

# A Distributed Algorithm for Managing Multi-Target Identities in Wireless Ad-hoc Sensor Networks

Jaewon Shin<sup>1</sup>, Leonidas J. Guibas<sup>1</sup>, and Feng Zhao<sup>2</sup>

<sup>1</sup> Computer Science Department  
Stanford University  
Stanford CA 94305, U.S.A  
{jwshin, guibas}@cs.stanford.edu  
<sup>2</sup> Palo Alto Research Center(PARC)  
3333 Coyote Hill Road  
Palo Alto, CA 94304, U.S.A  
zhao@parc.com

**Abstract.** This paper presents a scalable distributed algorithm for computing and maintaining multi-target identity information. The algorithm builds on a novel representational framework, Identity-Mass Flow, to overcome the problem of exponential computational complexity in managing multi-target identity explicitly. The algorithm uses local information to efficiently update the global multi-target identity information represented as a doubly stochastic matrix, and can be efficiently mapped to nodes in a wireless ad hoc sensor network. The paper describes a distributed implementation of the algorithm in sensor networks. Simulation results have validated the Identity-Mass Flow framework and demonstrated the feasibility of the algorithm.

## 1 Introduction

A wireless ad-hoc sensor network (WASN) is a network of sensor nodes with limited on-node sensing, processing, and communication capabilities. At the heart of many WASN applications such as object tracking and environmental monitoring is the problem of estimating non-local parameters or states of the physical phenomenon being observed using only local information available to each node. This problem poses unique challenges that are not addressed in a centralized setting or fixed network:

- Scalable distributed information fusion: The global parameters or states of interest must be estimated, updated, and maintained using only local information.
- Sensor tasking: A sensor node may be tasked to compute or store state information based on its relevance and utility to the current task, as well as cost constraints.

In this paper, we study the problem of distributed multi-target identity management in WASN. The goal is to maintain information about *who is who* over time given targets' position estimates. In addition to its central importance in many of the monitoring and surveillance applications, the problem highlights the need for distributed information fusion and sensor tasking. The study of this problem is an important step towards establishing a general methodology for the design of scalable WASN algorithms.

Multi-target identity management is closely related to the multi-target tracking problem. The main difficulty in both problems is the exponential complexity in associating target position estimates with target identities. In the past three decades, a number of approaches have been developed for the multi-target tracking problem, mostly for centralized computational platforms. MHT ([1]) explicitly maintains associations, or hypotheses, over time and prunes the associations using a rank function. JPDA ([3]) computes an association matrix at each time and updates it with a combination of all new measurements weighted by their marginal probabilities. While widely used in practice, both MHT and JPDA algorithms still suffer from their computational complexity, in the worst case exponential in the number of targets or time steps. Moreover, for WASN applications, a significant challenge lies in distributing the information and computation to each node.

This paper develops an efficient distributed approach to computing and updating multi-target identity information. The **main contribution** of this work is twofold. First, it introduces a new distributed representation called identity belief matrix, a doubly stochastic matrix, that describes how identity information of each target is distributedly represented (a row in the matrix). The key advantage of this representation is that when the matrix is mapped to a set of nodes in a WASN, a node could efficiently maintain possible identifies of a target it is tracking (a column in the matrix), using its local evidence only. Second, the paper develops a distributed algorithm for computing and updating the cross-node identity belief matrix in a WASN. The algorithm exploits doubly-stochasticity of the matrix to renormalize identity probability masses stored on different WASN nodes. As soon as a piece of local/marginal evidence is available to a node, the local belief of target identity is updated, and the information is propagated through the network to other nodes. The computational complexity of our algorithm is  $O(N^2)$ , where  $N$  is the number of targets, a significant advantage over the exponential complexity of MHT and JPDA.

The rest of the paper is organized as follows. Section 2 introduces identity-mass flow (IMF) framework to represent multi-target identity information. To ease the introduction of mathematical materials and focus on key representational issues, the identify representation is first developed in a centralized setting. Sections 3 and 4 develop algorithms to distribute the computation and map the algorithm to WASN nodes. Section 3 describes an algorithm for updating global identity information using local identity information. Section 4 maps the identity representation and algorithm to a set of WASN nodes. Finally, Section 5 describes an implementation of the algorithm, and presents simulation results

that validate the correctness of the representational framework and the basic structure of the algorithm.

## 2 Multiple Identity Management: A Mathematical Framework

Consider the following problem. There are  $N$  targets moving in the region of interest and their position-estimates are given periodically. Assuming the initial identities are known, the goal of the multiple identity management is to maintain and update the identity information of all the targets as they are moving.

Apparently, the dynamics of the target - how they are moving - seems to be the only information available to do the job. Unlike the radar system, however, the subset of the nodes in the sensor network are very close to the targets and are expected to sense the target signature information. This signature information is clear and dependable in the sparse target configuration, but is not informative when the targets are close to one another. Figure 1 illustrates the two target configurations, which are named as Configuration Of High Uncertainty(COHU) and Configuration Of Low Uncertainty(COLU), respectively. It is obvious that the information from COLU should be exploited to better understand the events in the sensor network and spending too much energy in COHU is not desirable.

