Valuing Persistent ISR Resources

Tod S. Levitt[*], Kellen G. Leister[*], Ronald F. Woodaman[*], Jerrit K. Askvig[*], Kathryn B. Laskey[*], Rick Hayes-Roth[†]

[*]George Mason University
4400 University Drive
Fairfax, VA 22030-4400
[tlevitt, kleister, rwoodama, jaskvig, klaskey]@gmu.edu

[†]Naval Postgraduate School
589 Dyer Road, Bldg. 235 Root Hall 223
Monterey, CA 93943
fahayesr@nps.edu

***Abstract*— Persistent ISR (PISR) systems are a potential game-changer for asymmetric warfare. Realizing this potential requires delivering actionable information when and where it is needed, while shielding decision-makers from information overload. This paper describes a methodology and prototype system for measuring the military value of Persistent ISR deployments and allocating ISR assets to maximize military value. The approach follows the *Valued Information at the Right Time* (VIRT) approach advocated by Hayes-Roth [1]. To any tactical scenario supported by a persistent ISR (PISR) system, an associated set of high-information world events is defined that warrant immediate attention from decision makers. Value of a PISR system is measured by its ability to detect these conditions of interest (COIs) so as to trigger "smart push" information operations. A methodology is presented for allocating PISR assets to maximize the ability to detect and communicate COIs.**

Keywords - Valued information at the right time; persistent ISR; resource alloction; agile procurement; integer programming; multi-entity Bayesian networks

I. INTRODUCTION

A new generation of sensing resources is creating an information glut that threatens to overwhelm human decision makers. Initial studies suggest that the *Valued Information at the Right Time* (VIRT) approach can raise operator productivity, reduce information overload, and enhance the performance of information networks [1, 2]. In the VIRT appproach, "smart push" operations are triggered by the occurrence of Conditions of Interest (COIs), or observable events whose occurrence warrants immediate attention from decision-makers. This allows human operators to direct their attention where it is most needed without having to sift through volumes of irrelevant data. Nowhere is VIRT more urgently needed than the employment of persistent ISR (PISR) to support asymmetric warfare operations. For example, the successful mission targeting Abu Musab al Zarquawi required hundreds of hours spent analyzing raw data from Predator flights [3]. More effective alerting of human analysts could dramatically increase the combat effectiveness of future such missions.

Consider a PISR system deployed in a given tactical scenario. A process description of the system specifies observables associated with detecting, recognizing, identifying and/or tracking entities of interest. These observables can be used to define COIs for the tactical scenario. In this manner, COIs can be defined for each tactical scenario of interest. A PISR system, set of associated COIs, and process description is called a "PISR deployment."

A key question faced by planners is how to allocate a set of PISR assets across Forward Operating Bases (FOBs) and within the terrain in each FOB Operating Region (FOBOR). Taking the VIRT approach, assets should be deployed in a manner that maximizes their ability to detect the most important COIs. This is achieved by: (1) defining an objective function to represent the value of a PISR deployment; (2) identifying constraints each deployment must satisfy; and (3) finding a feasible optimal or near-optimal allocation of assets into a PISR deployment across FOBs and within FOBORs.

The objective function measures the value of a PISR deployment across a set of FOBs and their FOBORs given the tactical scenarios, their associated COIs and the costs of the PISR components. Measures of effectiveness (MOEs) are defined to capture factors of importance to decision-makers. These MOEs are included as components of the objective function and are weighted according to tradeoffs established by decision-makers. The objective function estimates the terrain-situated performance of a PISR system given its process description as actuated in that specific terrain positioning of PISR assets.

The constraints represent conditions that must be satisfied by PISR deployments. Constraints define where it is permissible to situate PISR assets in or above the

The research reported in this paper was supported by the Naval Postgraduate School under U.S. Government Contract N00244-10-1-0059. NPS also provided technical support and collaboration. The views, claims and opinions of the authors expressed in this paper do not necessarily state or reflect those of the United States Government or any of its agencies, and shall not be represented as such.

terrain. For example, communications must be in place between components that send alerts or controls to each other; sensors must be within range, possess resolution and have a field of view necessary to detect, recognize, identify, track and/or communicate as required, etc.

In addition, the threat must be considered in allocating PISR assets. For example, in a C-IED scenario, surveillance should be performed along regions where attacks are likely to occur; in a HVI scenario, cameras should be placed at locations where it is more likely that a subject will be in range of the camera. Threat operations are modeled as a spatio-temporal stochastic process, called the Threat Stochastic Process (TSP). The TSP models the likelihood, for each small battle-space region and time interval, for each type of Red operation, that the given threat operation will take place at that region within that interval of time. The objective function weights the value of detecting a COI if it occurs by the likelihood that the COI will occur.

A set of PISR assets designed to operate collaboratively to detect and report a COI is termed a PISR asset admissible configuration (AC). An optimization method is used to find a feasible deployment of ACs that maximizes the objective function. Because of the dynamic nature of tactical situations, the goal is an optimization process that can support agile changes in PISR deployment. As the battlespace evolves, the TSP should be modified and PISR deployments re-computed.

II. OVERVIEW

The objective of the VPR System is to value allocations of Persistent ISR (PISR) assets across Forward Operating Bases (FOBs) and within the terrain in each FOB Operating Region (FOBOR), and to find optimal allocations of PISR assets. Allocation is optimized both across FOBs and situated in terrain within FOBORs. Optimization algorithms are applied to maximize an objective function representing the valuation of PISR assets.

The value of a terrain-situated set of PISR assets is modeled as a linear function of its ability to detect and report Conditions of Interest (COIs) that are incidents of the Red scenarios.

Factors in the objective function weight the relative importance of detecting alternative COIs within and across threat scenarios.

A set of PISR assets that are designed to operate collaboratively to detect and report a COI are termed a PISR asset admissible configuration (AC).

The probability of detecting and reporting a COI given an AC is modeled as the marginal probability of the node in a Bayesian network (BN) corresponding to the event that the COI is detected and reported.

The probability of detection is a function of how the PISR asset admissible configuration interacts with its environment, including the likelihood that an incident of a given threat scenario's type will occur, and the effect of the environment on the ability of the PISR assets to detect COIs.

The Threat Stochastic Process (TSP) summarizes the *a priori* likelihood for the space-time locations of relevant Red threat activities in a region of interest over a period of time. The TSP represents the likelihood of each modeled type of threat activity (e.g., IED emplacement, patrol ambush, high-valued individual), conditioned by terrain suitability, intelligence information (e.g., history of Red activity, sympathies of the local populace, informatino about typical Red TTPs), expected Blue activities (e.g. planned patrols).

A. Functional Design

A functional flow diagram of the VPR system is shown in Figure 1. The system is divided into operations performed off- and online. The TSP and COI detection models are developed offline. The online system is intended to be interactively operated by a user who is determining the number, types, allocation and terrain-situated assignment of PISR assets across a set of FOBs.

B. Control-Theoretic Basis for Optimizing Value of PISR Resources

Friendly Forces (FF) surveille a FOBOR (or other region), G, for a set of possible COIs, each of which for this application is a threat behavior that can create a threat incident, i.e. an IED emplacement, an ambush or a HVI.

