Hierarchical Track Association and Fusion for a Networked Surveillance System

J. ARETA
University of Connecticut

Y. BAR-SHALOM
University of Connecticut

M. LEVEDAHL
Raytheon Co.

K. R. PATTIPATI
University of Connecticut

In this paper we present a benchmark problem for data association based on a real-world networked surveillance system, and compare the behavior of several multidimensional assignment (MDA) algorithms. The problem consists of a set of N_s observers which transmit track/event reports to a fusion center through a particular (real-world based) communication network among one of N_n networks. The network discards the observer's track identity (ID), replacing it by a network-generated ID and the observer ID, thus losing the information on the origin of the tracks sent by each observer. The solution approach developed in this paper consists of a hierarchical decomposition of the problem. This hierarchical approach first eliminates the redundancy introduced by the communication network by using an MDA algorithm per each observer present, and then using another MDA algorithm to choose which 'non-redundant' reports to fuse. This decomposition drastically reduces the dimensionality of the problem from $N_s \times N_n$ to N_s problems of dimension N_n and one of dimension N_s.

A comparison of two association criteria, *normalized distance squared* (NDS) and *likelihood ratio* (LR), is carried out. *It is shown that the LR yields significantly superior results.* Also the selection of certain parameters in the likelihood ratio is discussed. Finally, to evaluate their performance, three different MDA algorithms are used in this setup, Lagrangean Relaxation based MDA, Sequential m-best 2D and Linear Programming. A thorough comparison of these algorithms in terms of the quality of their solutions as well as their run times is done showing some pitfalls and advantages of each.

Manuscript received October 25, 2006; revised July 03, 2007; released for publication August 9, 2007.

Refereeing of this contribution was handled by P. Singer.

Authors' addresses: J. Areta, Y. Bar-Shalom and K. Pattipati, Dept. of Electrical and Computer Engineering, University of Connecticut, Storrs, CT 06269-2157, E-mail: (areta ybs krishna@engr.uconn.edu); M. Levedahl, Raytheon Co., McLean, VA 22101, E-mail: (mlevedahl @raytheon.com).

1. INTRODUCTION

Data association is a major component of surveillance systems where several sensors and/or several targets are present. The origin uncertainty of reports from each of the sensors makes it critical for the fusion center to operate properly, especially in scenarios with closely-spaced targets. In such scenarios there is no clear-cut evidence whether a new report belongs to a previous track or not. The work in this paper deals with the formulation and the solution of a particular multidimensional assignment (MDA) problem using three different association algorithms for performance comparison.

The problem presented here corresponds to a real world surveillance network system for missile launch events. In it, a set of sources provide "event" (track) estimates via a number of communication networks to a Fusion Center (FC) which has to perform data association prior to fusion. This network model provides a realistic setup—a benchmark problem—that allows for a proper evaluation of the algorithms proposed to solve it. This differs from previous work [20], where such algorithms have been tested on randomly generated costs, which do not necessarily show the real performance of the algorithms in practical settings. The track generation model is a simplified one, without process noise.

The problem involves a set of N_s sources (observers) that provide event estimates (reports/tracks) of an unknown number of launches from their own observations. These reports are transmitted via N_n communication networks to a Fusion Center that has to perform data association prior to fusion. The network used to transmit each report is randomly chosen every time a new report is ready to be sent. The parameters estimated by the observers (and to be fused at the FC) are the launch time, launch point coordinates and heading. Figure 1 shows a possible scenario where 4 observers report through 3 networks to the fusion center.

A particular feature of the network model is that the information needed to distinguish among reports from the same source transmitted through different networks is not available at the FC: the track identity (ID) assigned by the source is not passed on, only a track ID assigned by the network and the source ID accompany the track. This makes it necessary to detect track duplications among the messages with the same source ID that arrive on different networks. This duplication elimination is performed based on the association criterion between tracks from the same origin, according to the recently developed general Likelihood Ratio (LR) approach in [3]. Also the selection of certain parameters in the LR is discussed. Out of the tracks deemed to be from the same event from the same source, the one with the smallest uncertainty is kept.

The resulting data, rearranged into sensor lists, is then associated using the likelihood ratio criterion from [3] with any of the proposed multidimensional assignment methods. The tracks obtained after association

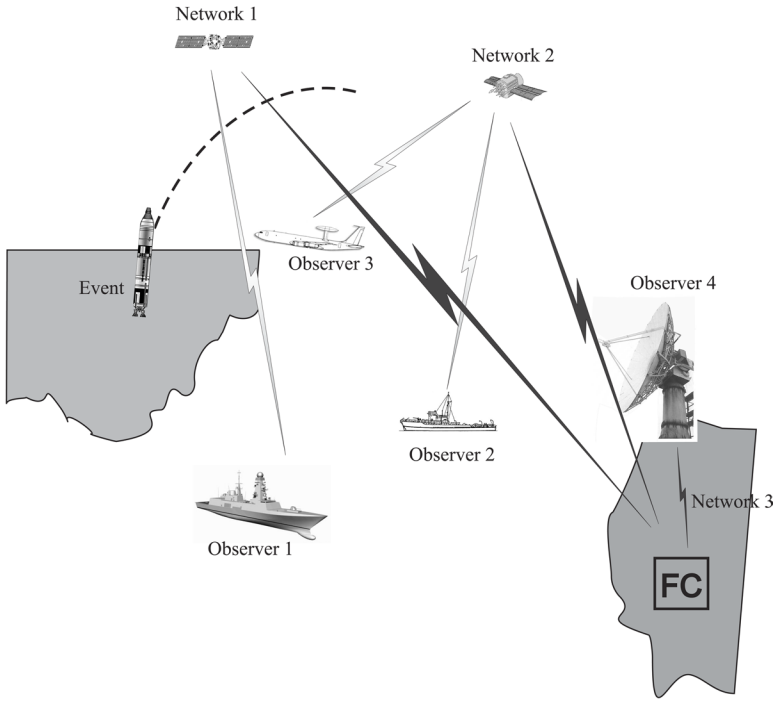


Fig. 1. Hypothetical scenario, where an event is reported by four observers to the FC using three communication networks.

are fused using a maximum likelihood (ML) approach as in [9]. An additional complication is that false reports can be also transmitted by the sources. Examples with several launches, sources and networks are presented to compare the performances of three assignment algorithms—the Lagrangean Relaxation based multidimensional (S-D) assignment [18], the "Sequential m-best 2D" assignment and the Linear Programming based assignment—on this realistic problem. The simpler Normalized Distance Squared (NDS) criterion is considered as well.

Section 2 describes the overall system with its layers and the information communicated between the layers, as well as a generic model for the local (source/observer) track/event estimates which are sent on the communication networks to the FC. Section 3 presents a hierarchical decomposition of the problem so that MDA algorithms can be applied in two stages. Section 4 presents two association criteria, the LR criterion, as well as the simpler normalized distance squared, also known as the Mahalanobis Distance or Chi-square criterion. Section 5 describes the three proposed methods to solve the MDA problem. Section 6 presents the results from the simulations and their analysis. Finally, Section 7 presents a discussion of the results and conclusions.

List of acronyms:

CC: Completely Correct
CI: Completely Incorrect
FC: Fusion Center
LaR: Lagrangean Relaxation
LR: Likelihood Ratio
LP: Linear Programming
MDA: Multidimensional (S-D) Assignment
ML: Maximum Likelihood
NDS: Normalized Distance Squared (Mahalanobis distance)
PC: Partially Correct
S-D: S-Dimensional Assignment
Sm2D: Sequential m-best 2D Assignment

2. PROBLEM FORMULATION

2.1. The Overall System

The system considered consists of the following layers:

1. Sources
2. Communication Networks
3. Fusion Center.

Each observer generates track reports and sends them to the FC through independent networks. These reports are based on individual observations made by the observers. Each event estimate consists of a vector and standard deviation (s.d.) for each component.[1] The reports received by the FC consist of

1. The event ID, t, which is *assigned by the network*. The source-assigned ID is not transmitted by the network to the FC, but replaced by a network-assigned

[1]The procedure easily generalizes to the case where one has full covariance matrices associated with the estimates, as long as they are available.

ID.[2] Nevertheless, the network-given ID continues to be used for all subsequent reports of the same event from the same observer.

2. The ID of the network, $n = 1,\ldots,N_n$.

3. The ID of the source, $s = 1,\ldots,N_s$.

4. The vector estimate and standard deviation for each component, as sent by the source.

The FC can receive reports on the same event from the same source via different networks—since the event ID is assigned by the network, it is not obvious upon reception at the FC which reports with different n and same s pertain to the same event/track. Also it is possible that reports with different n and different s can be on the same event. This makes it necessary to implement a data redundancy elimination stage prior to fusion, in order to eliminate duplicate reports on the same targets generated by the same observer.

2.2. The Sources

Each track report from a source is based upon all the measurements received up to the current time for that event. The measurements have uncorrelated errors and each report is the average of all the measurements received so reports on the same event have correlated errors.

It will be assumed that the measurement errors are zero mean and white with s.d. denoted as σ. While this value is not needed, it allows us to obtain the correlation coefficient of reports on the same event at the same source based on different (unknown) numbers of measurements. This is done as follows.

For the purpose of the present study, the estimate of an event based on k measurements is an n_x-vector with components assumed to be given by the (rather simple) expression[3]

$$\hat{x}_i(k) = \frac{1}{k}\sum_{l=1}^{k} z_i(l) \quad \text{with variance}$$
$$\sigma_i(k)^2 \triangleq E[(\hat{x}_i(k) - x_i)^2] = \frac{\sigma_i^2}{k}, \qquad i = 1,\ldots,n_x \tag{1}$$

where x_i is the true value of component i and the measurements are

$$z_i(l) = x_i + w_i(l), \qquad i = 1,\ldots,n_x \tag{2}$$

with the noises $w_i(l)$ zero mean, white and with variance σ_i^2. For the n_x components the noises are assumed to be uncorrelated. Note that the simple model (1) implies that there is no process noise and, consequently, no cross-correlation between the track estimates of the same event by different observers.

Then, given two estimates $\hat{x}_i(k_1)$ and $\hat{x}_i(k_2)$ from the same source, their correlation coefficient is [1]

$$\rho_i(t_1,t_2) = \frac{\min(\sigma_i(k_1)^{-2}, \sigma_i(k_2)^{-2})}{\sigma_i(k_1)^{-1}\sigma_i(k_2)^{-1}} \tag{3}$$

under the common origin assumption. This result will be used when carrying out certain track to track associations with correlated errors.

Note that estimates pertaining to the same event but obtained by different observers/sources have uncorrelated errors because there is no "common process noise" [4] since the above model (1) has no process noise at all.

On the other hand, if two tracks have different origins (i.e., they represent different events), their errors are uncorrelated. However, if one has two tracks with the same variance, the use of (3), which assumes common origin, would lead to unity correlation and the difference between the estimates—which is used in the test (see [4])—would then have zero variance. To avoid this, an upper bound (of, say, 0.95) could be used for (3).