In the next section, the mathematical formulation of the multiple identity management problem that allows us to exploit the above intuition is presented.

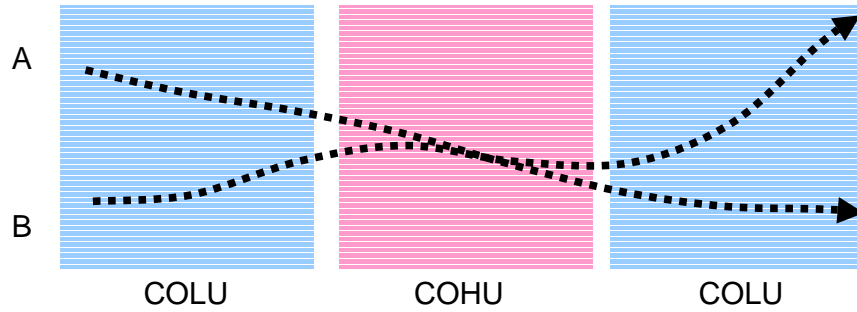


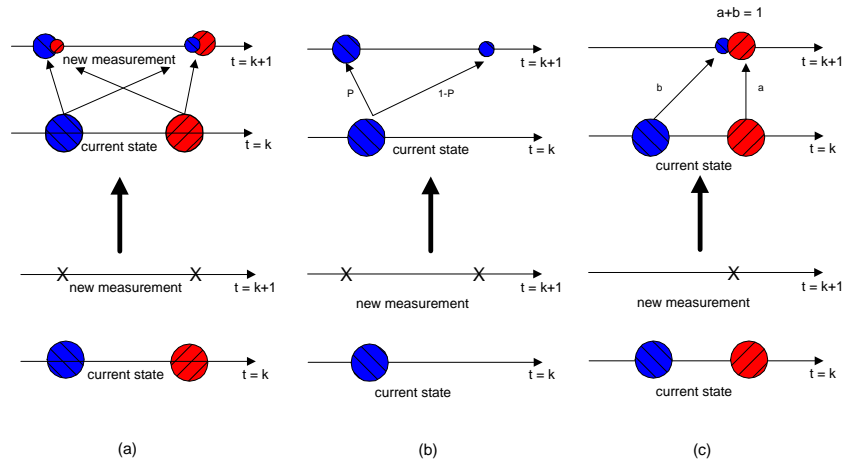
Fig. 1. Sparse(COLU) and Crowded(COHU) Target Configuration

### 2.1 Formulation

In this section, we formulate the problem of multiple identity management as identity-mass distribution problem. The identity set  $I = \{1, \dots, N\}$  is a set of identities of all the targets and the position-estimates of  $N$  targets at time  $k$

is  $X(k) = \{x_i(k) \in \mathbf{R}^2 | i = 1, \dots, N\}$ .<sup>3</sup> The identity management algorithm is supposed to compute the correct permutation of  $I$  given  $X(k)$ . The natural approach to this is to maintain all the possible permutations<sup>4</sup> at each time, although the number of possible permutations increases exponentially. Even with good rank functions and pruning heuristics, the number of possible permutations can easily go unmanageably large in a very short period of time.

To overcome the above combinatorial complexity and maintain the computational complexity as constant over time, we propose the idea of Identity-Mass Flow (IMF) to approximately represent all the possible permutations. Figure 2 (a) shows the basic idea behind our approach; Initially, a unit mass is assigned to each identity. Whenever the new position-estimate  $X(k)$  is available, the whole or partial masses from  $X(k-1)$  flow into  $X(k)$  and the identities are mixed in this way. There, however, need to be constraints regarding how masses flow to make the resulting mixed identities are physically meaningful. Figure 2 (b) and (c) explain the two constraints. (b) says no mass can be created or destroyed during the Identity-Mass Flow and (c) says the sum of all the masses arriving at  $x_i(k)$  is one.



**Fig. 2.** Identity Mass Flow

To formulate the above idea, we define the identity belief matrix  $B(k)$  and the mixing matrix  $M(k)$ .

<sup>3</sup>  $x_i(k)$  and  $x_i(k+1)$  are not related.  $i$ 's are just random labels that come with the position estimate.

<sup>4</sup> In multi-target tracking community, this is also called as *association*.

**Definition 1.** The identity belief matrix  $B(k)$  is a  $N \times N$  doubly-stochastic matrix<sup>5</sup>, whose entry  $b_{ij}(k)$  represents the amount of identity mass from  $i \in I$  that arrives at  $x_j(k)$ . The  $j$ th column  $\mathbf{b}_j(k)$  of  $B(k)$  is called the identity belief vector of  $x_j(k)$ .

$$B(k) = [\mathbf{b}_1(k) \ \mathbf{b}_2(k) \ \cdots \ \mathbf{b}_N(k)] \in [0, 1]^{N \times N}$$

where

$$\mathbf{b}_i(k) = \begin{bmatrix} p(x_i(k) \text{'s ID is 1}) \\ p(x_i(k) \text{'s ID is 2}) \\ \vdots \\ p(x_i(k) \text{'s ID is } N) \end{bmatrix} \in [0, 1]^{N \times 1}$$

**Definition 2.** The mixing matrix  $M(k)$  is a  $N \times N$  doubly-stochastic matrices, whose entry  $m_{ij}(k)$  represents the probability of  $x_j(k)$  being originated from  $x_i(k-1)$ , and is statistically independent with  $M(l)$  for all  $l \neq k$ .

