2010

Final report on JIEDDO research project optimally locating BETSS-C surveillance assets

Salmeron, Javier; Wood, R. Kevin

Monterey, California. Naval Postgraduate School

https://hdl.handle.net/10945/770

NPS-OR-10-010-PR

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

Final Report on JIEDDO Research Project
Optimally Locating BETSS-C Surveillance Assets

by

Javier Salmerón
R. Kevin Wood

November 2010

Approved for public release, distribution is unlimited.

Prepared for: Joint Improvised Explosive Device Defeat Organization (JIEDDO)
5000 Army Pentagon
Washington, D.C. 20301-5000

THIS PAGE INTENTIONALLY LEFT BLANK

NAVAL POSTGRADUATE SCHOOL
MONTEREY, CA 93943-5001

Daniel T. Oliver
President

Leonard A. Ferrari
Executive Vice President and
Provost

This report was prepared for and funded by the Joint Improvised Explosive Device Defeat Organization (JIEDDO), 5000 Army Pentagon, Washington, D.C. 20301-5000.

This report was prepared by:

JAVIER SALMERÓN
Associate Professor of
Operations Research

R. KEVIN WOOD
Distinguished Professor of
Operations Research

Reviewed by:

RONALD D. FRICKER
Associate Chairman for Research
Department of Operations Research

ROBERT F. DELL
Chairman
Department of Operations Research

Released by:

KARL VAN BIBBER
Vice President and
Dean of Research

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This is the final report for the project entitled "Optimally Locating BETSS-C Surveillance Assets" sponsored by the Joint Improvised Explosive Device Defeat Organization. The research has focused on developing optimization models for optimal placement of cameras and tower-mounted surveillance systems such as BETSS-C (Base Expeditionary Targeting and Surveillance Systems-Combined). These systems have proven themselves useful in detecting improvised explosive devices as they are being emplaced, and in making certain locations less desirable for emplacement. We have created models and solution software that locate a given set of camera towers (also observation towers or aerostats) to optimally cover "points of interest" on the ground. Computational results show that it is possible to obtain near-optimal solutions for problems with up to 30 cameras and 100 points of interest on a laptop computer in less than one minute.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

A. BACKGROUND

In Iraq and Afghanistan, Coalition Forces have found that camera towers such as "GBOSS" (Ground-Based Operational Surveillance System), BETSS-C (Base Expeditionary Targeting and Surveillance Systems-Combined), and JLENS/RAID PS2 systems, can help thwart the emplacement of improvised explosive devices (IEDs). These systems can also identify disturbances to which troops need to respond, follow suspicious vehicles, and so on. Their use, for similar purposes in the increasingly complex war in Afghanistan, is critical to the security of U.S. and allied forces, as well as to the civilian population. No tool currently exists, however, for assigning a limited number of camera towers to a large number of potential (secure) sites so as to (1) maximize the "value" of the surveilled "points of interest" (POIs), (2) maximize the probability of detecting specific threats, or (3) maximize the overall "coverage" of important sites that are surveilled. Developing such a tool is the purpose of this study.

The research focuses on developing, implementing, and solving a series of prototypic mathematical models for optimizing camera-tower placement. We have created nonlinear integer optimization models for this purpose, have reformulated those models for tractability, and are solving them using general-purpose optimization tools.

B. MATHEMATICAL MODELS

1. Basic Camera Location Models

We first develop the following nonlinear, camera location (NL-CL^1) optimization model:

$$\text{(NL-CL}^1\text{):}\ \min_{\mathbf{y}} \sum_{i\in I} v_i \prod_{l\in L} q_{il}^{y_l}$$

$$\text{subject to:}\quad \sum_{l} y_l \le m,$$

$$y_l \in \{0,1\}\ \forall l \in L,$$

where $l \in L$ is a given set of potential camera locations; $i \in I$ is a set of POIs that need to be kept under surveillance; m is the number of camera towers available; v_i is the "value" of POI i, which represents the damage or consequences that events (such as an IED emplacement) at that point would cause; q_{il} is the probability of not detecting an event at POI i from location l, if a camera is placed at that location; and the decision variable y_l is 1 if we locate a tower at l, and $y_l = 0$ otherwise. Since $\prod_{l\in L} q_{il}^{y_l}$ is the probability that an event at POI i is not detected by any of the installed cameras, NL-CL^1's objective, under an assumption of independence, minimizes overall expected "value" of undetected events across POIs, subject to the limit on available cameras. Henceforth, we use "expected 'value' of undetected events" and "expected damage" interchangeably. In particular, we refer to "expected damage at an individual POI i," $v_i \prod_{l \in L} q_{il}^{y_l}$, and to "overall expected damage," $\sum_{i \in I} v_i \prod_{l \in L} q_{il}^{y_l}$.

We note that (NL-CL^1) does assume independence of detections between events at a given POI, and requires some user inputs that may not be immediately available, namely v_i and q_{il}. It may be necessary to use subjective estimates if these data are unavailable. We believe that the model's solution is likely to provide useful insight even when using subjective input, however.

We also note that the notation in the model hides some of the practical aspects of an implementation. Suppose, for instance, that there is no line of sight between potential camera location ℓ and POI i. In this case, $q_{i\ell} = 1$, and the model is correct. However, our implementation would not even create the corresponding term in the objective function.

A second model, NL-CL^2, seeks to minimize the maximum expected damage at any POI, i.e., the worst-case damage among all POIs. This second model is:

$$
\begin{aligned}
(\text{NL-CL}^2)\ \min_{\mathbf{y},z}\ & z \\
\text{subject to:}\quad & \sum_l y_l \le m, \\
& z \ge v_i \prod_{l \in L} q_{il}^{y_l} \quad \forall i, \\
& y_l \in \{0,1\}\ \forall l \in L,
\end{aligned}
$$

where the new set of constraints ensures that z (the objective value sought) takes the maximum, among all POIs, of the expected damages at each individual POI. It is important to note that in this model we may minimize $Z = \log z$ without affecting the outcome. Also, since

$$\log z \ge \log\left(v_i \prod_{l \in L} q_{il}^{y_l} \right)\ \forall i \qquad \Rightarrow \qquad Z \ge \log(v_i) + \sum_{l \in L} (\log q_{il}) y_l \ \ \forall i,$$

NL-CL^2 can be restated as this mixed-integer program:

$$
\begin{aligned}
(\text{MI-CL}^2)\ \min_{\mathbf{y},Z}\ & Z \\
\text{subject to:}\quad & \sum_l y_l \le m, \\
& Z \ge \log(v_i) + \sum_{l \in L} (\log q_{il}) y_l \ \ \forall i, \\
& y_l \in \{0,1\}\ \forall l \in L.
\end{aligned}
$$

Unfortunately, the same type of simplification is not possible for NL-CL^1: The logarithm function cannot be used to decompose that model's objective function $\sum_{i \in I} v_i \prod_{l \in L} q_{il}^{y_l}$ into a linear expression of the y-variables. Although we may expect a strong correlation between the optimization of $\log\left(\sum_{i\in I} v_i \prod_{l\in L} q_{il}^{y_l}\right)$ and that of $\sum_{i\in I} v_i \log\left(\prod_{l\in L} q_{il}^{y_l}\right)$ in many practical instances, no guarantee of equivalence exists, and bounding the error in the approximation appears to be a difficult task.

