A Distributed Multiple-Target Identity Management Algorithm in Sensor Networks

Inseok Hwang*, Kaushik Roy†, Hamsa Balakrishnan*, and Claire Tomlin*
*Dept. of Aeronautics and Astronautics, Stanford University, CA 94305
† Dept. of Electrical Engineering, Stanford University, CA 94305
ishwang, kroy1, hamsa, tomlin@stanford.edu

***Abstract*— In this paper, we develop a distributed identity management algorithm for multiple targets in sensor networks. Each sensor is assumed to have the capability of managing identities of multiple targets within its surveillance region and of communicating with its neighboring sensors. We use the algorithm from our companion paper [1] to incorporate local information about the identity of a target when it is available to a local sensor and at the same time reduces the uncertainty of the target's identity as measured by entropy. Identity information fusion is crucial for distributed identity management to compute the global information of the system from information provided by local sensors. We formulate this problem as an optimization problem and present three different cost functions, namely, Shannon information, Chernoff information, and the sum of Kullback-Leibler distances, which represent different performance criteria. Using Bayesian analysis, we derive a data fusion algorithm that needs *a prior* probability of the given data. Finally, we demonstrate the performance of the distributed identity management algorithm using scenarios from multiple-aircraft tracking in a sensor (radar) network with different fusion criteria.**

I. Introduction

The last few decades have seen many advances in wireless communication techniques and in sensor technology. These advances, combined with growing interest in both military and civilian applications in using distributed sensors, have led to the concept of a sensor network. These applications include battlefield surveillance and enemy tracking in military applications, and habitat monitoring, environment observation, and traffic surveillance in civilian applications ([2], [3] and references therein). A sensor network is a network of sensor nodes which have local sensing, processing, and communication capabilities. Many applications of sensor networks, such as target tracking and habitat monitoring using only local information (*i.e.*, information obtained by each sensor), have a unique problem that does not arise in a centralized network: scalable distributed information fusion. This implies that the global states of the system must be estimated and maintained using only local information available to local sensors. In this paper, we present a Distributed Multiple-Target Identity Management (DMIM) algorithm which can estimate the identities of multiple maneuvering targets in sensor networks.

For DMIM in sensor networks, information about the identity of a target may become available to a local sensor, and thus we need methods which can incorporate this new information to reduce the uncertainty of the system. For the case in which the number of targets is constant, the Sinkhorn algorithm [4] is used in [5], [6], [7]. However, in distributed sensor networks, the number of targets in the surveillance region of each sensor may change over time and the Sinkhorn algorithm may not converge for this case. In [1], we have developed an algorithm which can solve this problem in polynomial time and in this paper, we use this algorithm for local information incorporation. A crucial part of the DMIM algorithm is distributed information fusion. In distributed sensor networks, identity information of multiple targets is maintained by individual sensors and each sensor can manage only identities of the targets within its surveillance region. Thus, each sensor has only a knowledge about its neighborhood, not the global picture of the whole system. To get the global information from information maintained by individual local sensors, we need an information fusion algorithm. To fully exploit the capability of sensor networks, this algorithm should be *scalable*, *i.e.*, adding/deleting sensors or targets into a sensor network can be handled efficiently, and *distributed*, *i.e.*, the algorithm can be implemented in individual sensors. We formulate the information fusion problem as an optimization problem and propose three different cost functions: Shannon information, Chernoff information, and the sum of Kullback-Leibler distances to represent different performance criteria. Using Bayesian analysis, we also derive an information fusion algorithm that needs *a prior* probability of the given data. Finally, we apply the DMIM algorithm to multiple-aircraft tracking problems in sensor networks and demonstrate the performance of the proposed information fusion algorithms under different scenarios.

This paper is organized as follows: In Section II, the Distributed Multiple-Target Identity Management (DMIM) algorithm including local information incorporation and belief information fusion is presented. Section III presents applications of the DMIM algorithm to multiple-aircraft tracking in sensor networks. Finally, our conclusions are presented in Section IV.

II. Distributed Multiple-Target Identity Management (DMIM)

In this section, we consider the problem of managing identities of multiple targets in sensor networks. Each sensor

is assumed to have its own surveillance region, and to communicate with its neighboring sensors. A two-sensor example is shown in Figure 1 in which the circles represent the surveillance regions of the sensors. We assume that each sensor has the capability to compute the position estimates and manage the identity of targets within its surveillance region. For distributed identity management, we have to consider the possibility that the number of targets within the surveillance region of a sensor could change over time. For example, a target might leave or enter the surveillance region of a sensor. Another important problem that has to be addressed is scalable and distributed information fusion to get the global estimate of the system from information computed by individual local sensors. These problems are unique for distributed sensor network applications. Therefore, we propose a scalable Distributed Multiple-Target Identity Management (DMIM) algorithm that can manage multiple-target identities efficiently in a distributed sensor network environment. The structure of DMIM is shown in Figure 2: let us start with the identity management algorithm.

