J Comb Optim
DOI 10.1007/s10878-009-9271-4

Robust multi-sensor scheduling for multi-site surveillance

Nikita Boyko · Timofey Turko · Vladimir Boginski · David E. Jeffcoat · Stanislav Uryasev · Grigoriy Zrazhevsky · Panos M. Pardalos

Abstract This paper presents mathematical programming techniques for solving a class of multi-sensor scheduling problems. Robust optimization problems are formulated for both deterministic and stochastic cases using linear 0–1 programming techniques. Equivalent formulations are developed in terms of cardinality constraints. We conducted numerical case studies and analyzed the performance of optimization solvers on the considered problem instances.

Keywords Mathematical programming · Multi-sensor scheduling · Combinatorial optimization · Robust optimization · Risk measures

1 Introduction

The task of area surveillance is important in a variety of applications in both military and civilian settings. One of the main challenges that one needs to address in these problems is the fact that the number of locations (sites) that need to be visited to gather potentially valuable information is often much larger than the number of available surveillance devices (sensors) that are used for collecting information.

The research was supported by AFOSR grants # 07MN01COR and FA9550-08-1- 0190 and Air Force contract # F08635-03-D-0130.

N. Boyko (✉) · T. Turko · S. Uryasev · G. Zrazhevsky · P.M. Pardalos
University of Florida, 303 Weil Hall, Gainesville, FL 32611, USA
e-mail: nikita@ufl.edu

V. Boginski
University of Florida REEF, 1350 N Poquito Road, Shalimar, FL 32579, USA

D.E. Jeffcoat
Air Force Research Laboratory, 101 West Eglin Boulevard, Bldg. 13, Eglin Air Force Base, FL 32542, USA

Published online: 30 December 2009

Under these conditions, one needs to develop efficient schedules for all the available sensors (that can be installed, for instance, on Unmanned Air Vehicles) such that the amount of valuable information collected by the sensors is maximized. One can formulate this problem in terms of minimizing the information losses associated with 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 depend on how long a site has not been previously visited by a sensor. In particular, taking into account variable losses of information is critical in the case of strategically important sites that need to be monitored as closely as possible.

In addition, the parameters that quantify fixed and variable information losses are in many cases *uncertain* by nature. In previous related works in this area, the uncertainties in these parameters were not explicitly taken into account (see, e.g., Yavuz and Jeffcoat 2007); however, the development of efficient techniques to minimize or restrict the information losses under uncertainty is the task of crucial importance. This paper proposes mathematical programming formulations that allow quantifying and restricting the risks of worst-case losses associated with uncertain parameters.

The mathematical programming formulations are first developed for the deterministic case, and the natural extensions of these formulations to the stochastic case (with uncertain information loss parameters) is made by utilizing quantitative risk measures allowing to control the conservativeness of the optimal strategy. In particular, the statistical concept referred to as Conditional Value-at-Risk (CVaR) is used in the proposed problem formulations under uncertainty. Using these techniques allows one to efficiently incorporate uncertainties in the considered optimization problems, as well as provides the means of balancing between the optimality and the robustness of the solutions. Equivalent reformulations and extensions of the considered problems are also presented.

We have also conducted numerical case studies to test the performance of the suggested algorithms. Since the considered problems are NP-hard and involve uncertain parameters, near-optimal solutions were found for large problem instances. It turned out that solving cardinality-based reformulations of the considered problems provided good quality solutions in a reasonable time.

2 Deterministic approach

This section introduces a general mathematical framework for multi-sensor scheduling problems. Initially, we utilize the concepts introduced in Yavuz and Jeffcoat (2007) that was developed for a deterministic case of a single-sensor scheduling problem. We then generalize and extend these formulations to the more realistic cases of multi-sensor scheduling problems, including the setups in uncertain environments. Assume that there are m sensors and n sites that need to be observed at every discrete time moment $t = 1, \ldots, T$. We 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, we have breaches in surveillance that

can cause losses of potentially valuable information. Our goal is to build a strategy that optimizes a potential loss that is associated with not observing certain sites at some time moments.

We then introduce binary decision variables

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

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

We 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

$$a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}). \tag{2}$$

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

$$\max_{i,t}\{a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t})\}. \tag{3}$$

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

$$\sum_{i=1}^{n} x_{i,t} \le m, \quad \forall t = 1, \ldots, T. \tag{4}$$

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:

$$0 \le y_{i,t} - y_{i,t-1} \le t x_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{5}$$

$$t x_{i,t} \le y_{i,t} \le t, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T. \tag{6}$$

Note that the above constraints automatically ensure that the feasible values of $y_{i,t}$ are integer. It is easy to verify by considering possible values of binary variables $x_{i,t}$. Therefore, in the following mathematical programming problems, we can set the variables $y_{i,t} \in \mathbb{R}$, and the inclusion of these constraints will make these variables integer in any feasible solution. This enables us to decrease the number of integer (binary) variables in the considered problems.

