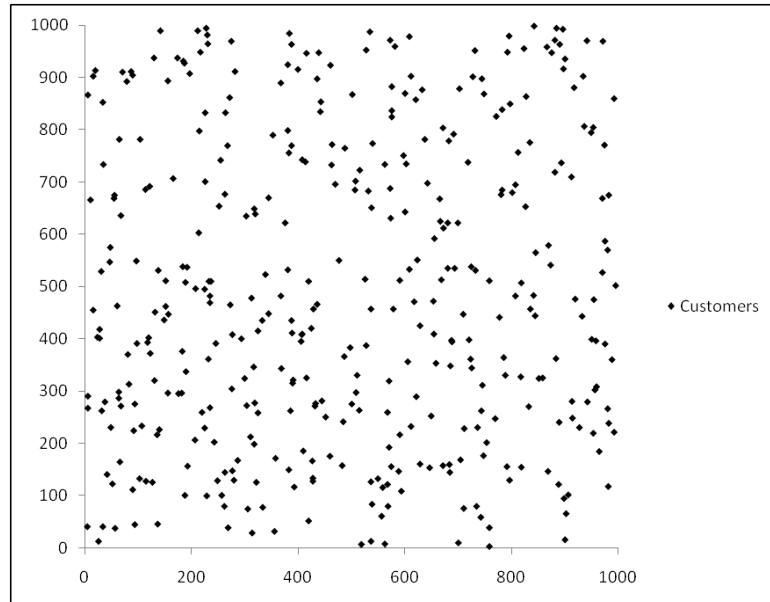


Figure B.2. U400 customer configuration

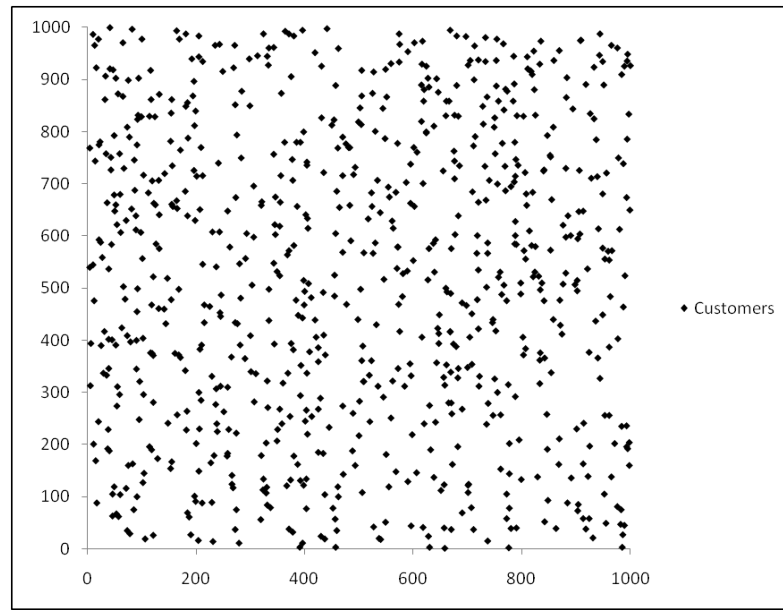


Figure B.3. U800 customer configuration

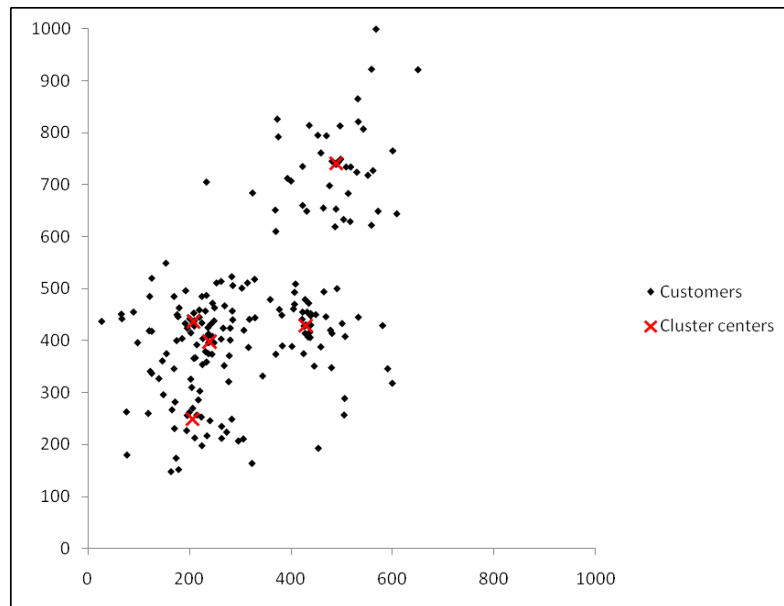


Figure B.4. C200 customer configuration

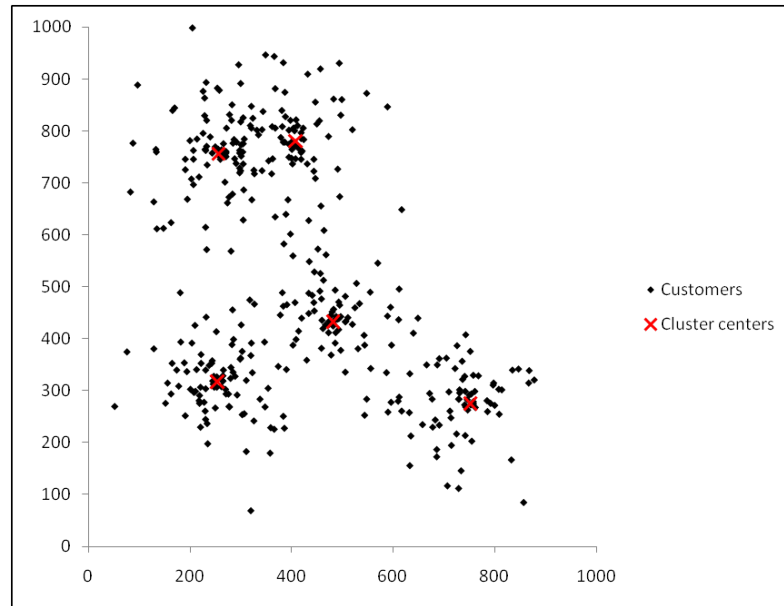


Figure B.5. C400 customer configuration

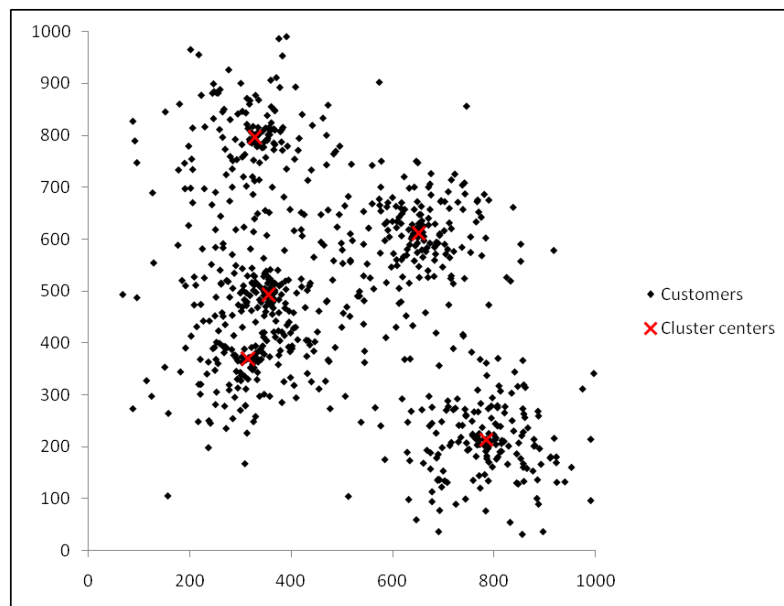


Figure B.6. C800 customer configuration

APPENDIX C: REAL WORLD CUSTOMER CONFIGURATIONS RESULTS

Table C.1. Solutions to problem instances based on SJC
324 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	3727.80	102	7	0	2	3727.80	56	0.00
10-2	4134.84	233	4	4	0	4134.84	69	0.00
10-3	4510.15	188	4	4	0	4510.15	92	0.00
10-4	4429.47	73	4	4	0	4429.47	53	0.00
10-5	4342.87	117	4	4	0	4342.87	89	0.00
10-6	4507.40	244	4	4	0	4507.40	85	0.00
10-7	3587.93	61	7	0	2	3587.93	57	0.00
10-8	3838.17	304	8	1	2	3838.17	75	0.00
10-9	4863.98	121	4	4	0	4863.98	71	0.00
10-10	4228.17	146	4	4	0	4228.17	87	0.00
15-1	4364.03	3400	4	4	0	4328.42	204	0.82
15-2	4558.63	1213	7	0	2	4558.63	211	0.00
15-3	4560.37	3066	4	4	0	4560.37	197	0.00
15-4	4397.98	1738	7	0	2	4397.98	286	0.00
15-5	4387.86	4360	8	0	2	4316.16	173	1.63
15-6	3824.93	3087	4	4	0	3824.93	176	0.00
15-7	3911.72	2440	8	0	2	3707.99	135	5.21
15-8	5087.84	3091	8	0	2	4829.50	194	5.08
15-9	4880.58	4411	7	2	2	4796.61	205	1.72
15-10	5169.68	3672	8	0	2	5169.68	219	0.00
20-1	—					4979.11	640	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table C.1. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-2	—					4041.34	574	—
20-3	—					5003.01	720	—
20-4	—					5178.47	437	—
20-5	—					4771.09	366	—
20-6	—					5254.05	688	—
20-7	—					4721.94	745	—
20-8	—					4125.54	736	—
20-9	—					5299.28	556	—
20-10	—					3486.44	626	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table C.2. Solutions to problem instances based on SJC
818 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	3549.49	882	5	4	1	3549.49	370	0.00
10-2	5886.37	924	5	4	1	5886.37	370	0.00
10-3	5759.05	856	4	4	0	5759.05	442	0.00
10-4	5416.88	634	4	4	0	5416.88	226	0.00
10-5	4428.10	1547	7	1	2	4428.10	294	0.00
10-6	5487.88	844	7	1	2	5487.88	232	0.00
10-7	4492.44	535	7	0	2	4492.44	204	0.00
10-8	3463.88	1920	5	4	1	3463.88	381	0.00
10-9	4656.29	499	4	4	0	4656.29	358	0.00
10-10	5929.36	678	4	4	0	5929.36	507	0.00
15-1	6608.20	12740	4	4	0	6608.20	901	0.00
15-2	4743.85	5539	4	4	0	4743.85	430	0.00
15-3	4428.24	8578	4	4	0	4428.24	847	0.00
15-4	5545.96	8429	8	0	2	5545.96	477	0.00
15-5	5311.43	4184	4	4	0	5311.43	982	0.00
15-6	6257.99	28695	7	1	2	6100.25	1,010	2.52
15-7	4823.98	6232	4	4	0	4823.98	528	0.00
15-8	6689.93	11521	4	4	0	6689.93	653	0.00
15-9	4115.00	3740	7	0	2	4115.00	552	0.00
15-10	6036.59	7108	4	4	0	6036.59	880	0.00
20-1	—					3815.62	1,223	—
20-2	—					5943.19	1,944	—
20-3	—					5916.81	760	—
20-4	—					6841.50	1,159	—
20-5	—					5929.31	1,562	—
20-6	—					4034.31	1,441	—
20-7	—					4514.36	1,244	—

Table C.2. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-8	—					5991.61	1,487	—
20-9	—					5405.72	2,169	—
20-10	—					4761.86	870	—

