Pacific Journal of Optimization
Proof : Deadline 5 May 2010

ROBUST CONNECTIVITY ISSUES IN DYNAMIC SENSOR NETWORKS FOR AREA SURVEILLANCE UNDER UNCERTAINTY*

Konstantin Kalinchenko, Alexander Veremyev, Vladimir Boginski, David E. Jeffcoat and Stan Uryasev

Abstract: We consider several classes of problems that deal with optimizing the performance of dynamic sensor networks used for area surveillance, in particular, in the presence of uncertainty. The overall efficiency of a sensor network is addressed from the aspects of minimizing the overall information losses, as well as ensuring that all nodes in a network form a robust connectivity pattern at every time moment, which would enable the sensors to communicate and exchange information in uncertain and adverse environments. The considered problems are solved using mathematical programming techniques that incorporate quantitative risk measures, which allow one to minimize or bound the losses associated with potential risks. The issue of robust connectivity is addressed by imposing explicit restrictions on the shortest path length between all pairs of sensors and on the number of connections for each sensor (i.e., node degrees) in a network. Specific formulations of linear 0-1 optimization problems and the corresponding computational results are presented.

Key words: *combinatorial optimization, sensor networks, surveillance, graph theory, robust connectivity, clique relaxations, optimization under uncertainty, conditional value-at-risk*

Mathematics Subject Classification: *90C27, 90C15, 90C35*

---

1 Introduction

In this paper, we address several problems and challenges arising in the task of utilizing dynamic sensor networks for area surveillance. This task needs to be efficiently performed in different applications, where various types of information need to be collected from multiple locations. In addition to obtaining potentially valuable information (that can often be time-sensitive), one also needs to ensure that the information can be efficiently transmitted between the nodes in a wireless communication/sensor network. In the simplest static case, the location of sensors (i.e., nodes in a sensor network) is fixed, and the links (edges in a sensor network) are determined by the distance between sensor nodes, that is, two nodes would be connected if they are located within their wireless transmission range. However, in many practical situations, the sensors are installed on moving vehicles (for instance, unmanned air vehicles (UAVs)) that can dynamically move within a specified area of surveillance. Clearly, in this case the location of nodes and edges in a network and the overall network topology can change significantly over time. The task of crucial importance in these settings is to develop optimal strategies for these dynamic sensor networks to operate efficiently in terms

*This research was partially supported by the U.S. Air Force contract # FA8651-08-D-0108.

of both collecting valuable information and ensuring robust wireless connectivity between sensor nodes.

In terms of collecting information from different locations (sites), one needs to deal with the challenge that the number of sites that need to be visited to gather potentially valuable information is usually much larger than the number of sensors. Under these conditions, one needs to develop efficient *schedules* for all the moving sensors such that the amount of valuable information collected by the sensors is maximized. A relevant approach that was previously used by the co-authors to address this challenge dealt with formulating this problem in terms of minimizing the *information losses* due to the fact that some locations are not under surveillance at certain time moments. In these settings, the information losses can be quantified as both *fixed* and *variable* losses, where fixed losses would occur when a given site is simply not under surveillance at some time moment, while variable losses would increase with time depending on how long a site has not been visited by a sensor. Taking into account variable losses of information is often critical in the cases of dealing with strategically important sites that need to be monitored as closely as possible. In addition, the parameters that quantify fixed and variable information losses are usually *uncertain*, therefore, the uncertainty and risk should be explicitly addressed in the corresponding optimization problems.

The other important challenge that will be addressed in this paper is ensuring *robust connectivity patterns* in dynamic sensor networks. These robustness properties are especially important in uncertain and adverse environments in military settings, where uncertain failures of network components (nodes and/or edges) can occur.

The considered robust connectivity characteristics will deal with different parameters of the network. First, the nodes within a network should be connected by paths that are not excessively long, that is, the number of intermediary nodes and edges in the information transmission path should be small enough. Second, each node should be connected to a significant number of other nodes in a network, which would provide the possibility of multiple (backup) transmission paths in the network, since otherwise the network topology would be vulnerable to possible network component failures.

Clearly, the aforementioned robust connectivity properties are satisfied if there are direct links between all pairs of nodes, that is, if the network forms a *clique*. Cliques are very robust network structures, due to the fact that they can sustain multiple network component failures. Note that any subgraph of a clique is also a clique, which implies that this structure would maintain robust connectivity patterns even if multiple nodes in the network are disabled. However, the practical drawbacks of cliques include the fact that these structures are often overly restrictive and expensive to construct.

To provide a tradeoff between robustness and practical feasibility, certain other network structures that "relax" the definition of a clique can be utilized. The following definitions address these relaxations from different perspectives. Given a graph $G(V, E)$ with a set of vertices (nodes) V and a set of edges E, a k-clique C is a set of vertices in which any two vertices are distance at most k from each other in G [3]. Let $d_G(i, j)$ be the length of a shortest path between vertices i and j in G and $d(G) = \max\limits_{i,j \in V} d_G(i, j)$ be the *diameter* of G.

Thus, if two vertices $u, v \in V$ belong to a k-clique C, then $d_G(u, v) \leq k$, however this does not imply that $d_{G(C)}(u, v) \leq k$ (that is, other nodes in the shortest path between u and v are *not required to belong to the k-clique*). This motivated Mokken [4] to introduce the concept of a *k-club*. A *k-club* is a subset of vertices $D \subseteq V$ such that the *diameter of induced subgraph* $G(D)$ is at most k (that is, there exists a path of length at most k connecting any pair of nodes within a k-club, where *all the nodes* in this path also belong to this k-club). Also, $\tilde{V} \subseteq V$ is said to be a k-plex if the degree of every vertex in the induced subgraph

$G(\tilde{V})$ is at least $|\tilde{V}| - k$ [8]. A comprehensive study of the maximum k-plex problem is presented in a recent work by Balasundaram et al. [1].