2. Converting the Basic Model into a Network-Flow-Based Model

This section shows how to convert NL-CL[1] into a mixed-integer linear program that has an underlying network-flow structure.

We use the concept of a generalized flow (see Ahuja et al., 1993, pp. 566-572). Specifically, for each POI *i*, we create a network whose nodes correspond to locations $l \in L_i \subset L$ that could detect an event should we install a camera at those locations, that is, $L_i = \{l \in L \mid q_{il} < 1\}$. For notational simplicity, assume L_i is ordered and denote its first element as l_i^F, its last element as l_i^L, and the predecessor to a given $l \in L_i$ ($l \neq l_i^F$) as l_{il}^P.

In our network model for POI *i*, each node $l \in L_i$ is connected to the next location node by two arcs: The flow on the first arc (denoted x_{il}) represents the probability that cameras up to that node have not detected the event, given that no camera is installed at the location. The flow on the second arc (denoted x_{il}^S) represents the same concept, but applies when a camera is installed at the node. That is, when $x_{il}^S > 0$, part of the flow going into node *l* (probability of non-detection up to that point) will be lost in the transition (flow) to the next location. The overall probability of non-detection, $Q_i = \prod_{l\in L} q_{il}^{y_l}$, is the flow into a fictitious node connected to the last node l_i^F.

The mixed-integer network model, MIN-CL[1], which is equivalent to NL-CL[1], may be stated as follows:

(MIN-CL[1]) $\min_{\mathbf{y,x,Q}} \sum_{i\in I} v_i Q_i$

subject to:

$$x_{i,l_i^F} + x_{i,l_i^F}^S = 1 \quad \forall i,$$

$$x_{il} + x_{il}^S = x_{i,l_{il}^P} + q_{i,l_{il}^P}\, x_{i,l_{il}^P}^S \quad \forall i, \forall l \in L_i \setminus \{l_i^F\},$$

$$Q_i = x_{i,l_i^L} + q_{i,l_i^L}\, x_{i,l_i^L}^S \quad \forall i,$$

$$0 \le x_{il} \le 1 - y_l \quad \forall i, \forall l \in L_i,$$

$$0 \le x_{il}^S \le y_l \quad \forall i, \forall l \in L_i,$$

$$\sum_l y_l \le m,$$

$$y_l \in \{0,1\} \forall l \in L.$$

MIN-CL[1] above (like the original NL-CL[1]) allows each location with a camera to surveil all POIs $\{i \in P \mid q_{il} < 1\}$ with a constant probability of detection. This assumption is, in most cases, optimistic, because a camera needs time to rotate, zoom in and out, and focus on each of the POIs, and personnel are needed to monitor the camera images from each POI. This research does not address the problem of incorporating a decrease in detection probability for a camera surveilling multiple POIs, or the inherent details about frequency of rotation between surveilled POIs, etc. The work of our colleagues Burton et al. (2008) may apply here to future extensions, however. That work determines the proportion of time that a single camera should dedicate to surveilling POI *i*, assuming events of interest occur according to a Poisson process with a location-dependent rate (and are independent of events at other POIs), and that detection times at each location are exponentially distributed.

To make our approach more realistic, we incorporate a parameter, *k*, denoting the maximum number of sites that any one camera may surveil simultaneously. In this case, additional variables y_{il}^S must control whether or not a camera installed at location *l* will be dedicated to surveil POI *i*. This can be accommodated in the existing models as follows:

Add the following constraint to both MIN-CL[1] and MI-CL[2]:

$$\sum_i y_{il}^S \le k\, y_l \quad \forall l \,.$$

- Replace the following constraints in MIN-CL[1]:

$$0 \le x_{il} \le 1 - y_l \quad \forall i, \forall l \in L_i \text{ and}$$

$$0 \le x_{il}^S \le y_l \qquad \forall i, \forall l \in L_i \,,$$

with these new constraints:

$$0 \le x_{il} \le 1 - y_{il}^S \quad \forall i, \forall l \in L_i \text{ and}$$

$$0 \le x_{il}^S \le y_{il}^S \qquad \forall i, \forall l \in L_i \,.$$

- Replace the following constraints in MI-CL[2]:

$$Z \ge \log(v_i) + \sum_{l \in L} (\log q_{il})\, y_l \quad \forall i$$

with this new constraint:

$$Z \ge \log(v_i) + \sum_{l \in L} (\log q_{i\ell})\, y_{il}^S \quad \forall i \,.$$

C. COMPUTATIONAL IMPLEMENTATION

1. Optimization Models

We have implemented the NL-CL^1 in GAMS (GAMS, 2010), and have solved a number of randomly generated instances using GAMS/CONOPT (GAMS/CONOPT, 2010) for benchmarking purposes.

For the mixed-integer models, MIN-CL^1 and MI-CL^2, we have used the Xpress-MP (FICO, 2010) development environment, and have solved them with the Xpress-MP and/or CPLEX (IBM, 2010) optimization engines. The remainder of the document refers to this implementation.

2. Database

The supporting database for our tool is implemented in Microsoft Access 2007 (Microsoft, 2010). Each database file contains one modeling example, representing a physical layout of POIs and locations. For that example, the file may include several scenario settings which differ, for example, in the number of available cameras, or in the type of model we would like to run. The structure of this database is as follows (Figure 1):

Tables:

LOC: Locations

POI: Points of interest

LOC_POI: Attributes for locations and points of interest

SCENARIO: Different scenarios to run, and associated solutions to store, for the incumbent “example” (see Section D) of locations and POIs

Queries:

Delete_LOC: Eliminates all records from LOC table

Delete_POI: Eliminates all records from POI table

LOC_POI_CreateMatrix: Creates the list of all possible combinations of locations and POIs to ease the input of associated probabilities