We will consider false tracks but not tracks corrupted by false measurements.

3. HIERARCHICAL DECOMPOSITION OF THE PROBLEM

As discussed in Section 2, each report is accompanied by 3 indices: t (event/track number, assigned by the network), n (network number) and s (source number). In order to use an assignment algorithm for track to track association (to be followed by fusion) across several lists, where each list will be a collection of track/event reports that arrived at the FC via the various networks from a particular source, one has to make sure no track/event appears more than once in each list. Consequently, before the final association and fusion, one has to eliminate redundant tracks.

Figure 2 shows a schematic of the system, where the observers may transmit through any of the N_n communication networks. The fusion center explicitly shows its two main components, the *Removal of redundant tracks from the same source that arrived through different networks* and the *Track association across the N_s source lists* followed by fusion of the selected tracks.

The following operations are performed at the FC:

0. Removal of redundant tracks from the same source that arrived on the same network.

If one has two tracks from network n with the same t (and s), then only the most recent one (with the smallest variances) is kept. This eliminates "old (redundant) tracks" communicated through the same network that have been superseded by their updated versions.

1. Removal of redundant tracks from the same source that arrived through different networks.

[2]This peculiarity of the network, while strange from the researcher's point of view, is a real-world fact. Even if the network would have transmitted the observer's track ID, the problem would still be challenging.

[3]More general estimates can be used as long as the corresponding likelihood functions (the pdfs of the estimates conditioned on their origin [3]) are available.

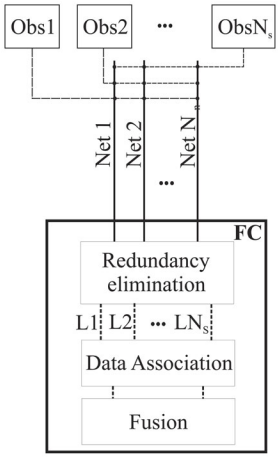


Fig. 2. Block schematic showing the hierarchical decomposition of the problem.

The remaining tracks after step 1 are reshuffled by common source s into lists according to the network they came on. For each s a search is done for "common origin" tracks across these lists using the LR association criterion of [3]. This criterion has to account for the dependence of the tracks (since common origin tracks have common measurements) according to (3). The search for these redundant tracks is done with MDA on (up to) N_n lists for each s. Tracks associated correspond to duplications of the same source reports sent via different networks. The best within each associated set (which is the most recent one) is kept. This step eliminates[4] the duplications in the set of tracks at the FC. Special attention should be given to incomplete associations across the N_n lists because event reports might not be sent via all the networks.

2. Track association across source lists.

At this point each source list contains the latest estimate for each event available from that source. The MDA will associate the elements across the N_s source lists (with at most one from each list) using the LR criterion function from [3]. The errors across the list elements are uncorrelated because there is no process noise and they have no common measurements.

3. Fusion of common origin tracks.

The tracks from different sources that have been designated by the MDA as having a common origin (same event) are fused. This is done according to the ML criterion from [9].

This decomposition drastically reduces the dimensionality of the problem from $N_s \times N_n$ to N_s problems of dimension N_n and one of dimension N_s. If the observer's track ID was available at the FC (via "ideal" networks), one would have only the second stage (item 2 above). The hierarchical approach avoids the need for an unnecessarily large single problem (the one formed by considering all the lists formed using the network IDs) in the case of the real-world networks and reduces the problem to two subproblems of the size one would have with ideal networks.

[4]Strictly speaking, this is a statistical testing approach subject to a maximum allowable (small) probability of error—incorrectly keeping a redundant track—according to which the test threshold is selected. This test then maximizes the power of the test—the probability of eliminating truly redundant tracks. However, this power (probability of eliminating truly redundant tracks) depends on the actual separation between neighboring distinct tracks.

4. ASSOCIATION CRITERIA

The following criteria can be used for track-to-track association:

i) Normalized distance squared (NDS). This criterion will be shown to be significantly inferior to the LR criterion. The reason for this is that large covariances reduce the NDS without imposing any penalty in view of the large uncertainty. Also, the use of NDS for associating tracks from more than 2 lists leads to necessarily heuristic approaches (see [12, 13]).

ii) Likelihood functions (LF). Since LF are pdf—with a physical dimension—they cannot be used for comparing associations of different number of tracks; this approach will not be pursued in this paper.

iii) Likelihood ratios (LR). These are physically dimensionless quantities and, consequently, allow comparison of associations of different number of tracks.

4.1. The Likelihood Ratio for Association

4.1.1. Removal of redundant tracks from the same source that arrived through different networks

Before using the MDA for removal of redundant tracks from the same source that arrived through different networks, a gating test should be used to select the candidates. The test will be based on the normalized distance squared (e.g., [5]) between pairs of tracks. Denoting the tracks now with full—triple—indexing, their normalized distance should be below a threshold, i.e.,

$$D(\hat{x}_{t_i,n_i,s}, \hat{x}_{t_j,n_j,s}) \triangleq (\hat{x}_{t_i,n_i,s} - \hat{x}_{t_j,n_j,s})' [T_{(t_i,n_i,s),(t_j,n_j,s)}]^{-1} (\hat{x}_{t_i,n_i,s} - \hat{x}_{t_j,n_j,s}) < c \tag{4}$$

where

$$T_{(t_i,n_i,s),(t_j,n_j,s)} \triangleq \operatorname{cov}(\hat{x}_{t_i,n_i,s} - \hat{x}_{t_j,n_j,s}) = P_{t_i,n_i,s} + P_{t_j,n_j,s} - P_{(t_i,n_i,s),(t_j,n_j,s)} - P'_{(t_i,n_i,s),(t_j,n_j,s)} \tag{5}$$

and (following [4], Sec. 8.4) $P_{t_i,n_i,s}$ is the track covariance corresponding to track $\hat{x}_{t_i,n_i,s}$ and $P_{(t_i,n_i,s),(t_j,n_j,s)}$ is the cross-covariance of the tracks with the indicated indexes. The elements of the cross-covariance are obtained according to (3) in the present problem.

The common origin likelihood function for a set of r tracks, composed of q non-dummies and $r-q$ dummies, from source s that arrived on different networks

$$\mathcal{T}_i = \{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_q}, n_{i_q}, s)\} \tag{6}$$

is given, under a diffuse prior assumption, by [3]

$$\Lambda(\mathcal{H}_{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_q}, n_{i_q}, s)}) = \frac{1}{V}\mathcal{N}[\hat{\mathbf{x}}_{\mathcal{T}_i}; 0, \mathbf{P}_{\mathcal{T}_i}]\mu_{\text{ex}}^{r-q}(P_d)^q(1-P_d)^{r-q} \tag{7}$$

where V is the (large) volume of the state space (in which the true common state is assumed uniformly distributed), P_d is the detection probability of an event, μ_{ex} is the "spatial density of extraneous targets," as shown in [2] using a spatial Poisson distribution. This density can be taken as the expected number of tracks (true and false) divided by V, and

$$\hat{\mathbf{x}}_{\mathcal{T}_i} \triangleq \begin{bmatrix} \hat{x}_{t_{i_2}, n_{i_2}, s} - \hat{x}_{t_{i_1}, n_{i_1}, s} \\ \vdots \\ \hat{x}_{t_{i_q}, n_{i_q}, s} - \hat{x}_{t_{i_1}, n_{i_1}, s} \end{bmatrix} \tag{8}$$

is a stacked $(q-1)n_x$-vector (with n_x the dimension of x), whose covariance has diagonal blocks

$$(\mathbf{P}_{\mathcal{T}_i})_{j-1,j-1} = T_{(t_{i_1}, n_{i_1}, s), (t_{i_j}, n_{i_j}, s)}, \qquad j = 2, \ldots, q \tag{9}$$

where $T_{(t_{i_1}, n_{i_1}, s), (t_{i_j}, n_{i_j}, s)}$ is given by (5), and the off-diagonal blocks are given by

$$(\mathbf{P}_{\mathcal{T}_i})_{k-1,j-1} = P_{t_{i_1}, n_{i_1}, s} - P_{(t_{i_k}, n_{i_k}, s), (t_{i_1}, n_{i_1}, s)} - P_{(t_{i_j}, n_{i_j}, s), (t_{i_1}, n_{i_1}, s)} + P_{(t_{i_k}, n_{i_k}, s), (t_{i_j}, n_{i_j}, s)}, \quad k, j = 2, \ldots, q; \quad k \neq j. \tag{10}$$

Since the comparisons might have to be made between hypotheses containing different numbers of tracks, likelihood functions cannot be used since, being pdf based, they have different physical dimensions for different numbers of tracks [5] and thus cannot be compared. Consequently, likelihood ratios have to be used. The likelihood ratio is obtained by dividing the above likelihood function by the joint pdf of the r track estimates under consideration, under the hypothesis that they are not of common origin. Given r tracks, the first one can be assumed uniformly distributed in V and the rest, which should be in its neighborhood, are again assumed to have a pdf given by the "spatial density of the extraneous targets," μ_{ex} (this is a consequence of the analysis presented in [2] using a spatial Poisson process).

Thus the LR will be

$$\mathcal{L}(\mathcal{H}_{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_q}, n_{i_q}, s)}) = \frac{1}{\mu_{\text{ex}}^{q-1}}\mathcal{N}[\hat{\mathbf{x}}_{\mathcal{T}_i}; 0, \mathbf{P}_{\mathcal{T}_i}](P_d)^q(1-P_d)^{r-q} \tag{11}$$

Note that the use of non-unity P_d "penalizes" incomplete associations. This becomes necessary for the costs obtained for this problem, as full tracks may yield better costs when split. For example, suppose a 4-D set of tracks having common origin $\{i_1, i_2, i_3, i_4\}$ yields cost C_{complete}, and a partition of two feasible partial track associations, $\{i_1, i_2, 0, 0\}$ and $\{0, 0, i_3, i_4\}$, yields costs C_{split_1} and C_{split_2} satisfying $C_{\text{complete}} < C_{\text{split}_i}$ for $i = 1, 2$ but $C_{\text{complete}} > C_{\text{split}_1} + C_{\text{split}_2}$. Then the split tracks can minimize the cost, although providing a less accurate solution. This undesirable phenomenon has been observed a number of times and the use of P_d in the LR cost function will avoid it.

The cost function to be used by the assignment algorithm in associating across the N_n lists is the negative of the logarithm of (11). This covers both complete associations (of N_n-tuples) as well as incomplete associations of q-tuples ($q < N_n$). In the latter case the "dummy element" [18] (i.e., no track) is chosen from $N_n - q$ lists; for these elements the likelihood ratios are taken as unity and consequently, they do not modify (11). The cost calculation for an association containing q non-dummy elements requires inversion of a $q \cdot n_x \times q \cdot n_x$ matrix.

