Preprints, 17th IFAC Workshop on
Control Applications of Optimization
Yekaterinburg, Russia, October 15-19, 2018

Comparison of Multi-Sensor Task Assignment Methods: Linear Matrix Inequalities vs. Brute Force ⋆

Victoria Erofeeva, * **Oleg Granichin,** ** **Anna Leonova,** ***

* *Saint Petersburg State University (Faculty of Mathematics and Mechanics), Institute of Problems in Mechanical Engineering, Russian Academy of Sciences, St. Petersburg, Russia*
e-mail: victoria@grenka.net
** *Saint Petersburg State University (Faculty of Mathematics and Mechanics, and Research Laboratory for Analysis and Modeling of Social Processes), Institute of Problems in Mechanical Engineering, Russian Academy of Sciences, St. Petersburg, Russia*
e-mail: o.granichin@spbu.ru
*** *Saint Petersburg State University (Faculty of Mathematics and Mechanics), St. Petersburg, Russia*
e-mail: annia.leonova@gmail.com

Abstract: Due to significant advancements in embedded systems, sensor devices, and wireless communication technology, sensor networks have been attracting widespread attention in areas such as target tracking, monitoring, and surveillance. Technological advancements made it possible to deploy a large number of inexpensive but technically advanced sensors to cover wide areas. However, when a tracking system has to track a large number of targets, the computation and communication loads arise. In this paper, we compare two task assignment methods that might be used in the multiple target tracking problem. The first one is the brute force method and the second one is based on linear matrix inequalities. We provide performance and load testing results for these methods.

Keywords: sensor network, task assignments, multiple target tracking, linear matrix inequalities, ellipsoidal approximation, brute force

1. INTRODUCTION

Sensor networks are widely used in various fields. Especially they are suitable for such applications as monitoring, target localization and tracking, space situational awareness (see Giannakis et al. (2013); Jia et al. (2016); Thite and Mishra (2016)). Significant advancements in embedded systems, sensor devices, and wireless communication technology make it possible to use large-scale networks, which provide more advantages over a single node or a small network. In particular, each sensor mostly receives incomplete observations (measurements) because of the noisiness of an environment and inaccuracy inherent to the sensor devices. Thanks to the use of multiple sensors one might obtain a more accurate estimation of the measured value through the information fusion. In other words, multi-sensor networks can be used to reduce uncertainties.

The use of the tracking systems, which are comprised of multiple inexpensive and small sensors, brings new challenges due to resource limitations of the network. Each sensor has limited sensing coverage and it might be ineffective for a target to be tracked by all available sensors or by a fixed subset of sensors through the entire tracking process. Moreover, sensors deployed in a large area of interest may not contribute much to the tracking quality since sensors might be far away from the targets. Nevertheless, they consume their own and network resources by collecting the data and communicating with the other nodes. These issues gave rise to the sensor selection or task assignment problem, in which the best subset of the available sensors needs to be chosen according to given performance constraints.

In general, the sensor selection problem is expressed as follows (Chepuri and Leus (2015)):

$$\arg\min_{\mathbf{q}\in\{0,1\}^N} h(\mathbf{Q}(\mathbf{q})) \;\; s.t. \;\; \mathbf{1}_N^{\mathrm{T}}\mathbf{q} = K, \tag{1}$$

where $\mathbf{q}$ is a selection vector of length N, $h(\mathbf{Q}(\mathbf{q}))$ is a scalar cost function related to the error covariance matrix $\mathbf{Q}$. The error covariance matrix is optimized to select the best subset of K sensors out of N available sensors. The problem (1) is combinatorial and one needs to do $\binom{K}{N}$ searches to find the solution. In the multi-target case this problem becomes even worse because we need to find a selection vector for every target.

There are various algorithms available in the literature to solve the problem of sensor selection or resource allocation. In Masazade et al. (2012) the authors propose a target

⋆ This work was supported by Russian Science Foundation (project 16-19-00057).

tracking algorithm based on extended Kalman filtering, in which the selection process is performed by designing a sparse gain matrix. Heuristic Yu and Prasanna (2005), stochastic heuristic Jin et al. (2012) and meta-heuristic intelligent optimization algorithms Yang et al. (2014) have also been considered to solve this problem. Exhaustive search Kaplan (2006), greedy search Kalandros (2002), auction algorithm Chen et al. (2006), are some other algorithms that are applied to resource allocation. Considered algorithms tend to be computationally expensive. To address the complexity issues, sparse convex optimization approaches are used in Joshi and Boyd (2009).

In Botts et al. (2016) the authors consider a stochastic multi-agent and multi-target surveillance problem and apply to it a cyclic stochastic optimization algorithm. Recently, researchers have been actively developing approaches based on randomization (see Tempo et al. (2012); Granichin et al. (2015)). Event-based tracking technologies are also widely considered due to throughput constraints and difficulty in analyzing a large amount of data Batmani et al. (2017).