APPENDIX D: RANDOM CUSTOMER CONFIGURATIONS RESULTS

Table D.1. Solutions to problem instances based on U200 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	1427.09	150	4	4	0	1427.09	32	0.00
10-2	1575.21	219	7	1	2	1552.55	25	1.44
10-3	1496.00	105	7	0	2	1496.00	33	0.00
10-4	1340.27	89	4	4	0	1340.27	21	0.00
10-5	1426.72	59	4	4	0	1426.72	30	0.00
10-6	1489.73	92	4	4	0	1489.73	29	0.00
10-7	1541.39	83	7	1	2	1541.39	26	0.00
10-8	1339.07	69	7	0	2	1339.07	21	0.00
10-9	1340.48	77	4	4	0	1327.82	31	0.94
10-10	1566.47	138	8	0	2	1566.47	32	0.00
15-1	1535.67	1987	7	0	2	1535.67	76	0.00
15-2	1620.01	1159	8	0	2	1620.01	69	0.00
15-3	1388.88	1418	8	0	2	1364.44	80	1.76
15-4	1833.36	610	8	0	2	1833.33	44	0.00
15-5	1703.82	1108	8	0	2	1703.82	72	0.00
15-6	1677.48	1089	8	0	2	1677.48	60	0.00
15-7	1762.07	1787	8	0	2	1762.07	87	0.00
15-8	1805.71	930	8	0	2	1793.79	66	0.66
15-9	1519.84	852	4	4	0	1519.84	56	0.00
15-10	1429.41	2818	4	4	0	1407.91	89	1.50

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table D.1. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-1	—					1508.57	200	—
20-2	—					1639.55	103	—
20-3	—					1484.27	174	—
20-4	—					1762.15	98	—
20-5	—					1542.81	155	—
20-6	—					1911.88	107	—
20-7	—					1710.47	96	—
20-8	—					1853.43	153	—
20-9	—					1655.26	103	—
20-10	—					1419.33	143	—

Table D.2. Solutions to problem instances based on U400 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	3085.19	157	8	0	2	3085.19	41	0.00
10-2	2532.94	89	7	1	2	2499.27	64	1.33
10-3	2780.09	165	7	0	2	2780.09	67	0.00
10-4	2640.76	182	7	0	2	2640.76	70	0.00
10-5	2420.19	130	7	0	2	2420.19	73	0.00
10-6	2767.29	58	7	0	2	2767.29	76	0.00
10-7	2502.38	377	8	0	2	2502.38	78	0.00
10-8	2562.12	211	7	1	2	2562.12	86	0.00
10-9	2295.64	276	8	0	2	2196.93	88	4.30
10-10	2756.96	289	8	0	2	2756.96	69	0.00
15-1	2836.21	2400	7	0	2	2836.21	127	0.00
15-2	3214.17	1894	8	0	2	3214.17	137	0.00
15-3	2967.21	3724	8	0	2	2967.21	201	0.00
15-4	3290.37	2698	8	0	2	3290.37	186	0.00
15-5	2810.31	4390	8	1	2	2785.36	139	0.89
15-6	3187.13	3179	8	0	2	3060.86	87	3.96
15-7	3304.89	1995	8	0	2	3304.89	126	0.00
15-8	2844.81	2547	8	0	2	2844.81	181	0.00
15-9	2816.47	2618	8	0	2	2816.47	136	0.00
15-10	3015.87	1379	8	0	2	3015.87	130	0.00
20-1	—					3229.70	203	—
20-2	—					3139.17	202	—
20-3	—					3059.98	283	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table D.2. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-4	—					3518.14	433	—
20-5	—					2865.69	343	—
20-6	—					3203.65	201	—
20-7	—					3465.28	273	—
20-8	—					2506.16	338	—
20-9	—					2839.49	229	—
20-10	—					3086.37	433	—

Table D.3. Solutions to problem instances based on U800 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	5670.82	481	8	0	2	5670.82	127	0.00
10-2	4384.49	418	4	4	0	4384.49	193	0.00
10-3	6253.06	360	8	0	2	6253.06	181	0.00
10-4	5163.88	814	4	4	0	5163.88	233	0.00
10-5	4507.04	503	7	0	2	4507.04	172	0.00
10-6	5729.79	461	8	0	2	5729.79	154	0.00
10-7	4906.62	653	4	4	0	4906.62	168	0.00
10-8	5958.82	648	8	0	2	5958.82	159	0.00
10-9	5650.91	567	8	0	2	5537.28	158	2.01
10-10	5098.43	576	8	0	2	5098.43	196	0.00
15-1	5687.56	2471	7	0	2	5687.56	381	0.00
15-2	5392.69	8901	8	0	2	5392.69	404	0.00
15-3	5703.02	4790	8	0	2	5570.58	427	2.32
15-4	6586.12	5145	8	0	2	6586.12	383	0.00
15-5	6106.78	10260	8	0	2	6106.78	495	0.00
15-6	6777.22	4710	8	0	2	6777.22	322	0.00
15-7	6345.27	6361	8	0	2	6345.27	313	0.00
15-8	5334.93	5938	8	0	2	5332.10	371	0.05
15-9	5792.45	4735	8	0	2	5792.45	229	0.00
15-10	6441.96	5794	9	0	2	6441.96	429	0.00
20-1	—					6120.20	703	—
20-2	—					6146.20	681	—
20-3	—					6848.84	925	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table D.3. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-4	—					5421.27	1,017	—
20-5	—					6187.12	999	—
20-6	—					5519.19	576	—
20-7	—					6442.53	807	—
20-8	—					6401.16	898	—
20-9	—					6079.94	1,115	—
20-10	—					5859.88	667	—

Table D.4. Solutions to problem instances based on C200 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	4109.95	80	4	4	0	4109.95	50	0.00
10-2	4405.27	239	4	4	0	4405.27	58	0.00
10-3	4235.89	166	5	4	1	4235.89	71	0.00
10-4	3925.42	29	4	4	0	3899.93	62	0.65
10-5	4661.01	52	4	4	0	4661.01	58	0.00
10-6	4646.23	191	4	4	0	4646.23	66	0.00
10-7	3935.72	422	5	4	1	3933.72	58	0.05
10-8	4472.88	92	4	4	0	4472.88	54	0.00
10-9	4818.70	60	4	4	0	4818.70	65	0.00
10-10	3613.83	151	4	4	0	3613.83	54	0.00
15-1	4246.29	2334	4	4	0	4211.63	138	0.82
15-2	4229.37	1083	4	4	0	4228.52	121	0.02
15-3	4228.76	891	4	4	0	4228.76	124	0.00
15-4	4519.04	2578	4	4	0	4519.04	130	0.00
15-5	4444.95	967	4	4	0	4444.95	123	0.00
15-6	4005.26	1365	4	4	0	4005.26	119	0.00
15-7	4733.76	1772	4	4	0	4733.76	163	0.00
15-8	4778.99	1613	4	4	0	4778.99	155	0.00
15-9	4776.78	1490	4	4	0	4776.78	161	0.00
15-10	4717.04	1087	4	4	0	4717.04	159	0.00
20-1	—					4634.09	253	—
20-2	—					4821.41	218	—
20-3	—					3992.47	259	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table D.4. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-4	—					4716.38	315	—
20-5	—					4497.45	227	—
20-6	—					4479.32	293	—
20-7	—					4689.33	205	—
20-8	—					4741.88	224	—
20-9	—					4663.43	233	—
20-10	—					4425.73	181	—

Table D.5. Solutions to problem instances based on C400 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	6479.93	155	4	4	0	6479.93	135	0.00
10-2	7783.98	196	4	4	0	7783.98	89	0.00
10-3	7128.68	177	4	4	0	7128.68	123	0.00
10-4	7025.69	93	4	4	0	7025.69	118	0.00
10-5	6318.00	145	4	4	0	6318.00	98	0.00
10-6	7106.78	248	4	4	0	7106.78	125	0.00
10-7	8197.68	393	4	4	0	8197.68	123	0.00
10-8	7046.98	123	4	4	0	7046.98	103	0.00
10-9	6751.94	242	4	4	0	6751.94	106	0.00
10-10	7609.32	238	4	4	0	7609.32	133	0.00
15-1	8084.03	1407	4	4	0	8084.03	293	0.00
15-2	8001.76	1334	4	4	0	8001.76	276	0.00
15-3	8226.97	2643	4	4	0	8226.97	258	0.00
15-4	7483.03	2538	4	4	0	7483.03	274	0.00
15-5	7004.11	3881	4	4	0	7004.11	348	0.00
15-6	7796.35	2582	4	4	0	7796.35	168	0.00
15-7	7748.70	4028	4	4	0	7748.70	299	0.00
15-8	7358.36	1612	4	4	0	7358.36	293	0.00
15-9	8011.98	3071	4	4	0	8011.98	254	0.00
15-10	8006.48	1988	4	4	0	8006.48	285	0.00
20-1	—					8049.51	611	—
20-2	—					8459.14	561	—
20-3	—					7850.95	557	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table D.5. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-4	—					8384.88	452	—
20-5	—					8236.59	553	—
20-6	—					7976.71	520	—
20-7	—					7576.33	522	—
20-8	—					7030.19	557	—
20-9	—					8100.52	616	—
20-10	—					7303.79	725	—

Table D.6. Solutions to problem instances based on C800 customer configuration

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF ¹	#PF ²	#IF ³	Coverage	CPU	
10-1	12717.37	541	4	4	0	12717.37	324	0.00
10-2	12651.21	306	4	4	0	12651.21	274	0.00
10-3	12478.82	835	4	4	0	12478.82	336	0.00
10-4	12368.53	738	4	4	0	12368.53	355	0.00
10-5	13499.00	862	4	4	0	13499.00	282	0.00
10-6	12883.43	206	4	4	0	12883.43	204	0.00
10-7	13610.22	751	4	4	0	13610.22	279	0.00
10-8	10511.73	780	4	3	1	10511.73	265	0.00
10-9	13371.17	489	4	4	0	13371.17	302	0.00
10-10	12546.28	483	4	4	0	12546.28	277	0.00
15-1	13153.76	3657	4	4	0	13153.76	987	0.00
15-2	12321.28	2960	4	4	0	12321.28	757	0.00
15-3	14867.64	9290	4	4	0	14867.64	1,004	0.00
15-4	13981.77	3452	4	4	0	13981.77	670	0.00
15-5	13371.63	4576	4	4	0	13371.63	631	0.00
15-6	12956.14	6454	4	4	0	12956.14	506	0.00
15-7	12119.15	8177	4	4	0	12119.15	682	0.00
15-8	13003.24	11328	4	4	0	13003.24	715	0.00
15-9	14114.68	4969	4	4	0	14114.68	670	0.00
15-10	13026.65	4195	4	4	0	13026.65	514	0.00
20-1	—					14994.66	2,066	—
20-2	—					11368.78	1,242	—
20-3	—					13940.98	1,472	—

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table D.6. Continued

Instance	BET					TSH		% Gap
	Coverage	CPU	#OF	#PF	#IF	Coverage	CPU	
20-4	—					14949.01	1,224	—
20-5	—					14470.68	1,898	—
20-6	—					13624.61	1,548	—
20-7	—					14625.47	1,248	—
20-8	—					14577.94	1,703	—
20-9	—					12747.33	1,536	—
20-10	—					12712.50	1,469	—

APPENDIX E: RESULTS OF THE TEST ABOUT K

Table E.1. Summarized overall results according to the number of potential facility locations

Instance Group	Average CPU		Average % Gap
	TSH	BET	
Uniform			
	10	91.10 283.20	0.32
	15	196.77 3322.90	0.36
	20	421.93	
Clustered			
	10	154.90 316.10	0.02
	15	375.90 3310.73	0.03
	20	782.93	