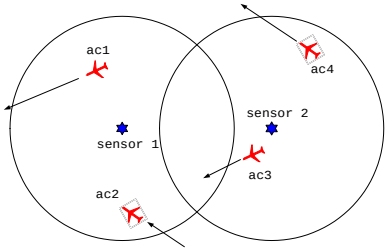


Fig. 1. A distributed multiple-target identity management scenario for a two-sensor network.

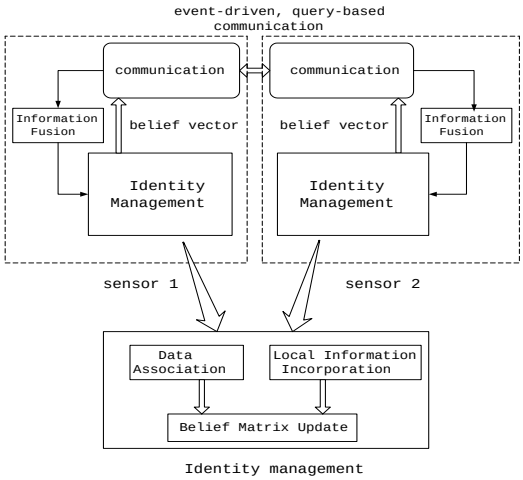


Fig. 2. The structure of the Distributed Multiple-Target Identity Management (DMIM) algorithm for a two-sensor example.

A. Data Association

Suppose there are T targets and T identities, for example, T aircraft with identities {piper, cherokee, cessna, $\cdots$ }, in the surveillance region of sensor i. Then, the problem of managing identities of multiple targets is to match each target to its identity over time. For this, we use the idea of the Identity-Mass-Flow in [5]. The idea of the Identity-Mass-Flow is that an identity is treated as a unit mass assigned to a target. These masses cannot be destroyed or created, and flow from a target into another through the *mixing matrix*, $M(k)$ at time k. The mixing matrix is an $T \times T$ matrix whose element $M_{ij}(k)$ represents the probability that target i at time $k-1$ has become target j at time k. Thus, the mixing matrix is a doubly stochastic matrix; that is, its column sums and row sums are equal to 1. The output of the *Data Association* block is the mixing matrix.

B. Belief Matrix Update

We use a *belief vector* to represent the identity of a target probabilistically. For multiple targets, we have a *belief matrix* $B(k)$ whose columns are belief vectors of the targets. Thus, entry $B_{ij}(k)$ represents the probability that target j can be identified as label (or name) i at time k. The *Belief Matrix Update* block maintains identity information stored in a $T \times T$ belief matrix $B(k)$ over time. The evolution of this belief matrix is governed by the equation [5]:

$$B(k) = B(k-1)M(k) \tag{1}$$

We can show that (1) keeps row and column sums of the belief matrix constant when the numbers of targets and identities are the same. However, this is not the case for distributed identity management since the number of the targets within the surveillance region of individual sensors may change over time. There are two possible cases: a target leaves or enters the surveillance region of a sensor. When a target leaves, we delete the corresponding column in the belief matrix managed by the sensor. When a target enters the surveillance region of a sensor, there are two possible cases: (i) the target comes from the surveillance region of another sensor, which may be queried, or (ii) the target comes from the outside of the surveillance region of a sensor network. For these cases, we propose a scalable, event-driven, query-based belief matrix update algorithm:

Algorithm 1: Event-driven, query-based Belief Matrix Update

- For sensor i and target t

if target t leaves the surveillance region of sensor i. **then**
  delete the corresponding column in the belief matrix.
end if
if a target enters the surveillance region of sensor i. **then**
  send a query about the identity of target t.
  if there is an answer "yes" and receive the belief vector of target t, **then**
    augment the belief matrix with the belief vector received.
  else
    augment the belief matrix with a belief vector with a new identity assigned to the target.
  end if
end if

For distributed identity management, a belief matrix managed by each sensor may not be a square matrix but might more likely be a skinny matrix which has more rows than columns. The belief matrix may not be a doubly stochastic matrix, but it should be a stochastic matrix with column sums equal to one. Its row sums remain constant because an identity mass cannot be destroyed or created. It also has the property that the sum of column sums is equal to the sum of row sums; that is, even though the number of targets in the surveillance region of each sensor changes, the identity mass is conserved in the surveillance region. Since the evolution of the belief matrix is governed by (1), these characteristics of the belief matrix are preserved over time.