In this paper, we utilize these concepts to develop rigorous mathematical programming formulations to model robust connectivity structures in dynamic sensor networks. Moreover, these formulations will also take into account various uncertain parameters by introducing quantitative risk measures that minimize or restrict information losses. Overall, we will develop optimal "schedules" for sensor movements that will take into account both the uncertain losses of information and the robust connectivity between the nodes that would allow one to efficiently exchange the collected information.

2 Multi-Sensor Scheduling Problems: General Deterministic Setup

This section introduces a preliminary mathematical framework for dynamic multi-sensor scheduling problems. The simplest deterministic one-sensor version of this problem was introduced in [9]. The one-sensor scheduling problem was then extended and generalized to more realistic cases of multi-sensor scheduling problems, including the setups in uncertain environments by Boyko et al. [2]. In the subsequent sections of this paper, this setup will be further extended to incorporate robust connectivity issues into the considered dynamic sensor network models.

To facilitate further discussion, we first introduce the following mathematical notations that will be used throughout the paper. Assume that there are m sensors that can move within a specified area of surveillance, and there are n sites that need to be observed at every discrete time moment $t = 1, \ldots, T$. One can initially assume that a sensor can observe only one site at one point of time and can immediately switch to another site at the next time moment. Since m is usually significantly smaller than n, there will be "breaches" in surveillance that can cause losses of potentially valuable information.

A possible objective that arises in practical situations is to build a strategy that optimizes a potential loss that is associated with not observing certain sites at some time moments.

One can introduce binary decision variables

$$x_{i,t} = \begin{cases} 1, & \text{if } i\text{-th site is observed at time } t, \\ 0, & \text{otherwise,} \end{cases} \tag{2.1}$$

and integer variables $y_{i,t}$ that denote the last time site i was visited as of the end of time t, $i = 1, \ldots, n,\ t = 1, \ldots, T,\ m < n$.

One can then associate a fixed penalty a_i with each site i and a variable penalty b_i of information loss. If a sensor is away from site i at time point t, the fixed penalty a_i is incurred. Moreover, the variable penalty b_i is proportional to the time interval when the site is not observed. We assume that the variable penalty rate can be dynamic; therefore, the values of b_i may be different at each time interval. Thus the loss at time t associated with site i is

$$a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}). \tag{2.2}$$

In the considered setup, we want to minimize the *maximum penalty* over all time points t and sites i

$$\max_{i,t}\{a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t})\}. \tag{2.3}$$

Furthermore, $x_{i,t}$ and $y_{i,t}$ are related via the following set of constraints. No more than

m sensors are used at each time point; therefore

$$\sum_{i=1}^{n} x_{i,t} \leq m, \ \forall t = 1, \dots, T. \tag{2.4}$$

Time $y_{i,t}$ is equal to the time when the site i was last visited by a sensor by time t. This condition is set by the following constraints:

$$0 \leq y_{i,t} - y_{i,t-1} \leq t x_{i,t}, \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T, \tag{2.5}$$

$$t x_{i,t} \leq y_{i,t} \leq t, \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T, \tag{2.6}$$

Further, using an extra variable C and standard linearization techniques, we can formulate the multi-sensor scheduling optimization problem in the deterministic setup as the following mixed integer linear program:

$$\min C \tag{2.7}$$

$$\text{s.t.} \quad C \geq a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}), \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T, \tag{2.8}$$

$$\sum_{i=1}^{n} x_{i,t} \leq m, \ \forall t = 1, \dots, T, \tag{2.9}$$

$$0 \leq y_{i,t} - y_{i,t-1} \leq t x_{i,t}, \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T, \tag{2.10}$$

$$t x_{i,t} \leq y_{i,t} \leq t, \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T, \tag{2.11}$$

$$y_{i,0} = 0, \ \forall i = 1, \dots, n, \tag{2.12}$$

$$x_{i,t} \in \{0, 1\}, \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T, \tag{2.13}$$

$$y_{i,t} \in I\!R, \forall i = 1, \dots, n, \ \forall t = 0, \dots, T. \tag{2.14}$$

We allowed relaxation (2.14) of variables $y_{i,t}$ to the space of real numbers, because the constraints (2.5) and (2.6) enforce the feasible values of variables $y_{i,t}$ to be integer.

2.1 Cardinality Formulation

Lemma 2.1. *Constraint* (2.13) *is equivalent to the following combination of two constraints:*

$$0 \leq x_{i,t} \leq 1 \ \forall i = 1, \dots, n, \ \forall t = 1, \dots, T \ , \tag{2.15}$$

$$card(\mathbf{x}_t) \leq \sum_{i=1}^{n} x_{i,t} \ \forall t = 1, \dots, T \ , \tag{2.16}$$

where $\mathbf{x}_t = (x_{1,t}, \dots, x_{n,t})^T$*, and* $card(\mathbf{x}_t)$ *denotes the cardinality function for the vector* $\mathbf{x}_t$*. By definition,* $card(\mathbf{x}_t)$ *equals the number of non-zero elements in the input vector* $\mathbf{x}_t$*.*

Proof. Assume the matrix $(x_{i,t})$ satisfies constraint (2.13). Obviously, it then satisfies (2.15). At the same time, for every t, sum of all elements is equal to the number of values 1 in it. And these are the only non-zero elements in it. Therefore, constraint (2.16) is also satisfied.