APPENDIX F: RESULTS ACCORDING TO THE NUMBER OF CUSTOMERS

Table F.1. Summarized overall results according to the number of customers

Instance Group	Average CPU		Average % Gap
	TSH	BET	
Uniform			
200	77.03	741.95	0.30
400	170.00	1437.90	0.50
800	462.77	3229.30	0.22
Clustered			
200	146.57	833.10	0.07
400	319.17	1354.70	0.00
800	848.00	3252.45	0.00

APPENDIX G: RESULTS OF THE TEST ABOUT BUDGET

Table G.1. Detailed results of the test about effects of
budget on instances of SJC324 configuration

Instance	BET		TSH		% Gap
	Coverage	CPU	Coverage	CPU	
b=25000					
10-7	3086.71	43	3086.71	59	0.00
10-8	3520.69	116	3520.69	67	0.00
10-9	3971.18	151	3971.18	64	0.00
10-10	3659.11	112	3659.11	99	0.00
15-7	3491.52	1110	3491.52	123	0.00
15-8	4224.33	2069	4221.25	224	0.07
15-9	4060.43	2484	4007.32	193	1.31
15-10	4351.00	2294	4351.01	185	0.00
20-7	–		4243.09	368	–
20-8	–		3555.16	387	–
20-9	–		4734.19	366	–
20-10	–		3075.54	376	–
b=30000					
10-7	3587.93	62	3587.93	58	0.00
10-8	3838.16	312	3838.16	75	0.00
10-9	4863.98	126	4863.98	69	0.00
10-10	4228.16	144	4228.16	88	0.00
15-7	3911.71	2307	3707.99	127	5.21
15-8	5087.83	2873	4844.75	232	4.78
15-9	4880.57	4351	4796.61	246	1.72
15-10	5169.67	3483	5169.67	202	0.00
20-7	–		4721.94	433	–

Table G.1. Continued

Instance	BET		TSH		% Gap
	Coverage	CPU	Coverage	CPU	
20-8	—		4125.53	491	—
20-9	—		5299.28	394	—
20-10	—		3486.43	440	—
b=35000					
10-7	3923.56	216	3923.56	61	0.00
10-8	4203.25	292	4203.25	73	0.00
10-9	4984.51	295	4984.51	83	0.00
10-10	4564.27	276	4271.85	103	6.41
15-7	4190.24	5863	4110.14	186	1.91
15-8	5779.58	5364	5583.73	245	3.39
15-9	5524.87	8973	5357.84	273	3.02
15-10	5959.16	7971	5959.16	235	0.00
20-7	—		5251.02	520	—
20-8	—		4608.97	379	—
20-9	—		5983.96	401	—
20-10	—		4025.75	435	—

APPENDIX H: RESULTS OF THE TEST FOR r

Table H.1. Detailed test results for r on instances of
SJC818 configuration

r	Instance	BET		TSH		% Gap
		Coverage	CPU	Coverage	CPU	
$r = 1$	10-1	3948.37	581	3948.37	248	0
	10-2	6572.11	765	6572.11	369	0
	10-3	6638.98	678	6638.98	294	0
	10-4	5743.85	478	5743.85	149	0
	10-5	5231.89	747	5231.89	195	0
	15-1	7953.90	9303	7953.90	664	0
	15-2	5321.10	6189	4985.89	192	6.30
	15-3	5007.07	10400	5007.07	660	0
	15-4	7001.65	5165	7001.65	182	0
	15-5	6235.92	3022	6235.92	454	0
	20-1	—		4442.39	1749	—
	20-2	—		6347.58	1351	—
	20-3	—		7167.54	598	—
	20-4	—		7758.09	929	—
	20-5	—		7100.31	888	—
$r = 2$	10-1	3549.49	882	3549.49	370	0
	10-2	5886.37	924	5886.37	370	0
	10-3	5759.05	856	5759.05	442	0
	10-4	5416.88	634	5416.88	226	0
	10-5	4428.10	1,547	4428.10	294	0
	15-1	6608.20	12,740	6608.20	901	0
	15-2	4743.85	5,539	4743.85	430	0
	15-3	4428.24	8,578	4428.24	847	0
	15-4	5545.96	8,429	5545.96	477	0
	15-5	5311.43	4,184	5311.43	982	0

Table H.1. Continued

r	Instance	BET		TSH		% Gap
		Coverage	CPU	Coverage	CPU	
	20-1	—		3815.62	1,223	—
	20-2	—		5943.19	1,944	—
	20-3	—		5916.81	760	—
	20-4	—		6841.50	1,159	—
	20-5	—		5929.31	1,562	—
$r = 3$	10-1	3549.49	397	3549.49	236	0
	10-2	5886.37	439	5886.37	221	0
	10-3	5759.05	377	5759.05	237	0
	10-4	5416.88	364	5416.88	151	0
	10-5	4386.53	354	4386.53	207	0
	15-1	6608.20	6465	6608.20	436	0
	15-2	4743.85	3879	4743.85	263	0
	15-3	4428.24	4307	4428.24	372	0
	15-4	5431.49	5675	5431.49	508	0
	15-5	5311.43	2702	5311.43	628	0
	20-1	—		3815.62	1131	—
	20-2	—		5943.19	1516	—
	20-3	—		5805.17	849	—
	20-4	—		6841.50	968	—
	20-5	—		5236.95	1165	—

Table H.2. Summarized results of test for r on instances of SJC818 configuration

Instance group	Average Coverage		Average CPU		Average Gap %
	TSH	BET	TSH	BET	
$r = 1$					
$N_2 = 10$	5627.04	5627.04	251.00	649.80	0.00
$N_2 = 15$	6236.93	6303.93	430.40	6815.80	1.06
$N_2 = 20$	6563.18	—	1103.00	—	—
$r = 2$					
$N_2 = 10$	5007.98	5007.98	340.40	968.60	0.00
$N_2 = 15$	5327.53	5327.53	727.40	7894.00	0.00
$N_2 = 20$	5689.29	—	1329.60	—	—
$r = 3$					
$N_2 = 10$	4999.66	4999.66	210.40	386.20	0.00
$N_2 = 15$	5304.64	5304.64	441.40	4605.60	0.00
$N_2 = 20$	5528.49	—	1125.80	—	—

Table H.3. Average number of protections and attacks

	$r = 0$	$r = 1$	$r = 2$	$r = 3$
Av. # of attacked facilities	0.00	1.00	0.73	0.00
Av. # of protected facilities	0.00	0.47	2.60	3.97

APPENDIX I: TEST RESULTS OF THE PROTECTION RESOURCES

Table I.1. Detailed test results with protection limit
where the locations are fixed on instances of SJC324

Instance	Prot. limit	Coverage	#OF ¹	#PF ²	#IF ³
10-2	0	1711.68	4	0	2
10-2	1	2161.49	4	1	2
10-2	2	2423.16	4	2	2
10-2	3	3310.11	4	3	1
10-2	4	4134.84	4	4	0
10-3	0	1768.83	4	0	2
10-3	1	2259.46	4	1	2
10-3	2	2741.32	4	2	2
10-3	3	3691.64	4	3	1
10-3	4	4510.15	4	4	0
10-4	0	1995.43	4	0	2
10-4	1	2434.05	4	1	2
10-4	2	2637.79	4	2	2
10-4	3	3593.17	4	3	1
10-4	4	4429.47	4	4	0
10-5	0	2144.07	4	0	2
10-5	1	2173.84	4	1	2
10-5	2	2389.92	4	2	2
10-5	3	3454.93	4	3	1
10-5	4	4342.87	4	4	0
10-6	0	2103.14	4	0	2

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

Table I.1. Continued

Instance	Prot. limit	Coverage	#OF	#PF	#IF
10-6	1	2404.26	4	1	2
10-6	2	2497.07	4	2	2
10-6	3	3632.52	4	3	1
10-6	4	4507.40	4	4	0
10-9	0	1923.48	4	0	2
10-9	1	2490.54	4	1	2
10-9	2	2940.50	4	2	2
10-9	3	3971.19	4	3	1
10-9	4	4863.98	4	4	0
10-10	0	1547.65	4	0	2
10-10	1	2509.13	4	1	2
10-10	2	2680.52	4	2	2
10-10	3	3494.40	4	3	1
10-10	4	4228.17	4	4	0
15-3	0	1909.11	4	0	2
15-3	1	2180.40	4	1	2
15-3	2	2651.26	4	2	2
15-3	3	3722.72	4	3	1
15-3	4	4560.37	4	4	0
15-5	0	2014.26	4	0	2
15-5	1	2248.39	4	1	2
15-5	2	2301.91	4	2	2
15-5	3	3310.85	4	3	1
15-5	4	4316.16	4	4	0
15-6	0	1651.32	4	0	2
15-6	1	1904.39	4	1	2
15-6	2	2240.71	4	2	2
15-6	3	3098.59	4	3	1

Table I.1. Continued

Instance	Prot. limit	Coverage	#OF	#PF	#IF
15-6	4	3824.93	4	4	0
15-9	0	1835.17	4	0	2
15-9	1	2507.93	4	1	2
15-9	2	2961.44	4	2	2
15-9	3	3886.42	4	3	1
15-9	4	4796.61	4	4	0
20-1	0	2099.93	4	0	2
20-1	1	2385.22	4	1	2
20-1	2	2890.10	4	2	2
20-1	3	4100.83	4	3	1
20-1	4	4979.11	4	4	0
20-2	0	1693.51	4	0	2
20-2	1	1891.40	4	1	2
20-2	2	2347.82	4	2	2
20-2	3	3368.51	4	3	1
20-2	4	4041.34	4	4	0
20-5	0	1830.59	4	0	2
20-5	1	2425.09	4	1	2
20-5	2	2940.50	4	2	2
20-5	3	3948.39	4	3	1
20-5	4	4771.09	4	4	0
20-8	0	1743.77	4	0	2
20-8	1	2142.45	4	1	2
20-8	2	2381.76	4	2	2
20-8	3	3332.08	4	3	1
20-8	4	4125.54	4	4	0
20-9	0	2339.71	4	0	2
20-9	1	2825.51	4	1	2

Table I.1. Continued

Instance	Prot. limit	Coverage	#OF	#PF	#IF
20-9	2	3014.01	4	2	2
20-9	3	4161.84	4	3	1
20-9	4	5299.28	4	4	0

Table I.2. Summarized results according to the marginal contribution of each protection

	Average Coverage	Marginal contribution
No protection	1894.48	—
1 st protection	2308.97	414.49
2 nd protection	2627.49	318.52
3 rd protection	3629.89	1002.40
4 th protection	4483.21	853.32

Table I.3. Summarized results according to the average contribution per protection

	Average Coverage	Average contribution
No protection	1894.48	—
1 st protection	2308.97	414.49
2 nd protection	2627.49	366.50
3 rd protection	3629.89	578.47
4 th protection	4483.21	647.18

APPENDIX J: RESULTS OF INACCURATE KNOWLEDGE EFFECTS

Table J.1. Detailed results on inaccurate knowledge on
instances of SJC324 configuration

Instance	r_p ¹	r_a ¹	Coverage	#OF ³	#PF ⁴	#IF ⁵
10-1	0	0	5492.01	7	0	0
10-1	0	1	4509.66	7	0	1
10-1	0	2	3583.35	7	0	2
10-1	0	3	2734.30	7	0	3
10-2	0	0	6037.84	8	0	0
10-2	0	1	4701.08	8	0	1
10-2	0	1	4664.38	4	0	3
10-2	0	2	3814.13	8	0	2
10-2	0	2	3829.85	4	0	3
10-2	0	3	2989.40	8	0	3
10-2	0	3	2611.27	4	0	4
10-3	0	0	6413.03	7	0	0
10-3	0	1	5112.66	7	0	1
10-3	0	2	3816.16	7	0	2
10-3	0	3	2865.84	7	0	3
10-4	0	0	6498.32	7	0	0
10-4	0	1	5175.82	7	0	1
10-4	0	2	4064.27	7	0	2
10-4	0	3	3068.09	7	0	3
10-5	0	0	6124.63	7	0	0
10-5	0	1	4990.83	7	0	1