For each incident i of a given type within G, there is an associated set of observables that can indicate the likely presence of the threat. FF seek to detect the threat behaviors via their observables. To detect threat behaviors, FF have at their disposal a set A of PISR assets: sensors, communicators, processors, and platforms.

A Total Terrain Situated PSIR Assignment (TTSPA) is an assignment $\Lambda(A, G, T)$ of the members of A to locations within G over a time period T with the goal of observing all COI observables. Locating an asset within G implies that we also have fixed its employment parameters: its orientation and coverage, for example.

The Unconstrained TTSPA (U-TTSPA) is a TTSPA that can detect all threat behaviors (via their associated observables) within G that will occur during T.

For a given time interval T, assume we know when and where all the threat behaviors that might manifest within G will do so. Assume that for each incident i of a given type detected by a TTSPA, the system gains a payoff for detecting i called the Incident Value, IV_{im} specific to the incident's type m. We define the Unconstrained PISR Value as the sum of all the IVs from

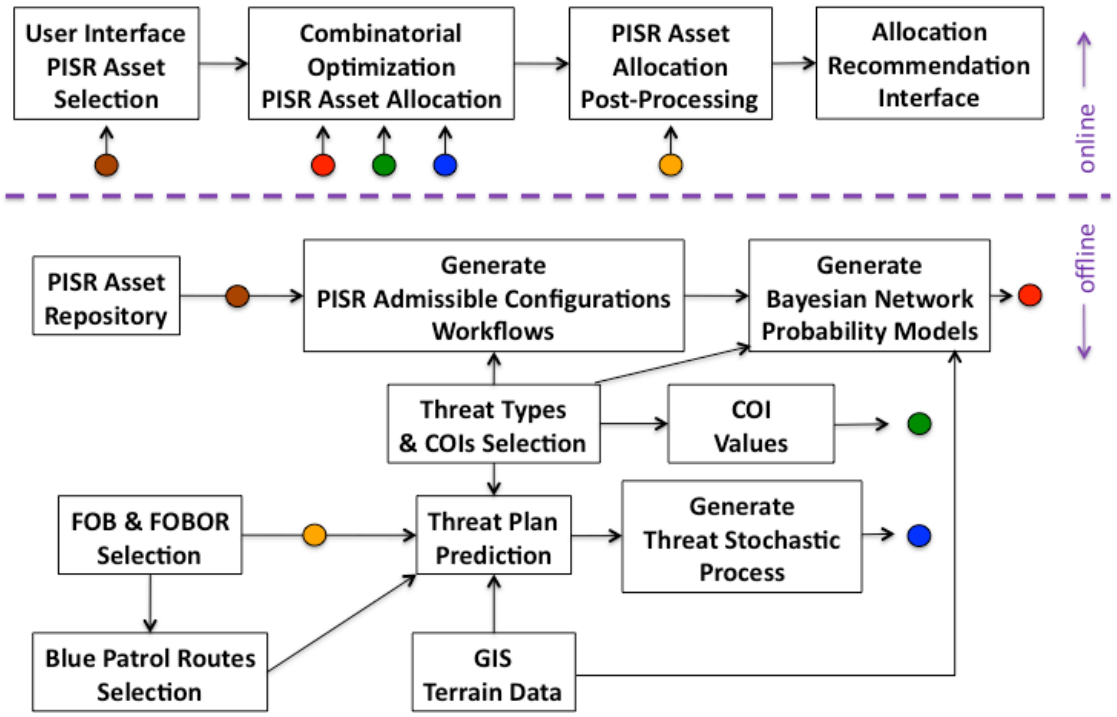


Figure 1 Valuing PISR System Functional Design

all the threat types that occur in *G* during *T*.

However, even given a U-TTSPA, it is impossible to know with certainty which incidents will occur within *G* during *T*. The TSP specifies the probability distribution of the specific incidents over *G* during *T*. The expected value under the TSP of the Unconstrained PISR value is called the Unconstrained Expected PISR Value (UEPV).

The U-TTSPA represents an ideal: that FF has as many perfect sensors as needed to guarantee detection of any observable that might occur with *G*. In actuality, any ISR asset set set *A* will be limited in both quantity and capability. The following constraints apply to TTSPA in real-world military scenarios:

- ***Sensors*** are limited in number, in range, in ability to detect observables even within their range, in endurance, environment, subject to threat counters, and are prone to false alarms.
- ***Communicators*** are limited in number, range, data throughput, time delay, endurance, by the environment, and are subject to threat counters.
- ***Processors*** are limited in number, processing throughput, reasoning capacity, errors, endurance, by the environment, and are subject to threat counters.
- ***Platforms*** are limited in number, maneuver, endurance, by the environment, and are subject to threat counters.

A Constrained TTSPA (C-TTSPA) is an assignment of the limited elements in *A* over *G* for the period *T*. An optimal assigment maximizes the Constrained Expected PISR Value (CEPV). In the U-TTSPA case, observables were detected with probability 1. In the C-TTSPA case, we must account not just for the likelihood of the incident manifesting at a particular point within *G* during *T*, but must now also account for the likelihood of the incident's observable being detected by the C-TTSPA.

The difference between UEPV and CEPV is termed the "PISR Gap". A metric for any C-TTSPA is the percent PISR Gap: [(UEPV – CEPV)/UEPV] x 100%. Clearly, maximizing CEPV minimizes the PISR Gap. A C-TTSPA that minimizes the PISR Gap is an optimal C-TTSPA.

Given a threat incident probability distribution over *G* and time period *T*, the objective is to determine the optimal C-TTSPA: the assignment(s) that maximize CEPV over all the feasible assignments Λ.

We followed a spiral development process in developing the optimization model. The initial spiral, which produces an optimal solution for a small, highly constrained problem. Subsequent spirals have increased the problem size, relaxed constraints and introduced additional complexities. For the first few spirals, in which optimal solutions are possible in short runtimes, we employed commercial off-the-shelf software, such as MPL, AMPL and CPLEX. In later spirals, as the problem

outgrows the capabilities of commercial software to solve the formulation in acceptable runtime, it will be necessary to adopt alternative heuristic approaches.

III. COMPONENTS

A. PISR System Workflows

A PISR system is any configuration of PISR assets that is defined by tactical users to be a "system". In practice this varies widely from individual PISR components such as a binocular imaging/video camera system manned by a warfighter, to complex systems of systems involving coordination of dozens of separate sensors, communication devices, networks and stationary and mobile platforms.

Regardless of the complexity of a PISR system, its usage in practice implicitly determines a formal workflow representation of how its components operate and inter-operate to accomplish the missions for which the PISR system is configured and employed.

Although the workflow of a specific PISR system can be complex, it is conceptually simple in that it is always a concatenation of elements that are essentially communications between any pair of: sensor, sensor platform, sensor controller, platform controller, exploitation module, communication device, user control panel or a user. Here a user is a warfighter playing any number of roles as an analyst, operator, controller, etc.