Further, using the notation $C = \max_{i,t}\{a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t})\}$ and standard linearization techniques, we can formulate the multi-sensor scheduling optimization problem in the deterministic setup as the following mixed integer linear program:

$$\min C \tag{7}$$

$$\text{s.t.} \quad C \geq a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}), \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{8}$$

$$\sum_{i=1}^{n} x_{i,t} \leq m, \quad \forall t = 1, \ldots, T, \tag{9}$$

$$0 \leq y_{i,t} - y_{i,t-1} \leq t x_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{10}$$

$$t x_{i,t} \leq y_{i,t} \leq t, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{11}$$

$$y_{i,0} = 0, \quad \forall i = 1, \ldots, n, \tag{12}$$

$$x_{i,t} \in \{0, 1\}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{13}$$

$$y_{i,t} \in \mathbb{R}, \quad \forall i = 1, \ldots, n, \ \forall t = 0, \ldots, T. \tag{14}$$

3 Deterministic setup with percentile objective

For every site i and every time moment t, we can calculate the penalty associated with the last time a sensor visited this site (see formula (2)). Let us pick $(1 - \alpha)\%$ of *worst cases* among these $n \times T$ penalty values. Then instead of minimizing the *maximum* loss, we can minimize the *average* loss taken over these $(1 - \alpha)\%$ percent of worst-case penalty values. Despite the fact that this formulation is deterministic, we will demonstrate that it is equivalent to computing $(1 - \alpha)$ Conditional Value-at Risk (CVaR) for a set of random outcomes having equal probabilities $p_{i,t} = \frac{1}{nT}$.

To facilitate the further discussion, let us give the definition of Conditional Value-at-Risk (CVaR) (Rockafellar and Uryasev 2002; Sarykalin et al. 2008). 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 $(1 - \alpha)$ (in many applications the value of $(1 - \alpha)$ is set rather high, e.g. 95%), the α-VaR is the lowest amount ζ such that with probability $(1 - \alpha)$, the loss will not exceed ζ, whereas for continuous distributions the α-CVaR is the conditional expectation of losses *above* that amount ζ. 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 Fig. 1).

Formally, α-CVaR can be expressed as

$$\text{CVaR}_\alpha(\mathbf{x}) = (1 - \alpha)^{-1} \int_{L(\mathbf{x},\mathbf{y}) \geq \zeta_\alpha(\mathbf{x})} L(\mathbf{x}, \mathbf{y})\, p(\mathbf{y})\, d\mathbf{y}, \tag{15}$$

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

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 Rockafellar and Uryasev (2002), Sarykalin et al. (2008).

It has been shown by Rockafellar and Uryasev (2000), Uryasev (2000) that minimizing $\phi_\alpha(\mathbf{x})$ is equivalent to minimizing the function

$$F_\alpha(\mathbf{x}, \zeta) \ = \ \zeta + (1 - \alpha)^{-1} \int_{y \in \mathbb{R}^m} [L(\mathbf{x}, \mathbf{y}) - \zeta]^+ \, p(\mathbf{y})\, d\mathbf{y}, \tag{16}$$

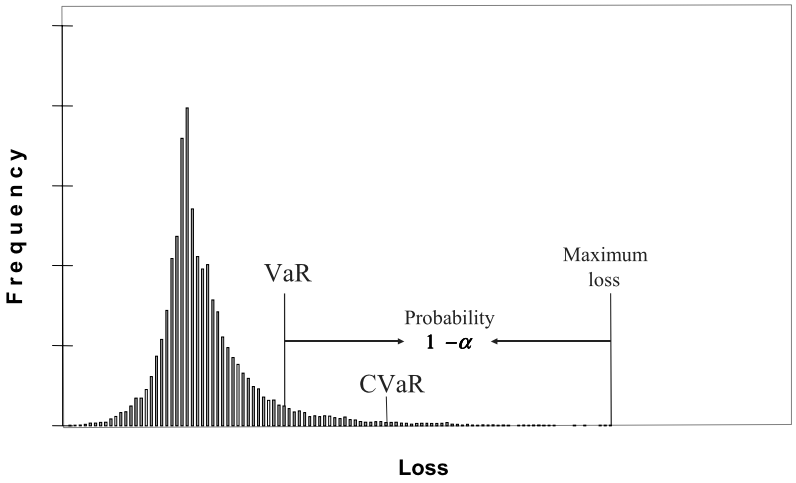

Fig. 1 Graphical representation of VaR and CVaR

where $[t]^+ = t$ when $t > 0$ but $[t]^+ = 0$ when $t \leq 0$, and the variable ζ corresponds to the VaR value, as introduced above.

Thus we can generalize our formulation and write the objective function for our problem as

$$\min_{x,y} \mathrm{CVaR}_\alpha \left[L(x, y, i, t) \right], \tag{17}$$

where

$$L(x, y, i, t) = a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}). \tag{18}$$

The particular extreme case when $\alpha \to 1$ corresponds to minimizing maximum penalty over all t-s and i-s. This case corresponds to the problem (7)–(14) formulated in Sect. 2. The other extreme case $\alpha = 0$ gives average taken over all time points and sites (if we assume uniform distribution). In the latter case we care about average loss and there is a high chance of not paying enough attention to particular bad outcomes.

Generally, α indicates the *level of conservatism* the decision maker is willing to accept. The closer α approaches to 1, the narrower the range of worst cases becomes in the corresponding optimization problem.