Given the definition of  $M(k)$  and  $B(k)$ , the following theorem presents the basic equation relating the two quantities.

**Theorem 1.** Let  $B(k)$  and  $M(k)$  be the identity belief matrix and the mixing matrix at time  $k$  as defined above, then the following is true.

$$B(k+1) = B(k)M(k+1)$$

*Proof.* From the above definitions, the identity belief of  $x_j(k+1)$  is computed as follows

$$\mathbf{b}_j(k+1) = \sum_{l=1}^N m_{lj}(k+1)\mathbf{b}_l(k) = B(k)\mathbf{m}_j(k+1)$$

where  $\mathbf{m}_j(k+1)$  is the  $j$ th column of  $M(k+1)$ . Therefore,

$$B(k+1) = B(k)M(k+1)$$

and this concludes the proof.

The above theorem shows that we can recursively compute the identity information  $B(k)$  by computing  $M(k)$  from  $X(k)$  and  $X(k-1)$  and is illustrated in Figure 3. The details on how to compute  $M(k)$  is investigated in the next section.

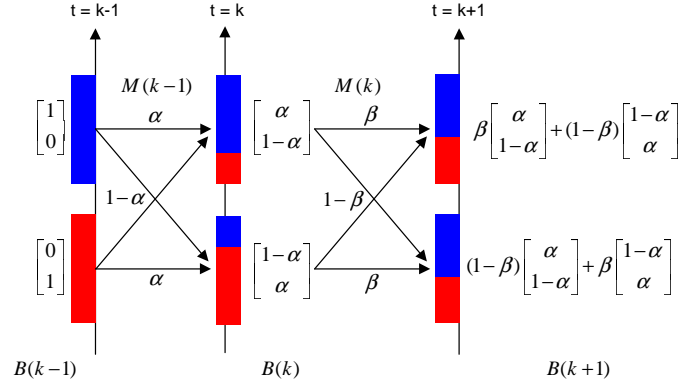
The following lemma explains how the uncertainty in the system changes over time in this formulation.

**Lemma 1.** Let  $\pi_{B(k)} \in [0, 1]^{N \times 1}$  be the probability mass function over all the possible identity association in  $B(k)$ , then

$$H(\pi_{B(k)}) \geq H(\pi_{B(k-1)})$$

where  $H(\cdot)$  is the statistical entropy of a probability mass function.

<sup>5</sup> The doubly-stochastic matrix is a  $N \times N$  non-negative matrix, whose rows and columns sum to one.



**Fig. 3.** Updating  $B(k)$  using  $M(k)$

*Proof.* The mixing matrix  $M(k)$  can be represented as a convex sum of permutation matrices as follows

$$M(k) = \sum_{i=1}^{N!} \alpha_i \Phi_i$$

where  $\sum_i \alpha_i = 1$  and  $\Phi_i$  is the  $i_{th}$   $N \times N$  permutation matrix. Then,

$$\begin{aligned} H(\pi_{B(k)}) &= H(\pi_{B(k-1)} M(k)) \\ &= H\left(\sum_{i=1}^{N!} \alpha_i \pi_{B(k-1)} \Phi_i\right) \\ &\geq \sum_{i=1}^{N!} \alpha_i H(\pi_{B(k-1)} \Phi_i) \\ &= \sum_{i=1}^{N!} \alpha_i H(\pi_{B(k-1)}) \\ &= H(\pi_{B(k-1)}) \end{aligned}$$

where the inequality comes from the strict concavity of the entropy function. This concludes the proof.

The statistical entropy is a quantitative measure of uncertainty in the probability distribution and the above lemma shows the uncertainty on the possible identity associations does not decrease over time. Therefore, the uncertainty will grow until every identity association becomes equally likely without any additional information available. As we have mentioned in section 2, the target identity information is more likely to be available in COLU and the proper use of this local information could make the uncertainty decrease in IMF formulation. The details on how to exploit the local identity information is investigated in section 3

## 2.2 Computing Mixing Matrix $M(k)$

The mixing matrix  $M(k)$  is a collection of marginal association probabilities and can be computed from the joint association probability<sup>6</sup> in theory. Computing the joint association probabilities is very expensive and should be avoided in practice. In this paper, we propose a simple heuristic with  $O(N^2)$  empirical complexity that only requires the information on how fast the targets are moving. Let's assume that the speed information is given as a probability density function  $p(s(k))$ , where  $s(k)$  is  $|x(k) - x(k-1)|/\Delta T$ .<sup>7</sup> Then, we compute a non-negative matrix  $L(k) \in \mathbf{R}^{N \times N}$ , whose  $(i, j)$  entry is  $l_{ij}(k) = p(s(k) = |x_j(k) - x_i(k-1)|/\Delta T)$ . In general,  $L(k)$  is not doubly-stochastic and an optimization need to be done to transform this into a doubly-stochastic matrix. We use the Iterative Scaling algorithm to transform  $L(k)$  into a doubly-stochastic matrix. The details on the Iterative Scaling algorithm are explained in section 3.2.