The existing works mainly address the problem of choosing K sensors from a set of available sensors in order to obtain the best tracking accuracy. However, in large-scale networks it is important to find a trade-off between accuracy and resource utilization.

1.1 Contribution

In this work we compare two task assignment methods that might be used in the multiple target tracking problem. The first one is the brute force method and the second one is based on linear matrix inequalities (LMI). We described the idea behind the second method in Erofeeva et al. (2018), where the selection problem was formulated as the design of a sparse resource allocation matrix $\mathbf{G}_t$ to choose the most informative sensors. The entries of $\mathbf{G}_t$ were designed to be as sparse as possible such that the tracking error and the amount of used sensors were minimized.

Mathematically, it means one should minimize the sum of non-zero entries of the vector $\mathbf{g}$ defined by the l_0-(quasi) norm: $\|\mathbf{g}\|_0 = \sum_{j=1}^{N} |\text{sign } g_j|$. Since the l_0-(quasi) norm optimization is NP-hard and nonconvex, one should use the convex surrogate, i.e. the l_1-norm heuristic, that gives the best approximation of the sparse solution (Barabanov and Granichin (1984); Polyak et al. (2014)):

$$\|\mathbf{g}\|_1 = \sum_{j=1}^{N} |g_j|.$$

In essence, we sought a sparse matrix $\mathbf{G}_t$ consisting of vectors $\mathbf{g}$ (i.e., vector with many zeros and a few non-zero entries) that minimizes the quality functional presented in the next section.

1.2 Outline and Notations

The remainder of the paper is organized as follows. In Section 2, we introduce the problem of multiple target tracking by a sensor network, consisting of identical devices. Section 3 provides the techniques of finding an intersection of the ellipsoids corresponding to the sensors measurements. In Section 4, we describe task assignment methods. In Section 5, we provide comparison results of the methods considered in the previous section. Finally, Section 6 concludes the paper.

The notation used in this paper can be described as follows. Upper and lower bold face letters are used for matrices and column vectors, respectively. $\mathrm{E}\{\cdot\}$ is the expectation operation. $\mathbf{I}_k$ is a $k \times k$ identity matrix with ones on the main diagonal and zeros elsewhere. $\preccurlyeq$ is a non-strict inequality for symmetric matrices that is understood in the sense of inequalities for quadratic forms. $(\cdot)^{\mathrm{T}}$ denotes transposition. $|\mathcal{U}|$ denotes the cardinality of the set $\mathcal{U}$. $\|\cdot\|$ is the Euclidean norm. $\mathrm{tr}\{\cdot\}$ is the matrix trace operator. $\det\{\cdot\}$ is the matrix determinant.

2. PROBLEM STATEMENT

Consider a distributed network of n sensors, randomly located in an area of interest. Let $N = \{1, 2, \dots, n\}$ be the set of sensors and $\mathbf{s}_t^j \in \mathbb{R}^k$ be the state of the sensor j. In the line of sight of the sensors are moving m targets. Our goal is to assign sensors to the targets in such a way that we could accurately predict the movement trajectories of the targets and use as less sensors as possible.

Let $M = \{1, 2, \dots, m\}$ be the set of targets, $\{\mathbf{r}_t^i\}_{t=0,1,2,\dots}$, $\mathbf{r}_t^i \in \mathbb{R}^l$, $i \in M$ be the movement trajectory of the target i, whose state changes according to the following equation:

$$\mathbf{r}_{t+1}^i = f^i(\mathbf{r}_t^i) + \mathbf{w}_t^i, \tag{2}$$

where $f^i(\cdot)$ is a state-transition function, $\{\mathbf{w}_t^i\}$ is the white Gaussian noise with zero mathematical expectation and covariance matrix $\mathbf{R}_w^i$: $\mathrm{E}\mathbf{w}_t^i = 0$, $\mathrm{E}\mathbf{w}_t^i(\mathbf{w}_t^i)^{\mathrm{T}} = \mathbf{R}_w^i \preccurlyeq \sigma_w^2 \mathbf{I}_l$.

The sensors estimate the state $\mathbf{r}_t^i$ of the object i based on measurements received in accordance with the following observation model

$$\mathbf{z}_t^{i,j} = \varphi(\mathbf{s}_t^j, \mathbf{r}_t^i) + \boldsymbol{\varepsilon}_t^{i,j}, \tag{3}$$

where $\mathbf{z}_t^{i,j} \in \mathbb{R}^q$ is a measurement of the state of the object i available to the sensor j at time instant t, $\varphi(\cdot,\cdot) : \mathbb{R}^k \times \mathbb{R}^l \to \mathbb{R}^q$ is an observation function, which depends on the current state of the object i and sensor j, $\{\boldsymbol{\varepsilon}_t^{i,j}\}$ is the additive external noise with zero mean $\mathrm{E}\boldsymbol{\varepsilon}_t^{i,j} = 0$ and the error covariance matrix $\mathrm{E}\boldsymbol{\varepsilon}_t^{i,j}(\boldsymbol{\varepsilon}_t^{i,j})^{\mathrm{T}} = \Sigma_t^{i,j}$.