Using the general approach outlined in formulas (15)–(16), our problem is now formulated as follows:

$$\min_{x,y,\zeta} \zeta + \frac{1}{1-\alpha} \sum_{i,t} p_{i,t} \left[a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}) - \zeta \right]^+ \tag{19}$$

$$\text{s.t.} \quad \text{constraints (9)–(14)},$$

$$\zeta \in \mathbb{R}, \tag{20}$$

where the values of $p_{i,t}$ can all be set equal to $1/nT$ as indicated in the beginning of this section.

Furthermore, this problem formulation can be easily transformed to a linear mixed integer problem by introducing a set of artificial variables $z_{i,t}$ that will lead to the following problem with a set of $n \times T$ additional constraints.

$$\min_{x,y,\zeta} \zeta + \frac{1}{nT(1-\alpha)} \sum_{i,t} y_{i,t} \tag{21}$$

$$\text{s.t.} \quad a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t}) - \eta \le y_{i,t}, \tag{22}$$

$$y_{i,t} \ge 0, \tag{23}$$

$$\text{constraints (9)–(14)},$$

$$\zeta \in \mathbb{R}. \tag{24}$$

4 Problem setup under uncertainty

To extend the proposed problem formulations to a more realistic setup, where the values of the penalty parameters are uncertain, we propose a new CVaR-based formulation of multi-sensor scheduling problems.

In this setup, assume that the fixed and variable penalty values $\mathbf{a_i}$ and $\mathbf{b_{i,t}}$ are random variables with given joint distributions. Further, we can consider a set of penalty values $(a_i^s, b_{i,t}^s)$, $s = 1, \ldots, S$ corresponding to S discrete samples (or *scenarios*) as an approximation of the joint distribution. Then for each $s = 1, \ldots, S$ the loss function can be written as:

$$L(x, y, i, t, s) = a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t}). \tag{25}$$

It is appropriate to consider $(1-\alpha)\%$ of worst-case penalties over all indices i, t, s. We can then chose a measure of loss as an average over these $(1-\alpha)\%$ worst cases and minimize the average. Namely we minimize

$$\text{CVaR}_\alpha \left[L(x, y, i, t, s) \right]. \tag{26}$$

Using the approach described in the previous section, we obtain the following robust optimization problem that explicitly takes into account the uncertain penalty parameters:

$$\min_{x,y,\zeta} \zeta + \frac{1}{nTS(1-\alpha)} \sum_{i,t,s} \left[\left(a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t}) \right) - \zeta \right]^+ \tag{27}$$

$$\text{s.t.} \quad \text{constraints (9)–(14), (20)}.$$

As mentioned above, this formulation can be linearized by introducing extra variables and constraints, and the linear mixed integer formulation is provided below.

$$\min_{x,y,\zeta} \zeta + \frac{1}{nTS(1-\alpha)} \sum_{i,t,s} y_{i,t,s} \tag{28}$$

$$\text{s.t.} \quad a_i^s(1 - x_{i,t}) + b_{i,t}^s(t - y_{i,t}) - \zeta \le y_{i,t,s}, \tag{29}$$

$$y_{i,t,s} \ge 0, \tag{30}$$

constraints (9)–(14), (20).

5 Equivalent formulations of the considered problems using cardinality constraints

In this section, we show that the developed linear mixed integer programming problems can be equivalently reformulated as problems with cardinality constraints. As it will be discussed later, solving these equivalent reformulations can provide better computational speed and performance in finding near-optimal solutions of the considered problems. It should be noted that due to the high dimensionality and complexity of these problems, it is often impossible to find exact optimal solutions in a reasonable time; however, it is often useful in practice to utilize heuristic techniques that can find near-optimal solutions fast.

There exist heuristics (Lobo et al. 2007) as well as software packages (AOrDA PSG 2008) which can solve optimization problems formulated in terms of cardinality constraints. Cardinality function simply equals to the number of non-zero components of its vector argument. More formally, for $x = (x_1, \ldots, x_n)^T \in \mathbb{R}^n$

$$\text{card}(x) = \sum_{i=1}^{n} I(x_i),$$

where I is an indicator function defined as:

$$I(z) = \begin{cases} 1, & z \neq 0; \\ 0, & \text{otherwise.} \end{cases} \tag{31}$$

This section presents a problem formulated in terms of cardinality function. This new problem is equivalent to the initial formulation (7)–(14) that can be written as

Problem (I):

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

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{i,t} \le m, \quad \forall t = 1, \ldots, T, \tag{32}$$

$$0 \le y_{i,t} - y_{i,t-1} \le t x_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{33}$$

$$t x_{i,t} \le y_{i,t} \le t, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{34}$$

$$y_{i,0} = 0, \quad \forall i = 1, \ldots, n, \tag{35}$$

$$x_{i,t} \in \{0, 1\}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{36}$$

$$y_{i,t} \in \mathbb{R}, \quad \forall i = 1, \ldots, n, \ \forall t = 0, \ldots, T. \tag{37}$$

The new problem can be formulated as:
Problem (II):

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

$$\text{s.t.} \quad \text{card}(\tilde{x}_t) \le m, \quad \forall t = 1, \ldots, T, \tag{38}$$

$$0 \le y_{i,t} - y_{i,t-1} \le t x_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{39}$$

$$y_{i,t} \le t, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{40}$$

$$y_{i,0} = 0, \quad \forall i = 1, \ldots, n, \tag{41}$$

$$0 \le x_{i,t} \le 1, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{42}$$

$$y_{i,t} \in \mathbb{R}, \quad \forall i = 1, \ldots, n, \ \forall t = 0, \ldots, T, \tag{43}$$

where $\tilde{x}_t = (x_{1,t}, \ldots, x_{N,t})^T$.