Note that an association should have a negative cost if the (non-dummy) tracks in it are more likely to have a common origin than not. For $q = 1$, i.e., when a single track is associated with dummies (hence it is unassociated), the cost for this should be larger than an association with negative cost. Consequently an "unassociation" will be given zero cost and also associations with positive costs will be discarded, implementing an implicit fine gating.

From the LR cost formulation above it can be seen that, in terms of the computational complexity required, Network MDA is a harder problem than sequential MDA/MHT. In the case of having S lists with m_S reports in each, and each report contains a vector of data of dimension n, the sequential MDA/MHT calculates the association cost adding one list at a time. Each element of the first list does pass the gating test with a certain proportion, α, of the reports in the following list, and for the cost calculation the inversion of a matrix of size n is required; hence the computation requirement is $m_S(\alpha m_S)n^3$ for the first two lists. When the third list is added the requirement is $m_S(\alpha m_S)^2 n^3$ as a consequence of the reports in the third list passing the gating test, what accounts for the exponential growth of the hypotheses. Finally when the last list is added, the number of operations required is $m_S(\alpha m_S)^{S-1}n^3$, so the number of operations required is $O(m_S^S \alpha^{S-1} n^3)$. On the other hand, for the networked MDA, the static nature of the problem makes the cost calculation require the inversion of a $(S-1)n$ matrix, which renders a complexity of $O(m_S^S \alpha^{S-1}((S-1)n)^3)$. The cost calculation complexity of the sequential m-best 2D algorithm is much lower, as each list incorporated does also give birth to αm_S^2 associations, out of which roughly m_S are kept after the

2D association. Thus the number of cost calculations involved is $O(m_S(\alpha m_S)(S-1))$.

4.1.2. Track association across source lists

The common origin likelihood ratio for a set of N_s tracks (one from each source list), composed of q non-dummies and $N_s - q$ dummies, from different sources s_{j_l}, $l = 1,\ldots,q$

$$\mathcal{T}_j = \{(t_{j_1}, n_{j_1}, s_{j_1}), \ldots, (t_{j_q}, n_{j_q}, s_{j_q})\} \tag{12}$$

is given, similarly to (11), by

$$\mathcal{L}(\mathcal{H}_{(t_{j_1},n_{j_1},s_{j_1}),\ldots,(t_{j_q},n_{j_q},s_{j_q})}) = \frac{1}{\mu_{\text{ex}}^{q-1}} \mathcal{N}[\hat{\mathbf{x}}_{\mathcal{T}_j}; 0, \mathbf{P}_{\mathcal{T}_j}](P_d)^q (1-P_d)^{N_s-q} \tag{13}$$

where $\hat{\mathbf{x}}_{\mathcal{T}_j}$ is a stacked $(q-1)n_x$-vector as in (8) except that here all the tracks are from different sources.

The covariance in (13) has diagonal blocks

$$(\mathbf{P}_{\mathcal{T}_j})_{k-1,k-1} = T_{(t_{j_1},n_{j_1},s_{j_1}),(t_{j_k},n_{j_k},s_{j_k})} = P_{t_{j_1},n_{j_1},s_{j_1}} + P_{t_{j_k},n_{j_k},s_{j_k}}, \quad k = 2,\ldots,q. \tag{14}$$

Differing from (5), there are no cross-covariance terms in (14) as these tracks are from different sources and there is no process noise. Consequently, the off-diagonal blocks are

$$(\mathbf{P}_{\mathcal{T}_j})_{k-1,l-1} = P_{t_{j_1},n_{j_1},s_{j_1}}, \qquad k,l = 2,\ldots,q; \quad k \neq l. \tag{15}$$

4.2. The Normalized Distance Criterion for Association

4.2.1. Removal of redundant tracks from the same source that arrived through different networks

In addition to using the LR criterion, the simpler NDS (normalized distance squared) [4], also known as "Chi-square," which is the (negative of the) exponent of the likelihood function, will be considered. The NDS between a pair of tracks coming from the same observer through two different networks is defined as

$$D(\hat{x}_{t_{i_j},n_{i_j},s}, \hat{x}_{t_{i_k},n_{i_k},s}) \triangleq (\hat{x}_{t_{i_j},n_{i_j},s} - \hat{x}_{t_{i_k},n_{i_k},s})' [T_{(t_{i_j},n_{i_j},s),(t_{i_k},n_{i_k},s)}]^{-1} (\hat{x}_{t_{i_j},n_{i_j},s} - \hat{x}_{t_{i_k},n_{i_k},s}) \tag{16}$$

which is the LHS of (4). The covariance matrix is defined as in (5) with nonzero cross-covariance matrices due to the common origin of the tracks.[5]

As before, a track to dummy association will be given zero cost. However, due to the positiveness of the distance, the use of zero cost for the association to dummies implies that the best (least cost) assignment will consist of only track to dummy associations with zero cost. To avoid this problem and to implement the gating (4) at the same time, the cost for each association pair is calculated as

$$C_{j,k} = D_{j,k} - \chi^2_{n_x}(\alpha) \tag{17}$$

where $\chi^2_{n_x}(\alpha)$ is a level α threshold (usually big enough, say $\alpha > .99$) obtained from χ^2 tables with n_x degrees of freedom, and $D_{j,k}$ corresponds to the distance between tracks j and k (with abbreviated notation) as specified in (16). Then, if the distance $D_{j,k}$ is greater than the threshold, an assignment with positive resulting cost will never be selected, since an assignment to a dummy offers better (lower) cost. In case when the distance is smaller than the threshold, smaller distances will yield more negative costs than bigger distances, making them more attractive for assignment.

Since this distance criterion, defined above for a single pair, will be shown to yield inferior performance compared to the LR criterion, we will not pursue it any further. The use of the distance criterion for more than two tracks was discussed in [12].

4.2.2. Track association across source lists

The cost of associating a pair of tracks from different lists is the same as defined in (16) (suitably modifying the list indices), with the covariance matrix defined as

$$T_{(t_{j_l},n_{j_l},s_{j_l}),(t_{j_k},n_{j_k},s_{j_k})} = P_{t_{j_l},n_{j_l},s_{j_l}} + P_{t_{j_k},n_{j_k},s_{j_k}} \tag{18}$$

which is similar to (14).

Similarly to the removal of redundant tracks case, the cost of each association pair is defined here by subtracting a suitably defined threshold as in (17).

5. MULTIDIMENSIONAL ASSIGNMENT ALGORITHMS

Once the reports are split into lists, for duplication elimination in the first stage and for selection of candidates for fusion in the second stage, a multidimensional assignment problem [18] needs to be solved. A comparison of three different methods is carried out, one based on Lagrangean Relaxation, another based on a sequential calculation of m-best 2D assignments and the last one based on Linear Programming Relaxation.

5.1 Multidimensional Assignment Problem

The Multidimensional Assignment problem, also known as the S-D Assignment problem, consists of partitioning $S \geq 3$ lists of reports into S-tuples of report associations (RA) in a way that every report of every list is used, and that it is used only once. This problem is known to be NP-hard, which motivates the use of suboptimal methods.

[5] Note that we are testing whether these tracks from the same source represent the same event.

To allow for missed detections and false alarms (hence lists of different size) each list also contains a "dummy" element on which the above constraints do not apply. The inclusion of these reports converts the problem into a Generalized S-D Assignment. For the generalized problem we can partition the set of candidate associations into two. One subset corresponds to real track report associations, that is, S-tuples containing at least two non-dummy reports. The second subset corresponds to associations, that contain only one non-dummy report, i.e., unassociated track reports.

Defining binary event variables $\rho_{i_1,\ldots,i_S}$ to take value 1 when $\mathcal{T}_i = \{(t_{i_1}, n_{i_1}, s_{i_1}), \ldots, (t_{i_S}, n_{i_S}, s_{i_S})\}$, the S-tuple of tracks, is associated and 0 otherwise, the problem can be recast as a Linear Binary Programming (LBP) problem

$$\min_{\{\rho\}} \sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_S=0}^{n_S} c_{i_1 i_2 \ldots i_S} \rho_{i_1 i_2 \ldots i_S} \tag{19}$$

subject to

$$\begin{aligned}
&\sum_{i_2=0}^{n_2} \sum_{i_3=0}^{n_3} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_1 = 1,2,\ldots,n_1 \\
&\sum_{i_1=0}^{n_1} \sum_{i_3=0}^{n_3} \cdots \sum_{i_S=0}^{n_S} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_2 = 1,2,\ldots,n_2 \\
&\qquad\qquad \vdots \\
&\sum_{i_1=0}^{n_1} \sum_{i_2=0}^{n_2} \cdots \sum_{i_{S-1}=0}^{n_{S-1}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_S = 1,2,\ldots,n_S
\end{aligned} \tag{20}$$

where $c_{i_1 i_2 \ldots i_S}$ is the cost of each association, given by the negative log-likelihood ratio (NLLR)

$$c_{i_1 i_2 \ldots i_S} = -\ln \mathcal{L}(\mathcal{H}_{(t_{i_1}, n_{i_1}, s), \ldots, (t_{i_m}, n_{i_m}, s)}) \tag{21}$$

where this LR is defined in (11). The dummy element in each list has index 0.

The following sub-sections briefly describe the three algorithms used for solving the described MDA assignment.

5.2 Lagrangean Relaxation Based S-D Assignment

The LBP problem can be suboptimally solved by relaxing the constraints and using Lagrange multipliers for them, until a 2D problem is obtained. This reduced dimension association problem can be solved exactly by well known algorithms such as JVC, Auction, Relax, etc. Then the relaxed constraints can be added one by one, using again a 2D association algorithm. This approach has been extensively described [17, 18, 19]. Sketchily, the S-D assignment problem is solved as a series of relaxed 2D subproblems in two phases: 1) relaxation of constraints, and 2) update of the Lagrange multipliers and constraint enforcement. In the first phase, each constraint set $r = S, S-1, \ldots, 3$ is appended to the cost function using Lagrange multiplier $\mathbf{u}_r$. After relaxing constraint set 3 we have a 2D assignment problem, which in our case is optimally solved using the generalized auction algorithm [7] (up to a certain accuracy—the granularity of the auction). After this the constraint enforcement phase begins by computing a solution of the 3D problem consisting of the previous assignment and the third constraint set, using again a 2D generalized auction algorithm. Then the multipliers $\mathbf{u}_3$ are updated using a subgradient method. Similarly, the successive constraints $r = 4,5,\ldots,S$ are enforced via a 2D assignment algorithm, and the multipliers $\mathbf{u}_r$ are updated. These two steps are repeated until all the constraints are satisfied in the relaxed problem, in which case the solution is optimal, or until the feasible solution is of acceptable quality.

It has been found that most of the running time (usually more than 95% of it) spent in solving the association problem with this approach is consumed by the calculation of the costs $c_{i_1 i_2 \ldots i_S}$. To alleviate this, clustering strategies may be used [10]. In our case performing gating during the construction of the association tree prevents the exponential growth of non-matching track branches, avoiding the corresponding cost calculations.