## 3 Updating Multi-target Identity Using Local Information

In WASN, the ability of using local information efficiently is critical in its algorithms since non-local information only come at the cost of communication. The IMF approach in multi-target identity management does provide a natural setting for exploiting local evidences. Figure 4 illustrate this in a simple two targets crossover scenario. Two targets are moving cross each other and their identity masses are mixed at the intersection of the two tracks. After the mixing, the identity belief matrix  $B(k)$  becomes un-informative - each association is almost equally likely. When the two targets are well-separated, i.e., in COLU, one of the nodes near the bottom target observes a local evidence<sup>8</sup> that the bottom target is more likely to be red. This observation increases the red-mass of the bottom target  $b_{red, bottom}(k)$  and the rest of the elements in  $B(k)$  can be updated from the doubly-stochasticity of  $B(k)$ . Therefore, the local information about the bottom target directly leads to the information about the top target in a unique way.

From the above example, only a local evidence seems to be enough to update the whole identity belief  $B(k)$  uniquely. For the general  $N$  target case, however, the doubly-stochasticity of  $B(k)$  is not enough to guarantee a unique solution since there are more unknowns  $O(N^2)$  than the equations  $O(N)$ . Therefore, we need more constraints or optimizations to compute a unique  $B(k)$  given a local evidence. The upcoming sub-sections deal with this problem of computing  $B(k)$  given a local evidence in a centralized setting. Section 4 discusses the distributed implementation of the algorithm for WASN.

<sup>6</sup> Probabilities of permutations/associations, i.e., how likely each permutation is.

<sup>7</sup> It is usually the case that  $p(s(k))$  is stationary, i.e., does not depend on  $k$ .

<sup>8</sup> A local evidence is the information enough to determine the whole or partial entries in  $b_i(k)$ . The simple example is " $x_i(k)$  is of ID  $j$ ".

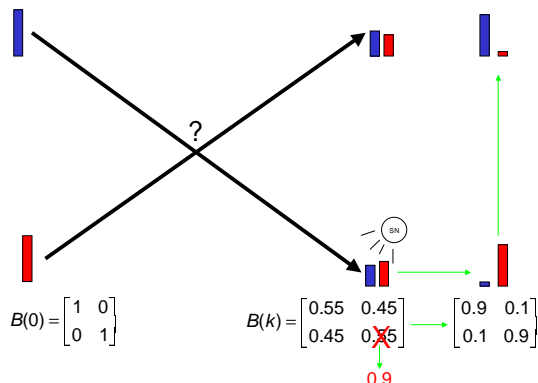


Fig. 4. Example of Using Local Evidence

### 3.1 Bayesian Normalization

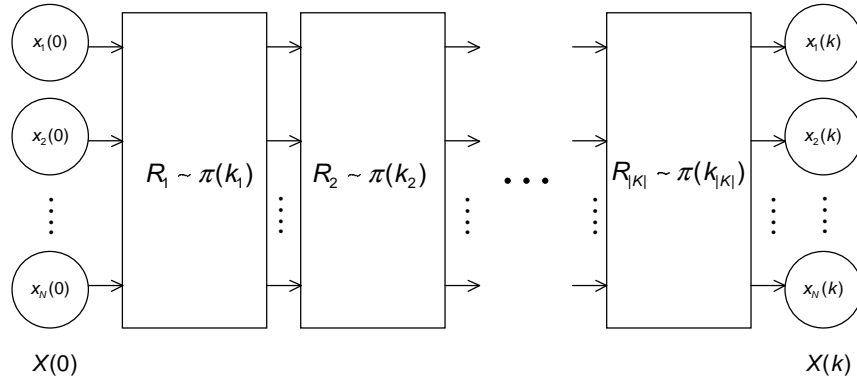
Before we introduce a practical solution to compute unique  $B(k)$  given a local evidence, we study the desirable properties of the *perfect solution* by computing  $B(k)$  as a Bayesian posterior belief distribution given a local evidence, assuming the whole history of the mixing events are known. In this case, we assume that the joint probability distribution  $\pi(k) \in \{Z \in \mathbf{R}^{N! \times 1} \mid \sum_i z_i = 1, z_i \geq 0\}$  over all the possible  $N!$  associations at time  $k$ , from which  $M(k)$  can be derived, is available for all  $k$ . Note that, for some  $k$ 's,  $\pi(k)$ 's are deterministic and their associated  $M(k)$  are permutation matrices. These  $\pi(k)$  do not contribute in computing the  $B(k)$  and its posterior. To consider only those random mixing events, we define  $K \subset \{0, 1, \dots, N\}$  be the set of time indices associated with  $|\pi(k_i)| > 2$ , where  $k_i$  is the  $i_{th}$  element in  $K$  and  $|\cdot|$  represents the cardinality of a set. We also introduce a sequence of random variables  $R_i$  associated with  $\pi(k_i)$ , which takes values  $j \in \{1, \dots, |\pi(k_i)|\}$  with probability of  $\pi_j(k_i)$ , i.e.,  $j_{th}$  value in  $\pi(k_i)$ .