We assume that for any $i \in M$, $j \in N$ and independent centered $\boldsymbol{\varepsilon}_t^{i,j}$ with the error covariance matrix $\Sigma_t^{i,j}$ there exists an inverse function $\varphi^{-1}(\mathbf{s}_t^j, \cdot) : \mathbb{R}^q \to \mathbb{R}^l$:

$$\varphi^{-1}(\mathbf{s}_t^j, \varphi(\mathbf{s}_t^j, \mathbf{r}_t^i) + \boldsymbol{\varepsilon}_t^{i,j}) = \mathbf{r}_t^i + \boldsymbol{\xi}_t^{i,j}, \tag{4}$$

where $\boldsymbol{\xi}_t^{i,j}$ is an independent component with zero mean $E\boldsymbol{\xi}_t^{i,j} = 0$, the error covariance matrix $\mathrm{E}\boldsymbol{\xi}_t^{i,j}(\boldsymbol{\xi}_t^{i,j})^{\mathrm{T}} = \Xi_t^{i,j}$ and the bounded fourth central moment $\mathrm{E}\|\boldsymbol{\xi}_t^{i,j}\|^4 \le M_4$.

Note that the measurements received by a single sensor might not be enough to reconstruct the state of an object. In this case a sequence of measurements collected by the sensor itself or through other sensors is usually utilized. Nevertheless, due to availability of technologically advanced equipment it is possible to satisfy the assumption (4). If there is no such single-valued inverse function, but there exists a subspace corresponding to $\mathbf{z}_t^{i,j} - \mathbf{U}_t^{i,j}\mathbf{r}_t^i = 0$, where $\mathbf{U}_t^{i,j}$ is a matrix mapping the state into the measurement, then we are able to estimate the true state on this subspace.

2.1 Confidence Region

Let a confidence region be represented as an ellipsoid around the point $\boldsymbol{\eta}_t^{i,j} = \varphi^{-1}(\mathbf{s}_t^j, \mathbf{z}_t^{i,j})$. The confidence region with the user-defined significance level p would include the point representing the "true" value $\mathbf{r}_t^i$ with the probability $1 - p$. We define this ellipsoid as follows:

$$\mathcal{E}_t^{i,j} = \{\mathbf{r}_t^i \;:\; (\mathbf{r}_t^i - \boldsymbol{\eta}_t^{i,j})^{\mathrm{T}}(\Xi_t^{i,j})^{-1}(\mathbf{r}_t^i - \boldsymbol{\eta}_t^{i,j}) \leq \chi_{p,d}^2\}, \quad (5)$$

where $\chi_{p,d}^2$ is the p-value matching to the χ^2 distribution for d degrees of freedom.

Example 1. Let one specify an ellipsoid around the point $\boldsymbol{\eta}_t^{i,j} \in \mathbb{R}^2$, i.e. $d = 2$. Let this ellipsoid include the value $\mathbf{r}_t^i$ with the 95% probability. In accordance with the table of χ^2 values vs p-values, one should set the p-value to 0.05.

For each target i we have a set of points $\boldsymbol{\eta}_t^i = \{\boldsymbol{\eta}_t^{i,1}, \ldots, \boldsymbol{\eta}_t^{i,n}\}$ and corresponding to them ellipsoids $\mathcal{E}_t^i = \{\mathcal{E}_t^{i,1}, \ldots, \mathcal{E}_t^{i,n}\}$. We assume that the "true" value $\mathbf{r}_t^i$ belongs to the intersection of the ellipsoids contained in $\mathcal{E}_t^i$ and we would like to find this intersection region.

2.2 Quality Function

We denote by $\boldsymbol{\theta}_t = col(\mathbf{r}_t^1, \ldots, \mathbf{r}_t^m)$ the joint vector of all target states. Let $\widehat{\mathbf{r}}_t^i$ be an estimate of the state of target i at time instant t and $\widehat{\boldsymbol{\theta}}_t = col(\widehat{\mathbf{r}}_t^1, \ldots, \widehat{\mathbf{r}}_t^m)$ be the joint vector of all estimates. Let $\mathcal{U}_t^i$ be the intersection region of the ellipsoids contained in $\mathcal{E}_t^i$ and $\widehat{\mathcal{U}}_t = \{\mathcal{U}_t^1, \ldots, \mathcal{U}_t^m\}$ be a set of the intersection regions.