C. Local Information Incorporation

In this section, we consider the case in which *local information* about the identity of a target is available to a local sensor. The local information has the form of a belief vector and when available, we use the local information to decrease the uncertainty of the belief matrix measured by entropy. The entropy of a $L \times T$ belief matrix is defined as $H(B(k)) \triangleq -\sum_{i=1}^{L}\sum_{j=1}^{T} B_{ij}(k)\log B_{ij}(k)$. Then, the problem is how to incorporate this information to the belief matrix. From the idea of the Identity-Mass-Flow and the characteristics of (1), we know that the belief matrix should have the following properties: its column sums are equal to one, its row sums remain constant, and the sum of row sums and the sum of column sums are equal. However, if we replace the column in the belief matrix with the local information, it is not guaranteed that the new belief matrix has the above properties. Thus, we have developed a polynomial-time algorithm that can check whether the available local information can be incorporated (*i.e.,* the new belief matrix is scalable or *almost* scalable) and if so, can make the new belief matrix have the above properties. We refer the reader to our companion paper [1] for the details. The local information incorporated may not necessarily decrease the uncertainty (entropy) of the belief matrix. Therefore, local information is incorporated only when it reduces the uncertainty of the belief matrix. The local information incorporation algorithm can be described as follows:

Algorithm 2: Local Information Incorporation

- **Given:** local information (belief vector) of a target and a belief matrix $B(k)$.
- Make a matrix $B'(k)$ by replacing the column corresponding to the target in $B(k)$ with the local information.

if $B'(k)$ is almost scalable **then**
  $B_{new}(k) := \mathsf{S}(B'(k))$
  if $H(B_{new}(k)) \leq H(B(k))$ **then**
    $B(k) := B_{new}(k)$
  else
    $B(k) := B(k)$
  end if
else
  $B(k) := B(k)$
end if

where is the operator S represents the matrix scaling process in [1].

D. Belief Information Fusion

In this section, we consider the problem of combining two belief vectors of the same target from two different sensors. Information fusion can be formulated as an optimization problem such that the fused information is the one that minimizes a cost function which represents a performance criterion. For optimization, we propose three different cost functions: Shannon information, Chernoff information, and the sum of Kullback-Leibler distances.

Shannon information*:* The Shannon information is defined as

$$H(b') = \sum_{i=1}^{n} -b'(i)\log b'(i) \tag{2}$$

where $b' \in [0,1]^n$ with $\sum_i b'(i) = 1$. The Shannon information (also known as *entropy*) is a measure of the uncertainty of a system. Thus, the minimization of the Shannon information selects a belief vector that is most informative in the sense of minimum entropy. Suppose b_1 and b_2 are belief vectors of target t computed by sensor 1 and sensor 2 respectively. Since the most common data fusion algorithms compute a *linear combination* of two data, we propose the following fusion strategy:

$$b' = \omega b_1 + (1-\omega)b_2 \tag{3}$$

where $\omega \in [0,1]$, $b_i \in [0,1]^n$ with $\sum_{j=1}^{n} b_i(j) = 1$ for $i \in \{1,2\}$, and $\sum_{j=1}^{n} b'(j) = 1$. Then, the problem of computing the fused belief vector becomes a problem to find a weight, ω, which minimizes the cost function in (2). If we use the fusion strategy in (3), the Shannon information of the new fused information is

$$H(b') = H(\omega b_1 + (1-\omega)b_2) \geq \omega H(b_1) + (1-\omega)H(b_2) \tag{4}$$

From (4), we can see that the minimum is always achieved at either $\omega = 0$ or $\omega = 1$. This means that a fused belief vector that has the minimum Shannon information is either of the two given belief vectors, which is a *hard* choice. For some applications such as identity management in this paper, the hard choice may not be desirable since it ignores one possibility completely and thus might quickly lead to a wrong answer over time if not immediately. Thus, we propose a *soft* choice method which has $\omega \in (0,1)$ for almost all cases. Motivated by the fact that Shannon information minimization chooses a belief vector which has the minimum entropy, we propose to use the inverse of the Shannon information of a belief vector as a weight. Thus, we put large confidence on a belief vector which has small Shannon information. Then, a new belief vector $b' = [b'(i)]$ is

$$b'(i) = \frac{H(b_1)^{-1}b_1(i)}{H(b_1)^{-1} + H(b_2)^{-1}} + \frac{H(b_2)^{-1}b_2(i)}{H(b_1)^{-1} + H(b_2)^{-1}} \tag{5}$$

From (3) and (5), we get

$$\omega = \frac{H(b_1)^{-1}}{H(b_1)^{-1} + H(b_2)^{-1}} = \frac{H(b_2)}{H(b_1) + H(b_2)} \quad (6)$$