Figure 5 illustrates the above formulation, where each box can be considered as a probabilistic switch governed by  $R_i$ , i.e., a specific permutation is chosen according to the value of  $R_i$  with some probability. Then, a single identity association at time  $k$  is a point  $\varepsilon$  in the joint event space  $S = \{(R_1, R_2, \dots, R_{|K|}) \mid R_i \in [1, |\pi(u_i)|]\}$  with  $|S| = \prod_i |\pi(u_i)|$  and the probability of this event can be easily computed due to the statistical independence assumption in section 2.1.

$$\begin{aligned} p((R_1, \dots, R_{|K|}) = \varepsilon) &= p(R_1 = \varepsilon_1) \cdots p(R_{|K|} = \varepsilon_{|K|}) \\ &= \pi_{\varepsilon_1}(u_1) \cdots \pi_{\varepsilon_{|K|}}(u_{|K|}) \end{aligned}$$

Using the above equation, we can compute the posterior  $B(k)$  given a local evidence  $L$ , which is a set of events in  $S$  satisfying the local observation, say,  $ID(x_i(k)) = j$ , using the following theorem.





**Fig. 5.** Bayesian Normalization

**Theorem 2.** Let  $E_{ij}(k)$  be the subset of  $S$  satisfying  $ID(x_j(k)) = i$  and  $L$  be the subset of  $S$  satisfying the local observation, then

$$p(b_{ij}(k)|L) = p(E_{ij}|L) = \frac{\sum_{\varepsilon_l \in E_{ij} \cup L} p(\varepsilon_l)}{\sum_{\varepsilon_l \in L} p(\varepsilon_l)}$$

*Proof.* The proof is trivial using the Bayes' rule.

Now that we know how to update  $B(k)$  given a local evidence  $L$ , but the effect of the evidence to the other columns is not obvious from the equation. The following lemma describes how the local evidence effect the uncertainty of the other columns.

**Lemma 2.** The local observation  $L$  does not increase the entropies of the columns, i.e.

$$H(\mathbf{b}_i(k)|L) \leq H(\mathbf{b}_i(k))$$

*Proof.* See [5] for the proof.

The lemma says that the local evidence does not increase the entropy of the other columns on the average.

The ideal solution obtained in the above theorem exhibits one very interesting characteristic, in which there are some elements in  $B(k)$  - in addition to the zero elements in  $B(k)$  - that are not affected by the local evidence. This can potentially save huge amount of communication energy in practice. The following theorem formally presents the property of the Bayesian solution.

**Theorem 3.** Let  $b_{pq}(k)$  be the entry that becomes 1 from the local evidence  $L$ , then the columns with zero at  $p$ th entry and the rows with zero at  $q$ th entry do not change.

*Proof.* Let's first prove that the columns with zero at  $p$ th entry do not change. If  $\mathbf{b}_i(k)$  is such a column, then there does not exist any event in  $S$  that guarantees a path from  $x_p(0)$  to  $x_i(k)$ , i.e., no path originated from  $x_p(0)$  reaches  $x_i(k)$ . The local evidence  $L$  defines a subset of  $S$ , which guarantees the existence of at least one path between  $x_p(0)$  and  $x_q(k)$ . None of these paths between  $x_p(0)$  and  $x_q(k)$ , however, affect the paths arriving at  $x_i(k)$ . Due to the statistical independence assumption on the mixing events,  $\mathbf{b}_i(k)$  does not change given the local evidence  $L$ . The row case can be proved in the same way. This concludes the proof.

The above theorem reduces the number of variables to be updated in  $B(k)$  given a local evidence. In addition to that, this can help the number of communications required to update  $B(k)$  given a local evidence assuming that each column  $\mathbf{b}_i(k)$  is maintained by a node in the sensor network. The details on the distributed computation is discussed in section 4.

In addition to these rows and columns, the zero elements in  $B(k)$  do not change given a local evidence  $L$ .

### 3.2 Iterative Scaling

In practice, the Bayesian formulation in the previous section is infeasible due to its exponential complexity. This section presents the practical alternative called the Iterative scaling. First, we present a version of the Iterative Scaling algorithm to achieve a doubly-stochastic matrix  $A$  given a  $N \times N$  non-negative matrix  $B$ .

```

B := A;
B_old := A;
for k = 1 to maximum_number_of_iteration
  for i = 1 to number_of_row
    row_sum := 0;
    for k = 1 to number_of_column
      row_sum := row_sum + B(i,k);
    end
    for j = 1 to number_of_column
      B(i,j) := B(i,j)/row_sum;
    end
  end
  for i = 1 to number_of_column
    column_sum := 0;
    for k = 1 to number_of_row
      column_sum := column_sum + B(i,k);
    end
    for j = 1 to number_of_row
      B(j,i) := B(j,i)/column_sum;
    end
  end
end
if |B - B_old| < error