¹presumed r

²actual r

³# of open facilities (total of existing and newly opened facilities)

⁴# of protected facilities

⁵# of interdicted facilities

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
10-5	0	2	3925.83	7	0	2
10-5	0	3	2898.78	7	0	3
15-1	0	0	6782.37	7	0	0
15-1	0	1	5118.27	7	0	1
15-1	0	2	3902.11	7	0	2
15-1	0	3	2974.35	7	0	3
15-2	0	0	7178.42	7	0	0
15-2	0	1	5704.67	7	0	1
15-2	0	2	4263.72	7	0	2
15-2	0	3	3164.19	7	0	3
15-3	0	0	6482.13	8	0	0
15-3	0	1	5282.69	8	0	1
15-3	0	2	4369.50	8	0	2
15-3	0	3	3311.01	8	0	3
15-4	0	0	6895.11	7	0	0
15-4	0	1	5526.99	7	0	1
15-4	0	2	4394.05	7	0	2
15-4	0	3	3280.27	7	0	3
15-5	0	0	6632.69	7	0	0
15-5	0	1	5389.62	7	0	1
15-5	0	2	4330.78	7	0	2
15-5	0	3	3325.46	7	0	3
20-1	0	0	7400.21	8	0	0
20-1	0	1	6047.97	8	0	1
20-1	0	2	4876.61	8	0	2
20-1	0	3	3722.07	8	0	3
20-2	0	0	5919.77	7	0	0
20-2	0	1	4664.38	7	0	1
20-2	0	2	3829.85	7	0	2

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
20-2	0	3	2611.27	7	0	3
20-3	0	0	7504.15	8	0	0
20-3	0	1	6003.94	8	0	1
20-3	0	2	5003.01	8	0	2
20-3	0	3	4022.68	8	0	3
20-4	0	0	7834.33	8	0	0
20-4	0	1	6124.13	8	0	1
20-4	0	2	5148.72	8	0	2
20-4	0	3	4308.15	8	0	3
20-5	0	0	6919.67	7	0	0
20-5	0	1	5321.93	7	0	1
20-5	0	2	3979.18	7	0	2
20-5	0	3	2796.11	7	0	3
10-1	1	0	5392.71	7	0	0
10-1	1	1	4554.20	7	0	1
10-1	1	2	3727.80	7	0	2
10-1	1	3	2876.63	7	0	3
10-2	1	0	5799.56	7	1	0
10-2	1	1	4912.61	7	1	1
10-2	1	1	4707.02	4	0	2
10-2	1	2	4087.88	7	1	2
10-2	1	2	3832.55	4	0	2
10-2	1	3	3352.74	7	1	3
10-2	1	3	3045.40	4	0	2
10-3	1	0	6413.03	7	0	0
10-3	1	1	5112.66	7	0	1
10-3	1	2	3816.16	7	0	2
10-3	1	3	2865.84	7	0	3
10-4	1	0	6498.32	7	0	0

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
10-4	1	1	5175.82	7	0	1
10-4	1	2	4064.27	7	0	2
10-4	1	3	3068.09	7	0	3
10-5	1	0	6124.63	7	0	0
10-5	1	1	4990.83	7	0	1
10-5	1	2	3925.83	7	0	2
10-5	1	3	2898.78	7	0	3
15-1	1	0	6453.23	7	0	0
15-1	1	1	5256.18	7	0	1
15-1	1	2	4328.42	7	0	2
15-1	1	3	3347.66	7	0	3
15-2	1	0	7178.42	7	0	0
15-2	1	1	5704.67	7	0	1
15-2	1	2	4263.72	7	0	2
15-2	1	3	3164.19	7	0	3
15-3	1	0	6516.42	8	0	0
15-3	1	1	5447.14	8	0	1
15-3	1	2	4533.95	8	0	2
15-3	1	3	3475.45	8	0	3
15-4	1	0	6895.11	7	0	0
15-4	1	1	5526.99	7	0	1
15-4	1	2	4394.05	7	0	2
15-4	1	3	3280.27	7	0	3
15-5	1	0	6689.77	8	0	0
15-5	1	1	5446.69	8	0	1
15-5	1	2	4387.86	8	0	2
15-5	1	3	3382.54	8	0	3
20-1	1	0	7400.21	8	0	0
20-1	1	1	6047.97	8	0	1

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
20-1	1	2	4876.61	8	0	2
20-1	1	3	3722.07	8	0	3
20-2	1	0	5662.57	7	0	0
20-2	1	1	4707.02	7	0	1
20-2	1	2	3832.55	7	0	2
20-2	1	3	3045.40	7	0	3
20-3	1	0	7416.50	8	0	0
20-3	1	1	6072.94	8	0	1
20-3	1	2	4986.66	8	0	2
20-3	1	3	4006.32	8	0	3
20-4	1	0	7277.19	7	1	0
20-4	1	1	6338.62	7	1	1
20-4	1	2	5261.41	7	1	2
20-4	1	3	4272.44	7	1	3
20-5	1	0	6919.67	7	0	0
20-5	1	1	5321.93	7	0	1
20-5	1	2	3979.18	7	0	2
20-5	1	3	2796.11	7	0	3
10-1	2	0	5392.71	7	0	0
10-1	2	1	4554.20	7	0	1
10-1	2	2	3727.80	7	0	2
10-1	2	3	2876.63	7	0	3
10-2	2	0	4134.84	4	4	0
10-2	2	1	4134.84	4	4	0
10-2	2	1	4041.34	3	3	0
10-2	2	2	4134.84	4	4	0
10-2	2	2	4041.34	3	3	0
10-2	2	3	4134.84	4	4	0
10-2	2	3	4041.34	3	3	0

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
10-3	2	0	4510.15	4	4	0
10-3	2	1	4510.15	4	4	0
10-3	2	2	4510.15	4	4	0
10-3	2	3	4510.15	4	4	0
10-4	2	0	4429.47	4	4	0
10-4	2	1	4429.47	4	4	0
10-4	2	2	4429.47	4	4	0
10-4	2	3	4429.47	4	4	0
10-5	2	0	4342.87	4	4	0
10-5	2	1	4342.87	4	4	0
10-5	2	2	4342.87	4	4	0
10-5	2	3	4342.87	4	4	0
15-1	2	0	6094.62	7	0	0
15-1	2	1	5166.86	7	0	1
15-1	2	2	4328.42	7	0	2
15-1	2	3	3347.66	7	0	3
15-2	2	0	6621.91	7	0	0
15-2	2	1	5573.08	7	0	1
15-2	2	2	4558.63	7	0	2
15-2	2	3	3641.93	7	0	3
15-3	2	0	4560.37	4	4	0
15-3	2	1	4560.37	4	4	0
15-3	2	2	4560.37	4	4	0
15-3	2	3	4560.37	4	4	0
15-4	2	0	6879.88	7	0	0
15-4	2	1	5511.76	7	0	1
15-4	2	2	4397.98	7	0	2
15-4	2	3	3457.86	7	0	3
15-5	2	0	4316.16	4	4	0

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
15-5	2	1	4316.16	4	4	0
15-5	2	2	4316.16	4	4	0
15-5	2	3	4316.16	4	4	0
20-1	2	0	4979.11	4	4	0
20-1	2	1	4979.11	4	4	0
20-1	2	2	4979.11	4	4	0
20-1	2	3	4979.11	4	4	0
20-2	2	0	4041.34	4	4	0
20-2	2	1	4041.34	4	4	0
20-2	2	2	4041.34	4	4	0
20-2	2	3	4041.34	4	4	0
20-3	2	0	7412.45	8	0	0
20-3	2	1	6068.90	8	0	1
20-3	2	2	5067.96	8	0	2
20-3	2	3	4087.63	8	0	3
20-4	2	0	7356.72	8	0	0
20-4	2	1	6156.15	8	0	1
20-4	2	2	5178.47	8	0	2
20-4	2	3	4203.05	8	0	3
20-5	2	0	4771.09	4	4	0
20-5	2	1	4771.09	4	4	0
20-5	2	2	4771.09	4	4	0
20-5	2	3	4771.09	4	4	0
10-1	3	0	3594.15	4	4	0
10-1	3	1	3594.15	4	4	0
10-1	3	2	3594.15	4	4	0
10-1	3	3	3594.15	4	4	0
10-2	3	0	4134.84	4	4	0
10-2	3	1	4134.84	4	4	0

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
10-2	3	1	4041.34	3	3	0
10-2	3	2	4134.84	4	4	0
10-2	3	2	4041.34	3	3	0
10-2	3	3	4134.84	4	4	0
10-2	3	3	4041.34	3	3	0
10-3	3	0	4510.15	4	4	0
10-3	3	1	4510.15	4	4	0
10-3	3	2	4510.15	4	4	0
10-3	3	3	4510.15	4	4	0
10-4	3	0	4429.47	4	4	0
10-4	3	1	4429.47	4	4	0
10-4	3	2	4429.47	4	4	0
10-4	3	3	4429.47	4	4	0
10-5	3	0	4342.87	4	4	0
10-5	3	1	4342.87	4	4	0
10-5	3	2	4342.87	4	4	0
10-5	3	3	4342.87	4	4	0
15-1	3	0	4325.36	4	4	0
15-1	3	1	4325.36	4	4	0
15-1	3	2	4325.36	4	4	0
15-1	3	3	4325.36	4	4	0
15-2	3	0	4369.13	4	4	0
15-2	3	1	4369.13	4	4	0
15-2	3	2	4369.13	4	4	0
15-2	3	3	4369.13	4	4	0
15-3	3	0	4560.37	4	4	0
15-3	3	1	4560.37	4	4	0
15-3	3	2	4560.37	4	4	0
15-3	3	3	4560.37	4	4	0

Table J.1. Continued

Instance	r_p	r_a	Coverage	#OF	#PF	#IF
15-4	3	0	4345.47	4	4	0
15-4	3	1	4345.47	4	4	0
15-4	3	2	4345.47	4	4	0
15-4	3	3	4345.47	4	4	0
15-5	3	0	4316.16	4	4	0
15-5	3	1	4316.16	4	4	0
15-5	3	2	4316.16	4	4	0
15-5	3	3	4316.16	4	4	0
20-1	3	0	4979.11	4	4	0
20-1	3	1	4979.11	4	4	0
20-1	3	2	4979.11	4	4	0
20-1	3	3	4979.11	4	4	0
20-2	3	0	4041.34	4	4	0
20-2	3	1	4041.34	4	4	0
20-2	3	2	4041.34	4	4	0
20-2	3	3	4041.34	4	4	0
20-3	3	0	4709.82	4	4	0
20-3	3	1	4709.82	4	4	0
20-3	3	2	4709.82	4	4	0
20-3	3	3	4709.82	4	4	0
20-4	3	0	5029.03	4	4	0
20-4	3	1	5029.03	4	4	0
20-4	3	2	5029.03	4	4	0
20-4	3	3	5029.03	4	4	0
20-5	3	0	4771.09	4	4	0
20-5	3	1	4771.09	4	4	0
20-5	3	2	4771.09	4	4	0
20-5	3	3	4771.09	4	4	0