In general, the main goal of the tracking process can be achieved by minimizing the following quality function:

$$\bar{F}_t(\widehat{\boldsymbol{\theta}}_t) = \frac{1}{2}\sum_{i\in M}\|\mathbf{r}_t^i - \widehat{\mathbf{r}}_t^i\|^2 \to \min_{\widehat{\boldsymbol{\theta}}_t}. \quad (6)$$

Equivalently, the problem (6) may be represented as follows:

$$\Phi_t(\widehat{\mathcal{U}}_t) = \sum_{i\in M}\mathrm{vol}(\mathcal{U}_t^i) \to \min_{\widehat{\mathcal{U}}_t}. \quad (7)$$

However, it might be hard to find the volume of the intersection region itself if the value n is large enough. In this case the intersection region becomes too complex. Alternatively, we may approximate this region by an ellipsoid (Matviychuk (2018)). Let $\hat{\mathcal{E}}_t = \{\hat{\mathcal{E}}_t^1, \ldots, \hat{\mathcal{E}}_t^m\}$ be the set of ellipsoids that approximate the intersections of ellipsoids contained in $\{\mathcal{E}_t^i\}_{i\in M}$. The problem (7) becomes as follows:

$$\Phi_t(\hat{\mathcal{E}}_t) = \sum_{i\in M}\mathrm{vol}(\hat{\mathcal{E}}_t^i) \to \min_{\hat{\mathcal{E}}_t}, \quad (8)$$

where $\mathrm{vol}(\cdot)$ is the volume.

In order to reduce the processing and communications loads, we are also going to minimize the number of selected sensors. We denote by $\mathbf{G}_t$ the resource allocation matrix that needs to be as sparse as possible. The entities $g_t^{i,j}$ of this matrix indicate whether the sensor j is assigned to the target i or not. Lastly, our quality function takes the following form:

$$\bar{\Phi}_t(\mathbf{G}_t) = \Phi_t(\hat{\mathcal{E}}_t) + \alpha\sum_{i\in M}\|\mathbf{G}_t^{(i,\cdot)}\|_1 \to \min_{\mathbf{G}_t}, \quad (9)$$

where α is the regularization coefficient, $\mathbf{G}_t^{(i,\cdot)}$ is the i-th row of the matrix $\mathbf{G}_t$.

3. AN INTERSECTION REGION OF ELLIPSOIDS

In this section we describe two methods of finding the volume of the intersection region. Monte Carlo method is used to find the volume of $\mathcal{U}_t^i, i \in M$. The linear matrix inequalities is adopted to approximate the region by an ellipsoid and then to calculate its volume.

3.1 Monte Carlo

Researchers started to widely use randomized approaches since the appearance of the Monte Carlo method of statistical simulation. This method was proposed by Metropolis and Ulam (see Metropolis and Ulam (1949)) during their work on the Manhattan Project at Los Alamos with von Neumann and Teller. The Monte Carlo method offers a simple scheme to estimate the mean value (expectation) of a function based on samples of its values for some random arguments (randomization).

More precisely, if the following formula is difficult to integrate analytically

$$F = \int \ldots \int_{\mathcal{D}} f(v)dv, \mathcal{D} \subset R^d,$$

then the Monte Carlo method is easier to use (Granichin et al. (2015)).

Let the set $\mathcal{D}$ be a parallelepiped with the volume $\mathrm{vol}(\mathcal{D})$, and function f is bounded: $0 \leq f(v) \leq f_{max}$. The geometric sense of the integral is a volume below the graph of function f.

The Monte Carlo method for this problem is as follows:

(1) Fix a positive integer T.
(2) Choose uniformly T independent identically distributed samples $z_1, z_2, \ldots z_T$ from the parallelepiped $\mathcal{D} \times [0, f_{max}] \subset R^{d+1}$ T.
(3) Among the samples $z_1, z_2, \ldots z_T$ count the number S of those $z_i, i \in 1..T$, whose last component does not exceed the value of the function f in the corresponding point defined by the first d coordinates.
(4) Compute the estimate

$$\hat{F} = \frac{S}{T}\mathrm{vol}(\mathcal{D})f_{max}$$

In the implementation of the method the square was a boundary area. The length of the side of the square was taken equal to the length of the smallest semi-major axis among all the ellipsoids participating in the intersection. The number of shots was taken equal to 25 for 1×1 conventional units (in terms of our system it is 1 km^2).

3.2 Linear Matrix Inequalities

Let the fusion center receive a set of points $\boldsymbol{\eta}_t^i = \{\boldsymbol{\eta}_t^{i,1}, \ldots, \boldsymbol{\eta}_t^{i,n}\}$ of the i-th target at time instant t. In Boyd et al. (1994) there are several methods that approximate the intersection region of ellipsoids. We are going to use outer approximation to find an ellipsoid $\hat{\mathcal{E}}_t^i$ such that

$$\hat{\mathcal{E}}_t^i \supseteq \bigcap_{j=1}^{n} \mathcal{E}_t^{i,j}. \tag{10}$$