We can abstract the essential processes of a PISR system to these:

- *Sensor scan*;
- *Control sensor*, e.g., turn sensor on or off or re-point sensor;
- *Move platform*, e.g., fly a UAV or relocate a SMSS;
- *Exploit signal*, e.g.: (*i*) UGS detecting a vehicle, its locating, speed or stopping; (*ii*) AEDT module recognizing humans dismounting from a vehicle; (*iii*) face recognition module recognizing a HVI;
- *Send message*, e.g.: (*i*) sensor sending an alert to a communication device, usually a radio; (*ii*) exploitation module sending information to a communication device; (*iii*) communication device repeating an alert or message to another communication device; (*iv*) communication device sending information to a user display; (*v*) communication device sending information to a user; (*vi*) user sending a control signal to a communication device; (*vii*) communication device sending a control signal to a platform; (*viii*) communication device sending a control signal to a sensor
- *Display message*, i.e., alert user.

This level of abstraction of PISR system workflow is useful to produce generic configurations of PISR assets that form different sorts of PISR systems. For example, Figure 2 shows three different generic PISR workflow configurations.

A configuration of PISR assets is a PISR system that is constructed by incrementally assembling PISR components (which can themselves be systems) such that the resulting PISR system has a well-defined workflow.

In the VPR context, detecting and reporting of COIs defines the top-level mission for any PISR system. Therefore, for the purposes of the VPR design, we define a PISR configuration to be "admissible" if its workflow is capable of detecting and reporting a COI.

Generally speaking, except for a user, there are few fielded capabilities that can reliably recognize an incident corresponding to a COI, i.e. an IED emplacement, preparation for an ambush or recognize an HVI. Therefore we interpret "detecting a COI" to mean that an observable is detected that is evidential to the occurrence of a COI.

To detect emplacement of an IED we must have a sensor or combination of sensor and exploitation module that can detect observables that are evidential to an IED emplacement. In a sufficiently isolated area such observables include, for example, a vehicle stopping and/or persons dismounting from a stopped vehicle.

It is important that an admissible configuration (AC) includes not only one or more processes for detecting a COI but also one or more process for reporting a COI to a user. A COI is considered to be reported if evidence of a possible occurrence of a COI is successfully sent to a manned platform of any type, or to a display intended to alert a user.

For example, an unattended ground sensor (UGS) can detect a vehicle stopping. Therefore it is capable of detecting an IED or an ambush, because a vehicle stopping in a non-urban location can be evidential to those incidents. However even though an UGS is a PISR system, an UGS by itself is not an AC because it does not send its detection message to a user.

The simplest AC containing an UGS must include a radio and a ground station such as a Squad Mission Support System (SMSS) vehicle that can receive the UGS radio signal. The PISR configuration UGS->Radio->SMSS is an AC.

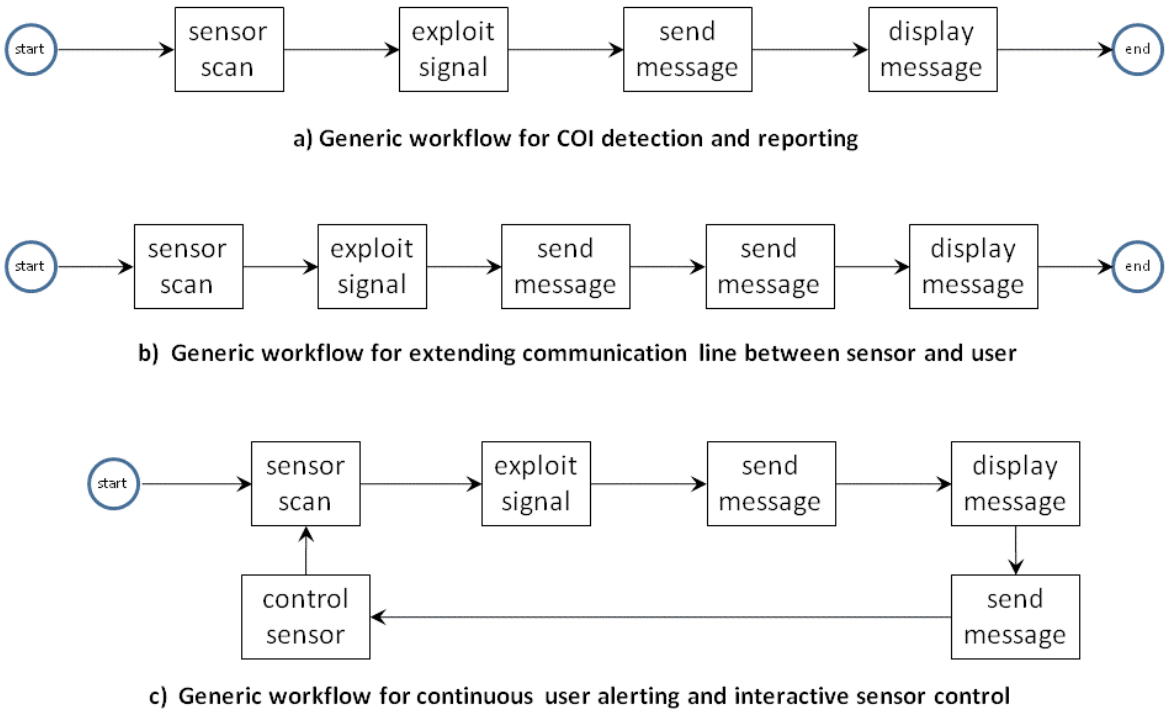


Figure 2 Generic PISR System Workflows

To optimize effective usage of PISR assets in the net-centric battle environment we would like a capability to dynamically configure plug-and-play PISR assets into ACs, in order to have the greatest flexibility to deploy them to achieve maximum value to the warfighter.

The formal XPDL semantics for workflow specified by the Workflow Management Coalition provide a standardized workflow representation methodology and implementation [4]. In theory this could be used in concert with a product line architecture (PLA) to produce a rigorous workflow representation with each PISR system configured under the PLA [5].

B. COI Detection & Reporting Probability Model

In order to value an AC as it is assigned to specific terrain, we must be able to estimate the probability that the AC as situated can detect and report a COI. This is accomplished by mapping the AC workflow to a probability model representing the chain of events resulting in reporting a COI. The probability model is specified as a Bayesian network (BN). An example BN is shown in Figure 3. Each consecutive pair of PISR assets in the AC workflow is mapped to parent-child pair of nodes in the BN. A conditional probability is specified for the event that the second asset in the pair triggers, given that the first asset has triggered. For our prototype system, the Bayesian networks were hand-constructed. The Bayesian networks for different admissible configurations involve combination of repeatable fragments, and are naturally represented as Multi-Entity Bayesian Network Fragments (MFrags), which can be assembled into situation-specific Bayesian networks (SSBN) [6]. Figure 3 shows an example of the transformation from AC workflow to SSBN. After SSBN construction, a standard BN inference algorithm can be used to calculate the likelihood of detecting and reporting the COI associated with the SSBN.

The term $Pd_{m,ac,sz,t}$ represents the probability that asset configuration *ac*, operating in zone *sz*, would detect the presence of an incident of type *m* (e.g., IED emplacement) during a time period *t*, given that an incident of type *m* occurred in *sz* during *t*. The steps in the construction of a SSBN used to calculate $Pd_{m,ac,sz,t}$ are described below:

1. A separate SSBN is generated for each combination of incident type, surveillance zone, time period, and asset configuration that can surveil *sz* in time period *t* and detect COIs indicative of the incident type (note we assume that just one asset configuration is assigned to a surveillance zone per time period; multiple asset configurations operating in a single surveillance zone are considered to constitute a single, larger configuration). Therefore, the first step in the construction of a BN is to specify the incident type, surveillance zone, time period and asset configuration.