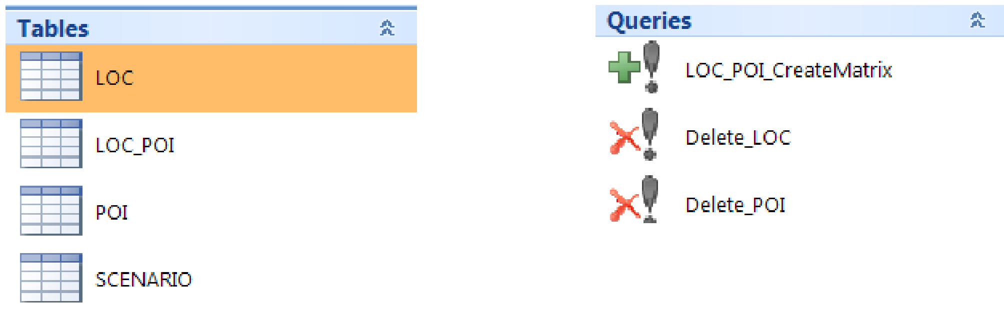

Figure 1. Database tables and queries

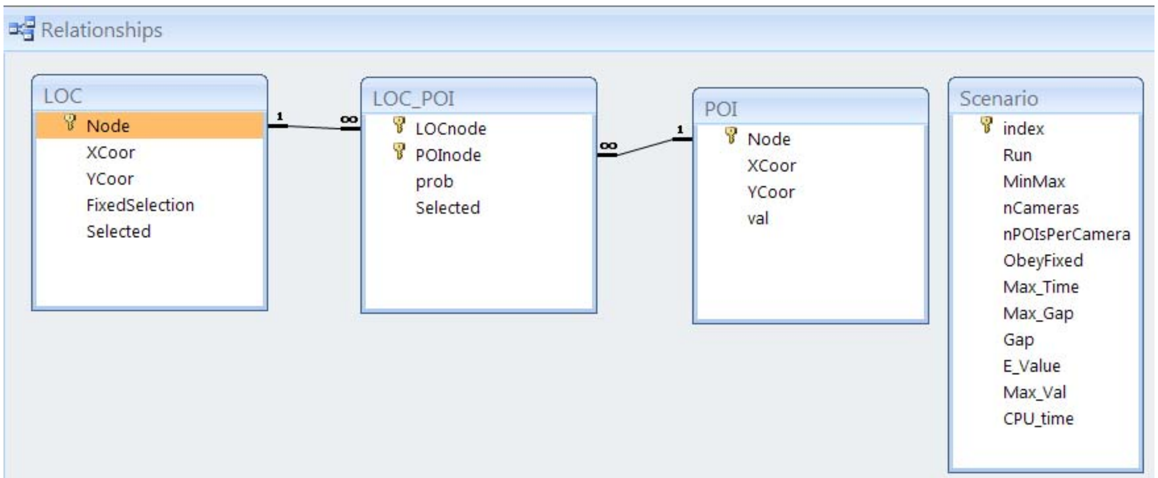

Figure 2. Fields for the database tables, and relationships among tables

Fields in each of the above tables and relationships are shown in Figure 2. The tables below describe these fields in more detail:

Table: LOC

Name	**Type**	**Default**	**Description**
Node	Text		Location code
XCoor	Double	0.0	X coordinate
YCoor	Double	0.0	Y coordinate
FixedSelection	Yes/No	No	If “Yes” and ObeyFixed=”Yes” in table SCENARIO, the location must be selected
Selected	Yes/No	No	Whether or not the location was selected by the optimization for emplacement of a camera (OUTPUT)

Table: POI

Name	**Type**	**Default**	**Description**
Node	Text		POI code
XCoor	Double	0.0	X coordinate
YCoor	Double	0.0	Y coordinate
val	Double	1.0	Value of the POI

Table: LOC_POI

Name	Type	Default	Description
LOCnode	Text		Location code
POInode	Text		POI code
prob	Double	0.0	Probability of detection of an event at the POI from the location
Selected	Yes/No	No	Whether or not the POI was selected by the optimization to be surveilled from the location (OUTPUT)

Table: SCENARIO

Name	Type	Default	Description
index	Long Integer		Scenario index (AUTOMATED)
Run	Yes/No	Yes	If "Yes," the scenario will be run next time the optimizer is executed
MinMax	Yes/No	No	If "Yes," the optimizer solves a min-max (MI-CL2) problem. If no, it solves the min overall average value (MIN-CL1) problem.
nCameras	Long Integer	0	Number of cameras allowed
nPOIsPerCamera	Double	2	Number of POIs each camera may surveil at a time
ObeyFixed	Yes/No	No	If "Yes," the fixed selections specified in table LOC must be followed
Max_Time	Long Integer	100	Maximum time (seconds) allowed to execute the run
Max_Gap	Double	0.0	Maximum optimality gap allowed (e.g., 0.05 means 5%)
Gap	Double		Actual gap achieved after the optimization (OUTPUT)
E_Value	Double		Overall expected damage, according to the objective of model MIN-CL[1] (OUTPUT)
Max_Val	Double		Maximum individual damage, according to the objective of model MI-CL[2] (but already corrected for the use of logarithms) (OUTPUT)
CPU_time	Double		Computational time (seconds) spent in the optimization (OUTPUT)

3. Other Inputs

Two other inputs are required to execute a problem: The database filename and the solver to be used. These are specified as parameters in the main program (BETSS_C.mos) of the application, which is the point from where the XPRESS-MP application is executed.

These parameters are called DBNAME and Use_CPLEX, respectively, and the excerpt in Figure 3 shows an example of how these can be set. Specifically, the highlighted example shows that the database to be used is under the "case\" folder (from the incumbent location of the BETSS_C.mos file) and the filename is "Test_Large.mdb." (Notice the extension ".mdb" should be omitted when specifying the file's name.)