5.3 Sequential m-best 2D Assignment

This method provides a heuristic approach to obtain a Generalized S-D assignment solution. It relies on the solution of a sequence of Generalized 2D assignment problems [18, 20], where a dummy report element is introduced in each list. These dummy reports, which allow for missed detections, are not subject to the constraints (as discussed above), and the cost of associating any report to them is defined to be zero (the cost of a "real association" is negative). As a result, this modification allows the association between lists with different numbers of reports, and also allows for poorly matching reports not to be associated among them but to dummy reports, i.e., stay unassociated.

The algorithm is started by associating two lists, using a 2D generalized assignment algorithm. In this step, not only the best solution is kept, but also the following $m-1$ best (in terms of the association cost) solutions are found using an adaptation of Murty's method [16, 20]. The goal is to find the m best solutions of this assignment problem. This is achieved by first finding the best solution, using the generalized 2D assignment algorithm, and then partitioning it into exhaustive non-overlapping subproblems of smaller dimension. These problems are solved by the previous assignment algorithm, and out of them the best one will correspond to the second best solution. Partitioning this problem again, and keeping the best solution of the list formed with all the previous partitions provides the desired best solutions of the problem.

An adaptation of Murty's algorithm is necessary to handle the use of dummy elements in the generalized 2D assignment problem. The main point of this adaptation

consists of keeping dummies after a dummy has been used as pivot in the partitioning process, as opposed to partitioning a solution when non-dummy reports are selected as pivots, in which case this non-dummy report is voided from some of the spawning assignments, and forced in the rest of them. The complexity of the algorithm can be reduced from $O(mn^4)$ to $O(mn^3)$ by a clever implementation of the partitioning process, the inheritance of variables from the assignment method, and by bounding the subproblem costs [15]. Further improvement can be obtained by switching the assignment solving algorithm as a function of the sparsity of the problem, and by parallelizing the algorithm [20].

The Sequential m-best 2D Assignment is started by associating 2 lists, usually lists 1 and 2, and obtaining the top m solutions. This initial problem is a plain Generalized 2D problem, where the cost of associating a pair of reports is calculated as discussed above. Each of the solutions of this problem consists of a set of 2-tuples:

$$T_2^{(r)} = \{\mathcal{T}_{2,1}, \ldots, \mathcal{T}_{2,q_{2,r}}\}, \qquad r = 1, \ldots, m \tag{22}$$

where $q_{2,r}$ is the number of associated pairs for solutions from the first 2 lists and

$$\mathcal{T}_{2,i} = \{(t_{i_1}, n_{i_1}, s_{i_1}), (t_{i_2}, n_{i_2}, s_{i_2})\}$$

is a 2D report association, the basic component of each solution.

To add a new list, another generalized 2D problem must be solved to match the new list elements with the associations in $T_2^{(r)}$, for each r. Each of the corresponding 2D assignment matrices will have dimension $q_{2,r} \times n_3$, where n_3 is the number of elements in list 3, and the cost for each element is calculated using the 3-tuple defined by

$$\mathcal{T}_{3,i} = \{\mathcal{T}_{2,j}, (t_{i_3}, n_{i_3}, s_{i_3})\} \tag{23}$$

where $j = 1, \ldots, q_{2,r}$ and $i = 1, \ldots, n_3$, with the cost function defined as before.

Rather than calculating m new solutions for each of the previous m best solutions, which would yield m^2 solutions from which to pick the top m, we initialize the list of problems with the m previous assignments. This makes the algorithm run only once, decomposing the solution which has the best cost at each time. Then the same m best associations will be obtained for considerably less computation [11].

After this second step, the obtained m best solutions are represented by

$$T_3^{(r)} = \{\mathcal{T}_{3,1}, \ldots, \mathcal{T}_{3,q_{3,r}}\}, \qquad r = 1, \ldots, m \tag{24}$$

where $q_{3,r}$ is the number of associated triplets for each solution, and

$$\mathcal{T}_{3,i} = \{(t_{i_1}, n_{i_1}, s_{i_1}), (t_{i_2}, n_{i_2}, s_{i_2}), (t_{i_3}, n_{i_3}, s_{i_3})\}$$

is a 3D report association.

The rest of the lists are incorporated using the same procedure. That is, for each of the m best solutions obtained after adding list k, a 2D association matrix with the costs of associating its RAs and the reports from list $k+1$ is calculated. Costs are again calculated using all the combinations of k-D RAs and list $k+1$ reports, together with (13) if there are more that two non-dummies, and setting 0 cost in case of having only one non-dummy element in the $(k+1)$-D RA. After all cost matrices corresponding to the previous best solutions and the new list are obtained, extended solutions formed by $(k+1)$-D RAs are calculated using the m-best algorithm with the mentioned cost matrices. After solving these problems, the m best solutions are found and the procedure is repeated with list $k+2$, and so on until the last list is incorporated. When the last list is finally incorporated, only the top solution is used as the resulting S-D association matrix, thus providing a "hard" solution to the association problem. Soft solutions that combine the m final associations may provide a better quality solution. This is currently under investigation.

In the results section, we will show that even for large values of m the quality of the solution for the problem considered does not improve from just taking the best solution at each step.

5.4 Linear Programming Based Assignment

The Linear Binary Programming problem defined by (19) and (20) can be relaxed to a Linear Programming (LP) problem by allowing non-integer values for the event variables ρ. This relaxed problem can be solved using several efficient LP algorithms, but the integrality of the solution is not ensured. This brings up the question of what to do with the fractional assignments. In general for the assignment problem, the occurrence of these fractional solutions (assignments) is rare. Thus for the present work we consider the fused fractional assignments as fractional tracks, i.e., after fusion an assignment with $0 < \rho_{i_1 i_2 \ldots i_S} < 1$ will count as (a fractional) $\rho_{i_1 i_2 \ldots i_S}$ track.

The number of variables involved in the LP problem is proportional to the product of the number of reports in each list. For example, for a problem with 4 lists and 10 tracks per list, the number of variables is greater than 10^4. To reduce the number of variables, we consider only those variables with zero or negative cost (i.e., we perform an implicit gating). The resulting reduced set of indexes is

$$\Xi = \{(i_1 i_2 \ldots i_S) : c_{i_1 i_2 \ldots i_S} \leq 0,\ i_1 = 0, \ldots, n_1,\ \ldots,\ i_S = 0, \ldots, n_S\}. \tag{25}$$

Then the reduced LP problem becomes

$$\min_{\{\rho\}} \sum_{(i_1 i_2 \ldots i_S) \in \Xi} c_{i_1 i_2 \ldots i_S} \rho_{i_1 i_2 \ldots i_S} \tag{26}$$

subject to

$$\begin{aligned}
&\sum_{\{(i_2 i_3 \ldots i_S):(i_1 i_2 \ldots i_S)\in\Xi\}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_1 = 1,2,\ldots,n_1 \\
&\sum_{\{(i_1 i_3 \ldots i_S):(i_1 i_2 \ldots i_S)\in\Xi\}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_2 = 1,2,\ldots,n_2 \\
&\qquad\qquad \vdots \\
&\sum_{\{(i_1 i_2 \ldots i_{S-1}):(i_1 i_2 \ldots i_S)\in\Xi\}} \rho_{i_1 i_2 \ldots i_S} = 1, \qquad i_S = 1,2,\ldots,n_S.
\end{aligned} \tag{27}$$

The solver used for this work is LPSolve [6], a free GNU program which implements a primal-dual method. This solver will be shown to provide fast solutions when the number of variables involved is below 10^4, although it is not of the self-dual family, which is reported to operate very efficiently for the 3D assignment problem [14].

6. SIMULATION RESULTS

The scenario consists of a set of events (missile launches) for which 2-D position, launch time and heading need to be estimated, using multiple sensors. The surveillance region covered by each sensors is $x \in [0,10000]$, $y \in [0,10000]$ for position, $\psi \in [0,30°]$ for heading. The launches occur randomly in the surveillance region, during a time interval $t \in [0,10]$. Each sensor receives measurements from each target on average every 10 units of time and transmits reports on average every 20 units of time. The time span of the scenarios is 200 units of time, and both redundancy elimination and fusion are performed every 20 units of time. The false report rate per sensor is P_f per unit time, so on the average there are $200P_f$ false track/event reports per sensor. Also, each of the communication networks has a probability P_r of delivering the report to the FC (its reliability). The measurements from each target received by each sensor are corrupted with white Gaussian noise, with standard deviations $\sigma_x = \sigma_y = 2500$, $\sigma_\psi = 10$, $\sigma_t = 3$ (uncorrelated components).

Scenario 1. The parameters that specify this scenario are: 2 events (launches), $N_n = 2$ networks with reliability $P_r = .5$, 2 sensors and false report rate (per unit time) $P_f = .01$. The differences in the values of interest between the two true event locations, headings and launch times, are $\Delta x = \Delta y = 1000$, $\Delta h = 6$, $\Delta t = 2$, respectively.

Figures 3 and 4 show the result of the use of the LR criterion and NDS criterion on this scenario, for 1000 Monte Carlo runs. They show the percentage of incorrect fusions (i.e., the fusion of two tracks with different origin) and incorrect eliminations (e.g., eliminating a report from a pair with different origins). It can be seen that when the LR criterion is used, as more data are obtained, the percentage of errors is reduced, despite the presence of false reports, which are handled correctly.

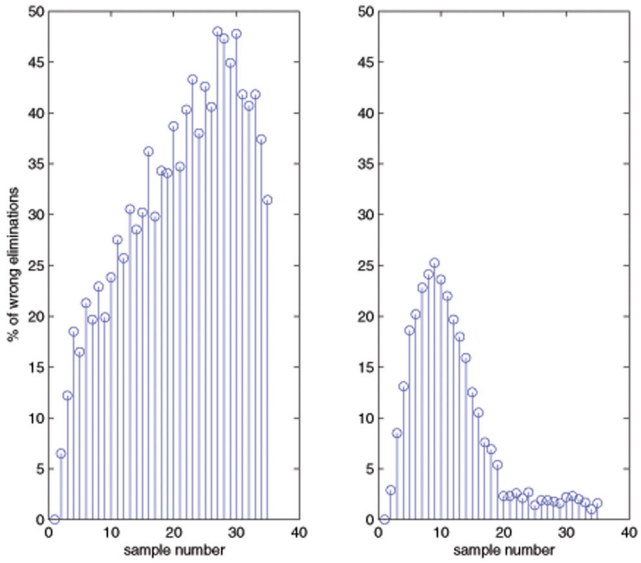


Fig. 3. Percentage of errors in the duplication elimination step using the NDS distance cost criterion (left) and LR cost criterion (right).