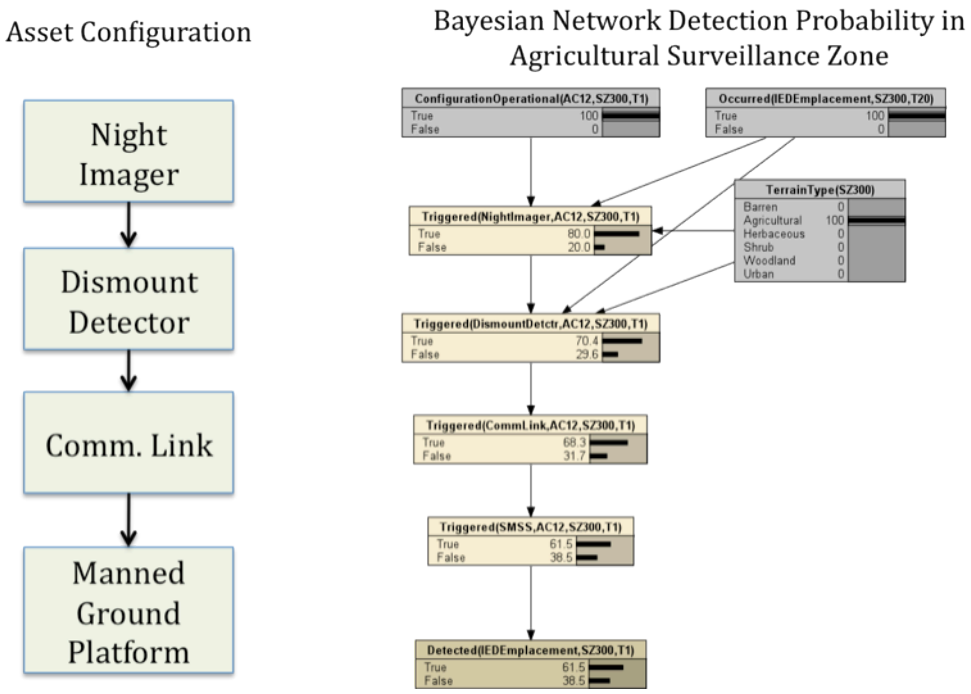
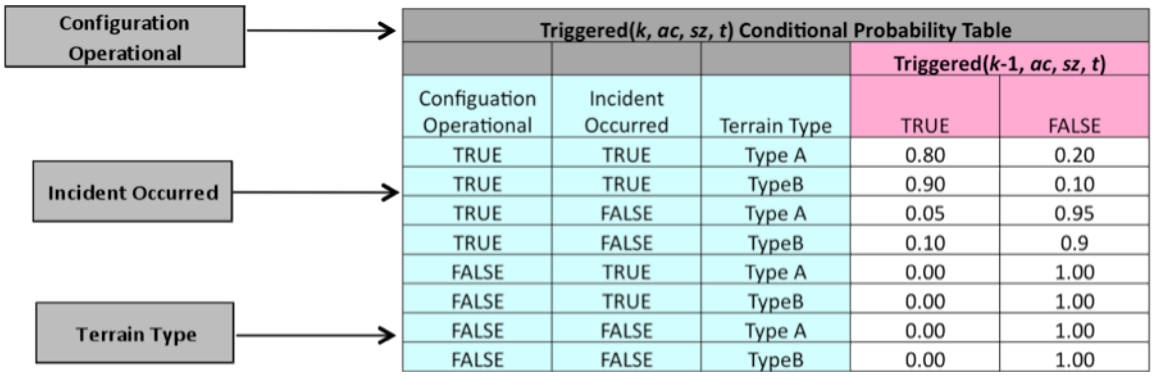


Figure 3: Admissible Configuration Workflow to Bayesian Network

2. A root node for the BN is created, labeled "Occurred(*m*, *sz*, *t*)", which represents presence or absence of an incident of type *m* in *sz* during *t*.
3. We next construct a root node for each admissible configuration that meets constraints for the surveillance zone. These nodes are labeled ConfigurationOperational(*ac*, *sz*, *t*). Exactly one of these nodes is set to True for each surveillance zone and time period, indicating the admissible configuration that is assigned to the surveillance zone during that time period. All others are set to False.
4. The surveillance zone incorporates terrain features (e.g., elevation, vegetation) that affect the probability that each ISR asset will function properly in *sz*. The BN, therefore, includes a node labeled TerrainType(*sz*), which incorporates states reflecting the various terrain types that might occur in *sz*. The terrain type node is set to the state that applies.
5. As described in section 4.1, each AC is composed of one or more components that have various functions (sensing, cuing, etc.), linked together to operate in series to detect and report the Incident. We assume that all of the components in *ac* must function correctly in order for *ac* to report the occurrence of an incident. Correct functioning of a component (which we refer to as "triggering") includes receiving a signal from the previous component in the *ac* series and performing any specific functions applicable to the (e.g., tracking), and forwarding a signal to the next serial component (if any) in *ac*. To model the probability that an asset configuration detects and reports an indident, we add to the SSBN a set of nodes representing the components of each asset configuration. These nodes are labeled Triggered(*k*, ac_j, *sz*, *t*) where *k* refers to a component of asset configuration *ac*. This node indicates whether or not the *k*th component in *ac* is triggered in surveillance zone *sz* during time interval *t*. The first such node is populated with the conditional probabilities that the component is or is not triggered given that the incident type that *ac* is able to detect occurs; if *ac* is not operational, then the first component does not trigger. The remaining such nodes are populated with the conditional probabilities that the component is or is not triggered given that the previous component triggered; the component is assumed not to trigger if its parent component does not trigger. As appropriate, these nodes may also be conditioned on terrain. Note that the conditional probabilities include the probabilities of false alarm. These probabilities would be provided through a combination of manufacturer's specifications for the ISR equipment and experience of the collections manager in using the ISR equipment. Figure Figure 4 shows the conditional probability table for one of the Triggered(*k*, *ac*, *sz*, *t*) nodes.
6. The prototype implementation did not consider



Triggered(*k*, *ac*, *sz*, *t*) Conditional Probability Table				
			Triggered(*k*-1, *ac*, *sz*, *t*)	
Configuation Operational	Incident Occurred	Terrain Type	TRUE	FALSE
TRUE	TRUE	Type A	0.80	0.20
TRUE	TRUE	TypeB	0.90	0.10
TRUE	FALSE	Type A	0.05	0.95
TRUE	FALSE	TypeB	0.10	0.9
FALSE	TRUE	Type A	0.00	1.00
FALSE	TRUE	TypeB	0.00	1.00
FALSE	FALSE	Type A	0.00	1.00
FALSE	FALSE	TypeB	0.00	1.00

Figure 4 Triggered(*k*, *ac*, *sz*, *t*) Node and its Parent Nodes

lighting conditions or weather; these could be modeled by adding additional nodes to the SSBN.