When $H(b_1) = H(b_2) = 0$, we set $\omega = \frac{1}{2}$. $\omega = 0$ if $H(b_2) = 0$ (no uncertainty in b_2) and $\omega = 1$ when $H(b_1) = 0$ (no uncertainty in b_1). In these cases, the fused belief vector computed by the proposed fusion algorithm is a belief vector which has no uncertainty. This fusion algorithm is a soft choice method since the fused data is a convex combination of the two given data with a larger weight on the data which has smaller entropy than the other. From (4) and (6), the Shannon information of the new belief $H(b')$ has the property that:

$$H(b') \geq \frac{2H(b_1)H(b_1)}{H(b_1) + H(b_2)} \; or \; 2H(b')^{-1} \leq H(b_1)^{-1} + H(b_2)^{-1} \quad (7)$$

Inequality (7) tells us that the achievable minimum uncertainty of the fused belief vector with the fusion strategy in (3) with (6) as a weight is under-bounded by uncertainties of the given information. In other words, the maximum achievable certainty (inverse of the Shannon information) is upper-bounded by the arithmetic mean of the inverse of the Shannon information of the given belief vectors. If we use the fusion strategy in (3), we can also derive the upper bound of the Shannon information of the new belief vector:

$$\begin{aligned} H(b') \quad \leq \quad & \omega^2 H(b_1) + (1-\omega)^2 H(b_2) + \omega(1-\omega)(H(b_1) \\ & + H(b_2) + D(b_1 \parallel b_2) + D(b_2 \parallel b_1)) \end{aligned} \quad (8)$$

where $D(p \parallel q) \triangleq \sum_i p(i) \log(\frac{p(i)}{q(i)})$ is the Kullback-Leibler distance [8]. If we use ω in (6), then

$$H(b') \leq \frac{2H(b_1)H(b_1)}{H(b_1) + H(b_2)} + \frac{H(b_1)H(b_2)[D(b_1 \parallel b_2) + D(b_2 \parallel b_1)]}{(H(b_1) + H(b_2))^2} \quad (9)$$

Thus, we can analytically compute the upper and lower bounds of the Shannon information of the new belief vector using the fusion strategy in (3) with (6). Thus, the Shannon information cost function would be useful when we have good knowledge about the performance and/or fidelity of each sensor, since we can get a solution which has lower entropy by weighing information that has smaller entropy more than the other. However, if we do not have such knowledge, we may get a biased solution by consistently putting more confidence on one piece of information (possibly the wrong one) than the other.

***Chernoff information**:* The Chernoff information is defined as

$$C(b_1, b_2) = -\min_{0 \leq \omega \leq 1} \log(\sum_{i=1}^{n} b_1(i)^{\omega} b_2(i)^{1-\omega}) \quad (10)$$

If ω^* minimizes (10), the new belief vector b' ($= [b'(i)]$ for $i = \{1, 2, \cdots, n\}$) is

$$b'(i) = \frac{b_1(i)^{\omega^*} b_2(i)^{1-\omega^*}}{\sum_{j=1}^{n} b_1(j)^{\omega^*} b_2(j)^{1-\omega^*}} \quad (11)$$

The new belief vector in (11) satisfies ([8], [9])

$$D(b' \parallel b_1) = D(b' \parallel b_2) \quad (12)$$

This fusion strategy is different from that in (3) which is a convex combination of the two data. From (12), the minimization of the Chernoff information is equivalent to finding a function that is in the *middle* of the two original functions, where the middle is defined in terms of the Kullback-Leibler distance. In other words, Chernoff information minimization could be interpreted as selecting a probability vector which is "equally close" in terms of the Kullback-Leibler distance to the original probability vectors. This fusion algorithm does not put more confidence on one than the other. Thus, this cost function could be useful when we do not know the quality of information obtained from individual sensors; by choosing the middle point of the two pieces of information, we could minimize the bias over time. However, the fused belief vector computed by the Chernoff information minimization algorithm may have larger entropy than that computed by the algorithm in (3) with (6).

***Sum of the Kullback-Leibler distances**:* Since the Kullback-Leibler distance is not symmetric, we consider two possible optimization problems:

$$\begin{array}{ll} \text{minimize} & D(b' \parallel b_1) + D(b' \parallel b_2) \\ \text{subject to} & \sum_{j=1}^{n} b'(j) = 1 \\ & b'(j) \geq 0 \end{array} \quad (13)$$

$$\begin{array}{ll} \text{minimize} & D(b_1 \parallel b') + D(b_2 \parallel b') \\ \text{subject to} & \sum_{j=1}^{n} b'(j) = 1 \\ & b'(j) \geq 0 \end{array} \quad (14)$$

where $b'(j)$ is the jth element of a vector b'. Let us first consider the optimization problem in (13). The Lagrangian is given by

$$L(b', \lambda) = D(b' \parallel b_1) + D(b' \parallel b_2) + \lambda(\sum_{j=1}^{n} b'(j) - 1) \quad (15)$$

To get an optimal solution, we set the derivatives of L with respect to $b'(i)$ and to λ to be equal to zero. Then, we get a new belief vector:

$$b'(i) = \frac{\sqrt{b_1(i)b_2(i)}}{\sum_{j=1}^{n} \sqrt{b_1(j)b_2(j)}} \quad (16)$$

From (16), we see that the fused data is a geometric mean of the given data. The fused data is the same as that in (11) for Chernoff information minimization when $\omega^* = \frac{1}{2}$. Thus, this data fusion strategy can be interpreted as a special case of the Chernoff information minimization method.