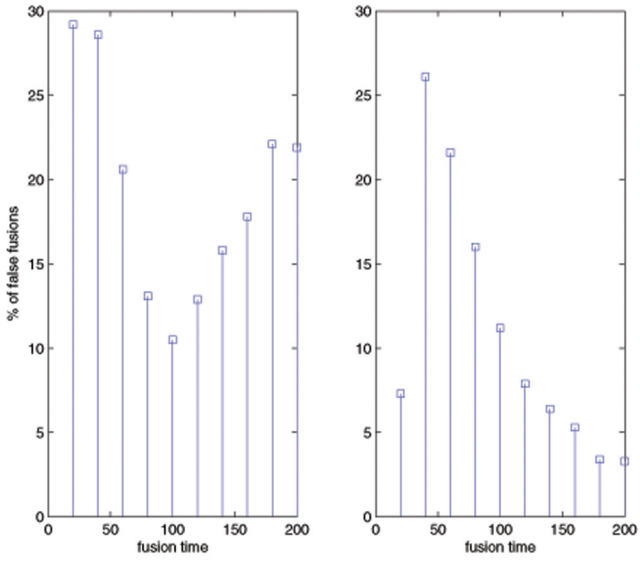


Fig. 4. Percentage of errors in the fusion candidate selection step using the NDS distance (left) cost criterion and LR cost criterion (right).

On the other hand, when the NDS criterion is used, the presence of false reports does affect both stages in a way that the errors committed reach very high levels. This is a consequence of the false reports, which have large standard deviations compared to the normal tracks and yield a low NDS with respect to almost any track. In the LR the presence of the standard deviation terms in the denominator compensates for this, and results in substantially fewer association errors.

Scenario 2. To characterize the behavior of the assignment methods presented in Section 5, different size problems are used, ranging from mid size problems (4D), to bigger problems, up to 7D. The parameters that govern this scenario are: 40 events, $N_n = 4$ networks,

and $N_s = 4, 5, 6, 7$ sensors. The impact of using P_d is investigated, as well as the selection of $\mu_{ex} = n_{ex}/V$ (expressed as only its numerator, n_{ex}), which will be varied from the theoretical value in [2] (expected number of total tracks per unit volume) to just the number of false tracks divided by the surveillance region volume. For each of these problems 50 Monte Carlo runs were performed for each of the proposed search methods.

Due to the initial variance of the estimates of each observer and the surveillance region volume, this scenario is dense in the sense that several tracks gate with each other, making the association problem hard to solve. A coarse (very optimistic) approximation to the number of possible non-overlapping events is to divide the surveillance region volume by the product of n standard deviations of the estimates. At the initial time this is $10000 \times 10000 \times 30 \times 10/(n \times 2500) \times (n \times 2500) \times (n \times 10) \times (n \times 3)$ which gives approximately 2 for $n = 3$ and 10 for $n = 2$. As new measurements come in, the track estimates reduce their variance, and the number of feasible associations reduce, then, considering that at the final time the s.d. is approximately reduced by a factor of $\sqrt{5}$, it is possible to have up to 50 non-overlapping tracks for $n = 3$. Simulations show that each track from a list gates on average 8 tracks from any other list when the track estimates are of poor quality (large variance, early in the game) and about 5 tracks when the uncertainty of the tracks is reduced. That is, around 15–20% of the tracks are gated, so this can be used as a rough estimate of the sparsity.

In general, for any number of lists, the method that finds the lower costs is the Linear Programming based S-D assignment, which does rarely come up with a fractional solution: however this is at a computation time expense that grows with the number of negative cost associations that are fed to the LP solver (this number increases with the number of lists, the number of elements per list and the value of n_{ex}). The solution cost found by the LP solver is almost always lower than the true cost, due to the noise that causes some associations to give better matches than the truth. The Sequential m-best 2D does also usually find lower costs than the truth, but this behavior is dependent on the parameter values. For high n_{ex} this happens both for the cases of using and not using the P_d term, but for low n_{ex} the usage of P_d makes the algorithm find costs lower than the truth, while not using it renders costs higher than the truth. This effect is reduced as the final time is approached. This behavior is a consequence of the myopicity of the algorithm, as will be explained later. The Lagrangean Relaxation based approach usually finds solutions with costs higher that the two previous methods, which are still lower than the truth, but with the advantage of getting a higher number of full associations, an effect that will be further discussed in the sequel.

To quantify the quality of the fused tracks, a partition into 3 fusion categories is done. The first category represents the fusion of reports having identical origin and using one report from each possible list (observer), i.e., completely correct (CC) fusions. The second category represents fusions which have at least 2 reports with common origin, and the rest may be from different origins (but not very distant, due to gating) or not detected, i.e., partially correct (PC) fusions. The last category consists of fusions without any pair of reports coming from the same origin, i.e., completely incorrect (CI) fusions. The sum of the PC and CC fusions is taken as the reference to quantify the performance of the fusion. For this scenario, a number of PC+CC fusions of 40 is optimal, together with a value of 0 for the CI fusions. The number of PC+CC fusions will be used hereafter to measure the quality of the solutions obtained. A more thorough measure would involve also the CI fusions, but these values tend to zero for all the methods presented, and are not presented here for brevity.

Figures 5–7 show PC+CC fusion results for the 7 observers scenario using each of the proposed MDA algorithms. In all cases the solution at earlier times improves as n_{ex} is reduced, and the effect of using P_d is noticeable. For $n_{ex} = 1$ all the algorithms provide good quality solutions, while for higher values the sequential m-best 2D finds lower quality solutions. The usage of the P_d term does alleviate this, although for higher values of n_{ex} this is not enough to get good solutions out of this algorithm. In general the increase in the number of PC+CC fusions comes from a phenomenon to be called "track splitting," in which a CC association is divided into two or more PC associations, which provide an overall lower cost. On the other hand, a decrease in the number of PC+CC fusions is caused for large n_{ex} by the unattractiveness of the track to track association, so the majority of the tracks are associated with dummies, resulting in a smaller number of fusions.

The Sequential m-best 2D (Sm2D) algorithm was used with two different values of m, 1 and 10, with practically the same results, and contrary to expectations, increasing m does not necessarily improve the cost. This happens mainly due to three reasons:

i) The algorithm associates one list at a time, and the m-best solutions for a problem size like the one considered here (at least 40×40) are very similar (usually differ in only one report). So in general all the m-best associations surviving one of these steps are minor variations of a single association. This can be mitigated by using a much larger m, which in general will be a function of the problem size; however, the increase in the problem size implies a need for huge values that usually render the algorithm time-infeasible, so values of m bigger than 10 are not used.

ii) The hierarchical approach and the suboptimality of the calculations due to the fact that the data to be fused has already undergone association, which has some probability of producing errors, and can contain duplicate tracks. These redundant tracks affect the fu-

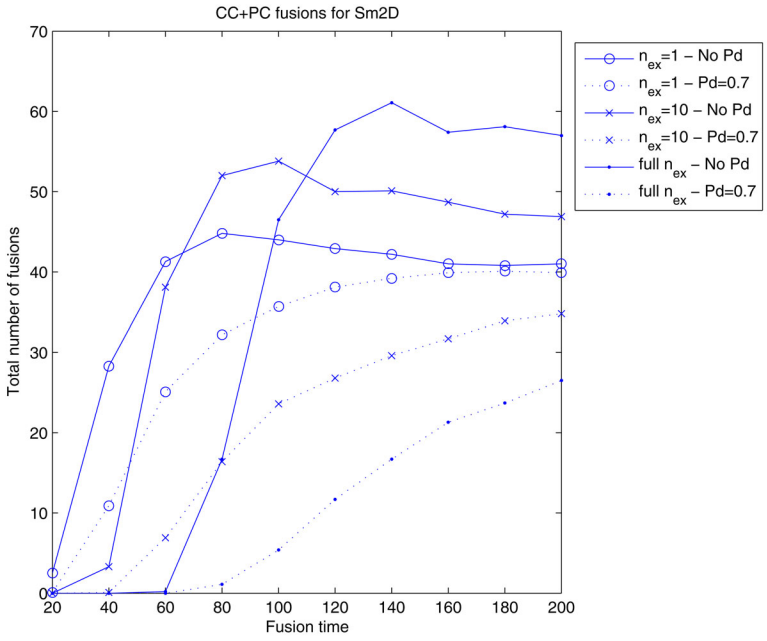


Fig. 5. Fusion quality for scenario 2, 7 observers, using S*m*2D.

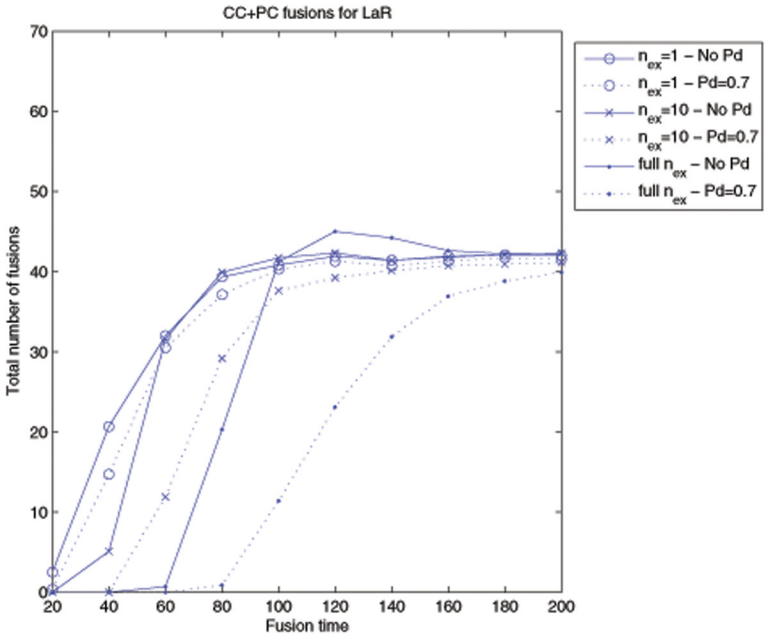


Fig. 6. Fusion quality for scenario 2, 7 observers, using LaR.

sion, and in general, the inclusion of redundant or noisy reports allows the cost to be lower than the true one. For example, if a track is observed by all 4 sensors, it is possible that the S*m*2D algorithm groups them into two different sets of 2, due to its "myopic (greedy) approach" for the sake of a lower cost. To alleviate this problem, the use of a "penalization" coefficient, P_d, was introduced in the cost definition, and proves to ameliorate the quality of the resulting tracks.

iii) The myopicity, which seems to be the main reason that, for reasonable values of m, the S*m*2D solution has a lower cost but more errors compared to the LaR. Specifically, the S*m*2D algorithm will not retain a positive cost that later could become negative (because it prefers "myopically" the zero or negative costs which fill the top m solutions). Another reason for the differences between S*m*2D and LaR S-D lies in the nature of the latter. There, in the relaxed cost calculations, a minimization operation is performed which usually rules out partial associations in favor of a full association. This full association is more likely to have better cost than the individual partials but not than the sum of the two complementary (split) partial associations that use the same tracks as the full association (see the example