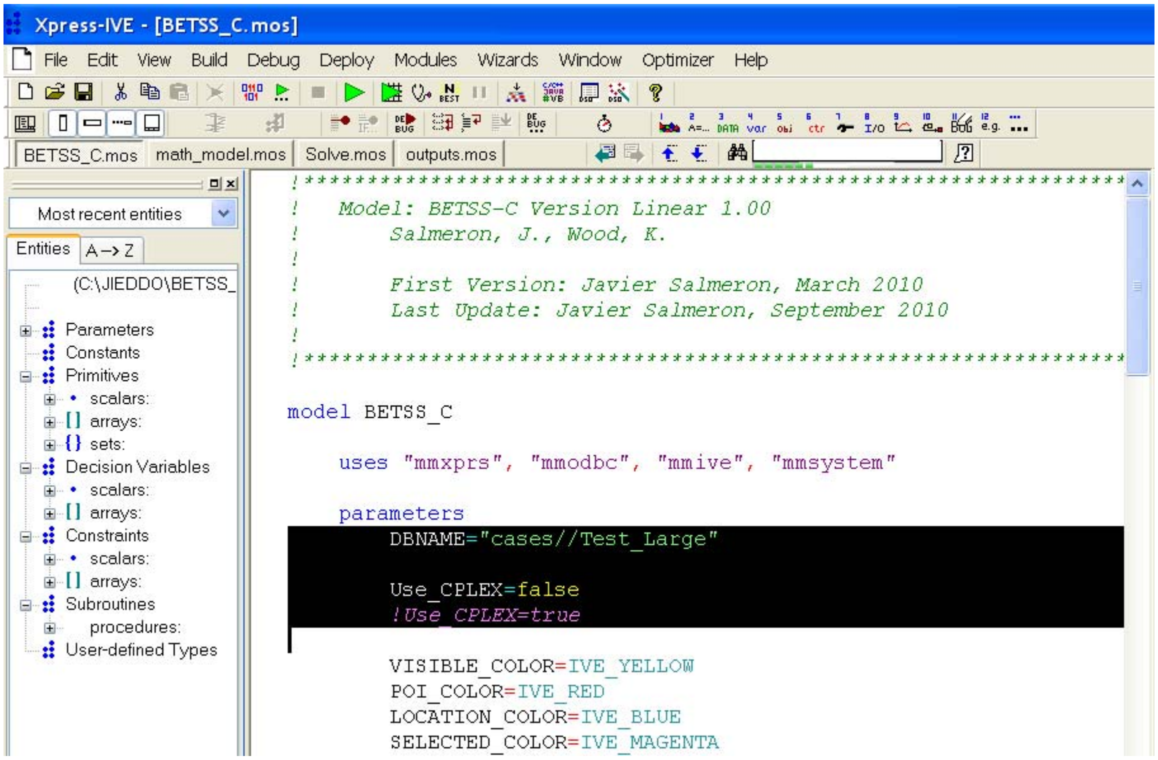

Figure 3. Specifying the database name and whether or not to use the CPLEX solver

Similarly, the example in the figure indicates the CPLEX solver is not going to be used. (Notice the "=true" line is commented out). Thus, all selected problems from the "case\Test_Large.mdb" database will be solved using the XPRESS-MP's internal solver.

4. Graphical Input and Output Environment

Xpress-MP's embedded graphical displays help visualize the problem and its solution.

For example, Figure 4 shows a snapshot that maps out POIs and candidate locations for cameras. For POIs we also see their value. By clicking on the "Visible" toggle, we would see a series of lines connecting candidate locations with those POIs that could be surveilled (with strictly positive probability of detection) from each location.

After the model is run, the "Selected" toggle shows the chosen selected locations for the case, indicated in parenthesis as average ("Avg") or min-max ("m-M") optimization, along with the number of cameras available for the case. The "Sel. Visible" (Selected Visible) toggle displays the assigned surveilled POIs from the selected locations.

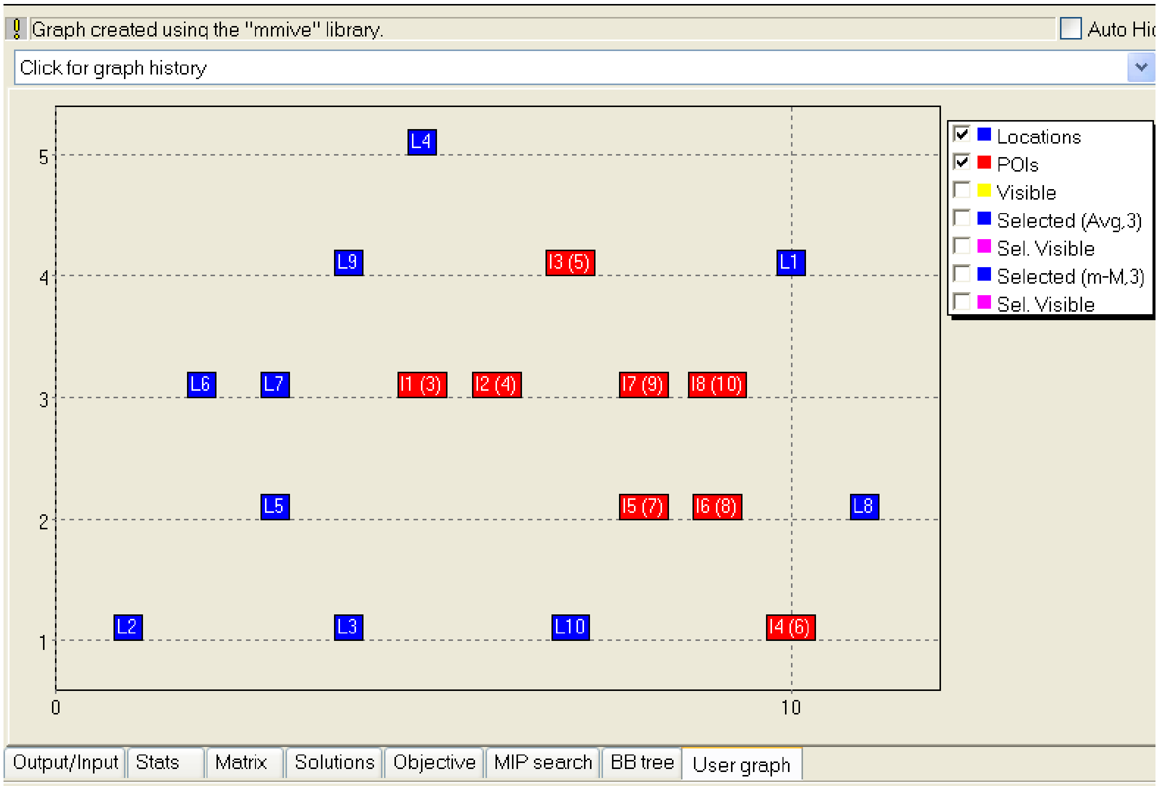

Figure 4. Preliminary display of locations (in blue) and points of interest (in red)

D. EXAMPLES

This section presents results for two hypothetical situations referred to as "Small example" and "Large example," respectively; each has several variants or "scenarios."

1. Small Example

The physical layout in this example (Figure 4) has ten potential camera locations and eight POIs. Figure 5 shows a portion of the example with "visibility" links activated, along with the associated probabilities of detection. For example, the probability of detecting POI "I4," if surveilled from "L8," is approximately 0.707.