7. After all of the Triggered(*k*, *ac*, *sz*, *t*) nodes have been created, a final node is created, labeled Detected(*m*, *sz*, *t*), which combines the results of the individual components of the admissible configuration that is operational, into an overall determination of whether the threat incident of the type *m* was detected and reported by the admissible system being deployed to detect the incident in *sz* during time period *t*. The parent nodes of the IncidentDetected node are the final Triggered(*k*, *ac*, *sz*, *t*) nodes from each of the admissible configurations *ac*. We assume that if and only if the final component of the operational admissible configuration *ac* was triggered properly, then *ac* detected and reported the incident. Therefore, the Detected(*m*, *sz*, *t*) node registers True if and only if the parent Triggered(*k*, *ac*, *sz*, *t*) node of the operational admissible configuration registers True.
8. All these nodes are combined into a Bayesian network. The probability of True shown in the IncidentDetected node is the probability of detection of the incident given that it is present and that the admissible configuration assigned value True is being deployed in *sz* during *t*.

An example SSBN is shown in Figure 5, for a scenario in which AC#11 and AC#12 may be operational in SZ#300 during time period T1, to detect IED emplacements. To obtain from the BN the probability that the incident has been detected given that it has occurred, we set the ConfigurationOperational(*ac*, *sz*, *t*) to True for the operational configuration (AC#12 in this example) and False for the others, set the appropriate terrain type node (agricultural in this example), and set the Occurred(IEDEmplacement) node to True.

C. *Threat Stochastic Process*

In order to utilize PISR assets to maximum value it is necessary to have a model of how likely COIs are to occur over space and time. The standard military process for determining this is called Intelligence Preparation of the Battlefield (IPB). While IPB is well-developed as a craft, it does not have a formal mathematical basis for specifying prior probabilities of Red activities.

By mapping an IPB-like process into a spatio-temporal mathematical model, threat likelihoods can be continuously updated as the battlespace evolves.

We have developed a representation called the Threat Stochastic Process (TSP) that summarizes the *a priori* likelihood of relevant red threat activities for a period of time going forward. It depends on mission and Blue operations, and is conditioned on expected Blue activities, e.g. planned patrols.

The TSP is fundamentally spatial, i.e. geographic, incorporating:

- Terrain features: elevation, ground cover, hydrology, roads, etc. as they affect Red's preference for sites to perpetrate threat incident (e.g. where to emplace IED)
- History of Red threat activities and incidents (e.g. IED emplacements, ambushes)
- Red location estimates (e.g. populations, camps, safe houses, red logistical stores, etc.)
- Blue operations plans (usually at a general level).

TSP can also incorporate relevant temporal environmental effects such as time of day (illumination), time of year (weather), cultural effects (rites, events) and the effect on anticipated red activities of bomb damage or other battlefield effects.

The TSP can be the output of a Red threat prediction or other estimation procedure as long as that procedure and its output is consistent with the mathematical and scientific constraints of the TSP representation.

Figure 6 shows an example IED likelihood map

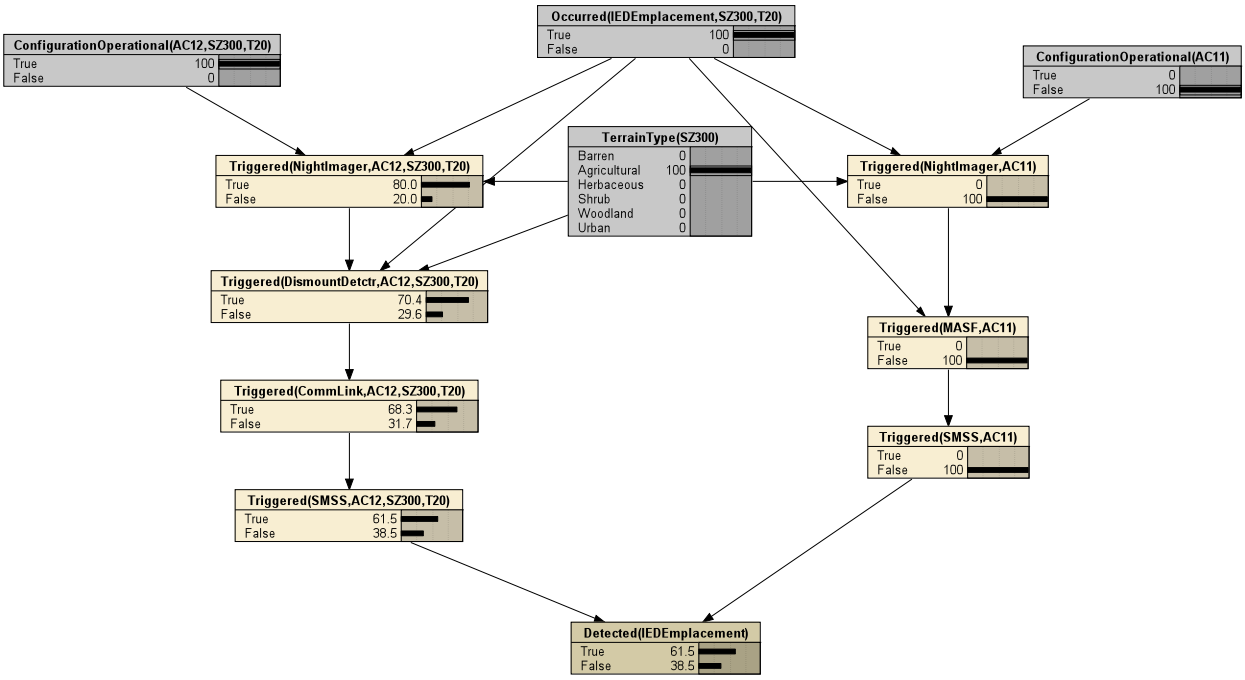


Figure 5: Constructed Bayesian Network

around three notional FOBs. Note that the likelihood is concentrated on the road network, because IEDs occur primarily on roads. Other types of threat would show different spatial patterns.

D. PISR Asset Value Optimization

The approach for measuring the value of a TTSPA is to calculate the expected value of assigning available PISR assets to specified surveillance zones within the region of interest, which can be, for example, the operating area for one FOB or for multiple FOBs. The expected value of assignment of PISR assets to terrain locations is calculated based on three factors:

- Vector of probabilities of occurrence of the threat activities within the surveillance zone;
- Matrix of probabilities of detection by the admissible configurations of PISR assets given threat incidents, within the surveillance zone; and
- Vector of values for successful detection of a threat instance for that surveillance zone.

The theory is based on a rational choice model that given the opportunity we should prefer the PISR assets assignment that maximizes aggregate expected value.

This can be viewed as a case of the Assignment Problem [7]: to find the optimal matching on a bipartite graph with weighted edges. A matching on a graph G is a set of non-adjacent edges. A bi-partite graph is a graph that can be partitioned into two sets of nodes, where within a set the nodes are not adjacent. In this case, the two sets of nodes are the PISR assets and the surveillance zones. The weights on the edges represent the expected assignment values. We wish the find the matching that assigns assets to zones so that the total expected assignment value is maximized.