Now assume the matrix $(x_{i,t})$ does not satisfy constraint (2.13). Thus there is a pair (i_δ, t_δ), for which $x_{i_\delta, t_\delta} = \delta$ and $\delta \neq 0$ and $\delta \neq 1$. If $\delta < 0$ or $\delta > 1$, then constraint (2.15) is violated. Thus, for all pairs (i, t), $0 \leq x_{i,t} \leq 1$, and $0 < \delta < 1$. Therefore, for all pairs (i, t), $\text{card}(x_{i,t}) \geq x_{i,t}$, and $\text{card}(\delta) > \delta$. Taking into account that $\text{card}(\mathbf{x}_t) = \sum_i \text{card}(x_{i,t})$ we conclude that (2.16) is violated. □

Now we can write alternative, cardinality formulation for the general deterministic sensor-scheduling problem.

$$\min C \tag{2.17}$$

$$\text{s.t.} \quad C \geq a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}),\ \forall i = 1, \ldots, n,\ \forall t = 1, \ldots, T, \tag{2.18}$$

$$\sum_{i=1}^{n} x_{i,t} \leq m,\ \forall t = 1, \ldots, T, \tag{2.19}$$

$$0 \leq y_{i,t} - y_{i,t-1} \leq t x_{i,t},\ \forall i = 1, \ldots, n,\ \forall t = 1, \ldots, T, \tag{2.20}$$

$$t x_{i,t} \leq y_{i,t} \leq t,\ \forall i = 1, \ldots, n,\ \forall t = 1, \ldots, T, \tag{2.21}$$

$$y_{i,0} = 0,\ \forall i = 1, \ldots, n, \tag{2.22}$$

$$0 \leq x_{i,t} \leq 1,\ \forall i = 1, \ldots, n,\ \forall t = 1, \ldots, T, \tag{2.23}$$

$$\text{card}(\mathbf{x}_t) \leq \sum_{i=1}^{n} x_{i,t},\ \forall t = 1, \ldots, T\ , \tag{2.24}$$

$$y_{i,t} \in I\!R, \forall i = 1, \ldots, n,\ \forall t = 0, \ldots, T. \tag{2.25}$$

Although the two formulations are equivalent, some optimization solvers, such as Portfolio Safeguard (that will be mentioned later in this paper), can provide a near-optimal solution faster if the formulation with cardinality constraints is used instead of the one with boolean variables, which may be important in time-critical systems in military settings.

3 Quantitative Risk Measures in Uncertain Environments: Conditional Value-at-Risk

To facilitate further discussion on the formulations of the aforementioned problems under uncertainty, in this section we briefly review basic definitions and facts related to the Conditional Value-at-Risk concept.

Conditional Value-at-Risk (CVaR) [5, 6, 7] is a quantitative risk measure that will be used in the models developed in the next section, which will take into account the presence of uncertain parameters. CVaR is closely related to a well-known quantitative risk measure referred to as Value-at-Risk (VaR). By definition, with respect to a specified probability level α (in many applications the value of α is set rather high, e.g. 95%), the α-VaR is the lowest amount η_α such that with probability α, the loss will not exceed η_α, whereas for continuous distributions the α-CVaR is the conditional expectation of losses *above* that amount η_α. As it can be seen, CVaR is a more conservative risk measure than VaR, which means that minimizing or restricting CVaR in optimization problems provides more robust solutions with respect to the risk of high losses (see figure 1).

Formally, α-CVaR for continuous distributions can be expressed as

$$\text{CVaR}_\alpha(\mathbf{x}) \ = \ (1-\alpha)^{-1} \int_{L(\mathbf{x},\mathbf{w}) \geq \eta_\alpha(\mathbf{x})} L(\mathbf{x}, \mathbf{w})\, p(\mathbf{w})\, d\mathbf{w}, \tag{3.1}$$

where $L(\mathbf{x}, \mathbf{w})$ is the random loss (penalty) variable driven by decision vector $\mathbf{x}$ and having a distribution in $I\!R$ induced by that of the vector of uncertain parameters $\mathbf{w}$.

CVaR is defined in a similar way for discrete or mixed distributions. The reader can find the formal definition of CVaR for general case in [6, 7].

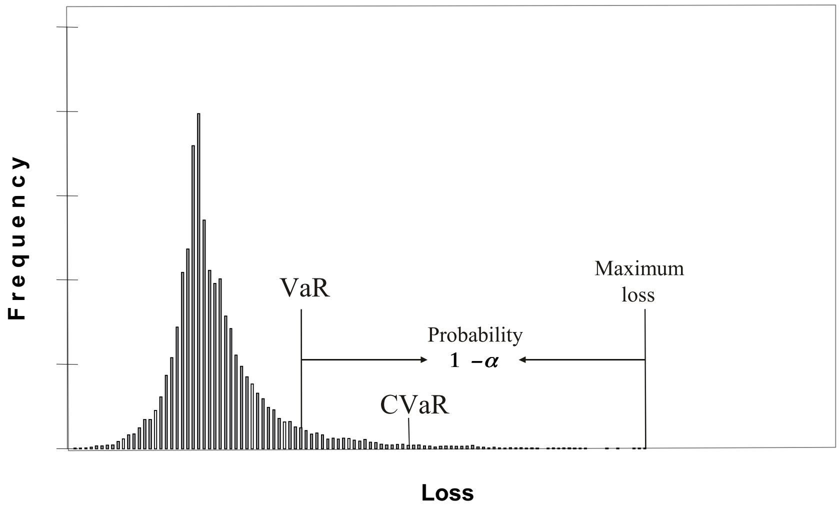

Figure 1: Graphical representation of VaR and CVaR.