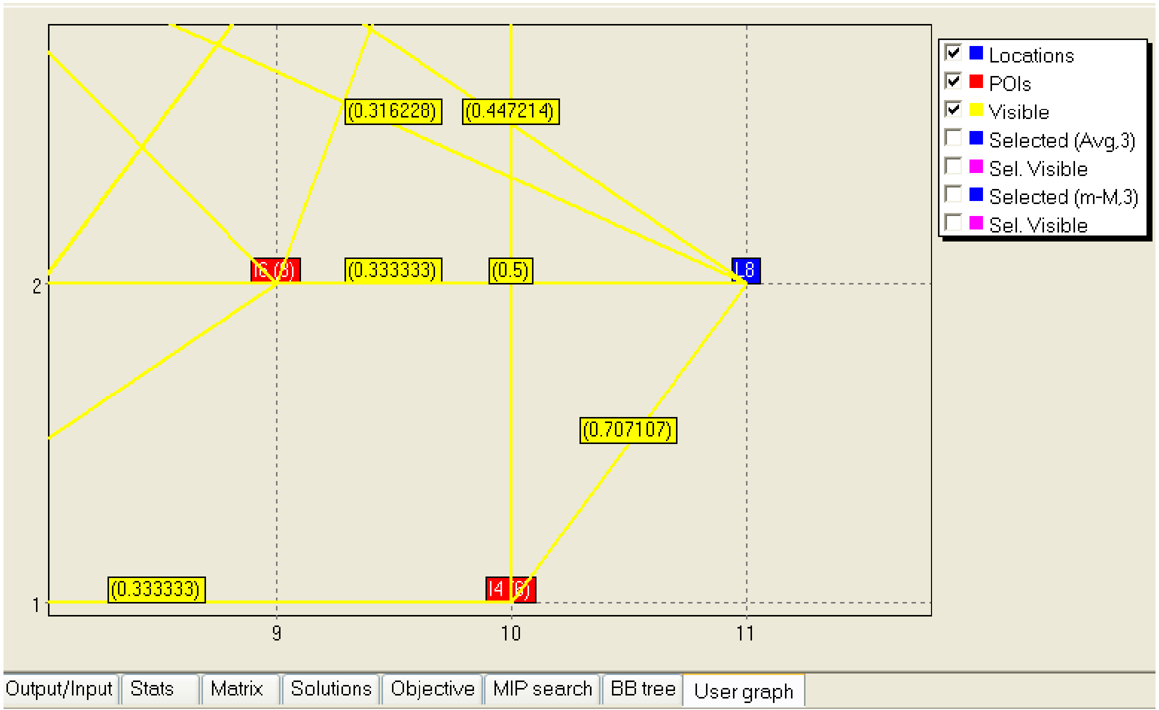

Figure 5. Small Example portion with line of sight (in yellow) and detection probabilities

We run four scenarios for this example, as indicated in Figure 6. The setting for the first scenario (indexed as "328" by the automated counter) seeks to minimize overall expected damage (model MIN-CL^1). We have three cameras available, and each camera can surveil an unlimited number of POIs simultaneously (indicated with a large limit of 100 in the data). The second scenario ("329") is the same as "328" except that the optimization model used is MIN-CL^2; that is, we seek to minimize the maximum damage at an individual POI. Scenario "338" and "339" resemble the first two scenarios, respectively, except that we reduce the number of POIs to be surveilled from any one location to a maximum of three. All scenarios are run for up to 100 seconds or when the optimality gap is zero.

SCENARIO

	index	Run	MinMax	nCameras	nPOIsPerCamera	ObeyFixed	Max_Time	Max_Gap
	328	☑	☐	3	100	☐	100	0
	329	☑	☑	3	100	☐	100	0
	338	☑	☐	3	3	☐	100	0
	339	☑	☑	3	3	☐	100	0
*	(New)	☑	☐	0	2	☐	100	0

Figure 6. Scenarios for Small Example

Results for the four scenarios are summarized in Figure 7. We note that the final gaps are zero (i.e., the solutions presented are guaranteed to be optimal) and the computational times are under one second in all cases.

The first two scenarios show that, when an unlimited number of POIs can be surveilled, both the overall-average and the worst-case optimal solutions are similar. The optimal objective for MIN-CL^1 yields an overall expected damage of 11.15 (scenario index “328”). The worst POI from this model has an expected damage of 1.93. Coincidentally, this is the optimal objective function value obtained when MI-CL^2 is used for that goal (scenario index “329”). The fact that the converse does not occur (i.e., the overall expected damage in scenario “329” does match that of “328”) is a consequence of multiple optimal solutions to the min-max objective in MI-CL^2. Figure 8 shows that the only difference between the two solutions is that MI-CL^2 does not surveil “I4” from “L1,” nor “I5” from “L8,” because that additional surveillance neither improves nor worsens the min-max objective.

SCENARIO

	index	Run	MinMax	nCameras	nPOIsPerCamera	Gap	E_Value	Max_Va	CPU_time
	328	☑	☐	3	100	9E-05	11.1535724224603	1.9393398	0.172
	329	☑	☑	3	100	0	11.9858896467254	1.9393398	0.016
	338	☑	☐	3	3	0	20.5991322548793	5	0.047
	339	☑	☑	3	3	0	20.8939885545277	3.40182	0.015
*	(New)	☑	☐	0	2				

Figure 7. Results for Small Contract scenarios

The last two scenarios, which limit the number of POIs that can be surveilled from a single location, have reduced detection probabilities. The optimal overall expected damage increases to 20.59, and the optimal worst-case individual damage increases to 3.40. Figure 9 displays these two scenarios.

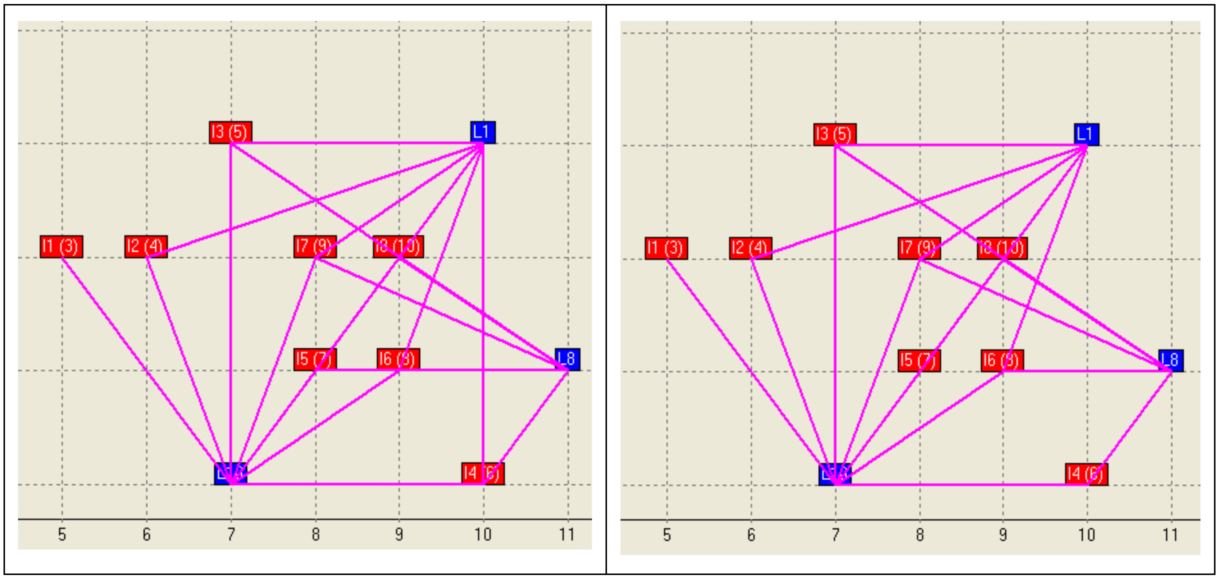

Figure 8. Graphical solution to scenarios “328” (left) and “329” (right)