```

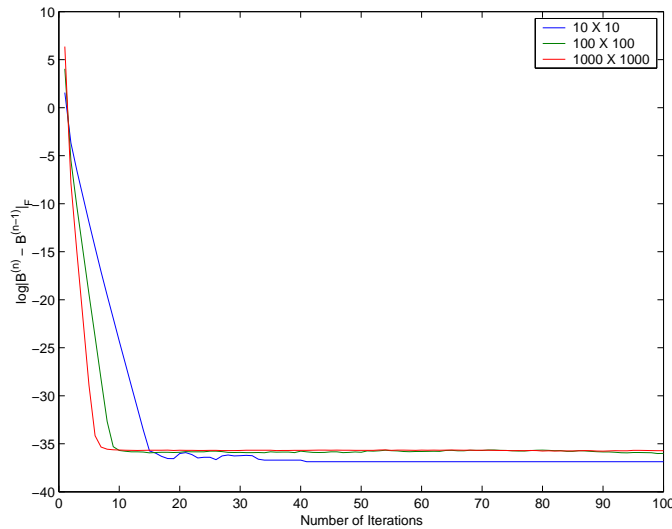
```

    terminate;
  end
  B_old := B;
end

```

Basically, the algorithm divides each element in  $i_{th}$  row(column) by the sum of the  $i_{th}$  row(column) and repeats the normalization until the error margin is small. The following observations are made from the numerical simulations and we list them here without proofs.

- The algorithm converges to a unique doubly-stochastic matrix given an initial matrix.
- The ordering of row/column normalization does not affect the convergence.
- The total number of iteration is not affected by the size of the matrix.



**Fig. 6.** Typical Convergence of Iterative Scaling: The flat part of the plots are due to the Matlab precision limit.

From these observations, the Iterative Scaling algorithm scales as  $O(N^2)$ . The proof of the complexity result remains as an immediate task for future research. Figure 6 shows an example of the convergence behavior of the algorithm. Three different sizes of matrices ( $10 \times 10$ ,  $100 \times 100$ ,  $1000 \times 1000$ ) are generated randomly using Matlab `rand(.)` function, in which each entry is generated according to the uniform probability density over  $[0, 1]$ . The plot shows that the Iterative Scaling method has fast convergence and the size of a matrix does not affect the convergence ratio. What seems to affect the convergence rate is how different

the (scaled) initial matrix from being the doubly-stochastic matrix, although we do not have a proper quantitative measure for this at this point. This is why the larger matrices in the above figure converges a little faster than the smaller ones, since all the row/column sums of the larger matrices are close to  $0.5N$  due to the Law of Large Numbers and effectively *close* to a doubly-stochastic matrix after scaling.

To benefit from the theorem 3, the Iterative scaling algorithm should be able to deal with the non-square matrix with the fixed row/column sums. Let  $\mathbf{r}$  and  $\mathbf{c}$  be  $N \times 1$  vector of row and column sum satisfying  $\sum_i c_i = \sum_j r_j$ , then we can modify the above pseudo code as follows.

$$\begin{aligned} B(i, j) &= B(i, j) / \text{row\_sum} * \mathbf{r}(i); \\ &\dots\dots \\ B(j, i) &= B(j, i) / \text{column\_sum} * \mathbf{c}(i); \end{aligned}$$

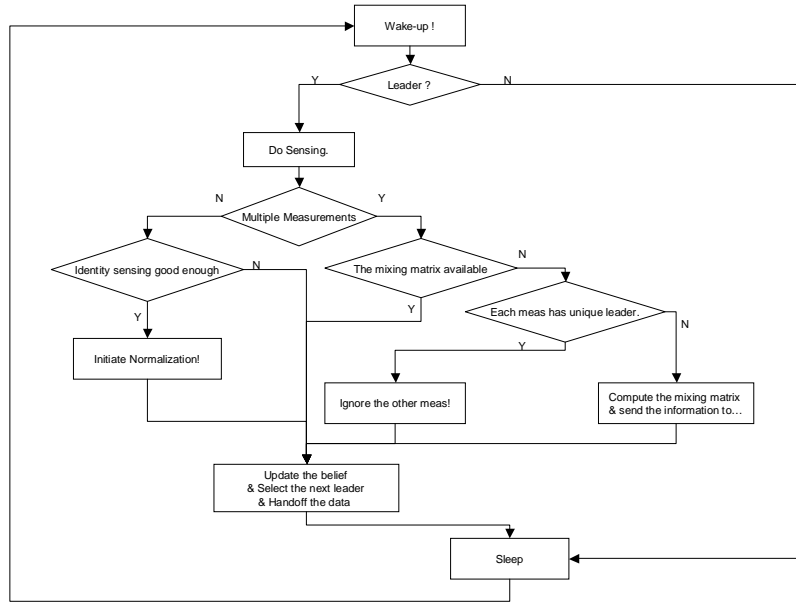
## 4 Distributed Implementation in WASN

The basic quantity to be distributed is  $B(k)$ , the belief matrix. One way of distributing  $B(k)$  is to let each node in the sensor network maintain its own version of  $B(k)$ . This method is very robust and fault-tolerant due to the information redundancy in the system. However, this idealistic distribution is infeasible and non-scalable for the following the reason. To update the information, each node need to compute its version of  $M(k)$ , which requires information from at least one of the other nodes. This is exactly the scenario in the landmark paper [13], where per-node throughput goes to zero as the number of nodes goes to infinity even under optimal routing and power control scheme. Therefore, the idealistic distribution of each node maintaining  $B(k)$  is impossible and this argument is true for all the algorithms of sensor networks that estimates the global quantity.