The following theorem provides the relation between the two problems.

Theorem 1 *The set of optimal solutions of problem* (I) *belongs to the set of optimal solutions of problem* (II). *Moreover, if a point* (x^{II}, y^{II}) *is an optimal solution for* (II) *then the optimal solution of* (I) (x^I, y^I) *can be constructed as*

$$x^I_{i,t} = \lceil x^{II}_{i,t} \rceil,$$
$$y^I_{i,t} = \max_{\tau \le t} \{\tau x^I_{i,\tau}\}.$$

In order to prove the theorem we will use an auxiliary formulation.
Problem (III):

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

$$\text{s.t.} \quad \text{card}(\tilde{x}_t) \le m, \quad \forall t = 1, \ldots, T,$$

$$0 \le y_{i,t} - y_{i,t-1} \le t x_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T,$$

$$y_{i,t} \le t, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T,$$

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

$$0 \le x_{i,t} \le 1, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T,$$

$$y_{i,t} \in \mathbb{R}, \quad \forall i = 1, \ldots, n, \ \forall t = 0, \ldots, T.$$

Denote by $z^{(I)}, z^{(II)}$ and $z^{(III)}$ the optimal objective values of problems (I)–(III) consequently.

Lemma 1 *For every optimal solution of* (III) *there exists a solution that will be both feasible and optimal in* (I) *and* (III) (*i.e. formulations* (I) *and* (III) *are equivalent in this sense*).

Proof Equations (33)–(34) enforce that

$$y_{i,t} = \max_{\tau \le t}\{\tau x_{i,\tau}\}.$$

If there exists an optimal solution for (III) such that $x_{i,t} = 1$ but $y_{i,t} < t$ for some i, t (i.e. it is not feasible in (I)) then you can build a new solution that has the same values of $x^*_{i,t} = x_{i,t}\ \forall i, t$ and $y^*_{i,t} = \max_{\tau \le t}\{\tau x_{i,\tau}\}$. This solution will be feasible for both formulations (I) and (III).

Moreover, $\forall i, t\ y_{i,t} <= y^*_{i,t}$, i.e. the objective value will not increase and, therefore, this solution will be also optimal for (III). Obviously this solution will be feasible and optimal for formulation (I) (since $z^{(III)} \le z^{(I)}$).

Thus, for every optimal solution of (III) there exists a solution that will be both feasible and optimal in (I) and (III) that means that these two formulations are equivalent. □

Proof of Theorem 1 Let us consider some optimal solution of (II) $x^0_{i,t},\ y^0_{i,t}$ and build a new solution $x^*_{i,t} = I\{x^0_{i,t} > 0\};\ y^*_{i,t} = y^0_{i,t}$. This solution will still be feasible and optimal for (II) since it will not increase the objective value.

Obviously, this solution will be feasible and optimal for (III) ($z^{(II)} \le z^{(III)}$).

According to Lemma 1 for every optimal solution of (III) there exists a solution $x^{**}_{i,t} = x^*_{i,t}$, $y^{**}_{i,t} = \max_{\tau \le t}\{\tau x^*_{i,\tau}\}$ that will be both feasible and optimal for (I) and (III).

This solution will be integer and feasible for (II). Since $\forall i, t\ y^*_{i,t} \le y^{**}_{i,t}$ then the objective value will not increase and, therefore, this solution will also be optimal for (II).

Thus, there always exists the optimal integer solution for (II) that will be also optimal for (I). □

In order to prevent the solution of (II) from being non-integral, we add a penalty to the objective of (II):

Problem (IV):

$$\min \max_{i,t}\{a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t})\} + \lambda \cdot \left(m \cdot T - \sum_{i,t} x_{i,t}\right)$$

s.t. constraints (38)–(43),

where $\lambda > 0$.

Corollary 1 *Problems* (I) *and* (IV) *have the same set of optimal values of* $\{x_{i,t}\}$.

Similar theorems can be proven for the other formulations, namely percentile deterministic and stochastic setups. For deterministic case problem (Ia), which is equivalent to (19) is related to reformulated in terms of cardinality problem (IIa):

Problem (Ia):

$$\min \mathrm{CVaR}_\alpha\{a_i(1 - x_{i,t}) + b_{i,t}(t - y_{i,t})\}$$

s.t. constraints (32)–(37).