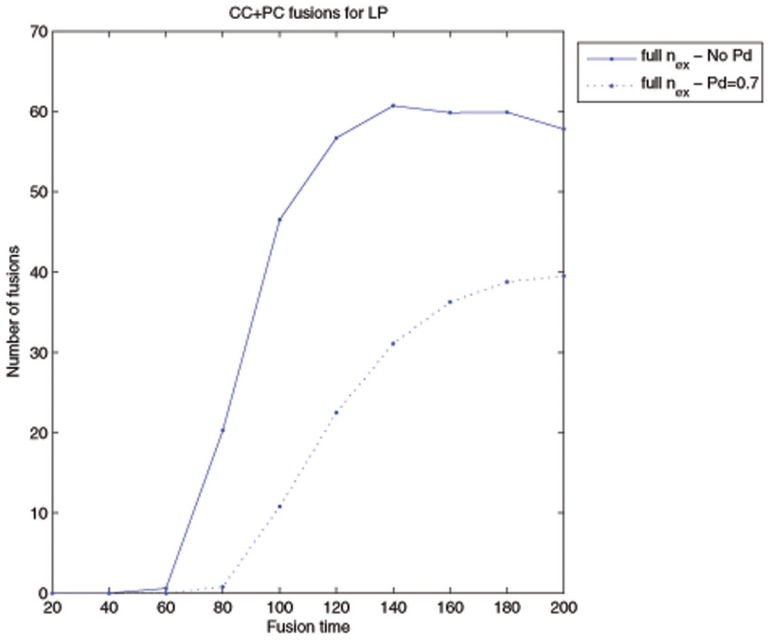


Fig. 7. Fusion quality for scenario 2, 7 observers, using LP (for low n_{ex} run-time is prohibitive—too many costs).

in Section 4.1.1). This feature naturally favors full associations to partial (split) ones, although the latter may yield lower costs when not penalized.

As previously stated, the inclusion of the P_d term in the cost function helps to alleviate the usual problem of having better costs for split tracks than for complete tracks (see Fig. 7), but it may overpenalize the incomplete associations, discarding some real associations (see Fig. 5). Continuing with the example in Section 4.1.1, and using costs obtained from a problem with $n_{\text{ex}} = 1$ and without using P_d, we have the following costs for a particular set of 4 tracks $\{i_1, i_2, i_3, i_4\}$ with common origin: $C_{\text{complete}} = C_{i_1,i_2,i_3,i_4} = -1.46$, while for the partition into two feasible partial associations one has $C_{\text{split}_1} = C_{i_1,i_2,0,0} = -.97$ and $C_{\text{split}_2} = C_{0,0,i_3,i_4} = -.83$. As before, we have $C_{\text{complete}} < C_{\text{split}_i}$ for $i = 1, 2$ but $C_{\text{complete}} > C_{\text{split}_1} + C_{\text{split}_2}$. Then the split associations can minimize the cost, although providing a less accurate solution. The Sm2D algorithm and the LP based method select this (cheaper but undesirable) track split, while the LaR algorithm does not. The reason for the LaR not selecting track splits lies in its suboptimality: at each constraint relaxation step a minimization is performed, fixing one track. In our example, the competition between the complete track and one of the partial tracks will happen when relaxing the constraints corresponding to i_3. At this point we will have costs $D_{i_1,i_2,i_3} = \min_{i_4}(C_{i_1,i_2,i_3,i_4} - \mu_{4,i_4})$, where μ_{4,i_4} is a Lagrange multiplier. It is very likely that both C_{i_1,i_2,i_3,i_4} and $C_{i_1,i_2,0,0}$ will survive and generate D_{i_1,i_2,i_3} and $D_{i_1,i_2,0}$. A further constraint relaxation will yield 2-D costs $B_{i_1,i_2,i_3} = \min_{i_3}(D_{i_1,i_2,i_3} - \mu_{3,i_3})$. Here the two associations will be competing one to one via the modified costs B, and it is usually the case that the Lagrange multipliers do not revert the cost majority relation between full and split associations, hence the association split_1 is discarded and the complete association is kept. As this association is feasible, and yields a better cost than split_2, the multiplier's update will penalize the usage of this split association, and hence causing the complete association to be selected after several iterations of the algorithm. This feature of the algorithm is undesirable in terms of getting the lowest possible cost, but in our case turns out to be an advantage, as it implicity forces a track completeness constraint. In other applications, where there are no full associations with common origin, this feature may yield degraded results.

REMARK The above discussion raises the issue of cost and optimization algorithm selection. The cost is really a surrogate for the "association accuracy" desideratum. However, no cost is an exact reflection of our desideratum and this motivates our detailed investigation of cost parameters and optimization algorithms.

Besides the quality of the solution, another fundamental aspect of these algorithms is run time, as suboptimality of the solution can be traded off for a speedup in the problem solution (otherwise a complete enumeration of the possibilities would find the optimal solution, at the expense of a huge run time). The algorithms used were coded in C++ and run on a P4-2.8 GHz computer.

Figures 8–10 correspond to run times for problems with 4 observers, 7 observers, and the ratio of computation time of the LaR based method and LP based method, over the time taken by the Sm2D method. Run times are split into two parts, time spent in cost calculation, and time spent purely in the optimization algorithm. As previously mentioned, all the algorithms have the best performance when $n_{\text{ex}} = 1$ and P_d is present,

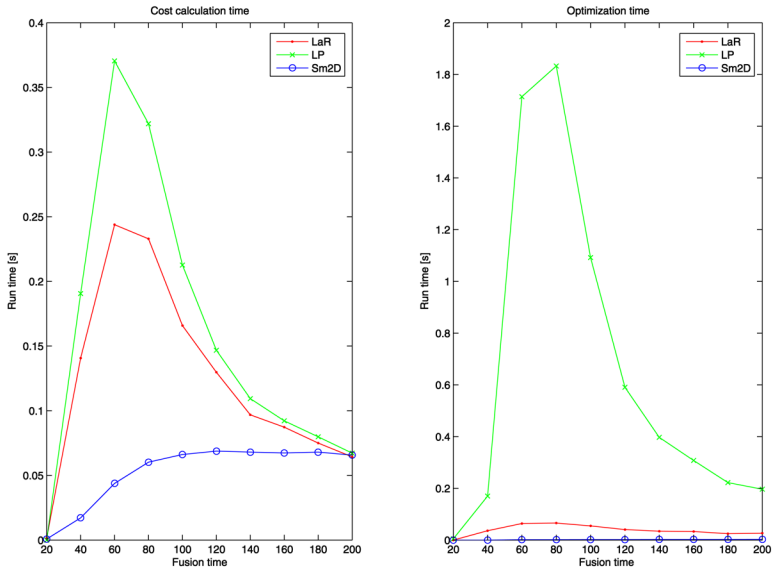


Fig. 8. Run time for a scenario with 4 observers, 4 networks and 40 launches, using $P_d = 0.7$ and $n_{\text{ex}} = 1$. The left figure shows the time spent in calculating the association costs, and the right figure shows the time spent in solving the minimization problem.

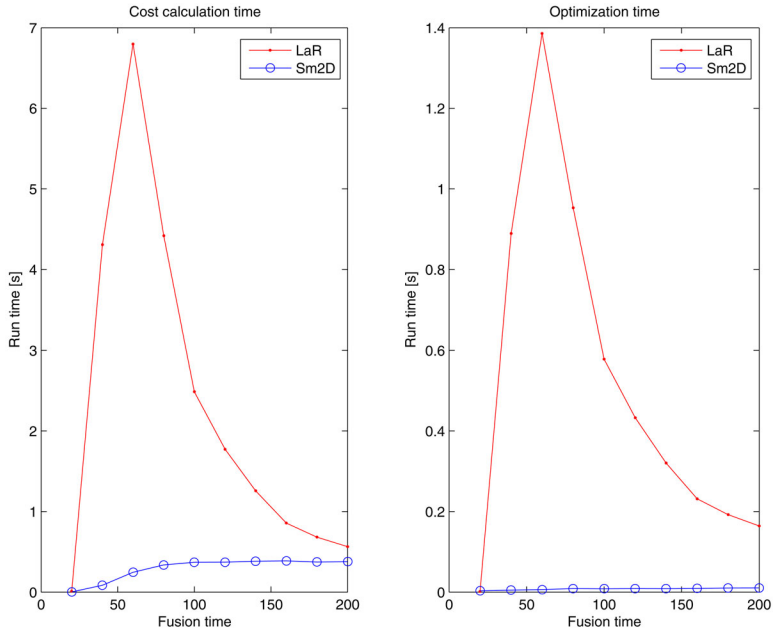


Fig. 9. Run time for a scenario with 7 observers, 4 networks and 40 launches, using $P_d = 0.7$ and $n_{\text{ex}} = 1$. Linear Programming time is not shown as it exceeds the times shown by at least 2 orders of magnitude—a consequence of the large number of negative cost associations for the scenario.

thus the presented results correspond to these parameter values.

The run time of the S*m*2D is the best, and its advantage over LaR and LP improves when the number of lists is increased. Also it does not change noticeably with the variation of the parameters P_d and n_{ex}, while for the other two methods the n_{ex} parameter affects the run time in diverse ways. For the LaR based method a smaller value of n_{ex} does increase the number of nonredundant tracks found after redundancy elimination, especially during the initial fusion times. Hence the feasible cost tree construction takes longer, as many reports gate with each other, making the time increase be polynomial. The time taken purely by the LaR minimization algorithm shows a linear increase with the tree size, so there is also a time increase in the minimization algorithm, but the cost calculation time clearly dominates the overall run time. For the LP based method the cost calculation follows the same pattern, but the minimization part does not show a linear increase as n_{ex} is decreased, as this has not only the effect of increasing the tree size, but also the number of negative association costs. For problems with more than $5 \cdot 10^4$ negative cost associations, the LP algorithm takes too long to run, and

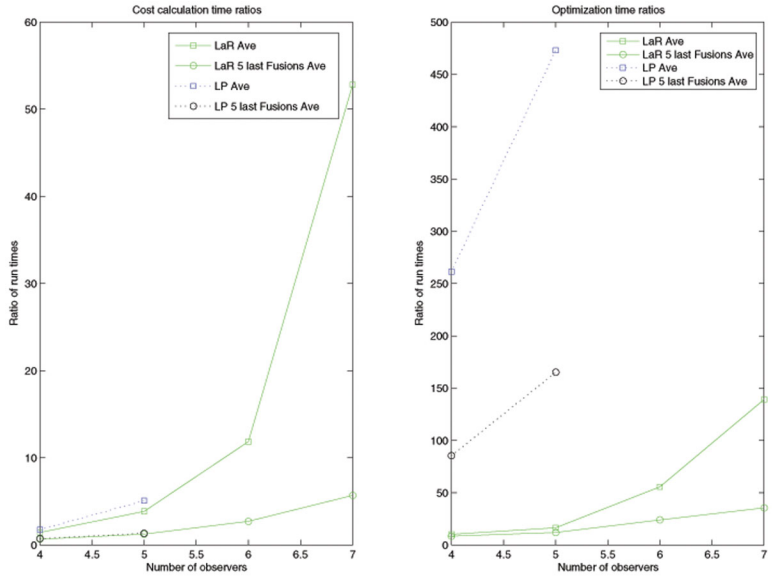


Fig. 10. Ratio of run times for a scenario with 7 observers, 4 networks and 40 launches, using $P_d = 0.7$ and $n_{ex} = 1$. The reference run time corresponds to the Sequential m-best 2D algorithm. There are 2 different measures of time, the average of the time ratios over the 10 fusion times, and the average of the time ratio over the last 5 fusion times, as from the previous two figures it can be seen that most of the time is spent in the first 5 fusion times, inflating the average.

dominates over the cost calculation time. This is not an uncommon situation for the case of having more than 4 lists, 40 elements per list and n_{ex} values below 10. In Figure 10 the run time ratios for the latter method is shown only for the case of having 4 and 5 lists, as for higher values the time taken by the LP algorithm is so large that its use is precluded for more than 5 observers.