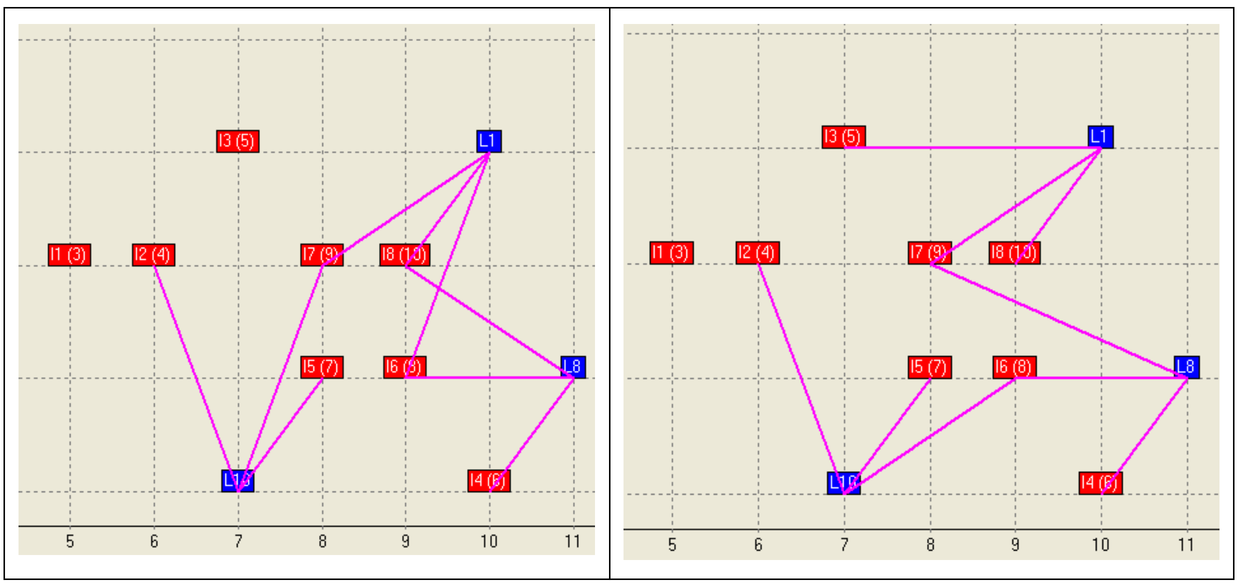

Figure 9. Graphical solution to scenarios “338” (left) and “339” (right)

2. Large Example

This setting has 30 candidate locations to surveil 100 POIs (Figure 10).

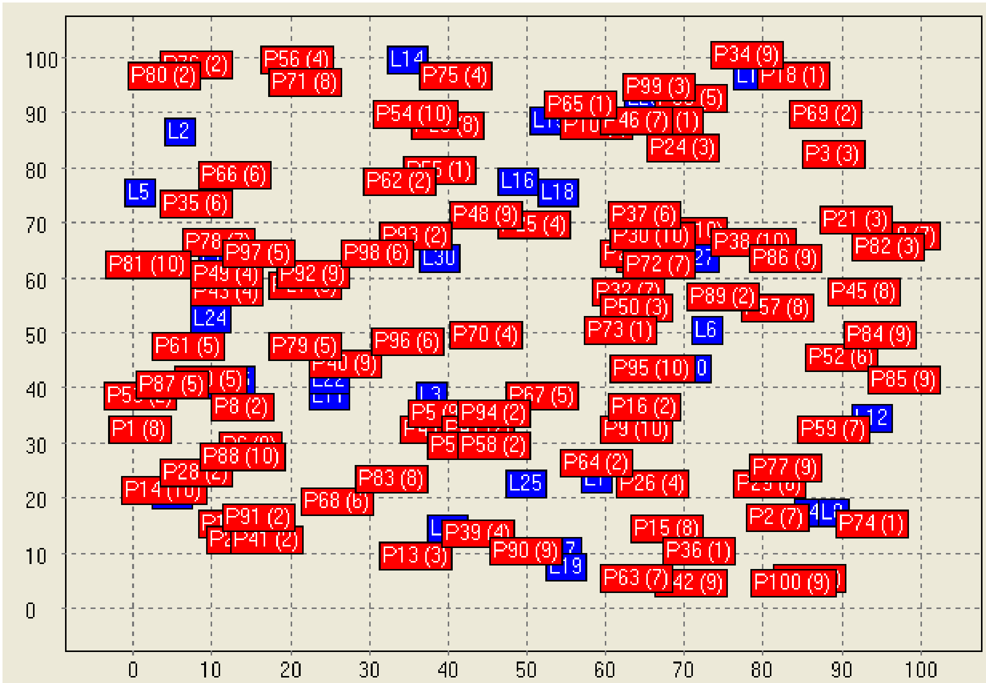

Figure 10. Large Example with 30 locations and 100 POIs

Figure 11 shows the scenarios. Scenarios "351" through "354" are solved via MIN-CL[1] with 5, 10, 15, and 20 cameras available, respectively, and a maximum of three POIs surveilled per camera. Scenario "355" is the same as scenario "354" (20 cameras available), but is solved with MI-CL[2], instead. Scenarios "356" though "359" explore a 20-camera case for a different number of POIs surveilled per camera. (Notice that "358" is the same scenario as "354.")

SCENARIO

index	Run	MinMax	nCameras	nPOIsPerCamera	ObeyFixed	Max_Time	Max_Gap
351	☑	☐	5	3	☐	100	0
352	☑	☐	10	3	☐	100	0
353	☑	☐	15	3	☐	100	0
354	☑	☐	20	3	☐	100	0
355	☑	☑	20	3	☐	300	0
356	☑	☐	20	1	☐	100	0
357	☑	☐	20	2	☐	100	0
358	☑	☐	20	3	☐	100	0
359	☑	☐	20	4	☐	100	0
(New)	☑	☐	0	2	☐	100	0

Figure 11. Scenarios for the Large Example

Results for all scenarios are summarized in Figure 12. As expected, there is a decrease in the overall damage as the number of available cameras increases (scenarios "351" through "354"). We can improve the worst-case individual damage (from 8.0 to 4.0 in this case) at the expense of increasing the overall expected damage from 156.14 to 207.86. Note that the solution to the min-max model MI-CL[2] takes all of the allotted time (five minutes), and stops with an optimality gap of 7.8%. Finally, we observe an improvement in the solution, as the number of POIs that can be surveilled per location increases.

SCENARIO

	index	Run	MinMax	nCam	nPOIsPerC	Gap	E_Value	Max_Val	CPU_time
	351	☑	☐	5	3	0	410.71	10	0.172
	352	☑	☐	10	3	0	306.09	9	0.437
	353	☑	☐	15	3	0	219.89	8	0.5
	354	☑	☐	20	3	0	156.14	8	5.094
	355	☑	☑	20	3	0.078	207.86	4	305.937
	356	☑	☐	20	1	0	364.56	10	0.14
	357	☑	☐	20	2	0	244.05	9	0.719
	358	☑	☐	20	3	0	156.14	8	5.141
	359	☑	☐	20	4	0	101.96	7	30.125
*	(New)	☑	☐	0	2				

Figure 12. Results for Large Example scenarios

Figures 13-17 present graphical displays for these solutions (but hide the POI names for the sake of clarity).