It has been shown by Rockafellar and Uryasev [5] that minimizing (3.1) is equivalent to minimizing the function

$$F_\alpha(\mathbf{x},\eta) \;=\; \eta + (1-\alpha)^{-1}\int_{\mathbf{w}\in \boldsymbol{R}^d} [L(\mathbf{x},\mathbf{w})-\eta]^+ \, p(\mathbf{w})\, d\mathbf{w} \tag{3.2}$$

over $\mathbf{w}$ and η, where $[t]^+ = t$ when $t > 0$ but $[t]^+ = 0$ when $t \leq 0$, and optimal value of the variable η corresponds to the VaR value η_α, introduced above.

4 Optimizing the Connectivity of Dynamic Sensor Networks Under Uncertainty

This section extends the previous sensors scheduling problem to a stochastic environment. We use CVaR measure to model and optimize various objectives associated with the risk of loss of information.

In the stochastic formulation, the penalties a_i and $b_{i,t}$ are random. We generate S discrete scenarios, which approximate implied joint distribution. Thus, every scenario consists of two arrays: one-dimensional $\{a_i\}^s$ and two-dimensional $\{b_{i,t}\}^s$.

Now, consider the term of the loss function corresponding to the site i, time t, and scenario s:

$$L^s(x,y;i,t) = a_i^s(1-x_{i,t}) + b_{i,t}^s(t-y_{i,t}).$$

Under uncertainty, it is often more important to mitigate the biggest possible losses, rather than the average damage. Following this idea, we take $(1-\alpha)$ biggest penalties, and minimize average penalty over all i, t and s. This objective function is exactly the conditional value-at-risk.

We now have the following class of optimization problems:

$$\min_{x,y} CVaR_\alpha\{L(x,y;i,t)\} \tag{4.1}$$

This class has one extreme case: $\alpha = 1$, when the problem becomes equivalent to minimizing maximum possible penalty over all scenarios, locations and time points:

$$\min_{x,y} \max_{i,t,s} (a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t})). \tag{4.2}$$

This problem has an equivalent LP formulation:

$$\min C \tag{4.3}$$

$$\text{s.t.} \quad C \geq a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t}), \tag{4.4}$$
$$\forall i = 1, \ldots, n, \forall t = 1, \ldots, T, \forall s = 1, \ldots, S.$$

In order to formulate a general CVaR optimization problem in LP terms we have to introduce additional variables $\tau_{i,t}^s$, $s = 1, \ldots, S$, $i = 1, \ldots, n$, $t = 1, \ldots, T$, and η. With these variables the problem of minimizing CVaR will be reduced to the following:

$$\min C \tag{4.5}$$

$$\text{s.t.} \quad C \geq \eta + \frac{1}{(1-\alpha)nST} \sum_{\substack{s=1,\ldots,S \\ i=1,\ldots,n \\ t=1,\ldots,T}} \tau_{i,t}^s, \tag{4.6}$$

$$\tau_{i,t}^s \geq a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t}) - \eta, \tag{4.7}$$
$$\forall i = 1, \ldots, n,\ t = 1, \ldots, T,\ s = 1, \ldots, S,$$

$$\tau_{i,t}^s \geq 0,\ \forall i = 1, \ldots, n,\ t = 1, \ldots, T,\ s = 1, \ldots, S. \tag{4.8}$$

We have discussed various objective functions with objective-specific constraints for sensor scheduling problems in the stochastic environment. In addition to that, every sensor scheduling problem, including those in stochastic environment, must have constraints limiting number of sensors (2.4) and defining variables of the last time of observation (2.5)–(2.6). These constraints are referred to as mandatory constraints for every sensor-scheduling problem.

Further we define a wireless connectivity network $G(V, E)$ on the set of locations V. We interpret it in terms of the 0-1 adjacency matrix $E = \{e_{ij}\}_{i,j=1,\ldots,n}$, where each e_{ij} is a 0-1 indicator of wireless signal reachability between nodes i and j, that is, if locations i and j are within direct transmission distance from each other, then they are connected by an edge, and $e_{ij} = 1$ ($e_{ij} = 0$ otherwise). We also define a subnetwork $\tilde{G}$ of $G(V, E)$ containing only those m nodes (locations) that are directly observed by sensors at a particular time moment.

Scheduling of observation often requires sensors to maintain a certain level of wireless connectivity robustness. If an enemy sends a jamming signal that breaks connectivity between a pair of nodes, then the subnetwork $\tilde{G}$ either should stay connected, or at least should maintain unity with probability close to 1. Further, we will utilize several types of network structures that can be applied to ensure that the network satisfies certain robustness constraints.

The most robust network structure is a clique, which implies that each pair of nodes is directly connected by an edge. Obviously, maintaining a clique structure of the subnetwork $\tilde{G}$ at every moment in time is very expensive in terms of penalty, and can be even impossible, if the overall wireless connectivity network is not dense enough. Hence, it is reasonable to utilize appropriate types of *clique relaxations* to ensure robust network connectivity at every time moment.

One of the considered concepts is a k-plex. By definition, as mentioned above, a k-plex is a subgraph in which every node is connected to at least $m-k$ other nodes in it (where m is the number of nodes in this subgraph). This network configuration ensures that each node is connected to multiple neighbors, which makes it more challenging for an adversary to disconnect the network and isolate the nodes by destroying (jamming) the edges.