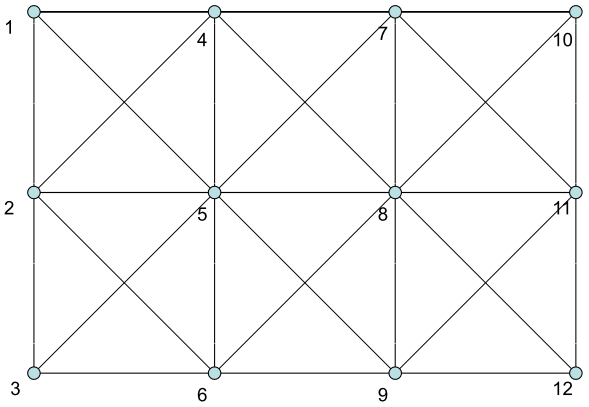

Fig. 2 Example of a possible network. Two nodes are connected by an arc if a sensor can move from one node to another in consequent time periods

Problem (IIa):

$$\begin{aligned} &\min \mathrm{CVaR}_{\alpha}\{a_i(1-x_{i,t})+b_{i,t}(t-y_{i,t})\} \\ &\text{s.t.} \quad \text{constraints (38)–(43).} \end{aligned}$$

Theorem 2 *The set of optimal solutions of problem* (Ia) *belongs to the set of optimal solutions of problem* (IIa). *Moreover if a point* (x^{II}, y^{II}) *is an optimal solution for* (IIa) *then the optimal solution of* (Ia) (x^{I}, y^{I}) *can be constructed as*

$$\begin{aligned} x^{I}_{i,t} &= \lceil x^{II}_{i,t} \rceil, \\ y^{I}_{i,t} &= \max_{\tau \le t}\{\tau x^{I}_{i,\tau}\}. \end{aligned}$$

For the stochastic case problem (Ib), which is equivalent to (27) is related to the reformulated in terms of cardinality problem (IIb):

Problem (Ib):

$$\begin{aligned} &\min \mathrm{CVaR}_{\alpha}\{a^{s}_{i}(1-x_{i,t})+b^{s}_{i,t}(t-y_{i,t})\} \\ &\text{s.t.} \quad \text{constraints (32)–(37).} \end{aligned}$$

Problem (IIb):

$$\begin{aligned} &\min \mathrm{CVaR}_{\alpha}\{a^{s}_{i}(1-x_{i,t})+b^{s}_{i,t}(t-y_{i,t})\} \\ &\text{s.t.} \quad \text{constraints (38)–(43).} \end{aligned}$$

Theorem 3 *The set of optimal solutions of problem* (Ib) *belongs to the set of optimal solutions of problem* (IIb). *Moreover if a point* (x^{II}, y^{II}) *is an optimal solution for* (IIb) *then the optimal solution of* (Ib) (x^{I}, y^{I}) *can be constructed as*

$$\begin{aligned} x^{I}_{i,t} &= \lceil x^{II}_{i,t} \rceil, \\ y^{I}_{i,t} &= \max_{\tau \le t}\{\tau x^{I}_{i,\tau}\}. \end{aligned}$$

6 Sensor scheduling in network-based settings

First, let us discuss a special case of one sensor ($m = 1$) to give an idea of this modeling approach. In case when surveillance requires sensors to physically move from one site to another their transition abilities are limited with distance or other constraint (for example a mountain can be a natural obstacle for UAV to move between sites). In this case each site can be modeled as a node of a network $G = (V, E)$. Whenever there is no arc between two nodes i and j we add the inequality

$$x_{i,t} + x_{j,t+1} \leq 1, \tag{44}$$

that prohibits the infeasible move $i \to j$ in consequent time periods t and $t + 1$. Formulations (7)–(14), (19) and (27) can be slightly modified to obtain corresponding formulations for one sensor ($m = 1$). If we demand the sensor to start and come back to a depot located at the certain site we can optionally set the initial ($i_0 \in \{1, \ldots, n\}$) and final ($i_T \in \{1, \ldots, n\}$) locations of the sensor.

Thus for the special case when $m = 1$ problem (7)–(14) can be formulated on the network:

$$\min \max_{i,t} \{a_i(1 - z_{i,t}) + b_{i,t}(t - y_{i,t})\} \tag{45}$$

s.t. constraints (9)–(14) $(m = 1)$,

$$x_{i,t} + x_{j,t+1} \leq 1 \quad \text{whenever } (i, j) \notin E,\ \forall t = 1, \ldots T,\ i, j = 1, \ldots, n, \tag{46}$$

$$x_{i_0,1} = 1, \quad \text{where } i_0 \in \{1, \ldots, n\} \text{ is the initial location of the sensor,} \tag{47}$$

$$x_{i_T,1} = 1, \quad \text{where } i_T \in \{1, \ldots, n\} \text{ is the final location of the sensor.} \tag{48}$$

In this formulation constraints (46) prohibit infeasible moves between not connected nodes. Equations (47) and (48) set initial and final destination for the sensor. The other formulations, namely deterministic (19) and stochastic (27), can be easily adapted for network case ($m = 1$) in the same way by adding network constraints (46)–(48) to the existing sets of constraints.