To overcome this problem, we adopt and extend the approach in [15] and [16], where a leader-based single target tracking in WASN is introduced. The basic idea is that, only a small number of nodes called *leaders* are active and responsible for maintaining/updating the information of interest. When the leaders are no longer the good subset of nodes for the information, they handoff the information to the other nodes based on a utility function and the nodes receiving the information become the new leaders. In this approach, whereabouts of the information is easily maintained at the risk of the reduced robustness and fault-tolerance.

Applying the leader-based approach to our algorithm, each column  $\mathbf{b}_i(k)$  of  $B(k)$  and its position estimate  $x_i(k)$  is maintained by each leader. When the mixing happens, the local version of  $M(k)$ <sup>9</sup> can be easily computed by the leader. When a leader node observes a local evidence about the ID of its target, say  $ID(x_i(k)) = j$ , then the leader needs to talk to some of the other leaders, who also think what they are tracking can be of the same ID. This type of multi-cast

<sup>9</sup> In the two target mixing, only non-zero entries in the  $i$ th and  $j$ th columns of  $M(k)$  are required to update  $b_i$  and  $b_j$  and they are locally computable.



**Fig. 7.** Flow Chart of the Distributed Algorithm for the Multi-Target Identity Management

communication in network is usually dealt by a group management protocol, which maintains and updates the different groups in the network according to a predefined membership. In our case, the  $i_{th}$  group is the set of the leader that have non-zero probability at  $i_{th}$  entry of their  $\mathbf{b}(k)$  and we assume there exists a good (or optimal) group management protocol for our purpose. Figure 7 shows the main procedures and their relations of the distributed algorithm that each node is running.

## 5 Experimental Results

We make the following assumptions for the simulation.

- Each node can sense the positions of the targets within its sensing range.
- Each node can talk to all the nodes within its communication range.
- Initially, the number of the targets are known to the initial leaders.
- Each node has a signal processing module for the signal classification and the module performs better in COLU.
- Each node knows the relative positions of all the nodes in the communication range.

The initial leaders are manually selected for the simulation, although it's possible to detect them as long as their initial positions are well separated. Each

leader updates its belief vector  $\mathbf{b}_i(\mathbf{k})$  using the local version of  $M(k)$  and handoffs the information to the next leader, which is selected based on its geographical location. The signal classification module of the leaders will keep collecting the information and initiates the identity normalization process by talking to the other leaders that are in the same group when the identity information from the signal classification is better than  $\mathbf{b}_i(k)$  and some threshold.

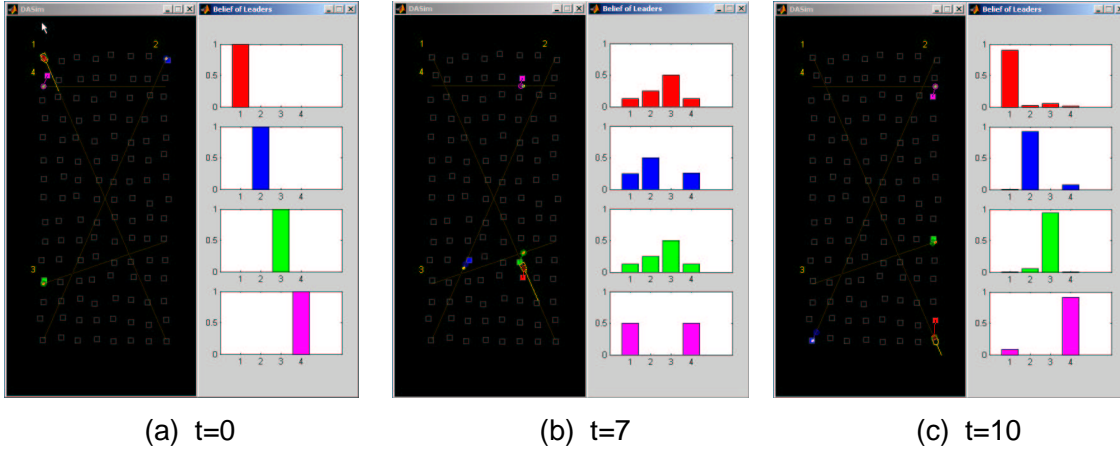


Fig. 8. Simulation Example

Figure 8 shows the three screen shots from the simulation of the algorithm. Four targets - one tank and three dots - are moving along the straight lines for 10 seconds. The tank signature is much different from those of the other three, so it can be identified with high probability in COLU. This local identity information about the tank is the only available information and used to normalize the belief  $\mathbf{b}_i(k)$  of the other leaders. The four leaders are colored differently and their corresponding beliefs are displayed with the same color. Figure 8 (a) is the initial configuration of the targets and their associated leaders. (b) shows that the belief  $\mathbf{b}_i(k)$  of each leader gets uncertain after some number of mixing events at  $t = 7$  and the leaders are no longer sure of the identities of the targets. (c), however, shows the beliefs get much better after the normalization using Iterative Scaling algorithm given local identity information.