Table J.2. Summarized results according to estimation error

Estimation			Average Coverage		Average	Final Average
error	r_p	r_a	$r_p \neq r_a$	$r_p = r_a$	Loss	Loss
-3	3	0	3204.80	4430.56	1225.76	1225.76
-2	2	0	4220.08	4489.64	269.56	698.24
	3	1	3303.63	4430.56	1126.92	
-1	1	0	5311.64	5374.42	62.77	190.63
	2	1	4297.76	4489.64	191.89	
	3	2	4113.34	4430.56	317.21	
0	0	0	6674.31	6674.31	0.00	0.00
	1	1	5374.42	5374.42	0.00	
	2	2	4489.64	4489.64	0.00	
	3	3	4430.56	4430.56	0.00	
1	2	3	4430.56	4489.64	59.09	219.19
	1	2	4874.42	5374.42	500.00	
	0	1	6575.82	6674.31	98.49	
2	1	3	4430.56	5374.42	943.86	1147.63
	0	2	5322.91	6674.31	1351.40	
3	0	3	4430.56	6674.31	2243.76	2243.76

APPENDIX K: ILLUSTRATIONS OF EXAMPLE SOLUTIONS

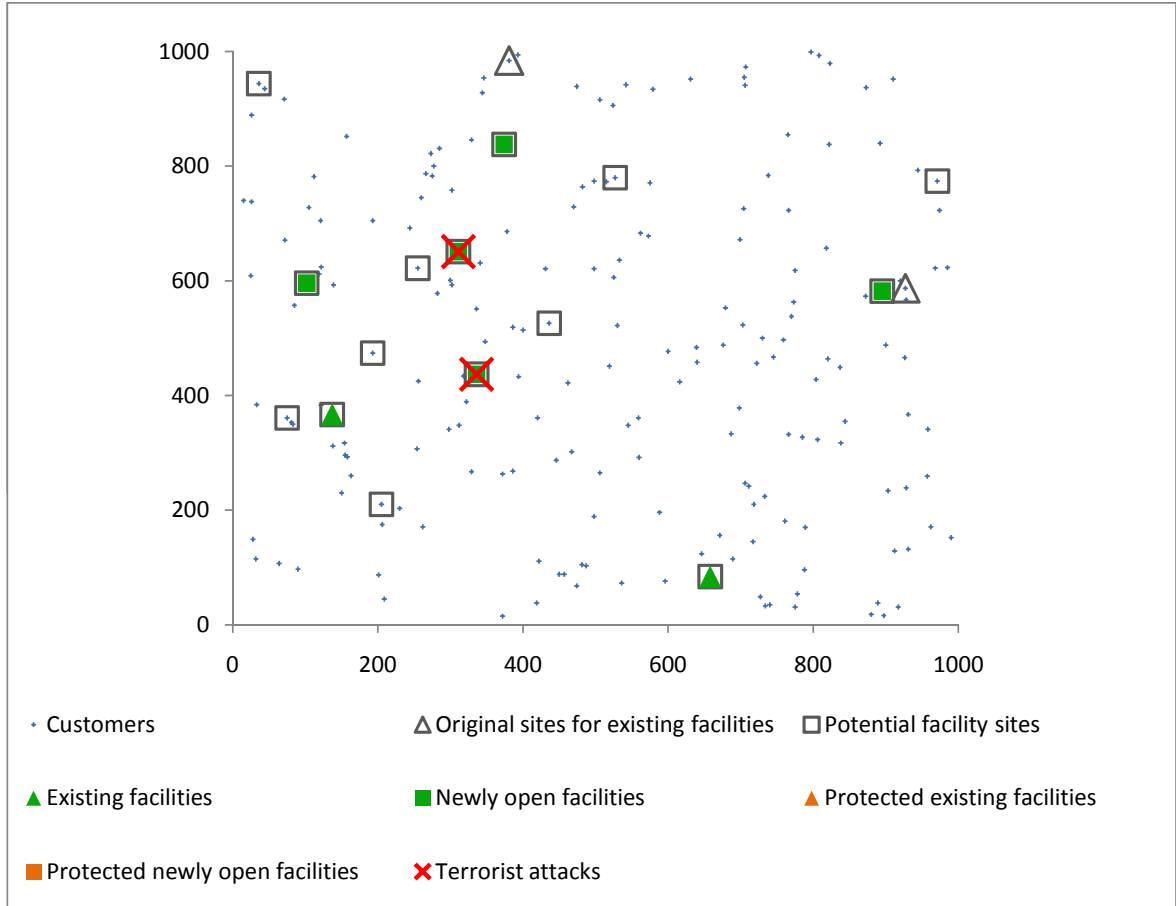


Figure K.1. Solution of the problem instance U200(15-1)

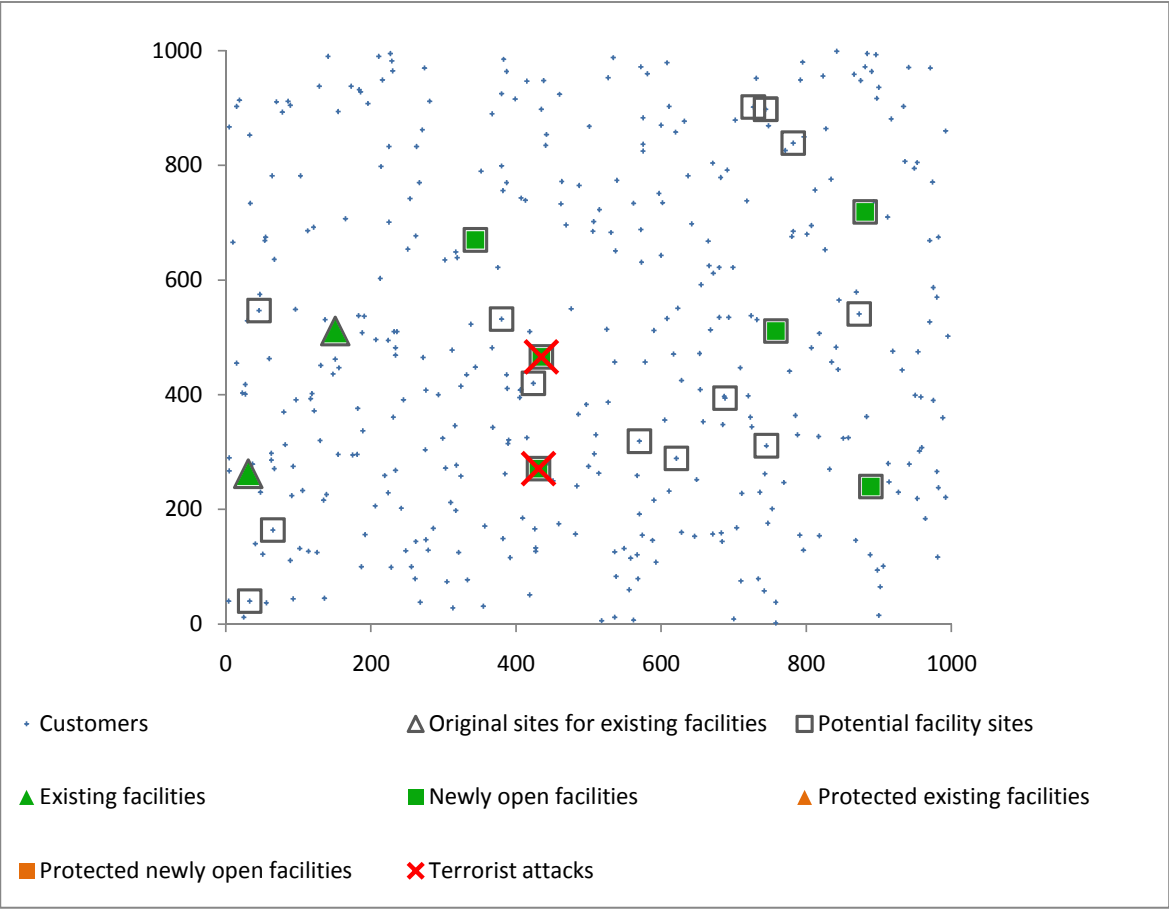


Figure K.2. Solution of the problem instance U400(20-3)

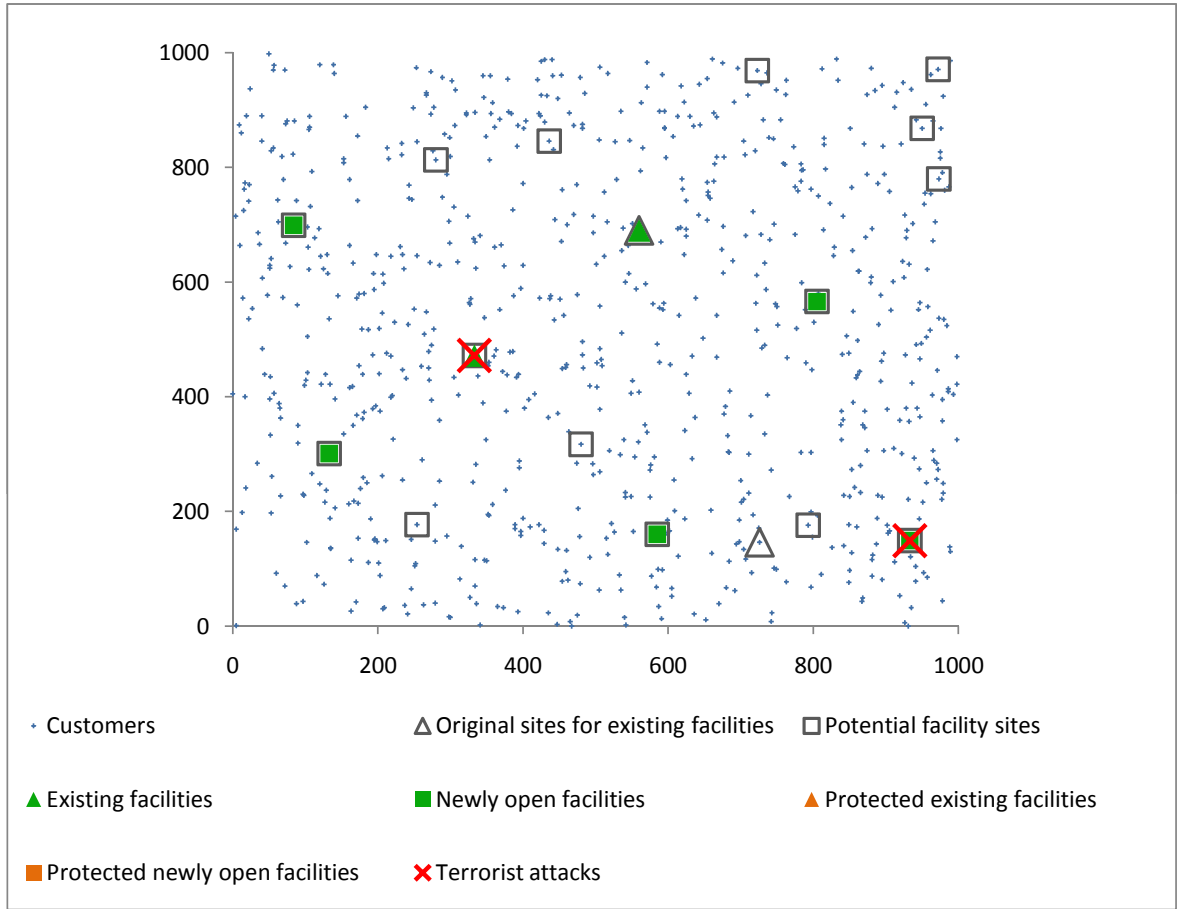


Figure K.3. Solution of the problem instance U800(15-1)