For this purpose we apply the S-procedure, which could be used to obtain a linear matrix inequality (LMI) that is sufficient for (10) to hold. Before the S-procedure application, we should convert the ellipsoid (5) into the following form:

$$\mathcal{E}_t^{i,j} = \{\mathbf{x} \mid H^{i,j}(\mathbf{x}) \le 0\},$$
$$H^{i,j}(\mathbf{x}) = \mathbf{x}^{\mathrm{T}}\mathbf{A}_t^{i,j}\mathbf{x} + 2\mathbf{x}^{\mathrm{T}}\mathbf{b}_t^{i,j} + \mathbf{c}_t^{i,j}, \tag{11}$$

where $\mathbf{A}_t^{i,j} = (\Xi_t^{i,j})^{-1}$, $\mathbf{b}_t^{i,j} = -(\Xi_t^{i,j})^{-1}\boldsymbol{\eta}_t^{i,j}$, $\mathbf{c}_t^{i,j} = (\boldsymbol{\eta}_t^{i,j})^{\mathrm{T}}(\Xi_t^{i,j})^{-1}\boldsymbol{\eta}_t^{i,j} - 1$. We can form the required representation of $\mathcal{E}_t^{i,j}$ by scaling $\mathbf{A}_t^{i,j}, \mathbf{b}_t^{i,j}$, and $\mathbf{c}_t^{i,j}$ by positive factors depending on $\chi^2_{p,d}$ value.

Each of these forms of representing an ellipsoid can be afterwards converted into each other. In this paper we are considering a special case of an ellipsoid, which is referred to as an ellipse, i.e. when a 2-D plane is considered. Nevertheless, the approach we are going to use is suitable for ellipsoids in higher dimensions as well.

From the S-procedure the following condition could be obtained: there exist positive scalars $\tau^{i,1}, \ldots, \tau^{i,n}$ such that

$$\begin{bmatrix} \hat{\mathbf{A}}^i & \hat{\mathbf{b}}^i \\ (\hat{\mathbf{b}}^i)^{\mathrm{T}} & (\hat{\mathbf{b}}^i)^{\mathrm{T}}(\hat{\mathbf{A}}^i)^{-1}\hat{\mathbf{b}}^i - 1 \end{bmatrix} - \sum_{j=1}^{n} \tau^{i,j} \begin{bmatrix} \mathbf{A}^{i,j} & \mathbf{b}^{i,j} \\ (\mathbf{b}^{i,j})^{\mathrm{T}} & \mathbf{c}^{i,j} \end{bmatrix} \preccurlyeq 0, \tag{12}$$

which can be written as the LMI (in variables $\hat{\mathbf{A}}^i$, $\hat{\mathbf{b}}^i$, $\hat{\mathbf{c}}^i = (\hat{\mathbf{b}}^i)^{\mathrm{T}}(\hat{\mathbf{A}}^i)^{-1}\hat{\mathbf{b}}^i - 1$, and $\tau^{i,1}, \ldots, \tau^{i,n}$). Finally, we will be able to find the ellipsoid $\hat{\mathcal{E}}_t^i$, which has the smallest volume, by solving the following convex problem:

$$\text{minimize} \quad \log\det(\hat{\mathbf{A}}^i)^{-1}$$
$$s.\,t. \quad \hat{\mathbf{A}}^i > 0, \quad \tau^{i,1} \ge 0, \ldots, \tau^{i,n} \ge 0, \tag{13}$$
$$\begin{bmatrix} \hat{\mathbf{A}}^i & \hat{\mathbf{b}}^i & 0 \\ (\hat{\mathbf{b}}^i)^{\mathrm{T}} & -1 & (\hat{\mathbf{b}}^i)^{\mathrm{T}} \\ 0 & \hat{\mathbf{b}}^i & -\hat{\mathbf{A}}^i \end{bmatrix} - \sum_{j=1}^{n} \tau^{i,j} \begin{bmatrix} \mathbf{A}^{i,j} & \mathbf{b}^{i,j} & 0 \\ (\mathbf{b}^{i,j})^{\mathrm{T}} & \mathbf{c}^{i,j} & 0 \\ 0 & 0 & 0 \end{bmatrix} \preccurlyeq 0.$$

The equation (13) was obtained from (12) using Schur complements (see Boyd et al. (1994)).

4. TASK ASSIGNMENT METHODS

In this section we consider brute force method and LMI based approach to solve the problem (9). Note that in the case of brute force method the first summand in (9) represented by (7).

4.1 Brute Force

The brute force algorithm belongs to the class of methods for finding solutions by searching through all possible options. All options of target distributions between sensors are looked over to analyze the effectiveness of such distributions. From a practical standpoint, the algorithm consumes a lot of time and resources, but provides the optimal solution. Subsequently it is used to compare the results with the method based on the LMI.

The values for all possible subsets of the sensor set are sequentially computed for each target to this brute force algorithm implementation. After that, the values of the sum of volumes and the value of the functional defined by formula (9) are calculated for all permutations of the obtained values in the previous stage.