Let a be the number of assets and s be the number of surveillance zones. It is reasonable to assume that $s >> a$. A brute force approach would require considering the permutations of a list of length a chosen from s. For 100 surveillance zones and 10 assets, that would require considering 6.3e+19 alternatives.

Efficient polynomial algorithms from network theory exist for exactly solving the assignment problem: including the:

- Hungarian algorithm
- Variations on Djikstra's shortest path, and
- Augmented flow algorithms.

However, the PISR asset assignment problem has complications, to include considering configurations of multiple PISR assets and configurations that cover more than one surveillance zone. Representing these possibilities requires constraints that take the solution beyond the reach of these polynomial run-time algorithms.

The algorithms and tools that we choose to employ in

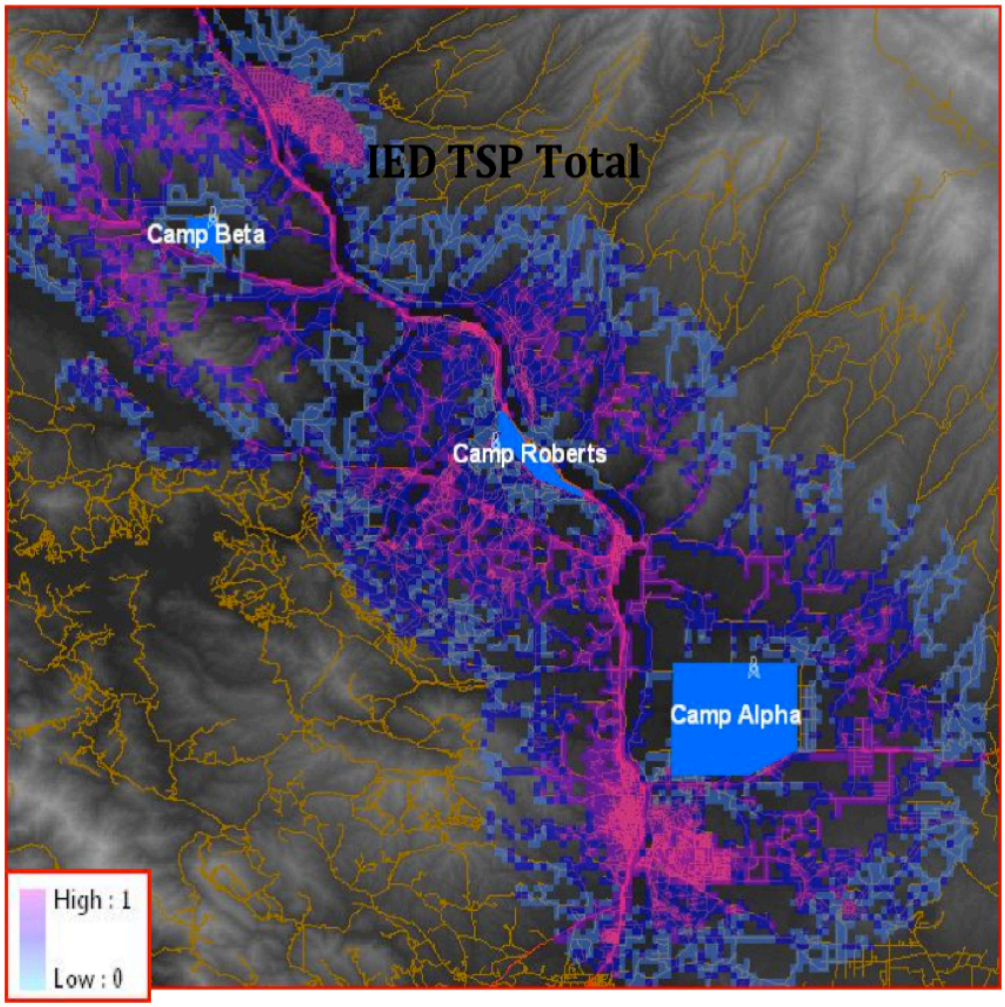


Figure 6: IED Likelihood Map

order to determine the TTSPA depend on several factors, including:

- Problem size - the numbers of available components, admissible configurations, minimal surveillance zones, and time periods will affect the number of assignment variables for which it is necessary to solve
- Assignment constraints – a spiral development approach to building an optimization model progresses from tight constraints that come from simplifying assumptions (e.g. one admissible configuration per surveillance zone and one surveillance zone per admissible configuration) to more practical constraints.
- Solution accuracy required – certain approaches may produce good, but not optimal assignment solutions in a short runtime. If operational conditions permit a longer runtime, a broader group of approaches is possible, including those guaranteed to produce optimal solutions.

When conditions permit use of a mathematical programming approach to solve the TTSPA, we use a binary integer program formulation of the assignment problem with side constraints, in which decision variables are equal to 1 when a specified component or admissible configuration is assigned to a specified surveillance zone in a given time period and 0 otherwise.

The basic integer programming algorithm employs a search-tree approach, implicitly enumerating the search space. Algorithm advancements have accelerated the search by identifying structure within the problem from which to generate cuts that rapidly prune the tree. This approach is guaranteed to yield provably optimal solutions. However, integer programming in general yields solution times that are exponential as a function of

the problem size. Thus for very large problems, the solution times may become prohibitive. Then it is necessary to have a means to identify feasible solutions to search heuristically.

Heuristic algorithms explore the search space of solutions without guaranteeing that an optimal solution is found. Typically they can be guaranteed to return solutions within a fixed period of time. Heuristic algorithms include several families of approaches:

- greedy search
- tabu search
- simulated annealing
- genetic and other evolutionary algorithms
- particle swarm algorithms
- memetic algorithms.

These approaches often yield results sufficient for their application purposes. However, the algorithms do not provide any information on the relative goodness of the solution generated in comparison with the optimal solution. Therefore, the integer programming formulation remains a critical tool for benchmarking the speed and degree of optimality of solutions obtained.

Sophisticated approaches to solving integer programs have combined the two, using heuristics to find good feasible starting solutions, and allowing the integer program to find an optimal solution. Commercial packages such as CPLEX employ some form of heuristic search as part of pre-processing of the problem before employing integer programming methods.

For this problem, other conceptual approaches may exist, such as employing a polynomial matching algorithm combined with a Lagrangian approach that penalizes infeasibilities and iterates towards a feasible solution. Commercial software is generally not available for these hybrid approaches, so substantial additional programming and development time and resources would be required to implement them.

For any heuristic approach chosen, the mathematical programming approach will remain critical for benchmarking the performance of the given heuristic using a set of test problems. Using the optimal benchmark, it is possible to make well-founded statements about the relative goodness of a given heuristic solution.