This approach, however, may not be easily extended on cases of two or more sensors. If we simply add (46)–(48) to existing non-network formulations we can arrive at the situation which prevents feasible moves when two or more sensors are involved. Figure 3 provides a counterexample. Let two sensors at time moment $t = 1$ are located at nodes 1 and 3. Though they could move to nodes 2 and 4 respectively at the next time point $t = 2$ the constraint $x_{1,1} + x_{2,4} \leq 1$ would prohibit this move. To avoid such a situation we need to add one more index for decision variable x:

$$x_{i,t,k} = \begin{cases} 1, & \text{sensor } k \text{ is surveiling site } i \text{ at time } t; \\ 0, & \text{otherwise.} \end{cases} \tag{49}$$

We can ensure that every sensor is assigned to a site at every time period T with the constraint:

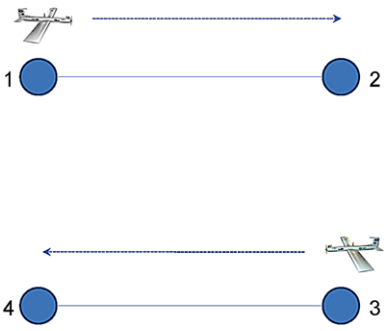

Fig. 3 Counterexample ($m = 2$): two sensors cannot perform simultaneous feasible move due to the constraint $x_{1,1} + x_{4,2} \leq 1$

$$\sum_{i=1}^{n} x_{i,t,k} = 1, \quad \forall k = 1, \ldots, m, \ \forall t = 1, \ldots, T. \tag{50}$$

Let us introduce $z_{i,t}$ indicating whether site i is observed at time t, namely

$$z_{i,t} = \begin{cases} 1, & \text{if any sensor is surveiling site } i \text{ at time } t; \\ 0, & \text{otherwise.} \end{cases} \tag{51}$$

Variables $z_{i,t}$ and $x_{i,t,k}$ can be related with the constraint

$$z_{i,t} \leq \sum_{i=1}^{n} x_{i,t,k} \leq m \cdot z_{i,t}, \tag{52}$$

which states that site i is being observed at time t ($z_{i,t} = 1$) if and only if at least one sensor is present at site i ($\sum_{i=1}^{n} x_{i,t,k} > 0$).

The loss function is written as

$$L(x, y, i, t) = a_i^s (1 - z_{i,t}) + b_{i,t}^s (t - y_{i,t}). \tag{53}$$

If we want to minimize maximum loss then using the formulated above constraints and based on the problem formulated for non-network setup we can reformulate the deterministic maximum loss minimization problem (7)–(14) on network as follows:

$$\min \max_{i,t} \{a_i (1 - z_{i,t}) + b_{i,t} (t - y_{i,t})\} \tag{54}$$

$$\text{s.t.} \quad \sum_{i=1}^{n} x_{i,t,k} = 1, \quad \forall k = 1, \ldots, m, \ \forall t = 1, \ldots, T, \tag{55}$$

$$z_{i,t} \leq \sum_{k=1}^{m} x_{i,t,k} \leq m \cdot z_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{56}$$

$$0 \leq y_{i,t} - y_{i,t-1} \leq t z_{i,t}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{57}$$

$$t z_{i,t} \leq y_{i,t} \leq t, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \tag{58}$$

$$y_{i,0} = 0, \quad \forall i = 1, \ldots, n, \tag{59}$$

$$x_{i,t,k} + x_{j,t+1,k} \leq 1,$$
$$\text{whenever } (i, j) \notin E,\ \forall t = 1, \ldots T,\ i, j = 1, \ldots, n,\ k = 1, \ldots, m, \tag{60}$$

$$x_{i_{0,k},1,k} = 1,$$
$$\text{where } i_{0,k} \in \{1, \ldots, n\} \text{ is the initial location of sensor } k, \tag{61}$$

Table 1 Performance results for deterministic model (7)–(14). n—number of sites; m—number of sensors. The number of discrete time steps is fixed: $T = 10$

		$n = 8$	$n = 9$	$n = 10$	$n = 11$	$n = 12$
$m = 1$	cplex value	320	330	330	332	332
	psg value	375	376	376	370	375
	%	14.7%	12.2%	12.2%	10.3%	11.5%
	time: cplex/psg	31.2/2.2	79.1/2.4	134.9/2.5	167.7/2.7	198.9/2.7
$m = 2$	cplex value	240	245	250	260	265
	psg value	305	310	304	310	310
	%	21.3%	21%	17.8%	16.1%	14.5%
	time: cplex/psg	38.4/2.2	142.7/2.3	928/2.5	2042.7/2.6	5898.9/2.8
$m = 3$	cplex value	206	210	215	217	224
	psg value	233	250	265	256	275
	%	11.6%	16%	18.9%	15.2%	18.5%
	time: cplex/psg	30.7/2.3	39.3/2.4	81.1/2.5	1003.6/2.7	9317.9/2.9
$m = 4$	cplex value	190	194	196	200	200
	psg value	215	217	242	237	242
	%	11.6%	10.6%	19%	15.6%	17.4%
	time: cplex/psg	6/2.3	76.5/2.5	64.6/2.6	231.4/2.7	589.3/2.8
$m = 5$	cplex value	183	185	188	190	190
	psg value	196	202	215	217	220
	%	6.6%	8.4%	12.6%	12.4%	13.6%
	time: cplex/psg	1.4/2.3	2.2/2.5	38/2.6	51.9/2.7	68/3
$m = 6$	cplex value	165	170	170	183	185
	psg value	185	185	197	195	200
	%	10.8%	8.1%	13.7%	6.2%	7.5%
	time: cplex/psg	1.1/2.4	1.5/2.5	2.9/2.7	29.1/2.8	123/3
$m = 7$	cplex value	155	160	163	168	170
	psg value	160	171	185	190	188
	%	3.1%	6.4%	11.9%	11.6%	9.6%
	time: cplex/psg	0.3/2.3	0.9/2.5	1.8/2.8	113.2/2.9	25.2/3