Now, let us consider the optimization problem in (14). The Lagrangian is given by

$$L(b', \lambda) = D(b_1 \parallel b') + D(b_2 \parallel b') + \lambda(\sum_{j=1}^{n} b'(j) - 1) \quad (17)$$

Similarly, we get an optimal solution:

$$b'(i) = \frac{b_1(i) + b_2(i)}{\sum_{j=1}^{n} [b_1(j) + b_2(j)]} = \frac{b_1(i) + b_2(i)}{2} \quad (18)$$

In this case, the fused data is the arithmetic mean of the given data. This fusion strategy is the same as that in (3)

when $\omega = \frac{1}{2}$. Thus, from (4) and (8), we get the lower and upper bounds of Shannon information of the new belief vector:

$$\begin{array}{rcl} H(b') & \geq & \frac{H(b_1)+H(b_2)}{2} \\ H(b') & \leq & \frac{H(b_1)+H(b_2)}{2} + \frac{D(b_1\|b_2)+D(b_2\|b_1)}{4} \end{array} \quad (19)$$

Therefore, the fusion algorithms obtained by solving the optimization problems in (13) or (14) are to average the given data either geometrically or arithmetically. This is similar to Chernoff information minimization and thus these fusion strategies would be useful when we want to get unbiased fused data in situations where we do not have good *a prior* information about a system. An example would be a case in which information from one sensor is wrong due to failure of the sensor or the malicious intent of the sensor that is unknown *a priori*. These information fusion strategies would be robust to this wrong information since they do not put more confidence on one (possibly incorrect information) than the other, but average them to compute a fused belief vector.

***Bayesian approach**:* In this section, we derive a fused belief vector using a Bayesian approach. Suppose the target's identity $\theta \in \{1, 2, \cdots, N\}$ and without loss of generality, suppose there are two sensors. Denote events X_1 and X_2 to be observations at sensor 1 and sensor 2 respectively. We are assumed to be given information $b_1(\theta) \triangleq P(\theta|X_1)$ from sensor 1 and $b_2(\theta) \triangleq P(\theta|X_1)$ from sensor 2 where $P(\cdot|\cdot)$ is a conditional probability. Then, the problem of information fusion is to find the *a posteriori* probability $P(\theta|X_1, X_2)$. We assume $P(X_1, X_2|\theta) = P(X_1|\theta)P(X_2|\theta)$ since given the identity of a target, the events that it is observed by sensor 1 or sensor 2 are independent in distributed identity management. Using the Bayes rule, we get

$$P(\theta|X_1, X_2) = \frac{P(X_1|\theta)P(X_2|\theta)P(\theta)}{P(X_1, X_2)} \quad (20)$$

Since $P(\theta|X_i) = \frac{P(X_i|\theta)P(\theta)}{P(X_i)}$ for $i \in \{1, 2\}$, we obtain

$$P(\theta|X_1, X_2) = \frac{b_1(\theta)b_2(\theta)}{P(\theta)} \frac{P(X_1)P(X_2)}{P(X_1, X_2)} \quad (21)$$

Therefore, a fused belief vector is

$$b'(\hat{\theta}) = \arg\max_{\theta} P(\theta|X_1, X_2) = \frac{b_1(\theta)b_2(\theta)}{P(\theta)} \cdot \frac{1}{c} \quad (22)$$

where c is a normalization constant. This is an interesting result because the fused data does depend only on the given data ($b_1(\theta)$, $b_2(\theta)$) and the *a priori* probability $P(\theta)$. Thus, if we knew *a priori* information, we could compute the *a posteriori* probability (*i.e.*, the fused data). However, since we may not know the *a priori* probability for some applications such as distributed identity management in this paper, we cannot compute the fused data from (22). In order to compute the fused data for this case, we have to assume $P(\theta)$ either from the characteristics of the systems or from that of applications. For example, due to the lack of information about the system, we assume that the *a priori* probability is a geometric mean of the given data ($P(\theta) = \sqrt{b_1(\theta)b_2(\theta)}$). Then, we can get exactly the same result as that in (16) which minimizes the sum of Kullback-Leibler distances to the original data in (13). Thus, the *a posteriori* probability is the same as the *a priori* probability; that is, we cannot extract any information from the given data. From Bayesian analysis, we can see that the data fusion strategies such as Chernoff information minimization and the minimization of the sum of Kullback-Leibler distances in (13) compute the solution in a similar form to the solution produced by the Bayesian approach.