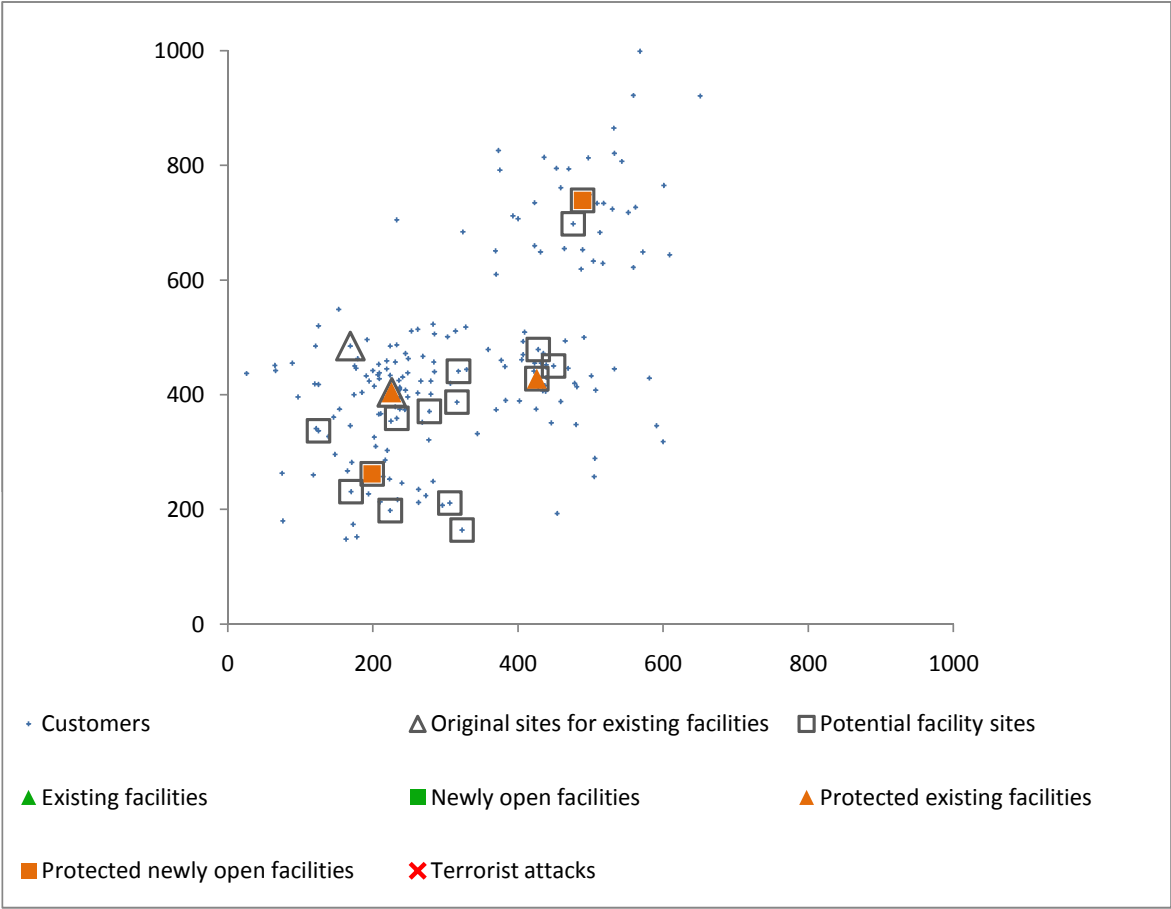


Figure K.4. Solution of the problem instance C200(15-7)

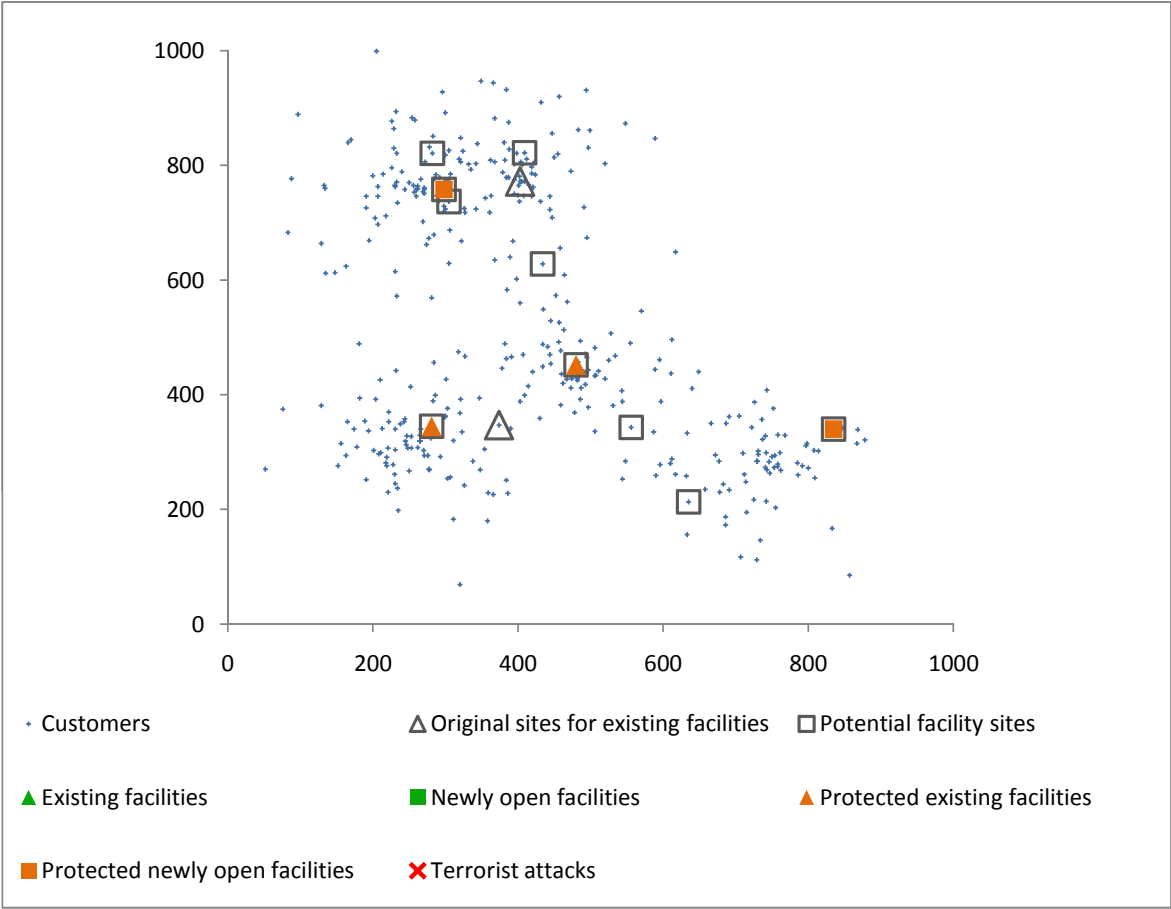


Figure K.5. Solution of the problem instance C400(10-2)

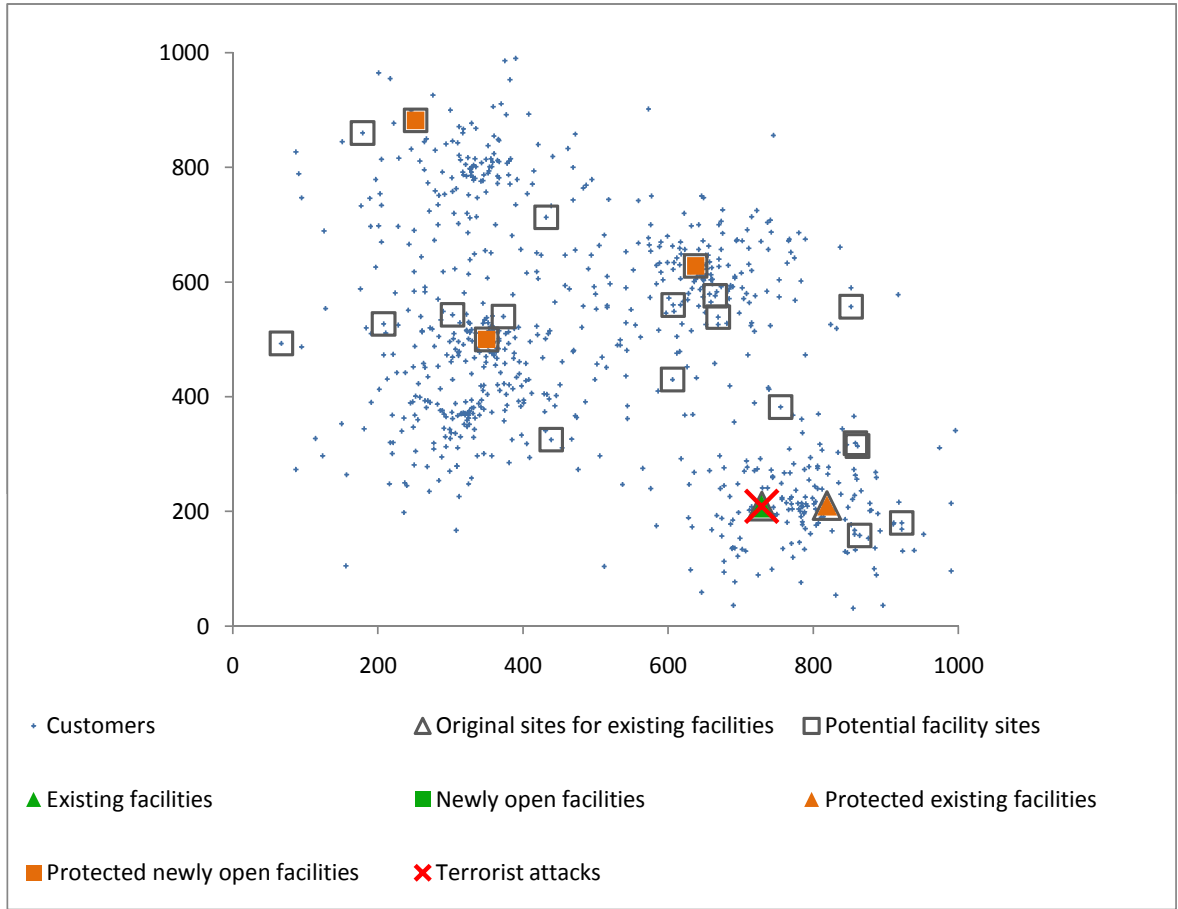


Figure K.6. Solution of the problem instance C800(20-9)

APPENDIX L: TEST RESULTS ON ALPHA

Table L.1. Detailed test results of alpha on instances of
uniform customer configuration

α	Instance	Coverage	#OF ¹	#PF ²	#IF ³	#PF/#OF	Average ⁴ #PF/#OF
$\alpha = 0.5$	U200,10-1	1698.00	5	5	0	1	0.83
	U200,10-2	1804.80	5	5	0	1	
	U200,15-1	1773.59	5	5	0	1	
	U200,15-2	1769.72	5	5	0	1	
	U200,20-1	1691.73	5	5	0	1	
	U200,20-2	1788.41	5	5	0	1	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2951.61	5	5	0	1	
	U400,15-1	3147.14	5	5	0	1	
	U400,15-2	3355.07	5	5	0	1	
	U400,20-1	3240.96	5	5	0	1	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5827.39	5	5	0	1	
	U800,10-2	5295.55	5	5	0	1	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5633.83	5	5	0	1	
	U800,20-1	6173.54	5	5	0	1	
	U800,20-2	6222.27	5	5	0	1	
$\alpha = 0.6$	U200,10-1	1427.09	4	4	0	1	
	U200,10-2	1804.80	5	5	0	1	
	U200,15-1	1582.69	4	4	0	1	
	U200,15-2	1742.56	5	5	0	1	

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

⁴Average for a certain α

Table L.1. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	U200,20-1	1624.36	5	5	0	1	0.61
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2951.61	5	5	0	1	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3355.07	5	5	0	1	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5777.01	5	5	0	1	
	U800,10-2	5267.82	5	5	0	1	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5633.83	5	5	0	1	
	U800,20-1	6120.20	8	0	2	0	
	U800,20-2	6222.27	5	5	0	1	
$\alpha = 0.7$	U200,10-1	1427.09	4	4	0	1	
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1582.69	4	4	0	1	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1552.44	7	1	2	0.142857	
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2670.99	7	1	2	0.142857	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4615.72	7	1	2	0.142857	