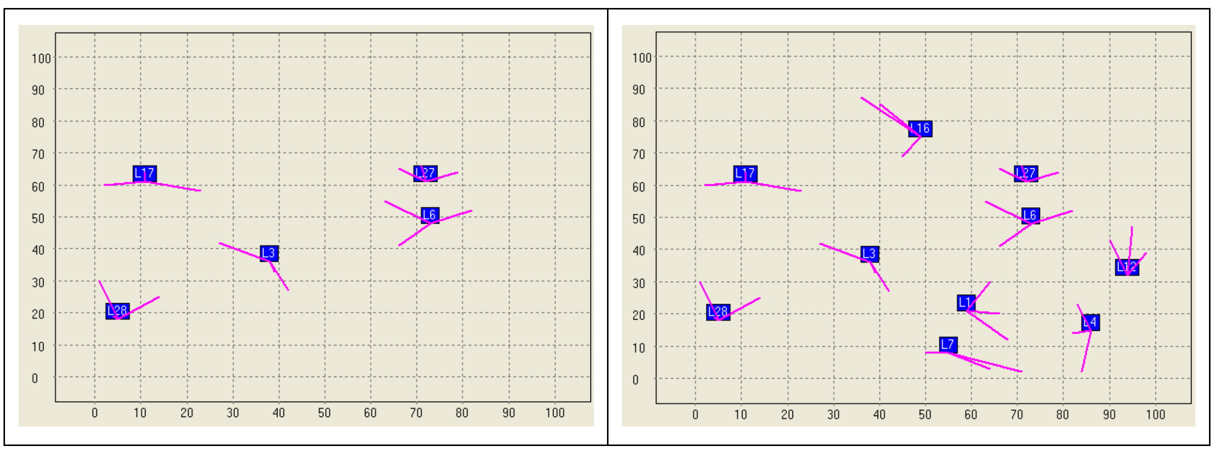

Figure 13. Graphical solution to scenarios “351” (left) and “352” (right)

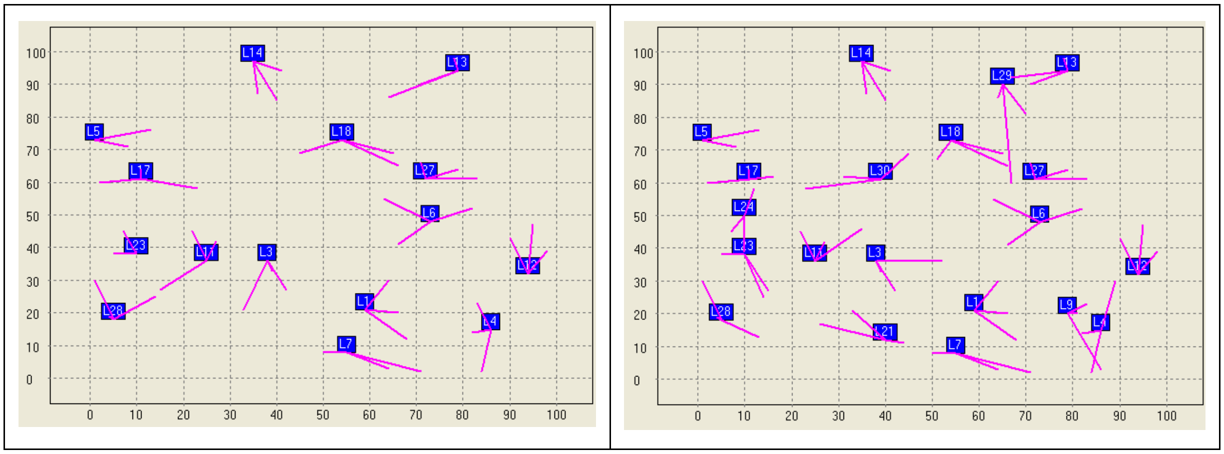

Figure 14. Graphical solution to scenarios “353” (left) and “354” (right)

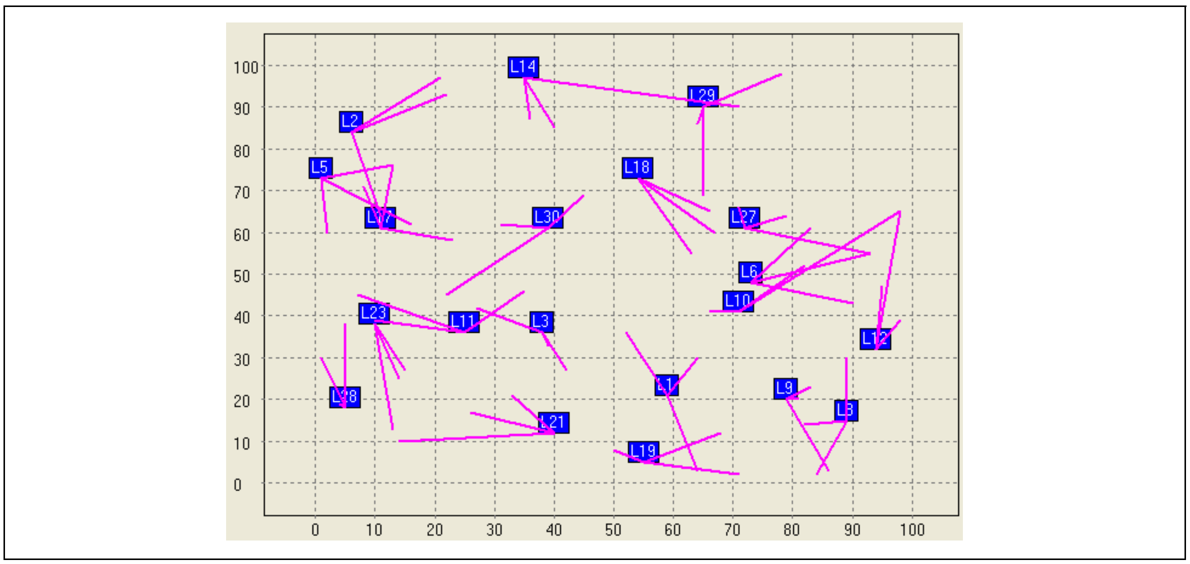

Figure 15. Graphical solution to scenario “355”