$$x_{i_{T,k},T,k} = 1,$$

where $i_{T,k} \in \{1, \ldots, n\}$ is the final location of sensor k, (62)

$$x_{i,t,k} \in \{0, 1\}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T, \ \forall k = 1, \ldots, m, \tag{63}$$

$$y_{i,t} \in \mathbb{R}, \quad \forall i = 1, \ldots, n, \ \forall t = 0, \ldots, T, \tag{64}$$

$$z_{i,t} \in \{0, 1\}, \quad \forall i = 1, \ldots, n, \ \forall t = 1, \ldots, T. \tag{65}$$

Table 2 Performance results for CVaR type deterministic model (19). n—number of sites; m—number of sensors. The number of discrete time steps is fixed: $T = 10$. CVaR confidence level $\alpha = 0.9$

		$n = 8$	$n = 9$	$n = 10$	$n = 11$	$n = 12$
$m = 1$	cplex value	307.1	318.4	317	318.7	318
	psg value	360.1	357.9	356.8	357.6	353.7
	%	14.7%	11%	11.2%	10.9%	10.1%
	time: cplex/psg	45.5/2.8	99.3/3	74/2.9	62.8/3.1	130.9/3
$m = 2$	cplex value	231.9	241.1	245.2	–	–
	psg value	297	299	297.5	293.5	291.6
	%	21.9%	19.4%	17.6%	–	–
	time: cplex/psg	1072.8/3	10924.6/3.1	16212.1/3.1	–/3.2	–/3.2
$m = 3$	cplex value	198.6	205.3	–	–	–
	psg value	223.6	237.3	245	255.4	260.9
	%	11.2%	13.5%	–	–	–
	time: cplex/psg	1910.7/2.8	45540.4/2.8	–/2.9	–/3	–/3.4
$m = 4$	cplex value	187.9	190.4	–	–	–
	psg value	211.4	207.2	230	218.2	221.8
	%	11.1%	8.1%	–	–	–
	time: cplex/psg	7989.2/3.2	23852.7/3	–/3.3	–/3.4	–/3.2
$m = 5$	cplex value	173.3	179.3	–	–	–
	psg value	193.4	192.6	197.8	207.7	210
	%	10.4%	6.9%	–	–	–
	time: cplex/psg	7672.6/2.9	25593/2.8	–/3.3	–/3.3	–/3.7
$m = 6$	cplex value	161.1	165.3	–	–	–
	psg value	179.5	182.8	185.2	195.2	190.9
	%	10.2%	9.5%	–	–	–
	time: cplex/psg	2250.3/2.8	36618.1/3.3	–/3.4	–/2.9	–/3.4
$m = 7$	cplex value	142.4	–	–	–	–
	psg value	150.9	166.6	173.4	179	183.8
	%	5.6%	–	–	–	–
	time: cplex/psg	6640.9/2.9	65122.8/3.3	–/3	–/3	–/3.7

Table 3 Performance results for CVaR type stochastic model (27). n—number of sites; m—number of sensors. The number of discrete time steps is fixed: $T = 10$, number of scenarios $S = 100$. CVaR confidence level $\alpha = 0.9$

		$n = 8$	$n = 9$	$n = 10$	$n = 11$	$n = 12$
$m = 1$	cplex value	–	–	–	–	–
	psg value	394	398.7	390.1	395.6	394.9
	%	–	–	–	–	–
	time: cplex/psg	–/25.9	–/28	–/35.8	–/40.2	–/47.7
$m = 2$	cplex value	–	–	–	–	–
	psg value	297.9	304	321.7	324.4	318.6
	%	–	–	–	–	–
	time: cplex/psg	–/32.7	–/38.6	–/51.4	–/67.8	–/76
$m = 3$	cplex value	–	–	–	–	–
	psg value	260.4	260.1	262.4	264.4	269
	%	–	–	–	–	–
	time: cplex/psg	–/38.4	–/45.4	–/56.3	–/71.4	–/84.6
$m = 4$	cplex value	–	–	–	–	–
	psg value	229.4	231.6	242.9	236.3	250.8
	%	–	–	–	–	–
	time: cplex/psg	–/47.2	–/60.2	–/72.5	–/84.6	–/98.4
$m = 5$	cplex value	–	–	–	–	–
	psg value	209.1	212.4	220.8	224.6	230.1
	%	–	–	–	–	–
	time: cplex/psg	–/56.1	–/72.3	–/83.7	–/96.6	–/119.5
$m = 6$	cplex value	–	–	–	–	–
	psg value	193.2	199.9	209.2	210.3	218.7
	%	–	–	–	–	–
	time: cplex/psg	–/51.8	–/67.3	–/86.1	–/112.9	–/121.4
$m = 7$	cplex value	–	–	–	–	–
	psg value	164.4	186.3	187.6	198.6	204.4
	%	–	–	–	–	–
	time: cplex/psg	–/45	–/66.5	–/89.8	–/111.3	–/133.5

Formulations for CVaR stochastic and deterministic cases as well as the linearized formulation can be obtained the same way as in non network case.