Table L.1. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	U800,15-1	5687.56	7	0	2	0	0.13
	U800,15-2	5392.69	8	0	2	0	
	U800,20-1	6120.20	8	0	2	0	
	U800,20-2	6146.20	8	0	2	0	
$\alpha = 0.8$	U200,10-1	1427.09	4	4	0	1	0.13
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1582.69	4	4	0	1	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1508.57	7	0	2	0	
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2613.18	7	1	2	0.142857	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4615.72	7	1	2	0.142857	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5392.69	8	0	2	0	
	U800,20-1	6120.20	8	0	2	0	
	U800,20-2	6146.20	8	0	2	0	
$\alpha = 0.9$	U200,10-1	1427.09	4	4	0	1	0.13
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1535.67	7	0	2	0	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1508.57	7	0	2	0	
	U200,20-2	1639.55	8	0	2	0	

Table L.1. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	U400,10-1	3085.19	8	0	2	0	0.11
	U400,10-2	2499.27	7	0	2	0	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4384.49	4	4	0	1	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5392.69	8	0	2	0	
	U800,20-1	6120.20	8	0	2	0	
	U800,20-2	6146.20	8	0	2	0	
$\alpha = 1$	U200,10-1	1397.75	4	4	0	1	
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1535.67	7	0	2	0	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1508.57	7	0	2	0	
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2499.27	7	0	2	0	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4356.77	4	4	0	1	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5392.69	8	0	2	0	

Table L.1. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	U800,20-1	6120.20	8	0	2	0	0.11
	U800,20-2	6146.20	8	0	2	0	
$\alpha = 1.1$	U200,10-1	1349.72	7	0	2	0	0.00
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1535.67	7	0	2	0	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1508.57	7	0	2	0	
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2499.27	7	0	2	0	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4276.72	7	0	2	0	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5392.69	8	0	2	0	
	U800,20-1	6120.20	8	0	2	0	
	U800,20-2	6146.20	8	0	2	0	
$\alpha = 1.2$	U200,10-1	1349.72	7	0	2	0	
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1535.67	7	0	2	0	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1508.57	7	0	2	0	
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2499.27	7	0	2	0	

Table L.1. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4276.72	7	0	2	0	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5392.69	8	0	2	0	
	U800,20-1	6120.20	8	0	2	0	
	U800,20-2	6146.20	8	0	2	0	
$\alpha = 1.3$	U200,10-1	1349.72	7	0	2	0	
	U200,10-2	1552.55	8	0	2	0	
	U200,15-1	1535.67	7	0	2	0	
	U200,15-2	1620.01	8	0	2	0	
	U200,20-1	1508.57	7	0	2	0	
	U200,20-2	1639.55	8	0	2	0	
	U400,10-1	3085.19	8	0	2	0	
	U400,10-2	2499.27	7	0	2	0	
	U400,15-1	2836.21	7	0	2	0	
	U400,15-2	3214.17	8	0	2	0	
	U400,20-1	3229.70	8	0	2	0	
	U400,20-2	3139.17	8	0	2	0	
	U800,10-1	5670.82	8	0	2	0	
	U800,10-2	4276.72	7	0	2	0	
	U800,15-1	5687.56	7	0	2	0	
	U800,15-2	5392.69	8	0	2	0	
	U800,20-1	6120.20	8	0	2	0	

Table L.1. Continued

α	Instance						Average
		Coverage	#OF	#PF	#IF	#PF/#OF	#PF/#OF
	U800,20-2	6146.20	8	0	2	0	0.00

Table L.2. Detailed test results of alpha on instances of clustered customer configuration

α	Instance	Coverage	#OF ¹	#PF ²	#IF ³	#PF/#OF	Average ⁴ #PF/#OF
$\alpha = 0.7$	C200,10-1	4109.95	4	4	0	1	1.00
	C200,10-2	4504.84	5	5	0	1	
	C200,15-1	4211.63	4	4	0	1	
	C200,15-2	4229.37	4	4	0	1	
	C200,20-1	4740.36	4	4	0	1	
	C200,20-2	4821.41	4	4	0	1	
	C400,10-1	6479.93	4	4	0	1	
	C400,10-2	8493.60	5	5	0	1	
	C400,15-1	8084.03	4	4	0	1	
	C400,15-2	8010.37	4	4	0	1	
	C400,20-1	8288.75	4	4	0	1	
	C400,20-2	8459.14	4	4	0	1	
	C800,10-1	13621.12	5	5	0	1	
	C800,10-2	12651.21	4	4	0	1	
	C800,15-1	13277.26	5	5	0	1	
	C800,15-2	12321.28	4	4	0	1	
	C800,20-1	14994.66	4	4	0	1	
	C800,20-2	13972.21	4	4	0	1	
$\alpha = 0.8$	C200,10-1	4109.95	4	4	0	1	
	C200,10-2	4405.27	4	4	0	1	
	C200,15-1	4211.63	4	4	0	1	
	C200,15-2	4229.37	4	4	0	1	
	C200,20-1	4740.36	4	4	0	1	
	C200,20-2	4821.41	4	4	0	1	

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

⁴Average for a certain α

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C400,10-1	6479.93	4	4	0	1	1.00
	C400,10-2	7783.98	4	4	0	1	
	C400,15-1	8084.03	4	4	0	1	
	C400,15-2	8010.37	4	4	0	1	
	C400,20-1	8288.75	4	4	0	1	
	C400,20-2	8459.14	4	4	0	1	
	C800,10-1	12717.37	4	4	0	1	
	C800,10-2	12651.21	4	4	0	1	
	C800,15-1	13153.76	4	4	0	1	
	C800,15-2	12321.28	4	4	0	1	
	C800,20-1	14994.66	4	4	0	1	
	C800,20-2	13819.72	4	4	0	1	
$\alpha = 0.9$	C200,10-1	4109.95	4	4	0	1	
	C200,10-2	4405.27	4	4	0	1	
	C200,15-1	4211.63	4	4	0	1	
	C200,15-2	4228.52	4	4	0	1	
	C200,20-1	4634.09	4	4	0	1	
	C200,20-2	4821.41	4	4	0	1	
	C400,10-1	6479.93	4	4	0	1	
	C400,10-2	7783.98	4	4	0	1	
	C400,15-1	8084.03	4	4	0	1	
	C400,15-2	8001.76	4	4	0	1	
	C400,20-1	8049.51	4	4	0	1	
	C400,20-2	8459.14	4	4	0	1	
	C800,10-1	12717.37	4	4	0	1	
	C800,10-2	12651.21	4	4	0	1	
	C800,15-1	13153.76	4	4	0	1	
	C800,15-2	12321.28	4	4	0	1	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C800,20-1	14994.66	4	4	0	1	1.00
	C800,20-2	11368.78	3	3	0	1	
$\alpha = 1$	C200,10-1	4109.95	4	4	0	1	0.82
	C200,10-2	4405.27	4	4	0	1	
	C200,15-1	4246.29	4	4	0	1	
	C200,15-2	4037.35	4	3	1	0.75	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	4773.22	4	4	0	1	
	C400,10-1	6479.93	4	4	0	1	
	C400,10-2	7783.98	4	4	0	1	
	C400,15-1	7987.43	4	4	0	1	
	C400,15-2	7580.62	4	4	0	1	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	7825.38	4	4	0	1	
	C800,10-1	12717.37	4	4	0	1	
	C800,10-2	10871.01	3	3	0	1	
	C800,15-1	13153.76	4	4	0	1	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	14625.88	4	4	0	1	
	C800,20-2	11368.78	3	3	0	1	
$\alpha = 1.1$	C200,10-1	3603.80	3	3	0	1	
	C200,10-2	4405.27	4	4	0	1	
	C200,15-1	3982.16	3	3	0	1	
	C200,15-2	4037.35	4	3	1	0.75	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3986.17	3	3	0	1	
	C400,10-1	5710.52	3	3	0	1	
	C400,10-2	7734.96	4	4	0	1	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C400,15-1	7005.78	4	4	0	1	0.76
	C400,15-2	6323.40	3	3	0	1	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	12717.37	4	4	0	1	
	C800,10-2	10871.01	3	3	0	1	
	C800,15-1	11663.57	3	3	0	1	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13561.11	4	4	0	1	
	C800,20-2	11368.78	3	3	0	1	
$\alpha = 1.2$	C200,10-1	3603.80	3	3	0	1	
	C200,10-2	4386.62	4	4	0	1	
	C200,15-1	3982.16	3	3	0	1	
	C200,15-2	4037.35	4	3	1	0.75	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3986.17	3	3	0	1	
	C400,10-1	5710.52	3	3	0	1	
	C400,10-2	7351.73	4	4	0	1	
	C400,15-1	6562.94	3	3	0	1	
	C400,15-2	6323.40	3	3	0	1	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	12596.75	4	4	0	1	
	C800,10-2	10871.01	3	3	0	1	
	C800,15-1	11663.57	3	3	0	1	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C400,15-2	6323.40	3	3	0	1	0.67
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11675.36	3	3	0	1	
	C800,10-2	10871.01	3	3	0	1	
	C800,15-1	11663.57	3	3	0	1	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	11216.29	3	3	0	1	
$\alpha = 1.5$	C200,10-1	3603.80	3	3	0	1	0.67
	C200,10-2	4062.70	3	3	0	1	
	C200,15-1	3982.16	3	3	0	1	
	C200,15-2	4037.35	3	3	0	1	
	C200,20-1	4039.07	3	3	0	1	
	C200,20-2	3986.17	3	3	0	1	
	C400,10-1	5710.52	3	3	0	1	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6483.74	3	3	0	1	
	C400,15-2	6323.40	3	3	0	1	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11675.36	3	3	0	1	
	C800,10-2	10871.01	3	3	0	1	
	C800,15-1	11663.57	3	3	0	1	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	
$\alpha = 1.6$	C200,10-1	3603.80	3	3	0	1	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C200,10-2	4062.70	3	3	0	1	0.50
	C200,15-1	3976.16	3	3	0	1	
	C200,15-2	4037.35	3	3	0	1	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3971.39	7	0	2	0	
	C400,10-1	5710.52	3	3	0	1	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6411.97	3	3	0	1	
	C400,15-2	6314.78	3	3	0	1	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11675.36	3	3	0	1	
	C800,10-2	10871.01	3	3	0	1	
	C800,15-1	10566.90	8	0	2	0	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	
$\alpha = 1.7$	C200,10-1	3603.80	3	3	0	1	
	C200,10-2	4062.70	3	3	0	1	
	C200,15-1	3942.70	3	3	0	1	
	C200,15-2	3924.45	3	3	0	1	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3971.39	7	0	2	0	
	C400,10-1	5710.52	3	3	0	1	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6204.46	8	0	2	0	
	C400,15-2	5970.62	7	0	2	0	
	C400,20-1	7178.04	8	0	2	0	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C400,20-2	6970.02	8	0	2	0	0.33
	C800,10-1	11675.36	3	3	0	1	
	C800,10-2	9791.53	7	0	2	0	
	C800,15-1	10566.90	8	0	2	0	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	
$\alpha = 1.8$	C200,10-1	3310.95	7	0	2	0	0.22
	C200,10-2	4062.70	3	3	0	1	
	C200,15-1	3457.80	7	0	2	0	
	C200,15-2	3662.77	3	3	0	1	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3971.39	7	0	2	0	
	C400,10-1	5710.52	3	3	0	1	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6204.46	8	0	2	0	
	C400,15-2	5970.62	7	0	2	0	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11675.36	3	3	0	1	
	C800,10-2	9791.53	7	0	2	0	
	C800,15-1	10566.90	8	0	2	0	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	
$\alpha = 1.9$	C200,10-1	3310.95	7	0	2	0	
	C200,10-2	4062.70	3	3	0	1	
	C200,15-1	3457.80	7	0	2	0	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C200,15-2	3662.77	3	3	0	1	0.11
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3971.39	7	0	2	0	
	C400,10-1	4694.06	7	0	2	0	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6204.46	8	0	2	0	
	C400,15-2	5970.62	7	0	2	0	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11121.75	8	0	2	0	
	C800,10-2	9791.53	7	0	2	0	
	C800,15-1	10566.90	8	0	2	0	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	
$\alpha = 2$	C200,10-1	3310.95	7	0	2	0	
	C200,10-2	4062.70	3	3	0	1	
	C200,15-1	3457.80	7	0	2	0	
	C200,15-2	3524.61	7	0	2	0	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3971.39	7	0	2	0	
	C400,10-1	4694.06	7	0	2	0	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6204.46	8	0	2	0	
	C400,15-2	5970.62	7	0	2	0	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11121.75	8	0	2	0	