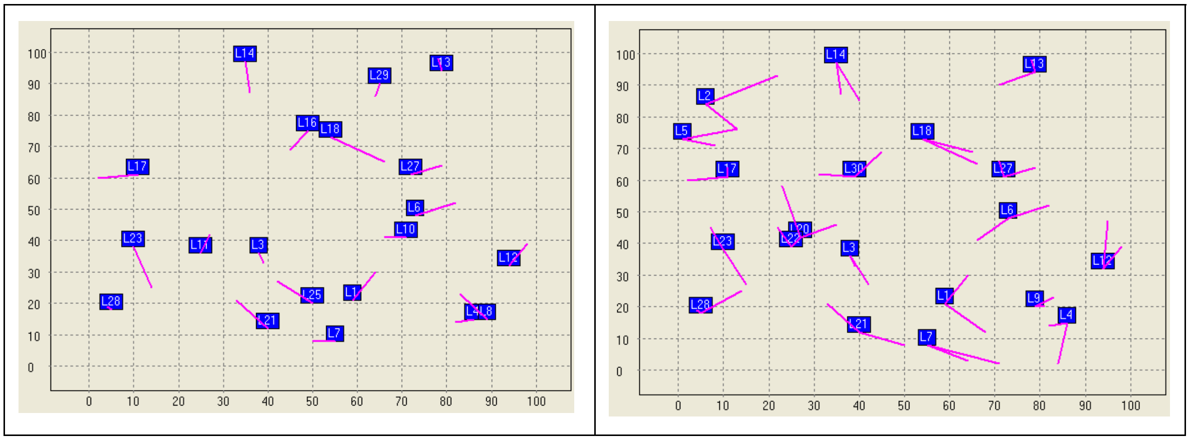

Figure 16. Graphical solution to scenarios “356” (left) and “357” (right)

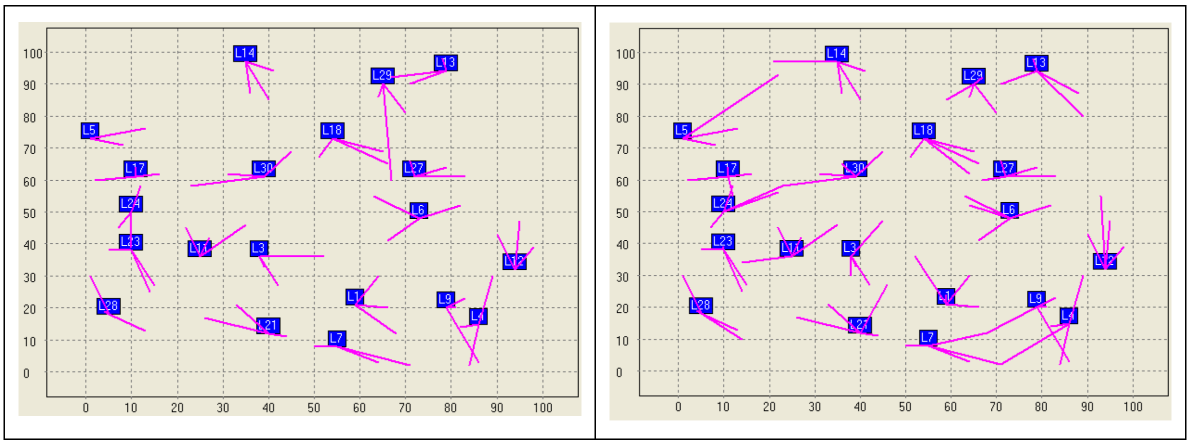

Figure 17. Graphical solution to scenarios “358” (left) and “359” (right)

E. FUTURE RESEARCH

The research performed in this project can be extended to more accurately assess and incorporate the “information value” of a collection of POIs that may be assigned to one or more camera towers. Our current implementation assumes a simple additive or separable value function that ignores “scheduling issues.” But, a camera that is set to surveil a given collection of POIs may be programmed to focus on, zoom in on, and surveil each POI for a given amount of time before transitioning to another POI. The corresponding surveillance and transition times affect the value of information collected (for example, the probability that an IED emplacement is detected), and should be part of the optimization process. A related problem is how those settings may affect the efficiency of multiple cameras surveilling a single POI “A.” For example, cameras C1 and C2 might have line of sight to a given POI “A,” but camera C1 could spend its time more fruitfully observing other POIs; thus, only camera C2 should observe POI “A,” or C1 should be scheduled to scan this POI only occasionally.

Secondly, our current models also assume constant conditions. However, day/night- or weather-related conditions may affect the probabilities of detection and could influence the optimal location of surveillance systems. In addition, we currently assume that the surveillance systems are fixed, i.e., once deployed, they cannot be relocated. By incorporating the dynamics of other surveillance systems (such as UAVs or the Eagle Eye mobile observation tower) in our analysis, it may be possible to provide better answers to problems with changing conditions.

Future research may also perform simulations to: (1) perform "what if" analysis for some key inputs which may be difficult to estimate by planners, such as POI values; and (2) analyze recommended locations under several potential terrorist behaviors. For example, a "greedy" (or risk-averse) behavior may be simulated as a terrorist attempting to emplace an IED at a high-value POI, which is being heavily surveilled. Other behaviors, from purely deterministic to random (including the selection of POI and time of day), can be integrated and may help assess, enhance, and validate the recommendations provided by the models described in this report.

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Ahuja, R.K., Magnanti, T.L., and Orlin, J.B. (1993). *Network Flows. Theory, Algorithms and Applications*. Prentice Hall, Upper Saddle River, NJ.

Burton, D., Kress, M., Lin, K., Rowe, A., and Szechtman, R. (2008). Optimal Mix and Employment of Sensors for Persistent Surveillance – Final Report. Working Report (obtained from authors at the Naval Postgraduate School, Monterey, CA).

FICO. (2010). FICO Xpress Optimization Suite [computer software]. Online: http://www.fico.com/en/Products/DMTools/Pages/FICO-Xpress-Optimization-Suite.aspx.

GAMS. (2010). GAMS [computer software]. Online: http://www.gams.com.

GAMS/CONOPT. (2010). GAMS/CONOPT [computer software]. Online: http://www.gams.com/solvers/solvers.htm#CONOPT.

IBM. (2010). IBM ILOG CPLEX Optimizer [computer software]. Online: http://www-01.ibm.com/software/integration/optimization/cplex-optimizer/.

Microsoft. (2010). Microsoft Access 2007 [computer software]. Online: http://office.microsoft.com/en-us/access/.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

1. Research Office (Code 09)..1
   Naval Postgraduate School
   Monterey, CA 93943-5000

2. Dudley Knox Library (Code 013)..2
   Naval Postgraduate School
   Monterey, CA 93943-5002

3. Defense Technical Information Center 2
   8725 John J. Kingman Rd., STE 0944
   Ft. Belvoir, VA 22060-6218

4. Richard Mastowski (Technical Editor) 2
   Graduate School of Operational and Information Sciences (GSOIS)
   Naval Postgraduate School
   Monterey, CA 93943-5219

5. Joint Improvised Explosive Device Defeat Organization (JIEDDO) 1
   5000 Army Pentagon, Washington, D.C. 20301-5000

6. Associate Professor Javier Salmerón 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5219

7. Distinguished Professor R. Kevin Wood 1
   Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5219

8. Professor Susan Sanchez...1
   SEED Center, Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5219

9. Associate Professor Tom Lucas 1
   SEED Center, Department of Operations Research
   Naval Postgraduate School
   Monterey, CA 93943-5219