Another considered class of network configurations is a k-club. Recall that every pair of nodes in k-club is connected in it through a chain of no more than k arcs (edges). The motivation for studying this type of constraints is based on the fact that if two sensors are connected through a shorter path, it lowers the probability of errors in information transmission through intermediaries, since the number of intermediaries is smaller. Later in the paper, we will specifically use a stronger requirement on the length of these paths. We require that any two nodes are connected either directly by an edge, or through *at most one* intermediary node, which is often a desired robustness requirement under the conditions when the number of intermediary information transmissions needs to be minimized due to adversarial conditions. Clearly, a *2-club* is a structure that satisfies this requirement. In the next subsection, we show that this condition can be incorporated in the considered optimization models.

4.1 Ensuring Short Transmission Paths via 2-club Formulation

The general requirement for a subnetwork $\tilde{G}$ to represent a k-club can be formulated as the following set of constraints:

$$\begin{aligned} e_{ij} + \dots \\ &+ \sum_{q=1}^{n} e_{iq}e_{qj}x_{q,t} + \dots \\ &+ \sum_{q=1}^{n}\sum_{l=1}^{n} e_{iq}e_{ql}e_{lj}x_{q,t}x_{l,t} + \dots \\ &+ \sum_{q=1}^{n}\sum_{l=1}^{n}\sum_{p=1}^{n} e_{iq}e_{ql}e_{lp}e_{pj}x_{q,t}x_{l,t}x_{p,t} + \dots \\ &+ \sum_{i_1=1}^{n}\sum_{i_2=1}^{n}\cdots\sum_{i_{k-2}=1}^{n}\sum_{i_{k-1}=1}^{n} e_{ii_1}e_{i_1i_2}\dots e_{i_{k-2}i_{k-1}}e_{i_{k-1}j}x_{i_1,t}\dots x_{i_{k-1},t} \geq \\ &\geq x_{i,t} + x_{j,t} - 1, \end{aligned} \tag{4.9}$$

where $i = 1, \dots, n-1,\ j = i+1, \dots, n,\ t = 1, \dots, T$. For every k these constraints can be linearized, however, the size of the problem may substantially increase. In this paper, we limit our discussion only to 2-club constraints due to the practical reasons mentioned earlier in this section and due to the fact that the formulation for the case of $k = 2$ will not add too many new entities (no more than $O(n^2)$) to the problem formulation. They require every pair of nodes (i, j) to be connected directly, or through some other node p. Such type of communication between sensors (i, j) has a concise formulation:

$$e_{ij} + \sum_{p=1}^{n} e_{ip}e_{pj}x_{p,t} \geq x_{i,t} + x_{j,t} - 1\ \forall i = 1, \dots, n-1,\ \forall j = i+1, \dots, n,$$
$$\forall t = 1, \dots, T.$$

Here, the left-hand side is always nonnegative. The right-hand side becomes positive only if both locations i and j are observed by sensors, and then it equals 1. According to the 2-club definition, these sensors have to be connected (and exchange information) either directly, or through one other intermediary sensor node. In the first case e_{ij} equals 1. In the second case, the sum $\sum_{p=1}^{n} e_{ip}e_{pj}x_{p,t}$ will also be positive.

It is also important to note that those constraints, for which $e_{ij} = 1$, can be omitted. Thus, a 2-club wireless network configuration can be ensured by the following set of constraints:

$$\sum_{p\in\delta(i)\cap\delta(j)} x_{p,t} \geq x_{i,t} + x_{j,t} - 1,$$

$$\forall i = 1,\ldots,n-1,\ \forall j = i+1,\ldots,n,\ j \notin \delta(i),\ \forall t = 1,\ldots,T,$$

where $\delta(i)$ and $\delta(j)$ are the sets of neighbors of nodes i and j, respectively.

Below we present the complete general formulation for the dynamic sensor scheduling optimization problem in a stochastic environment with 2-club wireless connectivity constraints.

$$\begin{aligned}
&\min C \\
\text{s.t.}\quad & C \geq \eta + \frac{1}{(1-\alpha)nST} \sum_{\substack{s=1,\ldots,S\\ i=1,\ldots,n\\ t=1,\ldots,T}} \tau_{i,t}^{s} \\
& \tau_{i,t}^{s} \geq a_i^s(1 - x_{i,t}) + b_{i,t}^{s}(t - y_{i,t}) - \eta \\
& \forall i = 1,\ldots,n,\ t = 1,\ldots,T,\ s = 1,\ldots,S, \\
& \tau_{i,t}^{s} \geq 0,\ \forall i = 1,\ldots,n,\ t = 1,\ldots,T,\ s = 1,\ldots,S, \\
& \sum_{i=1}^{n} x_{i,t} \leq m,\ \forall t = 1,\ldots,T, \\
& 0 \leq y_{i,t} - y_{i,t-1} \leq t x_{i,t},\ \forall i = 1,\ldots,n,\ \forall t = 1,\ldots,T, \\
& t x_{i,t} \leq y_{i,t} \leq t,\ \forall i = 1,\ldots,n,\ \forall t = 1,\ldots,T, \\
& y_{i,0} = 0,\ \forall i = 1,\ldots,n, \\
& \sum_{p\in\delta(i)\cap\delta(j)} x_{p,t} \geq x_{i,t} + x_{j,t} - 1 \\
& \forall i = 1,\ldots,n-1,\ \forall j = i+1,\ldots,n,\ j \notin \delta(i),\ \forall t = 1,\ldots,T, \\
& x_{i,t} \in \{0,1\},\ \forall i = 1,\ldots,n,\ \forall t = 1,\ldots,T, \\
& y_{i,t} \in \mathbb{R},\ \forall i = 1,\ldots,n,\ \forall t = 0,\ldots,T.
\end{aligned}$$