4.2 LMI Based Approach

In order to solve problem (9) we need to modify (13) in such a way that the method takes into account the resource allocation matrix $\mathbf{G}_t$. We made a slight change in the problem (13), adding the new conditions as follows:

$$\text{minimize} \quad \delta$$
$$s.\,t. \quad \forall i \quad \hat{\mathbf{A}} > 0, \quad g^{i,1} \ge 0, \ldots, g^{i,n} \ge 0, \tag{14}$$
$$\begin{bmatrix} \hat{\mathbf{A}}^i & \hat{\mathbf{b}}^i & 0 \\ (\hat{\mathbf{b}}^i)^{\mathrm{T}} & -1 & (\hat{\mathbf{b}}^i)^{\mathrm{T}} \\ 0 & \hat{\mathbf{b}}^i & -\hat{\mathbf{A}}^i \end{bmatrix} - \sum_{j=1}^{n} g^{i,j} \begin{bmatrix} \mathbf{A}^{i,j} & \mathbf{b}^{i,j} & 0 \\ (\mathbf{b}^{i,j})^{\mathrm{T}} & \mathbf{c}^{i,j} & 0 \\ 0 & 0 & 0 \end{bmatrix} \preccurlyeq 0.$$
$$\sum_{i=1}^{m} \log\det(\hat{\mathbf{A}}^i)^{-1} + \alpha \sum_{i=1}^{m} \|\mathbf{G}_t^{(i,\cdot)}\|_1 \le \delta.$$

The problem (14) means that we would like to find an ellipsoid with the volume as small as possible while using as few sensors as possible. In that case, we agree to get an estimation with some quality loss, but instead we reduce computational and communication loads.

In real applications there may be some restrictions regarding the maximum number of targets that can be tracked by each sensor, i.e. $|G_t^j| \le g^j_{\max}$. The solution of (14) does not guarantee that this restriction will hold. It only minimizes the value of $|G_t^j|$. To deal with this issue one may use the tracking algorithm that holds this restriction, like the parameter estimation method presented in Granichin and Erofeeva (2018).

5. COMPARISON RESULTS

5.1 Observation Model

We consider a 2D-plane, in which the state of the stationary sensor j is $\mathbf{s}_t^j = [s_t^{j,1} \; s_t^{j,2}]^{\mathrm{T}}$. The state consists of position components at time instant t. The tracking system estimates the position of the target i, i.e. $\mathbf{r}_t^i = [r_t^{i,1} \; r_t^{i,2}]^{\mathrm{T}}$. Suppose the sensors are able to determine the angle and distance to the objects, then:

$$\varphi(\mathbf{s}_t^j, \mathbf{r}_t^i) = \begin{bmatrix} \psi(\mathbf{s}_t^j, \mathbf{r}_t^i) \\ \rho(\mathbf{s}_t^j, \mathbf{r}_t^i) \end{bmatrix} \in \mathbb{R}^2, \tag{15}$$

where

$$\psi(\mathbf{s}_t^j, \mathbf{r}_t^i) = \operatorname{arctg}\left[\frac{r_t^{i,1} - s_t^{j,1}}{r_t^{i,2} - s_t^{j,2}}\right] \tag{16}$$

is the angle to the object i,

$$\rho(\mathbf{s}_t^j, \mathbf{r}_t^i) = \sqrt{\left(r_t^{i,1} - s_t^{j,1}\right)^2 + \left(r_t^{i,2} - s_t^{j,2}\right)^2} \tag{17}$$

is the distance to the object i.

In this case, the inverse function $\varphi^{-1}(\mathbf{s}_t^j, \cdot)$ is as follows

$$\varphi^{-1}(\mathbf{s}_t^j, \mathbf{z}_t^{i,j}) = \mathbf{s}_t^j + \begin{bmatrix} z_t^{i,j,2} \sin z_t^{i,j,1} \\ z_t^{i,j,2} \cos z_t^{i,j,1} \end{bmatrix}, \tag{18}$$

where $z_t^{i,j,1}$ and $z_t^{i,j,2}$ are the first and second coordinates of the vector $\mathbf{z}_t^{i,j}$, respectively. If the covariance matrices $\boldsymbol{\varepsilon}_t^{i,j}$ are equal to $\Sigma_t^{i,j} = \begin{bmatrix} \sigma_\psi^2 & 0 \\ 0 & (z_t^{i,j,2}\sigma_\rho)^2 \end{bmatrix}$, then the covariance of $\boldsymbol{\xi}_t^{i,j}$ is

$$\Xi_t^{i,j} = R(z_t^{i,j,1}) \begin{bmatrix} (z_t^{i,j,2}\sigma_\psi)^2 & 0 \\ 0 & (z_t^{i,j,2}\sigma_\rho)^2 \end{bmatrix} R(z_t^{i,j,1})^{\mathrm{T}}, \tag{19}$$

where $R(\psi) = \begin{bmatrix} \sin\psi & -\cos\psi \\ \cos\psi & \sin\psi \end{bmatrix}$ is the rotation matrix through the angle ψ.

5.2 Experiment

Next, we consider one possible experiment. Four targets move uniformly and rectilinearly in a square area of interest with identical and constant velocities. The area is of size $300\times300\ km^2$ and velocities are equal to $2500\ km/h$.