7. SUMMARY AND CONCLUSIONS

This work compared three MDA algorithms when applied to a benchmark problem. A hierarchical scheme was developed to eliminate the duplicate reports from the same observer transmitted through N_n different networks, and then to select which reports from each of the N_s observers will be fused. This approach drastically reduces the dimensionality of the problem from $N_s \times N_n$ to N_s problems of dimension N_n and one of dimension N_s. This dimensionality reduction allows the use of algorithms like Lagrangean Relaxation and Linear Programming, which are infeasible for high density and high dimension problems due to run time limitations. The results for the comparison of different association criteria indicate a significant performance improvement when the Likelihood Ratio criterion is used vs. the NDS. Due to this, and the intrinsic 2D nature of the NDS cost criterion, this method is not used for the higher dimension problems presented. The Sequential m-best 2D assignment algorithm was found to be the fastest. However, its performance does not improve as the number m of best solutions kept at each stage is increased, and the cost does not necessarily improve with this increase in the number of best solutions kept. This behavior is mainly due to the "myopicity" of the approach, which also showed a tendency for preferring incomplete associations. The usage of a penalization term, P_d, for the cost calculation alleviates this drawback, by discouraging incomplete associations. The Linear Programming approach provides the lowest cost solutions, but they are not necessarily the best quality solutions (in terms of association accuracy), at a time expense similar to the Lagrangean Relaxation approach for cost calculation, but much higher for the optimization algorithm when the number of negative association costs is large. The LP method also has the drawback of giving non-integer solutions from time to time (less than 5% of the time a non-integer solution was observed). Both the LP and Sequential m-best 2D algorithms yield solutions that depend highly on the value of the number of extraneous targets (a parameter in the LR cost) n_{ex} used, giving better results for small n_{ex}, close to the number of false tracks. On the other hand, the Lagrangean Relaxation approach has proven to be the most robust method that works consistently for almost all parameter values.

Since, in general, the cost function parameters n_{ex} and P_d are not exactly known, the Lagrangean Relaxation algorithm is proposed as the algorithm to use as its robustness pays off for the run time increase when compared to the Sequential m-best 2D algorithm, and it exceeds the performance of the Linear Programming algorithm both in run time and solution quality when the number of lists grows above 4.

APPENDIX

A. Gating implementation

Given S lists with m_S event reports in each, we form an association tree which will contain all the

possible S-tuples. During the construction of the tree, a coarse gating strategy is used to discard unfeasible associations, and reduce the size of the tree. The tree constructed in this implementation is of depth S, with each level corresponding to a list.

The construction of the tree is recursive. Beginning at list 1, report 1 $(1,1)$, we continue with $(2,1)$ and so on until $(S,1)$. Each time a report is incorporated we test if it falls inside the gates defined by the previous reports in the tuple. Next we exhaust list S reports, fixing the $S-1$ previous reports, and generate $m_S - 1$ branches by selecting (S,i) for $i = 2,\ldots,m_S$. In the same way, fixing the first $S-2$ reports we exhaust list $S-1$, and at each value of $(S-1,j)$ all the values of (S,i) are also exhausted. We continue with the process until all the lists are incorporated.

There are two possible gating methods, one based on the NDS of Section 4.2.1, and other based on the individual coordinates, where the distance between scalar components of the report is compared to a threshold proportional to its standard deviation. In this way, if a report does not pass the gating test all the branches fanning from this node are discarded.

B. Inversion of the stacked covariance matrix

In equations (11) and (13), the LR cost calculation for association requires the inversion of the correlation matrix $\mathbf{P}_{\mathcal{T}_i}$ and the determinant of this inverse. The dimension of this matrix grows linearly with the number of tracks, so efficient ways of calculating this inverse are of great interest.

Using a permutation matrix as in [2] the vector of differences $\hat{\mathbf{x}}_{\mathcal{T}_i}$ can be transformed to a vector where the differences are among consecutive tracks

$$\hat{\mathbf{x}}_{\mathcal{T}_i} \triangleq \begin{bmatrix} \hat{x}_{t_{i_2},n_{i_2},s_{i_2}} - \hat{x}_{t_{i_1},n_{i_1},s_{i_1}} \\ \hat{x}_{t_{i_3},n_{i_3},s_{i_3}} - \hat{x}_{t_{i_2},n_{i_2},s_{i_2}} \\ \vdots \\ \hat{x}_{t_{i_q},n_{i_q},s_{i_q}} - \hat{x}_{t_{i_{q-1}},n_{i_{q-1}},s_{i_{q-1}}} \\ \hat{x}_{t_{i_1},n_{i_1},s_{i_1}} - \hat{x}_{t_{i_q},n_{i_q},s_{i_q}} \end{bmatrix}. \tag{28}$$

For the particular case of data association prior to fusion, the reports are uncorrelated (see Section 4.1.2), so the corresponding correlation matrix will be block tridiagonal, with block size n_x equal to the dimension of x_{t_i,n_i,s_i}. Following [8], the correlation matrix corresponding to q tracks can be written as

$$\mathbf{P}_{\mathcal{T}_i} = \begin{bmatrix} R_{11} & R_{12} & & & \\ R_{21} & R_{22} & R_{23} & & \\ & R_{32} & \ddots & \ddots & \\ & & \ddots & R_{q-2q-2} & R_{q-2q-1} \\ & & & R_{q-1q-2} & R_{q-1q-1} \end{bmatrix}$$

$$= \begin{bmatrix} D_1 & * & & & \\ E_1 & D_2 & * & & \\ & E_2 & \ddots & \ddots & \\ & & \ddots & D_{q-2} & * \\ & & & E_{q-2} & D_{q-1} \end{bmatrix} \tag{29}$$

and decomposed into block Cholesky factors

$$\mathbf{L}_{\mathcal{T}_i} = \begin{bmatrix} \tilde{D}_1 & & & & \\ \tilde{E}_1 & \tilde{D}_2 & & & \\ & \tilde{E}_2 & \ddots & & \\ & & \ddots & \tilde{D}_{q-2} & \\ & & & \tilde{E}_{q-2} & \tilde{D}_{q-1} \end{bmatrix} \tag{30}$$

where $\tilde{D}_i$, $i = 1,\ldots,q-1$ are lower triangular and

$$\begin{aligned} D_1 &= \tilde{D}_1\tilde{D}_1^T \\ E_i &= \tilde{E}_i\tilde{D}_i^{-T}, \qquad i = 1,\ldots,q-2 \\ D_i - \tilde{E}_i\tilde{E}_i^T &= \tilde{D}_i\tilde{D}_i^T, \qquad i = 2,\ldots,q-1. \end{aligned} \tag{31}$$

Then, an algorithm for overwriting the block tridiagonal matrix blocks with the corresponding block Cholesky factors is given by

$$\begin{aligned} &\textbf{for} \quad i = 1,\ldots,q-2 \\ &\quad D_i \leftarrow \hat{D}_i = \mathrm{Chol}(D_i); \qquad n_x^3/3 \text{ ops.} \\ &\quad E_i \leftarrow \hat{E}_i = E_i D_i^{-T}; \qquad n_x^3 \text{ ops.} \\ &\quad D_{i+1} \leftarrow D_{i+1} - E_i E_i^T; \qquad n_x^3 \text{ ops.} \\ &\textbf{end} \\ &D_{q-1} \quad \leftarrow \hat{D}_{q-1} = \mathrm{Chol}(D_{q-1}); \qquad n_x^3/3 \text{ ops.} \end{aligned} \tag{32}$$

Thus the total operation count of the algorithm is

$$\tfrac{1}{3}(q-1)n_x^3 + 2*(q-2)n_x^3 \approx \tfrac{7}{3}(q-1)n_x^3 \tag{33}$$

that is, the required number of operations for obtaining the Cholesky decomposition is linear in the number of blocks. The inverse of $\mathbf{P}_{\mathcal{T}_i}$ is not explicitly required, as it is used in a quadratic form problem

$$\begin{aligned} \hat{\mathbf{x}}_{\mathcal{T}_i}^T \mathbf{P}_{\mathcal{T}_i}^{-1} \hat{\mathbf{x}}_{\mathcal{T}_i} &= \hat{\mathbf{x}}_{\mathcal{T}_i}^T (\mathbf{L}_{\mathcal{T}_i}\mathbf{L}_{\mathcal{T}_i}^T)^{-1} \hat{\mathbf{x}}_{\mathcal{T}_i} \\ &= (\mathbf{L}_{\mathcal{T}_i}^{-1}\hat{\mathbf{x}}_{\mathcal{T}_i})^T (\mathbf{L}_{\mathcal{T}_i}^{-1}\hat{\mathbf{x}}_{\mathcal{T}_i}) \\ &= \mathbf{y}_{\mathcal{T}_i}^T \mathbf{y}_{\mathcal{T}_i} \end{aligned} \tag{34}$$

so back-substitution can be used to solve for $\mathbf{y}_{\mathcal{T}_i}$ in

$$\mathbf{L}_{\mathcal{T}_i}\mathbf{y}_{\mathcal{T}_i} = \hat{\mathbf{x}}_{\mathcal{T}_i} \tag{35}$$

requiring approximately $n_x(q-1)$ operations.

The determinant of $\mathbf{P}_{\mathcal{T}_i}^{-1}$ can be calculated as one over the determinant of $\mathbf{P}_{\mathcal{T}_i}$, and this is obtained from the diagonal of $\mathbf{L}_{\mathcal{T}_i}$ in just $(q-1)$ operations.

Overall, for the LR cost calculation for the selection of tracks to fuse the dominating term in the operation count is $7/3(q-1)n_x^3$, which, as previously said, is linear in the number of blocks.

For the case of the redundancy elimination, the correlation matrix is full, and thus the operation count corresponds to the Cholesky decomposition of the matrix, $((q-1)n_x)^3/3$, which is no longer linear in the number of blocks.