4.2 Ensuring Backup Connections via k-plex Formulation

Constraints that require a wireless network to have the k-plex structure, can be defined using a symmetric adjacency matrix $E = \{e_{ij}\}_{i,j=1,\ldots,n}$, as defined above. Recall that $\mathbf{x}_t = (x_{1,t},\ldots,x_{n,t})^T$. Consider the vector $\mathbf{z}_t = (z_{1,t},\ldots,z_{n,t}) = E\mathbf{x}_t$. The element $z_{i,t}$ can be interpreted as the number of sensors which have a wireless connection with node i at time t. Thus, the constraint $E\mathbf{x}_t \geq \mathbf{x}_t$ or $(E - I)\mathbf{x}_t \geq 0$ ensures that each sensor node has at least one neighbor, i.e., it is not isolated. If we want each sensor to have at least $(m - k)$ wireless connections (edges) with other sensors, then we should make the constraints more restrictive: $E\mathbf{x}_t \geq (m-k)\mathbf{x}_t$, or

$$(E - (m-k)\, I)\, \mathbf{x}_t \geq 0 \;\; \forall t = 1, \ldots, T.$$

These restrictions by definition ensure that a subnetwork $\tilde{G}$ is a k-plex.

Below we present the complete general formulation for the dynamic sensor scheduling optimization problem in a stochastic environment with k-plex wireless connectivity constraints.

$$\begin{aligned}
& \min C \\
\text{s.t.} \quad & C \geq \eta + \frac{1}{(1-\alpha)nST} \sum_{\substack{s=1,\ldots,S \\ i=1,\ldots,n \\ t=1,\ldots,T}} \tau_{i,t}^s \\
& \tau_{i,t}^s \geq a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t}) - \eta \\
& \forall i = 1,\ldots,n,\ t = 1,\ldots,T,\ s = 1,\ldots,S, \\
& \tau_{i,t}^s \geq 0,\ \forall i = 1,\ldots,n,\ t = 1,\ldots,T,\ s = 1,\ldots,S, \\
& \sum_{i=1}^{n} x_{i,t} \leq m,\ \forall t = 1,\ldots,T, \\
& 0 \leq y_{i,t} - y_{i,t-1} \leq t x_{i,t},\ \forall i = 1,\ldots,n,\ \forall t = 1,\ldots,T, \\
& t x_{i,t} \leq y_{i,t} \leq t,\ \forall i = 1,\ldots,n,\ \forall t = 1,\ldots,T, \\
& y_{i,0} = 0,\ \forall i = 1,\ldots,n, \\
& (E - (m-k)\, I)\, \mathbf{x}_t \geq 0,\ \forall t = 1,\ldots,T \\
& x_{i,t} \in \{0,1\},\ \forall i = 1,\ldots,n,\ \forall t = 1,\ldots,T, \\
& y_{i,t} \in I\!R,\ \forall i = 1,\ldots,n,\ \forall t = 0,\ldots,T.
\end{aligned}$$

5 Computational Experiments

Computational experiments on sample problem instances have been performed on Intel Xeon X5355 2.66 GHz CPU with 16GB RAM, using two commercial optimization solvers: ILOG CPLEX 11.2 and AORDA PSG 64 bit (MATLAB 64 bit environment). It should be noted that due to the nature of the considered class of problems, they are computationally challenging even on relatively small networks. Therefore, in many practical situations, finding near-optimal solutions in a reasonable time would be sufficient. The PSG package was used in addition to CPLEX because it has attractive features in terms of coding the optimization problems, and therefore it may be more preferable to use in practical time-critical settings. In particular, in addition to linear and polynomial functions, PSG supports a number of different classes of functions, such as CVaR and cardinality functions. For the purposes of the current case study we defined in PSG the objective using the CVaR function, and we also used cardinality function for the cardinality constraint on $x_{i,t}$ instead of boolean constraint.

For comparison purposes, multiple experiments have been performed. All experiments were divided into two groups: with 2-club connectivity constraints on subnetwork $\tilde{G}$, and k-plex constraints with $k = \left[\frac{m}{2}\right]$. In each of these groups, number of locations $n = 10, 11, 12, 13, 14, 15$ and number of sensors $m = 4, 5, 6, 7, 8$. All problems have CVaR-type objective with $\alpha = 0.9$, deterministic setup (1 scenario), 20 time intervals. The edge density of the considered overall wireless connectivity network was $\rho = 0.8$ (80% pairs of nodes are connected).