In figure 9, how the identity uncertainty of each target evolves during the simulation is depicted using the entropy of each identity belief  $\mathbf{b}_i$ . The increases in the uncertainty are due to the mixing events and the decreases are by the local evidences. The two pieces of the local evidence on target 1 have reduced the uncertainties of all the other targets in this example, since the identity mass from the target 1 is mixed with all the other identities during the mixing events.

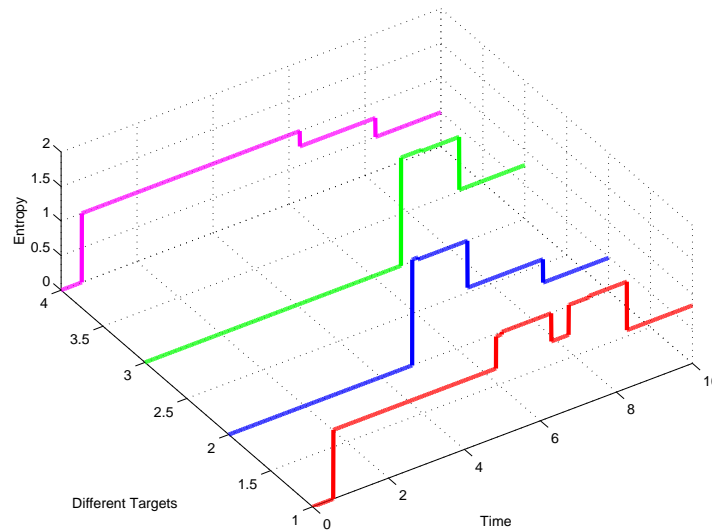


Fig. 9. An Example of How Uncertainty of Each Belief Changes

## 6 Summary and Future Work

In this paper, we presents a scalable distributed algorithm for multi-target identity management problem, first in a mathematical/centralized setting and then a experimental/distributed case. We would like to further investigate this problem under more practical assumptions and models for WASN for example. A signal processing model for target localization and signal classification and the ability to deal with the addition/deletion of the targets are among the top priorities. Theoretically, the convergence proof of the normalization given a multiple local evidence regardless of their chronological order is among tasks we have in mind.

## References

1. Reid, D.B., "An Algorithm for Tracking Multiple Targets", *IEEE Trans. on Automatic Control*, No. 6, December 1979, pp. 843-854
2. R. R. Tenny and N. R. Sandell Jr., "Detection with Distributed Sensors", *IEEE Transactions on Aerospace and Electronic Systems*, Vol. AES-17, No. 4, July 1981
3. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, New York, 1988
4. H. R. Hashemi and I. B. Rhodes, "Decentralized Sequential Detection", *IEEE Transactions on Information Theory*, Vol. 35, No. 3, May 1989
5. T. M. Cover and J. A. Thomas, "Elements of Information Theory", *Wiley*, 1991
6. I. J. Cox and S. L. Hingorani, "An Efficient Implementation of Reid's Multiple Hypothesis Tracking Algorithm and its Evaluation for the Purpose of Visual Tracking," *IEEE Trans. on PAMI*, Vol 18., No. 2, pp. 138-150, Feb. 1996

7. P. L. Combettes, "Hilbertian Convex Feasibility Problem: Convergence of Projection Methods" *Appl. Math. Optim.* 35:311-330, 1997
8. M. Isard and A. Blake, "CONDENSATION - Conditional Density Propagation for Visual Tracking", *Int. J. Computer Vision*, 1998
9. L. J. Guibas, "Kinetic data structures - a state of the art report," *Proc. Workshop Algorithmic Found. Robot.*, pages 191-209, A. K. Peters, Wellesley, MA, 1998
10. H. Tao, H.S. Sawhney and R. Kumar, "A Sampling Algorithm for Tracking Multiple Objects," *Proc. of the IEEE Wkshp. on Vision Algorithms, Corfu, Greece*, Sep. 1999
11. H. Pasula, S. Russel, M. Ostland and Y. Ritov, "Tracking Many Objects with many sensors," *Int. Joint Conf. on Artificial Intelligence (IJCAI), Stockholm*, pages 1160-1171, 1999.
12. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next century challenges: Scalable coordination in sensor networks", *Proceedings of the Fifth Annual International Conference on Mobile Computing and Networks*, Seattle, Washington, August 1999
13. P. Gupta and P.R. Kumar, "The Capacity of Wireless Networks", *IEEE Trans. Inform. Theory*, 46(2):388-404, 2000.
14. J. MacCormick and A. Blake, "Probabilistic exclusion and partitioned sampling for multiple object tracking.", *Intl J. Computer Vision*, 39(1):57-71, 2000.
15. M. Chu, H. Haussecker, and F. Zhao, "Scalable Information-Driven Sensor Query and Routing for ad hoc Heterogeneous Sensor Network", *International Journal of High Performance Computing Applications*, 2002
16. F. Zhao, J. Shin, and J. Reich, "Information Driven Dynamic Sensor Collaboration for Tracking Application", *IEEE Signal Processing Magazine* 2002 March
17. Wei Ye, John Heidemann and Deborah Estrin, "An Energy-Efficient MAC Protocol for Wireless Sensor Networks" ,*IEEE INFOCOM 2002*, June, 2002.