7 Computational experiments

The computational experiments were performed on the test problems using two commercial optimization software solvers: ILOG CPLEX (2008) and AOrDA PSG

Table 4 Comparing PSG and CPLEX performance for obtaining approximate solution of CVaR type stochastic problem (27) ($n = 12$ sites and $T = 10$ time periods)

	PSG value	PSG time (s)	CPLEX time (s)
$m = 1$	389.2	103	32
$m = 2$	318.532	110	185
$m = 3$	276.332	133	230
$m = 4$	246.648	136	247
$m = 5$	229.234	208	210
$m = 6$	218.166	158	330
$m = 7$	204.1	201	260
$m = 8$	193.199	191	220
$m = 9$	180.976	155	250
$m = 10$	164.746	191	280
$m = 11$	146.607	180	200

(2008). The performance of the solvers is compared in Tables 1–3 (each table corresponds to one of the three problem formulations). It can be observed that CPLEX finds exact solutions, however, it takes too much time for large instances, especially for the problems under uncertainty. PSG allows sacrificing quality for time, i.e. the obtained solutions for cardinality formulations are not globally optimal, but the computational time is negligibly small. The numerical experiments shows that the local solutions differ from global in 10–20% for most cases.

Table 4 compares performance of CPLEX and PSG in finding approximate solutions for stochastic case ($n = 12$ sites and $T = 10$ time periods). We stopped CPLEX when it found the objective as small as the PSG objective value (and recorded the computation time). It appears that PSG outperforms CPLEX for problems with large number of stochastic scenarios while they have similar performance for small size problems. Therefore, based on the size of the problem and user requirements, one can determine the appropriate equivalent problem formulation and the optimization solver that can be used to find an optimal or a near-optimal solution.

We performed experiments on network formulation using the network provided on Fig. 2. Table 5 provides CPU times in seconds for obtaining exact solution for the deterministic network case (54)–(65) in ILOG CPLEX. Computation was performed for number of sites from 6 to 12 and 10 discrete time steps.

8 Conclusion

The paper develops a mathematical programming techniques for solving a class of multi-sensor scheduling problems. Three robust optimization problems have been formulated: two for deterministic and one for stochastic case. The obtained 0–1 problems have also been reformulated in terms of cardinality functions.

Numerical experiments are conducted using two commercial solvers ILOG CPLEX and AOrDa PSG. CPLEX gives exact solutions for small problems. Both solvers give an approximate solution in reasonable time for large problems. However, for large stochastic problems with many scenarios, solving the reformulated problems

Table 5 ILOG CPLEX CPU time (s) for network deterministic model (54)–(65). n—number of sites; m—number of sensors. The number of discrete time steps is $T = 10$

	$n=6$	$n=7$	$n=8$	$n=9$	$n=10$	$n=11$	$n=12$
$m=1$	0.31	0.77	0.76	2.16	0.90	0.50	1.21
$m=2$	3.96	3.96	8.00	15.85	238.42	174.93	315.76
$m=3$	0.93	21.48	22.20	9.17	340.86	350.85	2037.73
$m=4$	0.20	0.31	0.74	34.26	14.45	926.64	436.80
$m=5$	0.32	1.79	2.62	31.40	91.38	62.20	3359.69
$m=6$		0.58	1.79	4.36	69.76	19.05	238.59
$m=7$			2.12	4.08	22.87	19.57	47.20
$m=8$				0.79	6.29	7.14	38.69
$m=9$					1.17	3.19	6.81
$m=10$						0.79	2.15
$m=11$							1.94

with cardinality constraints using PSG provided good quality approximate solutions faster than CPLEX. Therefore, this approach can be beneficial in the settings of time-critical systems where computational speed of finding good approximate solutions is the crucial factor.

References

American optimal decision, portfolio safeguard (2008) http://www.aorda.com

Ilog cplex (2008) http://www.ilog.com/products/cplex/

Lobo MS, Fazel M, Boyd S (2007) Portfolio optimization with linear and fixed transaction costs. Ann Oper Res 152(1):376–394

Rockafellar RT, Uryasev SP (2000) Optimization of conditional value-at-risk. J Risk 2:21–42

Rockafellar RT, Uryasev SP (2002) Conditional value-at-risk for general loss distributions. J Bank Financ 26:1443–1471

Sarykalin S, Serraino G, Uryasev S (2008) Var vs cvar in risk management and optimization. In: INFORMS tutorial

Uryasev S (2000) Conditional value-at-risk: optimization algorithms and applications. Financ Eng News 14:1–5

Yavuz M, Jeffcoat DE (2007) Single sensor scheduling for multi-site surveillance. Technical report, Air Force Research Laboratory