III. Simulations

Several simulations are presented in this section to highlight the performance and capabilities of the DMIM algorithm. Specifically, the scenarios consist of stationary sensors (*e.g.*, air traffic control radars) tracking multiple aircraft through two-dimensional space. Individual sensors are assumed to have the capability to compute the position estimates of aircraft. Measurements are available to sensors when inside a sensing radius, set to 10 km, while two sensors can communicate if inside the communication radius, set to 20 km.

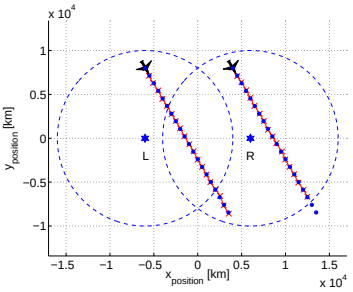
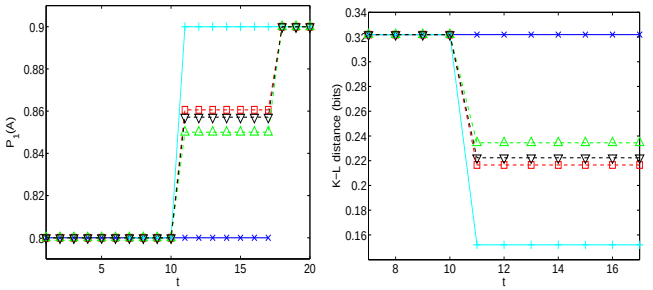


Fig. 3. Aircraft trajectories for two-aircraft, two-sensor scenario.



(a) (b)

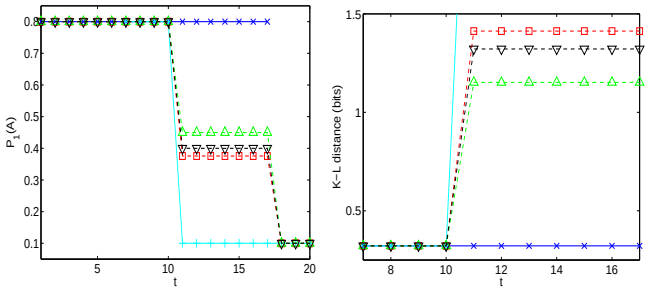
Fig. 4. A cooperative two-sensor scenario. (a) Belief of aircraft 1 having identity A. (b) Efficiency of belief estimates as measured in bits. (Solid lines are local estimates and dashed lines are global estimates. Symbols 'x' and '+' are used for sensor L and R respectively. Symbols △, ▽, and □ are used for the identity fusion algorithms using the sum of Kullback-Leibler distances between the local estimates and the global estimate (corresponding to an arithmetic mean), the sum of Kullback-Leibler distances between the global estimate and the local estimates (corresponding to a geometric mean), and Shannon information, respectively.)

The first two simulations involve a system of two sensors, denoted sensor L for left and R for right, observing two

aircraft, denoted 1 and 2. Aircraft 1 starts in sensor L's surveillance region, while aircraft 2 starts in R's surveillance region, as shown in Figure 3. This figure shows the true positions of the aircraft (solid lines and x's) and the global position estimates made by the sensors (dashed lines and o's). Both aircraft travel southeast with velocity 200 m/s. The true identity of aircraft 1 and 2 are A and B respectively, but simulation is initialized with belief vectors $[0.8\ 0.2]^T$ for aircraft 1 and $[0.2\ 0.8]^T$ for aircraft 2 respectively. $[0.8\ 0.2]^T$ means that aircraft 1 is thought to have identity A with 80% probability and B with 20% probability. Three times correspond to important events in the simulation. At time 7, aircraft 1 enters R's surveillance region, allowing global belief estimates based on fusion techniques discussed in the previous section and sensor R receives the identity information about aircraft 1 from sensor L using Algorithm 1. At time 11, local information is obtained by sensor R. Aircraft 2 is now thought to have belief $[0.1\ 0.9]^T$. Since this local information makes a new belief matrix scalable and decreases entropy (uncertainty) of the belief matrix, we replace the belief vector of aircraft 2 in the belief matrix with this local information. Since row sums of the belief matrix should remain constant, the belief vector for aircraft 1 in the new belief matrix becomes $[0.9\ 0.1]^T$. This local information incorporation is performed using Algorithm 2. Finally, at time 18, aircraft 1 leaves sensor L's surveillance region.