In the area of interest we randomly locate five sensors. The noise in the measurements obtained by each sensor is set to the following values:

$$\Sigma_t^{i,j} = \begin{bmatrix} \sigma_\psi^2 & 0 \\ 0 & (z_t^{i,j,2}\sigma_\rho)^2 \end{bmatrix} = \begin{bmatrix} 0.3^2 & 0 \\ 0 & \left(\dfrac{z_t^{i,j,2}}{100}\right)^2 \end{bmatrix}$$

The targets begin their movement starting at the initial positions, which are defined as $\mathbf{r}_0^1 = [70, 210]$; $\mathbf{r}_0^2 = [100, 190]$; $\mathbf{r}_0^3 = [120, 180]$; $\mathbf{r}_0^4 = [250, 180]$. We assume that the sensor network is homogeneous, i.e. the characteristics of each sensor are the same. By characteristics we mean, for example, the field of view, which is assumed to be 360 degrees. The field of view also covers the whole considered area of interest. The duration of experiments is 200 iterations.

5.3 Performance Testing

An experiment was performed for $n = 5$ and $m = 4$. The areas of real intersections by the Monte Carlo method were calculated for the compared algorithms. The results of this experiment are shown in the table below.

Table 1. Experiment results

	Brute force	LMI
Total volume	163.6	217.7
Total volume / Confidence	249.4	152.6
	Tracking group	
Target 1	1, 2, 3, 4, 5	1, 2
Target 2	1, 3, 5	2, 5
Target 3	1, 2, 3, 4, 5	2, 5
Target 4	1, 2, 4, 5	2, 5

Comparison of the results from the table shows that the brute force method certainly gives the best answer. The sum of the intersection volumes in this case is equal to 163.6. Meanwhile, the LMI based method gives the worse result because the volume is equal to 217.7. It is important to note that the sets of tracking groups differ greatly in power. These sets consist of a maximum or almost maximum number of sensors for brute force, but two sensors are used for the LMI based method.

Considering this, the groups formed by LMI based method are more attractive in terms of communication load. The probability that the true value belongs to the intersection decreases when the ellipsoids intersect than in the original confidence ellipsoids. It seems that the volume of the area becomes smaller if a large number of ellipsoids are used. However, in fact, the area becomes less significant if we use confidence ellipsoids, that is, the level of confidence becomes lower. Therefore, it is advisable to compare the volume normalized to the probability that the true value belongs to the intersection. This value is significantly lower for the method based on LMI, which gives a significant win for this indicator.

5.4 Load Testing

We performed a series of experiments for the algorithms considered in the previous section. For each experiment, the mathematical expectation was calculated. To calculate the values, we used the formula:

$$E = \frac{1}{h} * \sum_{i=1}^{h} X_i,$$

where X_i is the time obtained through experiments, h is the number of experiments. The tests were conducted on the system:

Windows 10 Home
CPU: Intel Core i7-7700HQ 2.8 GHz
RAM: 8 GB
OS type: 64-bit

Below is a table that contains the test results.

Table 2. Load testing results

Series	Number of sensors	Number of targets	Math. expectation, sec	
			Brute force	LMI
1	3	2	0.7062	0.4418
2	5	4	7.4825	1.1645
3	8	8	119.704	3.0956
4	16	16	>1200	18.2321
5	39	39	—	439.398
6	40	40	—	—

The results of the comparison showed incommensurable values for the brute force and LMI based method with increasing number of sensors and targets. Brute force ceases to work in a reasonable time at values $n = 16$ and $m = 16$. Meantime, the LMI based method shows good result at values $n = 39$ and $m = 39$. However, the method based on LMI for $n = 40$ and $m = 40$ does not provide a final result. This happens due to the implementation of the algorithm and the use of auxiliary packages.

6. CONCLUSIONS AND FUTURE DIRECTIONS

The purpose of the paper is to compare the brute force method with the algorithm based on linear matrix inequalities. The first method gives an optimal solution in the sensor selection problem but requires a lot of computational resources if a large-scale network is considered. The second method provides us with a suboptimal result, which we are able to compute in a much lesser time interval even for more complex networks.

At the moment, we minimize the number of sensors assigned to each target along with the uncertainty regarding the "true" state of the targets. In this case, the results presented in Table 1 show that some sensors track all targets and might be overloaded. In the follow-up works, we are going to add the third summand in (9) in order to minimize the number of targets assigned to each sensor. The preliminary results of this work were published in Erofeeva (2018).

The algorithm based on linear matrix inequalities was implemented with the use of the library named CVX (see Grant and Boyd (2014)). As stated on the official website, the free version is not suitable for the development of large-scale methods. Because of that, we were not able to obtain the results for the cases, when there are more than 40 sensors and targets. Due to this problem, we are going to develop a new method to approximate an intersection of ellipsoids based on randomized algorithms of stochastic approximation (see Granichin (2002); Granichin et al. (2015)).

REFERENCES

Barabanov, A. and Granichin, O. (1984). Optimal controller for linear plants with bounded noise. *Automation and Remote Control*, 45(5), 39–46.