Table L.2. Continued

α	Instance	Coverage	#OF	#PF	#IF	#PF/#OF	Average #PF/#OF
	C800,10-2	9791.53	7	0	2	0	0.06
	C800,15-1	10566.90	8	0	2	0	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	
$\alpha = 2.1$	C200,10-1	3310.95	7	0	2	0	0.00
	C200,10-2	3287.03	8	0	2	0	
	C200,15-1	3457.80	7	0	2	0	
	C200,15-2	3524.61	7	0	2	0	
	C200,20-1	4000.38	8	0	2	0	
	C200,20-2	3971.39	7	0	2	0	
	C400,10-1	4694.06	7	0	2	0	
	C400,10-2	6989.33	8	0	2	0	
	C400,15-1	6204.46	8	0	2	0	
	C400,15-2	5970.62	7	0	2	0	
	C400,20-1	7178.04	8	0	2	0	
	C400,20-2	6970.02	8	0	2	0	
	C800,10-1	11121.75	8	0	2	0	
	C800,10-2	9791.53	7	0	2	0	
	C800,15-1	10566.90	8	0	2	0	
	C800,15-2	11699.04	8	0	2	0	
	C800,20-1	13103.66	8	0	2	0	
	C800,20-2	10558.20	7	0	2	0	

APPENDIX M: TEST RESULTS ON RELOCATION OPTION

Table M.1. Detailed test results of relocation option on
instances of SJC818 configuration

Instance	No relocation case					Original	Average
	Coverage	#OF ¹	#PF ²	#IF ³	#IEF ⁴	Coverage	Loss
10-1	3549.49	4	4	0	0	3549.49	118.50
10-2	5886.37	4	4	0	0	5886.37	
10-3	5759.05	4	4	0	0	5759.05	
10-4	5065.26	5	4	1	1	5416.88	
10-5	4323.84	5	4	1	1	4428.10	
10-6	5313.26	5	4	1	1	5487.88	
10-7	4492.44	7	0	2	0	4492.44	
10-8	3463.88	5	4	1	1	3463.88	
10-9	4284.75	4	4	0	0	4656.29	
10-10	5746.39	4	4	0	0	5929.36	
15-1	6586.50	5	4	1	1	6608.20	481.39
15-2	4179.71	4	4	0	0	4743.85	
15-3	3986.10	5	4	1	1	4428.24	
15-4	5153.56	8	0	2	0	5545.96	
15-5	4896.09	4	4	0	0	5311.43	
15-6	5723.81	8	1	2	0	6100.25	
15-7	4521.33	7	1	2	0	4823.98	
15-8	5986.33	8	0	2	0	6689.93	
15-9	3528.47	7	0	2	0	4115.00	
15-10	5027.66	5	3	2	2	6036.59	

¹# of open facilities (total of existing and newly opened facilities)

²# of protected facilities

³# of interdicted facilities

³# of interdicted existing facilities

Table M.1.

Instance	No relocation case					Original	Average
	Coverage	#OF	#PF	#IF	#IEF	Coverage	Loss
20-1	3506.52	8	0	2	0	3815.62	513.81
20-2	5564.73	4	4	0	0	5943.19	
20-3	4649.95	5	3	2	2	5916.81	
20-4	5376.10	5	3	2	2	6841.50	
20-5	5929.31	8	0	2	0	5929.31	
20-6	2851.91	5	3	2	2	4034.31	
20-7	4422.78	8	0	2	0	4514.36	
20-8	5603.53	8	0	2	0	5991.61	
20-9	5405.72	8	0	2	1	5405.72	
20-10	4705.68	7	1	2	1	4761.86	

REFERENCES

1. Perl, R., "Trends in Terrorism: 2006", *Congressional Research Service Report for Congress*, 2007.
2. Chalk, P., B. Hoffman, R. Reville and A. Kasupski, "Trends in Terrorism", *RAND Center for Terrorism Risk Management Policy*, 2005.
3. Radio Free Europe - Radio Liberty: *Afghanistan: Mobile-Phone Towers Are Taliban's New Target*, <http://www.rferl.org/content/article/1347757.html>, 2008.
4. BBC News UK: *Arsonists target ambulance station*, BBC Online Network, http://news.bbc.co.uk/2/hi/uk_news/378254.stm, 1999.
5. Motto, A. L., J. M. Arroyo and F. D. Galiana, "MILP for the analysis of electric grid security under disruptive threat" *IEEE Trans. on Power Syst.*, Vol. 20, No. 3, pp. 1357-1365, 2005.
6. Ben-Ayed, O. and C. E. Blair, "Computational Difficulties of Bilevel Linear Programming", *Operations Research*, Vol. 38, No. 3, pp. 556-560, May-June 1990.
7. Owen, S. H. and M. S. Daskin, "Strategic Facility Location: A Review", *European Journal of Operational Research*, Vol. 111, No. 3, pp. 423-447, December 1998.
8. Ghosh, A. and G. Rushton (editors), *Spatial analysis and location-allocation models*, Van Nostrand Reinhold Company, New York, 1987.
9. Daskin, M. S., *Network and discrete location: models, algorithms, and applications*, Wiley, New York, 1995.
10. Toregas, C , R. Swain, C. Reville, and L. Bergmann, "The Location of Emergency Service Facilities," *Oper. Res.*, Vol. 19, No. 6, pp. 1363-1373 ,October 1971.

11. Church, R. and C. ReVelle, "The maximal covering location problem", *Papers of the Regional Science Association*, Vol. 18, pp. 101-118, 1974.
12. Current, J. and J. Storbeck, "Capacitated Covering Models", *Environment and Planning B.*, Vol 15, pp. 153-164, 1988.
13. Pirkul, H. and D. A. Schilling, "The maximal covering location problem with Capacities on total workload", *Management Science*, Vol. 37, No. 2, pp. 233-248, February 1991.
14. Berman, O. and D. Krass, "The generalized maximal covering location problem", *Computers & Operations Research*, Vol. 29, pp. 563-581, 2002.
15. Berman, O., D. Krass and Z. Drezner, "The gradual covering decay location problem on a network", *European Journal of Operational Research*, Vol. 151, pp. 474-480, 2003.
16. Karasakal, O. and E. Karasakal, "A maximal covering location model in the presence of partial coverage", *Computers and Operations Research*, Vol. 31, pp. 15-26, 2004.
17. Mehrez, A. and A. Stulman, "The maximal covering location problem with facility placement on the entire plane", *Journal of Regional Science*, Vol. 22, No.3, pp. 361-365, 1982.
18. Berman, O., Z. Drezner, D. Krass and G. O. Wesolowsky, "The variable radius covering problem", *European Journal of Operational Research*, Vol. 196, pp. 516-525, 2009.
19. Drezner, Z., "Heuristic Solution Methods for Two Location Problems With Unreliable Facilities", *Journal of Operational Research Society*, Vol. 38, No. 6, pp. 509-514, June 1987.
20. Snyder, L. V. and M. S. Daskin, "Reliability Models for Facility Location: The

- Expected Failure Cost Case”, *Transportation Science*, Vol. 39, No. 3, pp. 400-416, August 2005.
21. Berman, O., D. Krass and M. B. C. Menezes, “Facility Reliability Issues in Network p -Median Problems: Strategic Centralization and Co-Location Effects”, *Operations Research*, Vol. 55, No. 2, pp. 332-350, March-April 2007.
 22. Snyder, L. V. and M. S. Daskin, “Models for Reliable Supply Chain Network Design”, in A. T. Murray and T. H. Grubescic (eds.), *Advances in Spatial Science, Critical Infrastructure Reliability and Vulnerability*, pp. 257-289, Springer, Berlin Heidelberg, 2007.
 23. Snyder, L. V., M. P. Scaparra, M. S. Daskin and R. L. Church, ”Planning for Disruptions in Supply Chain Networks”, *Tutorials in Operations Research*, pp. 234-357, 2006.
 24. Wollmer, R., “Removing Arcs from a Network”, *Operations Research*, Vol. 12, No. 6, pp. 934-940, 1964
 25. Smith, J. C., C. Lim and F. Sudargho, “Survivable network design under optimal and heuristic interdiction scenarios”, *Journal of Global Optimization*, Vol. 38, pp. 181-199, 2007.
 26. Berman, O., T. Drezner, Z. Drezner and G. O. Wesolowsky, “A defensive maximal covering problem on a network”, *International Transactions in Operations Research*, Vol. 16, pp. 69-86, 2009.
 27. Lim, C. and J. C. Smith, “Algorithms for Discrete and Continuous Multicommodity Flow Network Interdiction Problems”, *IIE Transactions*, Vol. 39, No. 1, pp. 15-26, January 2007.
 28. Matisziw, T. C. and A. T. Murray, “Modeling s-t Path Availability to Support Disaster Vulnerability Assessment of Network Infrastructure”, *Computers and Op-*

- erations Research*, Vol. 36, No.1, pp. 16-26, 2009.
29. Church, R. L., M. P. Scaparra, R. S. Middleton, "Identifying Critical Infrastructure: The Median and Covering Facility Interdiction Problems, *Annals of the Association of American Geographers*, Vol. 94, No. 3, pp. 491-502, 2004.
 30. Church, R. L. and M. P. Scaparra, "Protecting Critical Assets: The r-interdiction Median Problem With Fortification", *Geographical Analysis*, Vol. 39, No. 2, pp. 129-146, 2007.
 31. Scaparra, M. P. and R. L. Church, "An Exact Solution Approach for The Interdiction Median Problem With Fortification", *European Journal of Operational Research*, Vol. 189, No. 1, pp. 76-92, August 2008.
 32. Scaparra, M. P. and R. L. Church, "A Bilevel Mixed-integer Program for Critical Infrastructure Protection Planning", *Computers and Operations Research*, Vol. 35, No. 6, pp. 1905-1923, June 2008.
 33. Aksen, D., N. Piyade and N. Aras, "The budget constrained r-interdiction median problem with capacity expansion", *Central European Journal of Operations Research*, Vol. 18, No. 3, pp. 269-291, 2010.
 34. O'Hanley, J.R., R. L. Church, and J. K. Gilles, "Locating and protecting critical reserve sites to minimize expected and worst-case losses", *Biological Conservation*, Vol. 34, pp. 130-141, 2007
 35. O'Hanley, J.R., R. L. Church, J. K. Gilles, "The importance of in situ site loss in nature reserve selection: Balancing notions of complementarity and robustness" *Biological Conservation*, Vol. 35, pp. 170-180, 2007.
 36. O'Hanley, J. and R. L. Church, "Designing Robust Coverage Networks to Hedge Against Worst-Case Facility Losses", Working Paper No. 173, ISSN 1748-7595, 2008.
 37. H. von Stackelberg. *The Theory of the Market Economy*, William Hodge & Co.,

London, UK, 1952.

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