Table 1: CPLEX Results: Problem with 2-club Constraints

Case		PSG CAR		PSG TANK		CPLEX 26 sec		CPLEX 1 min	
n	m	value	time	value	time	value	time	value	time
10	4	97.96	22.3	100.61	21.6	84.62	26.1	84.62	60.1
10	5	79.29	22.8	83.69	25.6	74.30	26.0	71.57	60.0
10	6	74.79	24.2	71.15	23.6	63.32	26.1	63.32	60.1
10	7	64.82	25.6	61.68	26.5	57.54	26.1	57.54	60.0
10	8	52.27	26.7	51.86	25.2	50.76	26.1	50.21	60.0
11	4	106.19	23.2	105.16	24.2	92.21	26.1	92.21	60.1
11	5	86.61	23.0	85.21	22.4	75.54	26.1	75.54	60.1
11	6	75.08	23.9	76.21	22.5	70.79	26.1	66.32	60.1
11	7	68.43	25.4	69.93	24.0	58.37	26.0	58.37	60.1
11	8	60.44	24.4	61.05	23.8	57.01	26.1	57.01	60.1
12	4	122.32	24.1	124.00	22.8	105.45	26.1	105.45	60.1
12	5	91.25	22.8	98.02	22.6	82.08	26.2	81.96	60.1
12	6	84.69	22.8	81.76	23.0	72.65	26.0	72.65	60.1
12	7	73.44	23.6	76.95	22.1	64.11	26.1	64.11	60.1
12	8	64.78	25.6	61.95	24.0	56.10	26.1	56.10	60.1
13	4	126.37	24.5	119.76	28.9	98.46	26.0	98.46	60.1
13	5	94.48	24.0	104.78	26.5	86.31	26.0	88.20	60.1
13	6	82.46	24.2	83.29	27.2	76.61	26.1	76.61	60.1
13	7	73.97	25.5	74.59	30.8	70.53	26.1	67.93	60.0
13	8	71.57	26.4	69.79	33.9	59.92	26.1	59.92	60.1
14	4	135.75	25.6	139.06	27.3	118.41	26.0	112.74	60.1
14	5	109.75	27.1	114.27	27.2	95.01	26.1	94.87	60.1
14	6	89.58	24.3	93.82	27.2	79.54	26.1	79.54	60.1
14	7	80.70	25.9	80.89	23.3	70.53	26.2	70.37	60.1
14	8	75.31	26.0	76.88	26.6	65.26	26.1	61.67	60.1
15	4	155.67	27.9	145.00	26.7	127.00	26.2	126.82	60.1
15	5	113.18	25.8	115.65	28.8	104.06	26.1	102.34	60.1
15	6	95.51	24.8	99.11	28.1	90.80	26.1	82.74	60.1
15	7	85.96	25.0	86.49	27.8	74.22	26.1	74.22	60.1
15	8	77.81	26.1	76.83	34.4	68.09	26.1	68.22	60.0
			avg		avg		avg		avg
			24.8		26.0		26.1		60.1

We have run PSG using two built-in solvers: CAR and TANK. These solvers took on average 26 seconds to deliver solution over all cases with 2-club constraints, and 27 seconds for the cases with k-plex constraints. After that we run CPLEX on cases with 2-club constraints with time limit 26 seconds, on cases with k-plex constraints with time limit 27 seconds. Then, we additionally run CPLEX on all cases with time limit 1 minute. Computational results are presented in two tables, for the cases with 2-club constraints and k-plex constraints, respectively.

The results show that on average the best solution is produced by CPLEX 1 minute run. Values, obtained by CPLEX runs with 26 and 27 seconds limits are by 1.2% and 2.2% greater for 2-club and for k-plex respectively. In most cases solutions obtained by two runs were equal. Therefore, CPLEX obtains solution close to optimal in about less than

Table 2: CPLEX Results: Problem with k-plex Constraints

Case		PSG CAR		PSG TANK		CPLEX 27 sec		CPLEX 1 min	
n	m	value	time	value	time	value	time	value	time
10	4	106.52	25.1	102.94	22.7	85.34	27.1	85.34	60.1
10	5	79.98	23.7	83.48	22.0	72.40	27.1	72.40	60.1
10	6	74.22	25.0	78.25	26.0	63.22	27.0	63.22	60.1
10	7	65.71	27.0	62.27	26.9	56.73	27.0	56.73	60.0
10	8	55.70	25.8	50.91	24.3	50.26	27.0	50.26	60.0
11	4	117.61	23.7	101.28	26.8	87.57	27.0	87.57	60.1
11	5	89.16	24.7	92.77	22.9	75.97	27.1	75.97	60.0
11	6	77.27	24.3	84.08	26.2	68.25	27.1	67.03	60.0
11	7	70.4	26.0	67.24	23.5	58.61	27.1	58.61	60.0
11	8	67.98	25.9	64.84	23.4	53.80	27.1	53.15	60.0
12	4	134.62	31.1	128.60	26.1	109.89	27.1	102.52	60.1
12	5	100.80	24.4	103.70	23.0	80.11	27.1	80.11	60.1
12	6	83.21	24.3	87.56	25.5	70.54	27.1	70.54	60.1
12	7	69.91	25.8	74.99	26.2	63.26	27.1	63.26	60.1
12	8	70.03	27.9	69.28	22.7	56.75	27.1	56.73	60.1
13	4	134.39	32.2	121.72	28.5	103.87	27.1	103.87	60.1
13	5	97.24	25.4	103.89	23.7	84.63	27.1	84.57	60.1
13	6	90.51	25.7	89.40	29.5	77.94	27.1	77.94	60.1
13	7	78.15	26.2	77.34	28.7	68.52	27.1	68.52	60.1
13	8	77.49	26.7	72.61	24.3	63.36	27.1	59.69	60.1
14	4	134.01	30.3	140.12	32.1	119.17	27.1	112.18	60.1
14	5	114.72	26.9	113.61	27.6	90.00	27.1	89.34	60.1
14	6	97.83	27.4	96.71	29.1	78.37	27.1	78.37	60.1
14	7	86.09	26.4	87.00	31.4	70.46	27.1	70.24	60.1
14	8	77.78	27.2	75.75	26.0	62.48	27.0	62.48	60.1
15	4	153.18	34.4	189.41	32.3	136.89	27.1	120.81	60.1
15	5	123.61	28.6	123.84	27.0	110.94	27.1	98.15	60.2
15	6	97.53	27.2	101.72	31.4	83.11	27.0	82.59	60.1
15	7	93.21	27.3	86.48	30.1	74.43	27.1	74.43	60.0
15	8	80.29	28.9	75.70	25.7	67.08	27.1	67.37	60.1
			avg		avg		avg		avg
			26.9		26.5		27.1		60.1

30 seconds. PSG TANK solution value is greater than CPLEX 1 minute solution value by 15.8% and 22.4% for 2-club and for k-plex respectively. PSG CAR performs slightly better than another solver, providing the solution values greater than CPLEX 1 minute solution values by 15.0% and 22.0%.