IV. Prototype Implementation

A prototype system was implemented to demonstrate feasibility of the approach. Figure 7 shows the flow of input data and output results through the VPR system. The input control panel is an Excel file used to input data for available asset quantities, detection probabilities for admissible configurations, and the relative value to Blue

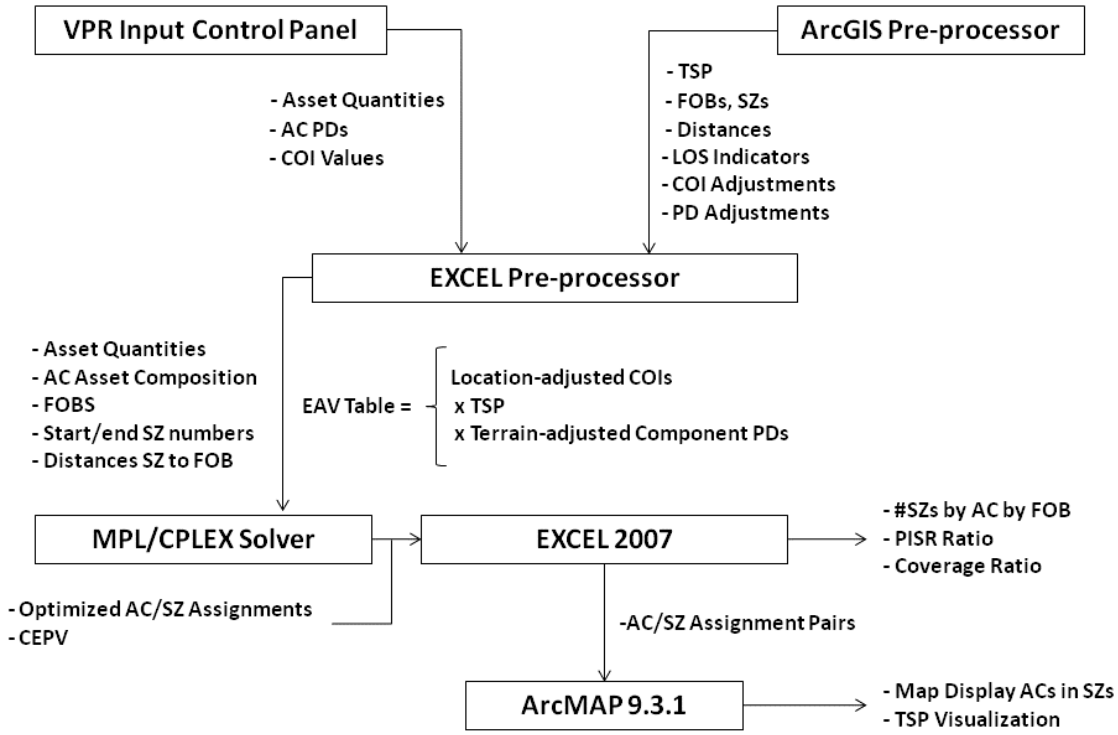


Figure 7 VPR Prototype System Dataflow

of the different COIs. The ArcGIS preprocessor is used to generate the threat stochastic process, to identify the surveillance zones and FOBs, and to incorporate distance, line-of-sight, location adjustments to the COI values defined in the input control panel, and terrain adjustments to detection probabilities. The Excel preprocessor accepts inputs from the input control panel and ArcGIS preprocessor, and produces output suitable for use by the MPL/CPLEX optimization solver. The MPL/CPLEX solver finds the optimal way to configure assets into admissible asset configurations and assign them to surveillance zones. The Excel output file accepts MPL/CPLEX output and produces a table showing the number of SZs surveilled by each AC within each FOBOR. These outputs are then read into ArcMAP to display the results on a map.

V. Experiments

A series of experiments was designed and conducted to examine the performance of the VPR methodology.

A. Objectives

Experiments were performed to: (*i*) assess practical limits on the number and size of the input parameters the optimization model can handle; (*ii*) quantify the benefit gained from optimal PISR assignments, compared to realistic simulated manual approaches; and (*iii*) study the effect of limiting the budget available for purchase of PISR assets.

B. Metrics

Metrics for the first objective were (1) the time required to load input data into the solver, and (2) the time required by the solver to identify the optimal assignment solution. We assume valid results are produced by the solver.

The primary metric for the second objective is the Constrained Expected PISR Value (CEPV). The method of analysis is to compare the CEPV for the optimal asset assignment to the CEPV for suboptimal assignment approaches. The ratio of the CEPV values quantifies the incremental value produced by VPR.

For the third objective, the CEPV is used to measure the value of optimal PISR assignments achievable with a given dollar investment in PISR assets. The ratio of CEPV for different budget levels indicates the relative value provided by the assets purchased under each of the budgets.

C. Experiments

We performed optimization runs on a base case of 6,178 SZs, each measuring 300m x 300m. To investigate scalability, we ran excursions that increased the number of surveillance zones, as described below. We used a fixed set of components, a basic quantity of each component, and a fixed set of 29 admissible configurations. We defined three notional FOBs in the vicinity of Camp Roberts, CA, and used the actual terrain around these FOBs to determine roads, adjust detection probabilities for terrain, and define the threat stochastic process. For the experiments, we used one Condition of Interest, IED detection.

To quantify the value added by VPR, we compared the optimal asset assignment with a heuristic we developed to simulate the manual process used by humans to allocate PISR components across a group of FOBs, assemble the components into admissible configurations, and assign the resulting configurations to surveillance zones. We then compared the resulting CEPV to the CEPV produced by PISR assignments using VPR. The heuristic allocates assets in proportion to the number of Blue personnel assigned to a FOB, creates ACs to maximize the number of SZs that can be monitored, and assigns ACs to SZs with the highest probability of IED emplacement.

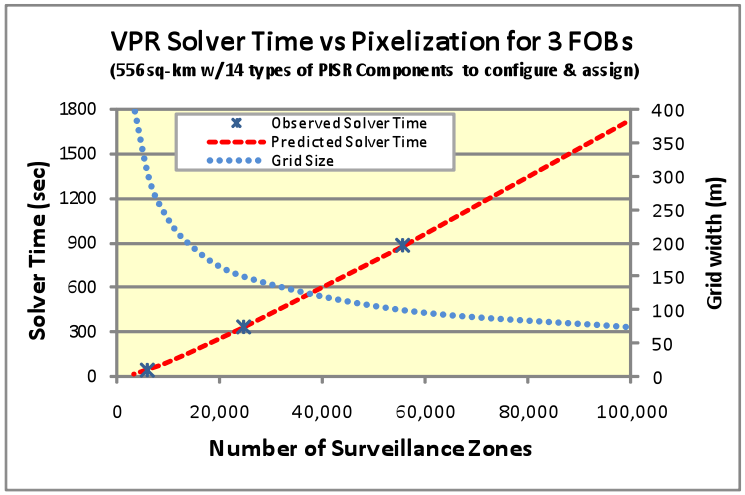


Figure 8 VPR Solver Time

To determine the effect of budget constraints on CEPV, we assigned a dollar cost to each component, and added a total dollar budget constraint to VPR. CEPVs were calculated under various budget limit scenarios. We then studied the resulting CEPVs as a function of budget limit.

D. Results

We consider each of the three objectives in turn.

Estimate Model Limits. We investigated scaling by shrinking the size of each SZ, resulting in a larger number of SZs and a correspondingly larger number of variables nd constraints. In addition to the base case of 6,178 SZs measuring 300m x 300m, we ran a case with 24,712 SZs measuring 150m x 150m, and 55,602 SZs measuring 100m x 100m. Figure 8 shows solver run times as a function of the number of SZs. Experiments were run on a Macbook Pro with 4GB RAM, solving the VPR problem via AMPL/Gurobi 3.0 optimization solver within a Parallels Windows XP virtual machine. We fit a low-degree polynomial to the data to extrapolate runtime for larger problems. We expect that, with a sufficiently powerful computer, most problems involving a practical number of surveillance zones could be handled. We did not vary the number of different components or ACs, or the quantity of each component, which could affect runtime. These variations could be pursued in future research.