Batmani, Y., Davoodi, M., and Meskin, N. (2017). Event-triggered suboptimal tracking controller design for a class of nonlinear discrete-time systems. *IEEE Transactions on Industrial Electronics*, 64(10), 8079–8087.

Botts, C.H., Spall, J.C., and Newman, A.J. (2016). Multi-agent surveillance and tracking using cyclic stochastic gradient. In *American Control Conference (ACC), 2016*, 270–275. IEEE.

Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. (1994). *Linear matrix inequalities in system and control theory*. SIAM.

Chen, J., Zang, C., Liang, W., and Yu, H. (2006). Auction-based dynamic coalition for single target tracking in wireless sensor networks. In *Intelligent Control and Automation, 2006. WCICA 2006. The Sixth World Congress on*, volume 1, 94–98. IEEE.

Chepuri, S.P. and Leus, G. (2015). Sparsity-promoting sensor selection for non-linear measurement models. *IEEE Transactions on Signal Processing*, 63(3), 684–698.

Erofeeva, V. (2018). State estimation and task distribution optimization in sensor networks based on cyclic approach. *Stochastic Optimization in Informatics*, 14(1), 3–30.

Erofeeva, V., Granichin, O., and Granichina, O. (2018). Multi-sensor task assignment using linear matrix inequalities in the multiple target tracking problem. In *18th IFAC Symposium on System Identification, 2018*, 880–885.

Giannakis, G.B., Kekatos, V., Gatsis, N., Kim, S.J., Zhu, H., and Wollenberg, B.F. (2013). Monitoring and optimization for power grids: A signal processing perspective. *IEEE Signal Processing Magazine*, 30(5), 107–128.

Granichin, O. and Erofeeva, V. (2018). Cyclic stochastic approximation with disturbance on input in the parameter tracking problem based on a multiagent algorithm. *Automation and Remote Control*, 79(6), 996–1011.

Granichin, O., Volkovich, Z., and Toledano-Kitai, D. (2015). *Randomized algorithms in automatic control and data mining*. 251. Springer.

Granichin, O.N. (2002). Randomized algorithms for stochastic approximation under arbitrary disturbances. *Automation and Remote Control*, 63(2), 209–219.

Grant, M. and Boyd, S. (2014). CVX: Matlab software for disciplined convex programming, version 2.1. `http://cvxr.com/cvx`.

Jia, B., Pham, K.D., Blasch, E., Shen, D., Wang, Z., and Chen, G. (2016). Cooperative space object tracking using space-based optical sensors via consensus-based filters. *IEEE Transactions on Aerospace and Electronic Systems*, 52(4), 1908–1936.

Jin, Y., Jin, J., Gluhak, A., Moessner, K., and Palaniswami, M. (2012). An intelligent task allocation scheme for multihop wireless networks. *IEEE Transactions on Parallel and Distributed Systems*, 23(3), 444–451.

Joshi, S. and Boyd, S. (2009). Sensor selection via convex optimization. *IEEE Transactions on Signal Processing*, 57(2), 451–462.

Kalandros, M. (2002). Covariance control for multisensor systems. *IEEE Transactions on Aerospace and Electronic Systems*, 38(4), 1138–1157.

Kaplan, L.M. (2006). Global node selection for localization in a distributed sensor network. *IEEE Transactions on Aerospace and Electronic systems*, 42(1), 113–135.

Masazade, E., Fardad, M., and Varshney, P.K. (2012). Sparsity-promoting extended kalman filtering for target tracking in wireless sensor networks. *IEEE Signal Processing Letters*, 19(12), 845–848.

Matviychuk, O. (2018). State estimation for bilinear impulsive control systems under uncertainties. *Cybernetics and Physics*, 7(10), 35–40.

Metropolis, N. and Ulam, S. (1949). The monte carlo method. *Journal of the American statistical association*, 44(247), 335–341.

Polyak, B.T., Khlebnikov, M.V., and Shcherbakov, P.S. (2014). Sparse feedback in linear control systems. *Automation and Remote Control*, 75(12), 2099–2111.

Tempo, R., Calafiore, G., and Dabbene, F. (2012). *Randomized algorithms for analysis and control of uncertain systems: with applications*. Springer Science & Business Media.

Thite, A. and Mishra, A. (2016). Optimized multi-sensor multi-target tracking algorithm for air surveillance system. In *Advances in Electrical, Electronics, Information, Communication and Bio-Informatics (AEEICB), 2016 2nd International Conference on*, 637–642. IEEE.

Yang, J., Zhang, H., Ling, Y., Pan, C., and Sun, W. (2014). Task allocation for wireless sensor network using modified binary particle swarm optimization. *IEEE Sensors Journal*, 14(3), 882–892.

Yu, Y. and Prasanna, V.K. (2005). Energy-balanced task allocation for collaborative processing in wireless sensor networks. *Mobile Networks and Applications*, 10(1-2), 115–131.