REFERENCES

[1] J. Areta, Y. Bar-Shalom and M. Levedahl
A hierarchical benchmark association problem in missile defense.
In *Proceedings SPIE Conference on Tracking Small Targets*, vol. 5913–38, San Diego, CA, Aug. 2005.

[2] Y. Bar-Shalom, S. S. Blackman and R. J. Fitzgerald
The dimensionless score function for measurement to track association.
IEEE Transactions on Aerospace and Electronic Systems, **43**, 1 (Jan. 2007), 392–400.

[3] Y. Bar-Shalom and H. Chen
Multisensor track-to-track association for tracks with dependent errors.
Journal of Advances in Information Fusion, **1**, 1 (July 2006), 3–17.

[4] Y. Bar-Shalom and X. R. Li
Multitarget-Multisensor Tracking: Principles and Techniques.
YBS Publishing, 1995.

[5] Y. Bar-Shalom, X. R. Li and T. Kirubarajan
Estimation with Applications to Tracking and Navigation: Theory, Algorithms and Software.
Wiley, 2001.

[6] M. Berkelaar, K. Eikland and P. Notebaert
lp_solve, an open source (mixed-integer) linear programming system.
Homepage: http://lpsolve.sourceforge.net/5.5/.

[7] D. Bertsekas
The auction algorithm: A distributed relaxation method for the assignment problem.
Annals of Operations Research: Special Issue on Parallel Optimization, **14** (1988), 105–123.

[8] T. Cao, J. Hall and R. van de Geijn
Parallel Cholesky factorization of a block tridiagonal matrix.
International Conference on Parallel Processing Workshops (ICPPW'02), 2002, 327–335.

[9] H. Chen, T. Kirubarajan and Y. Bar-Shalom
Performance limits of track-to-track fusion vs. centralized estimation.
IEEE Transactions on Aerospace and Electronic Systems, **39**, 2 (Apr. 2003), 386–400.

[10] M. Chummun, T. Kirubarajan, K. Pattipati and Y. Bar-Shalom
Fast data association using multidimensional assignment with clustering.
IEEE Transactions on Aerospace and Electronic Systems, **37**, 3 (July 2001), 898–913.

[11] I. Cox and M. Miller
On finding ranked assignments with application to multitarget tracking and motion.
IEEE Transactions on Aerospace and Electronic Systems, **31**, 1 (Jan. 1995), 486–489.

[12] L. Kaplan and W. Blair
Assignment costs for multiple sensor track-to-track association.
In *Proceedings of the 7th International Conference on Information Fusion*, Stockholm, Sweden, June 2004.

[13] L. M. Kaplan, W. Blair and Y. Bar-Shalom
Simulation studies of multisensor track association and fusion methods.
In *Proceedings of the 2006 IEEE/AIAA Aerospace Conference*, Big Sky, MT, Mar. 2006.

[14] X. Li, Z. Luo, K. Wong and E. Bossé
An interior point linear pogramming approach to two-scan data association.
IEEE Transactions on Aerospace and Electronic Systems, **35**, 2 (Apr. 1999), 474–490.

[15] M. L. Miller, H. S. Stone and I. J. Cox
Optimizing Murty's ranked assignment method.
IEEE Transactions on Aerospace and Electronic Systems, **33**, 3 (July 1997), 851–862.

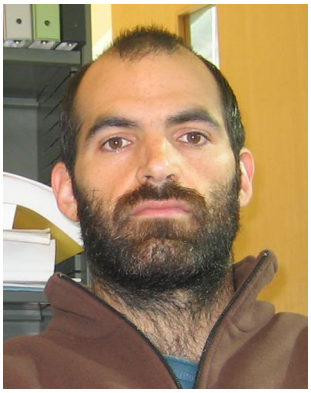
[16] K. Murty
An algorithm for ranking all the assignments in order of increasing cost.
Operations Research, **16** (1968), 682–687.

[17] K. R. Pattipati, S. Deb, Y. Bar-Shalom and R. B. Washburn
Passive Multisensor Data Association Using a New Relaxation Algorithm.
Multitarget-Multisensor Tracking: Advanced Applications, Artech House, 1990, ch. 7. Reprinted by YBS Publishing, 1998.

[18] K. R. Pattipati, R. Popp and T. Kirubarajan
Survey of assignment techniques for multitarget tracking.
In Y. Bar-Shalom and W. Dale Blair (Eds.), *Multitarget-Multisensor Tracking: Applications and Advances*, vol. III, Artech House, 2000.

[19] A. Poore and A. Robertson
A new class of Lagrangian relaxation based algortithms for a class of multidimensional assignment problems.
Computational Optimization and Applications, **8** (1997), 129–150.

[20] R. Popp, K. R. Pattipati and Y. Bar-Shalom
An m-Best 2-D assignment algorithm and multilevel parallelization.
Transactions on Aerospace and Electronic Systems, **35**, 4 (Oct. 1999), 1145–1160.



Javier Areta received the electrical engineer degree in 2001 from the University of La Plata, Argentina (UNLP). He is currently pursuing his Ph.D. in electrical and computer engineering at the University of Storrs, Connecticut.

His main interests lie in the areas of tracking, data fusion and signal processing.

Yaakov Bar-Shalom (S'63—M'66—SM'80—F'84) was born on May 11, 1941. He received the B.S. and M.S. degrees from the Technion, Israel Institute of Technology, in 1963 and 1967 and the Ph.D. degree from Princeton University, Princeton, NJ, in 1970, all in electrical engineering.

From 1970 to 1976 he was with Systems Control, Inc., Palo Alto, CA. Currently he is Board of Trustees Distinguished Professor in the Dept. of Electrical and Computer Engineering and Marianne E. Klewin Professor in Engineering. He is also director of the ESP Lab (Estimation and Signal Processing) at the University of Connecticut. His research interests are in estimation theory and stochastic adaptive control and he has published over 350 papers and book chapters in these areas. In view of the causality principle between the given name of a person (in this case, "(he) will track," in the modern version of the original language of the Bible) and the profession of this person, his interests have focused on tracking.

He coauthored the monograph *Tracking and Data Association* (Academic Press, 1988), the graduate text *Estimation with Applications to Tracking and Navigation* (Wiley, 2001), the text *Multitarget-Multisensor Tracking: Principles and Techniques* (YBS Publishing, 1995), and edited the books *Multitarget-Multisensor Tracking: Applications and Advances* (Artech House, Vol. I 1990; Vol. II 1992, Vol. III 2000). He has been elected Fellow of IEEE for "contributions to the theory of stochastic systems and of multitarget tracking." He has been consulting to numerous companies, and originated the series of Multitarget Tracking and Multisensor Data Fusion short courses offered at Government Laboratories, private companies, and overseas.

During 1976 and 1977 he served as associate editor of the *IEEE Transactions on Automatic Control* and from 1978 to 1981 as associate editor of *Automatica*. He was program chairman of the 1982 American Control Conference, general chairman of the 1985 ACC, and cochairman of the 1989 IEEE International Conference on Control and Applications. During 1983–1987 he served as chairman of the Conference Activities Board of the IEEE Control Systems Society and during 1987–1989 was a member of the Board of Governors of the IEEE CSS. Currently he is a member of the Board of Directors of the International Society of Information Fusion and served as its Y2K and Y2K2 President. In 1987 he received the IEEE CSS distinguished Member Award. Since 1995 he is a distinguished lecturer of the IEEE AESS. He is co-recipient of the M. Barry Carlton Awards for the best paper in the *IEEE Transactions on Aerospace and Electronic Systems* in 1995 and 2000, and received the 1998 University of Connecticut AAUP Excellence Award for Research and the 2002 J. Mignona Data Fusion Award from the DoD JDL Data Fusion Group.

Mark Levedahl B. Engineering (mechanical), McGill University, 1979.

Mr. Levedahl has over 25 years experience working in systems engineering, guidance, navigation, and control system development, battle management, multi-target multi-sensor tracking, and simulation. He has been employed by Raytheon since 1991 and working primarily on missile defense systems during that time. He is currently the Algorithm Team Leader for the Missile Defense National Team C2BMC program that is developing the integrated Command/Control and Battle Management system for world-wide ballistic missile defense. In this role, he has overall responsibility for the design and delivered performance of tracking and engagement management algorithms for the system. His work in missile defense has also included several interceptor systems including GBI-EKV, LEAP, MEADS, SM2-IVA programs, with focus on guidance methods, seeker requirements, and resulting system performance. In the 1980s, he was employed by Northrop designing and testing guidance, navigation, and control systems for unmanned air vehicles. Mr. Levedahl holds patents in Dynamic Wireless Utilization and in Sensor Data Association in the presence of biases, and is author or coauthor of a number of papers in multi-sensor tracking and systems interoperability. He is the primary author or a key contributor to a number of simulation systems with significant usage, including missile simulations at Northrop and Raytheon, and most recently the MDA sponsored BMD Benchmark used for missile defense.

Krishna R. Pattipati received the B.Tech. degree in electrical engineering with highest honors from the Indian Institute of Technology, Kharagpur, in 1975, and the M.S. and Ph.D. degrees in systems engineering from the University of Connecticut in 1977 and 1980, respectively.

From 1980–86 he was employed by ALPHATECH, Inc., Burlington, MA. Since 1986, he has been with the University of Connecticut, where he is a Professor of Electrical and Computer Engineering. His current research interests are in the areas of adaptive organizations for dynamic and uncertain environments, multi-user detection in wireless communications, signal processing and diagnosis techniques for complex system monitoring, and multi-object tracking. Dr. Pattipati has published over 300 articles, primarily in the application of systems theory and optimization (continuous and discrete) techniques to large-scale systems. He has served as a consultant to Alphatech, Inc. Aptima, Inc., and IBM Research and Development, and is a cofounder of Qualtech Systems, Inc., a small business specializing in intelligent diagnostic software tools.

Dr. Pattipati was selected by the IEEE Systems, Man, and Cybernetics Society as the Outstanding Young Engineer of 1984, and received the Centennial Key to the Future award. He was elected a Fellow of the IEEE in 1995 for his contributions to discrete-optimization algorithms for large-scale systems and team decision making. He has served as the Editor-in-Chief of the *IEEE Transactions on SMC: Part B—Cybernetics* during 1998–2001, vice-president for technical activities of the IEEE SMC Society (1998–1999), and as vice-president for conferences and meetings of the IEEE SMC Society (2000–2001). He was co-recipient of the Andrew P. Sage award for the Best SMC Transactions Paper for 1999, Barry Carlton award for the Best AES Transactions Paper for 2000, the 2002 NASA Space Act Award for "A Comprehensive Toolset for Model-based Health Monitoring and Diagnosis," the 2003 AAUP Research Excellence Award and the 2005 School of Engineering Teaching Excellence Award at the University of Connecticut. He also won the best technical paper awards at the 1985, 1990, 1994, 2002, 2004 and 2005 IEEE AUTOTEST Conferences, and at the 1997 and 2004 Command and Control Conferences.