For the experimental conditions, the graph shows that grid widths in the 200m – 300m range would appear to offer a good combination of resolution and runtime, wth runtimes of under 5 minutes. Different PISR assets may be better-suited to different-sized surveillance zones, so the mix of PISR assets to be optimized might affect the selection of grid width as well.

Another practical consideration is the time available for a PISR asset collections manager to run an optimization like VPR. In some situations, available runtime may be quite short, in which case it might not be practical to use small grid-widths. Smaller grid widths permit more precise positioning of PISR assets, so it would seem to be advantageous to use smaller grid widths, if possible. This may have the downside, however, of being susceptible to inaccuracies in the geo-spatial data at this level of resolution.

Quantify Value Added by VPR. The CEPV generated by VPR, along with the heuristic-generated CEPVs using various percentages of top-TSP SZs, are shown in Table 1.

Scenario	CEPV	Ratio VPR CEPV to Scenario CEPV
VPR- optimized	5723	1.0
Heuristic - Top 1% of SZs by TSP	1312	4.4
Heuristic - Top 10% of SZs by TSP	1225	4.7
Heuristic - Top 100% of SZs by TSP	830	6.9

Table 1 CEPV for VPR vs. Heuristic

The heuristic-generated CEPV using the top X% of SZs by TSP assumes that the human attempts to identify the optimal *n* SZs to surveille in a given FOB, and that he uses his perception of TSP as the basis for selecting the SZs. We assume that the human is not able to identify the top *n* SZs, but rather, a set of *n* SZs, the mean TSP of which matches the mean TSP of the highest X% of SZs. Thus, a very skilled human might identify SZs the mean TSP of which is the same as that of the top 1% of SZs by TSP. Random assignment of ACs to SZs would be expected to result in mean TSP value equal to that of all of the SZs (i.e., the top 100%).

There is a positive correlation between TSP for a SZ and the expected assignment value (EAV) for the SZ Recall that EAV is the product of TSP, probability of detection of a COI by an AC in the SZ, and the Value of the threat type occurring in the SZ. Terrain features of the SZ, especially distance from the FOB, affect all three of these values, which increases the correlation between TSP and EAV. Nevertheless, the correlation is less than 1.0, so selection of the top-TSP zones does not equate directly to selection of the top-EAV zones. VPR is able to identify the top SZs by EAV but the human is assumed not to possess this capability. Because of this advantage, as well as its ability to optimize the component-allocaton and AC assembly processes, VPR is able to produce a higher CEPV than can the heuristic process. In this experiment, VPR out-performs the heuristic by 340% to 690%, depending on the assumed ability of the human to identify top-TSP SZs.

Determine Effect of Budget Constraints. A final set of experiments was performed to determine the optimal mix of PISR assets for a given level of dollar investment. Using notional cost data for PISR assets, we selected a maximum ("Baseline") number of each of 14 PISR assets that are available for purchase. With a $13.5B budget, all the assets can be purchased. With a smaller budget, one must select some percentage of the maximum number of each component to purchase. We choose to purchase those assets that will maximum CEPV under optimal SZ assignment of the ACs that can be assembled using the assets purchased. Using alternative budget values ranging from $250,000 to $16,000,000, we used VPR to determine the optimal asset mix. The resulting CEPV values as a function of The budget are graphed and shown in Figure 9. As expected, returns diminish as the budget is increased.

Of interest is how the asset mix changes as the budget changes. At small budget levels, the GBOSS, GBLite and

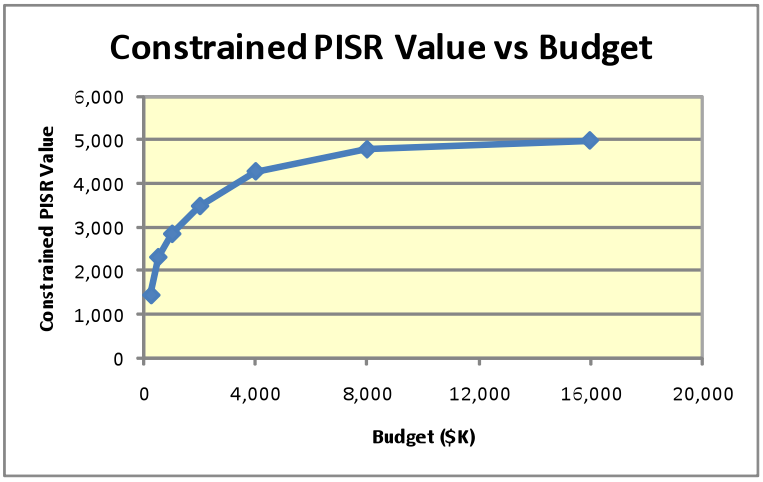


Figure 9 Component Quantities and Costs

UAV are not affordable relative to the value they generate based on the notional values used for this experiment. The less expensive assets are fully utilized (i.e. 100% are purchased) at even the lowest budget levels tested.

This type of analysis could prove useful in PISR asset acquisition and deployment exercises.

VI. Conclusion

A methodology was developed and prototype system implemented for valuing persistent ISR deployments and allocating PISR assets across FOBs and in terrain within FOBORs. Experiments indicate that optimal allocations can be found in minutes, freeing human operators from the tedious manual process of constructing allocations. The approach is practical for problems of reasonable size, and can scale to very large problems if heuristic optimization methods are used. The methodology can be used both to deploy existing assets and to prioritize acquisition of additional assets. The methodology is suitable for rapid modification of allocations to support agile deployment in a changing environment and rapid procurement to meet changing conditions on the ground.

References

[1] Rick Hayes-Roth, Two Theories of Process Design for Information Superiority: Smart Pull vs. Smart Push, *Command and Control Research and Technology Symposium,* San Diego, CA, US Department of Defense, Command and Control Research Program (CCRP), 2006.

[2] Rick Hayes-Roth, *Valued Information at the Right Time (VIRT): Why Less Volume Is More Value In Hastily Formed Networks*. Monterey, CA: Naval Postgraduate School Cebrowski Institute, 2006, retrieved from http://faculty.nps.edu/fahayesr/docs/VIRTforHFNs.pdf.

[3] Marc V. Schanz, The Indispensible Weapon. *Air Force Magazine*, *93*(2), 2010.

[4] http://www.wfmc.org/xpdl.html

[5] Barbara Hayes-Roth, Karl Pfleger, Philippe Lalanda, Philippe Morignot, and Marko Balabanovic. 1995. A Domain-Specific Software Architecture for Adaptive Intelligent Systems. *IEEE Transactions on Software Engineering 21*(4), 1995, pp. 288-301. DOI=10.1109/32.385968 http://dx.doi.org/10.1109/32.385968

[6] K.B. Laskey, MEBN: A language for First-Order Bayesian Knowledge Bases. *Artificial Intelligence*, *172*(2-3), 2008, pp. 140-178.

[7] Lawrence A. Woolsey, *Integer Programming*, John Wiley & Sons Inc. 1998.