In addition to deterministic setup, we have run the aforementioned optimization problems under uncertainty on several stochastic problem instances with the number of sensors $m = 6$, the number of locations $n = 12$, the CVaR-type objective with $\alpha = 0.9$, $T = 10$ time intervals, for different numbers of scenarios: $S = 10, 20, 50, 100$. As before, the wireless connectivity network edge density was $\rho = 0.8$. The time limit was set to 5 minutes. PSG solvers in most cases provided solution before the time limit was reached. However, the quality of solution was worse then provided by CPLEX by 15% on average.

Table 3: CPLEX and PSG Results: Stochastic Setup

		CPLEX			PSG CAR		PSG TANK	
type	S	value	time	gap	value	time	value	time
k-plex	10	85.14	300.1	27.0%	96.78	25.3	96.81	29.4
k-plex	20	89.36	300.2	34.6%	95.10	35.3	96.10	37.1
k-plex	50	92.27	300.5	44.4%	110.06	154.7	97.40	273.1
k-plex	100	93.81	301.7	49.7%	104.57	300.6	115.69	300.6
2-club	10	86.92	300.1	30.5%	100.13	229.9	100.13	300.1
2-club	20	84.92	300.1	32.0%	97.55	79.4	97.55	300.1
2-club	50	89.75	300.5	44.3%	104.30	300.1	103.42	300.1
2-club	100	95.87	301.8	50.7%	116.63	300.2	116.23	300.2

6 Conclusions

We have defined and extended a class of dynamic sensor scheduling problems by introducing explicit robust connectivity requirements, specifically, k-club and k-plex constraints, taking into account wireless connectivity requirements for sensors at every time moment.

We have also presented computational results for moderate-size instances in both deterministic and stochastic problem setups. Since the size of the stochastic version of the problem is S times larger than for the deterministic version (where S is the number of implied penalty scenarios), solving these stochastic problems is clearly challenging from the computational perspective.

Problem formulations can be further extended by adding movement network and corresponding constraints as introduced in Boyko et al. [2], thus modeling the map of possible sensor movements.

We have also compared the performance of the two optimization software packages: ILOG CPLEX 11.2 and AORDA Portfolio Safeguard 64 bit. Although CPLEX generally outperforms PSG on the considered problem instances in terms of the quality of solutions by 15–22%, PSG may have its own advantages in practical settings due to a more user-friendly approach and more options to formulate a problem.

The classes of problems considered in this paper are primarily motivated by military applications; however, the developed formulations are general enough so that they can be applied in a variety of settings.

References

[1] B. Balasundaram, S. Butenko and I. Hicks, Clique relaxations in social network analysis: the maximum k-plex problem, *Oper. Res.* in press, 2010.

[2] N. Boyko, T. Turko, V. Boginski, D.E. Jeffcoat, S. Uryasev, G. Zrazhevsky and P.M. Pardalos, Robust multi-sensor scheduling for multi-site surveillance, *J. Comb. Optim.* in press, 2010.

[3] R.D. Luce. Connectivity and generalized cliques in sociometric group structure. *Psychometrika* 15 (1950) 169–190.

[4] R.J. Mokken, Cliques, clubs and clans, *Qual. Quant.* 13 (1979) 161–173.

[5] R.T. Rockafellar and S. Uryasev, Optimization of conditional value-at-risk, *J. Risk* 2 (2000) 21–42.

[6] R.T. Rockafellar and S. Uryasev, Conditional value-at-risk for general loss distributions, *J. Banking Finance* 26 (2002) 1443–1471.

[7] S. Sarykalin, G. Serraino and S. Uryasev, VaR vs CVaR in risk management and optimization, in *INFORMS Tutorials in Operations Research*, 2008, pp. 270–294.

[8] S.B. Seidman and B.L. Foster, A graph theoretic generalization of the clique concept, *J. Math. Sociol.* 6 (1978) 139–154.

[9] M. Yavuz and D.E. Jeffcoat. Single sensor scheduling for multi-site surveillance. Technical report, Air Force Research Laboratory, 2007.

Manuscript received
revised
accepted for publication

Konstantin Kalinchenko
Department of Industrial and Systems Engineering, University of Florida
303 Weil Hall, Gainesville, FL 32611
E-mail address: kalinchenko@ufl.edu

Alexander Veremyev
Department of Industrial and Systems Engineering, University of Florida
303 Weil Hall, Gainesville, FL 32611
E-mail address: averemyev@ufl.edu

Vladimir Boginski
Department of Industrial and Systems Engineering, University of Florida
303 Weil Hall, Gainesville, FL 32611;
University of Florida Research and Engineering Education Facility (UF-REEF)
1350 N Poquito Road, Shalimar, FL 32579 E-mail address: boginski@reef.ufl.edu

David E. Jeffcoat
AFRL/RW 101 West Eglin Boulevard, Bldg. 13, AFRL Room 341 Eglin Air Force Base, FL 32542-6810
E-mail address: david.jeffcoat@eglin.af.mil

Stan Uryasev
Department of Industrial and Systems Engineering, University of Florida
303 Weil Hall, Gainesville, FL 32611
E-mail address: uryasev@ufl.edu