Because the two aircraft must share identity A and B, an estimate of the probability of aircraft 1 having identity A determines fully the belief matrix for the system. Thus, only $P_1(A)$ (probability that aircraft 1 has identity A) is plotted in Figure 4-(a), which presents belief information according to different estimators. Solid lines indicate $P_1(A)$ for each observer, while the various dashed lines indicate $P_1(A)$ for three global estimate methods discussed in the previous section. The three methods are using Shannon information, minimizing the sum of Kullback-Leibler distances between the local estimates and the global estimate (corresponding to an arithmetic mean of the local beliefs), and minimizing the sum of Kullback-Leibler distances between the global estimate and the local estimates (corresponding to a geometric mean of the local beliefs). The Bayesian approach and the Chernoff approach have a similar form to the geometric mean and are thus not plotted.

Figure 4-(b) presents the efficiency of each method of estimating the belief of aircraft 1. This figure only covers times 7 to 17, since these are the times when the information fusion algorithms are used. We use the Kullback-Leibler distance between the correct belief vector and the estimated belief vector (*i.e.*, $D(b_{true} \parallel b_{est})$) as a performance measure since this Kullback-Leibler distance measures the inefficiency of estimators [8], [10], [11]. As shown in the figure, the inefficiency in sensor L's belief of aircraft 1 is constant, since it always has belief $P_1(A) = 0.8$. Sensor R's inefficiency drops at time 11 when local information is incorporated. The various global estimates lie between the two local belief estimates. For this scenario, global belief estimates are best obtained through Shannon information mixing in (3) with (6) as expected.



(a) (b)

Fig. 5. A malicious two-sensor scenario. (a) Belief of aircraft 1 having identity A. (b) Efficiency of belief estimates as measured in bits. (Solid lines are local estimates and dashed lines are global estimates. Symbols 'x' and '+' are used for sensor L and R respectively. Symbols △, ▽, and □ are as in Figure 4.)

The second scenario presents the same aircraft and trajectory, but sensor R, through error or malice, reverses the local information received at time 11. That is, it believes $P_1(A) = 0.1$, rather than $P_1(A) = 0.9$. The trajectory plot is exactly the same as in Figure 3, but belief estimates $P_1(A)$ are now those shown in Figure 5-(a). The inefficiency of the belief estimate is shown in Figure 5-(b). As in the previous scenario, sensor L's belief of aircraft 1 and thus its inefficiency in that belief remain constant. However, sensor R receives incorrect local information, thereby increasing its inefficiency to 3.32 bits, above the scale of Figure 5-(b). The global belief estimates again fall somewhere in between. However, because sensor R contributes incorrect information, the global belief estimate with the least inefficiency is now the arithmetic average of the local estimates in (18), while the method based on the Shannon information results in the highest inefficiency. That is, the arithmetic average method is most robust to incorrect information. To analyze the performance of the various

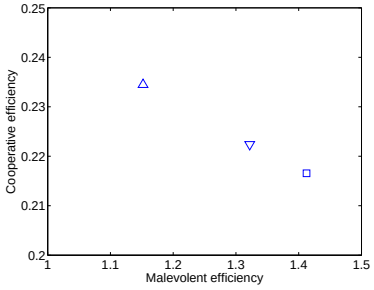


Fig. 6. Cooperative efficiency versus malicious efficiency for each global belief estimate, for the two scenarios presented above. Symbols △, ▽, and □ are as in Figure 4.)

belief fusion methods, we present Figure 6 that plots the ordered pairs (D_{coop},D_{mal}) for each method of global belief fusion. The quantities D_{coop} and D_{mal} refer respectively to the efficiency of the global estimates for the cooperative

and malicious scenarios and are taken from Figures 4-(b) and 5-(b). A point closer to the origin in both axes is more efficient overall than another point. From Figure 6, one can argue that using Shannon information is more efficient for scenarios in which all sensors are known to be cooperative, while arithmetic combination is more efficient for scenarios in which there is a high probability of malicious sensors. However, this conclusion depends on the configuration of sensor networks: the number of malicious sensors and their locations.

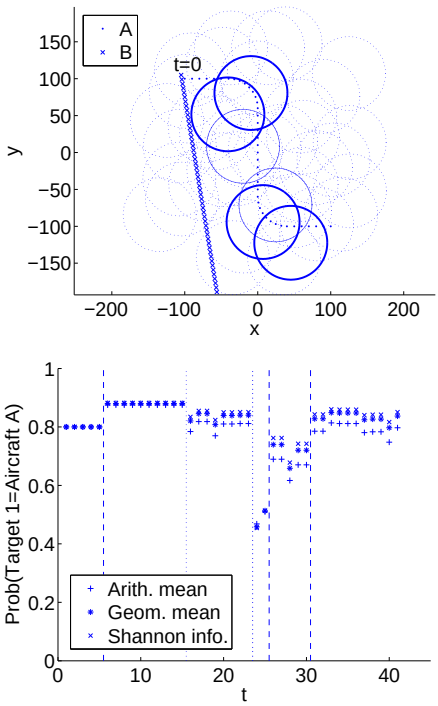


Fig. 7. A general sensor network scenario. (Top) Aircraft trajectories for two-aircraft, 41-sensor scenario. (Bottom) Global belief estimates of aircraft 1 having identity A.

Figure 7 shows a complex scenario involving many sensors. It is assumed that sensors that can see the targets are all initialized to the same belief vectors. Target 1 is initialized to $[0.8\ 0.2]^T$ and target 2 to $[0.2\ 0.8]^T$. Figure 7-(Top) shows the track of each target and the coverage areas of the 41 sensors. Sensors with solid lines are malicious, meaning their local estimates are $[\beta\ \alpha]^T$ when $[\alpha\ \beta]^T$ is sensed. Sensors with thick solid lines gather local information; this information is always that target 1 is aircraft A with probability 0.9. Target 2 exists to create confusion at the start of the scenario; however, identity information is not presented for this target. For target 1, local estimates are made at each time step by those sensors that can observe the target. The global estimates are made at each time step by combining the local identity estimates. In Figure 7-(Bottom), the global estimates, using each of the three methods described in the previous section, are plotted over time. Initially, all sensors have local belief $[0.8\ 0.2]^T$, yielding the same global estimate. At times 6, 26, and 31, sensors with local information start observing target 1, leading to improved global estimates; these events are noted by dashed lines in Figure 7-(Bottom). At times 16 and 24, malicious sensors begin observing target 1, leading to degraded global estimation, as noted by dotted lines in the figure. This scenario exhibits the scalability of the DMIM algorithm applied to a large set of sensors tracking the identity of a maneuvering target.

IV. Conclusions

We have developed a scalable Distributed Multiple-Target Identity Management (DMIM) algorithm which can manage identities of multiple maneuvering targets in sensor networks and efficiently incorporate local information about the identity of a target, when available, to reduce the uncertainty of the system. For identity fusion to obtain global information using local sensor information, we have formulated an optimization problem and have presented three different cost functions: Shannon information, Chernoff information, and the sum of Kullback-Leibler distances, which represent different performance criteria that could be useful for different applications. Using Bayesian analysis, we have also derived an information fusion algorithm that needs *a priori* probability of the given data. Finally, we have applied the DMIM algorithm to the problem of managing identities of multiple aircraft in sensor networks and demonstrated the performance of the proposed fusion algorithms.

References

[1] H. Balakrishnan, I. Hwang, and C. Tomlin. Polynomial approximation algorithms for belief matrix maintenance in identity management. In *Proceedings of the 43nd IEEE Conference on Decision and Control*, Atlantis, Paradise Island, Bahamas, December 2004.

[2] I.F. Akyidliz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. A survey on sensor networks. *IEEE Communications Magazine*, 40(8):102–114, August 2002.

[3] Ning Xu. A survey of sensor network applications. http://enl.usc.edu/ ningxu/papers, 2003.

[4] R. Sinkhorn. Diagonal equivalence to matrices with prescribed row and column sums. *American Mathematical Monthly*, 74:402–405, 1967.

[5] J. Shin, L.J. Guibas, and F. Zhao. A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks. In F. Zhao and L. Guibas, editors, *Information Processing in Sensor Networks*, Lecture Notes in Computer Science 2654, pages 223–238, Palo Alto, CA, April 2003.

[6] I. Hwang, H. Balakrishnan, K. Roy, J. Shin, L. Guibas, and C. Tomlin. Multiple-target Tracking and Identity Management algorithm for Air Traffic Control. In *Proceedings of the Second IEEE International Conference on Sensors*, Toronto, Canada, October 2003.

[7] I. Hwang, H. Balakrishnan, K. Roy, and C. Tomlin. Multiple-target Tracking and Identity Management algorithm in clutter, with application to aircraft tracking. In *Proceedings of the AACC American Control Conference*, Boston, MA, June 2004.

[8] T.M. Cover and J.A. Thomas. *Elements of Information Theory*. Wiley-Interscience, New York, 1991.

[9] J.N. Kapur and H.K. Kesavan. *Entropy Optimization Principles with Application*. Academic Press, Inc., 1992.

[10] D. Fox. Adapting the sample size in particle filters through KLD-sampling. *International Journal of Robotics Research*, 22, October 2003.

[11] J. Kasturi, R. Acharya, and M. Ramanathan. An information theoretic approach for analyzing temporal pattern of gene expression. *Bioinformatics*, 19(4):449–458, 2003.