This paper appeared at 2003 SPIE AeroSense Conference.

Multitarget-Multisensor Data Association Using the Tree-Reweighted Max-Product Algorithm

Lei Chen[†], Martin J. Wainwright[‡], Mujdat Cetin[†], Alan S. Willsky[†]

[†]Laboratory for Information and Decision Systems, EECS, MIT, Cambridge, MA
[‡]EECS, Univ. of California, Berkeley, CA

ABSTRACT

Data association is a fundamental problem in multitarget-multisensor tracking. It entails selecting the most probable association between sensor measurements and target tracks from a very large set of possibilities. With N sensors and n targets in the detection range of each sensor, even with perfect detection there are $(n!)^N$ different configurations which renders infeasible a solution by direct computation even in modestly-sized applications. We describe an iterative method for solving the optimal data association problem in a distributed fashion; the work exploits the framework of graphical models, which are a powerful tool for encoding the statistical dependencies of a set of random variables and are widely used in many applications (e.g., computer vision, error-correcting codes). Our basic idea is to treat the measurement assignment for each sensor as a random variable, which is in turn represented as a node in an underlying graph. Neighboring nodes are coupled by the targets visible to both sensors. Thus we transform the data association problem to that of computing the maximum *a posteriori* (MAP) configuration in a graphical model to which efficient techniques (e.g., the max-product/min-sum algorithm) can be applied. We use a tree-reweighted version of the usual max-product algorithm that either outputs the MAP data association, or acknowledges failure. For acyclic graphs, this message-passing algorithm can solve the data association problem directly and recursively with complexity $\mathcal{O}\left((n!)^2 N\right)$. On graphs with cycles, the algorithm may require more iterations to converge, and need not output an unambiguous assignment. However, for the data association problems considered here, the coupling matrices involved in computations are inherently of low rank, and experiments show that the algorithm converges very fast and finds the MAP configurations in this case.

Keywords: data association, max-product/min-sum algorithm, multitarget-multisensor tracking, Markov random fields, graphical models

1. INTRODUCTION

Data association, with the goal of partitioning observations to match up with a particular origin, is a fundamental problem of multiple target tracking.[1,2] The specifics of the data association problem vary according to the different tracking approaches and the different modalities of the sensors being used. However, the common aspect of the problem is to identify a pair of points, one from each of two random point sets, that share the same origin.[3] The points in the first set $\mathbf{X}$ provide the prior knowledge about the distribution of target states that we are interested in, and are usually obtained from the prediction step of the tracking algorithm. Thus $\mathbf{X}$ is called the prediction set. The second set $\mathbf{Z}$ is the measurement set and consists of noisy measurements of the target states produced by the sensors. The correspondence between points in $\mathbf{X}$ and points in $\mathbf{Z}$ is unknown and needs to be resolved. The central task of data association is to figure out the correct mapping relationship based on the observation $\mathbf{Z} = Z$, subject to the constraints that each $\mathbf{z} \in \mathbf{Z}$ is assigned to at most one $\mathbf{x} \in \mathbf{X}$, and each $\mathbf{x}$

Further author information: (Send correspondence to Lei Chen)
Lei Chen: E-mail: lchen@mit.edu, Address: LIDS, MIT, 77 Mass. Ave, Room 35-439, Cambridge, MA 02139
Martin J. Wainwright: E-mail: martinw@eecs.berkeley.edu, Address: Soda Hall 485, EECS Dept., Univ. of California at Berkeley, Berkeley, CA 94720
Mujdat Cetin: E-mail: mcetin@mit.edu
Alan S. Willsky: E-mail: willsky@mit.edu

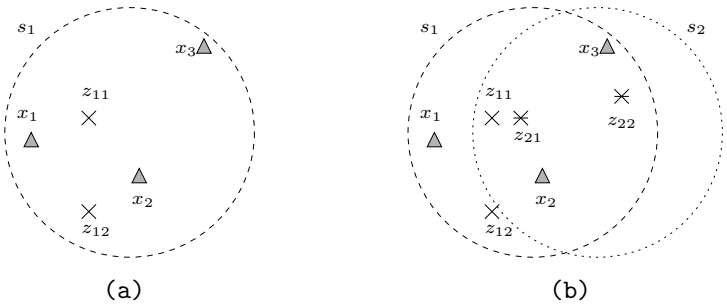

Figure 1. An example of multitarget-multisensor data association. Dashed circles represent the sensor detection range. Gray $\triangle$ are targets with mean positions x_1, x_2 and x_3. $\times$ and $*$ are measurements generated by sensors s_1 and s_2 respectively. (a) Data association scenario with only sensor s_1. If we use the Euclidean distance between the measurements and target positions as the association criterion, z_{11} should go to target 1 and z_{12} should go to target 2. (b) Data association scenario with two sensors of overlapping spatial coverage. The proximity of z_{11} and z_{21} suggests the possibility that they came from the same target, which is most likely to be x_2.

gets no more than one $\mathbf{z}$. Fig. 1(a) shows a trivial example with one sensor and two measurements for the three targets in the range. The three targets have mean positions at x_1, x_2 and x_3 respectively. Assuming there are no false alarms, then there are 6 possible association configurations. For simplicity if we use the Euclidean distance as the association criterion in this example, it is easy to see measurement z_{11} should be associated with target 1 and measurement z_{12} with target 2. However, when the target density as well as measurement density increases, the number of possible associations increases combinatorially with the number of targets and an answer to the data association problem is not obvious. The problem becomes even more complicated when the surveillance systems utilize multiple sensors to provide overlapping coverage on the targets. In such a case, the association in different sensors are coupled, and the locally optimal assignment in general is not globally optimal. Back to the example in Fig. 1, if we have a second sensor which can only detect target 2 and target 3, as shown in Fig. 1(b), we might consider flipping our earlier assignment based on Fig. 1(a), so that target 2 is associated with a pair of *consistent* measurements from sensor s_1 and sensor s_2. This suggests that we should take the association for the other sensors into consideration when multiple sensors are used. However, the number of different association configurations increases exponentially with the number of sensors in use. Even in the case of perfect detection (without false alarms and missed detections), with N sensors and n targets in the range of each sensor, there are $(n!)^N$ different configurations, which renders infeasible a solution by direct computation even in modestly-sized applications. For example, for $N = 7$ and $n = 5$, there are more than 300 trillion possible configurations. To solve a problem with such complexity, the traditional centralized data association approach imposes a very heavy burden on the central node, in terms of both computation and communication. This difficulty motivates the distributed data association approach that performs the task at the sensor level and makes use of the computational resources of all sensors in parallel. Compared with the centralized approach, distributed data association has the advantage of reduced data-bus loading, reduced computational loading at a single processor and high survivability.[4] These benefits have drawn much interest to research on the architecture of distributed tracking systems using sensor networks. Yet the development of practical distributed data association algorithms is still a topic with numerous open research issues.

In this paper, we describe an iterative method for solving the optimal data association in a distributed fashion; the work makes use of the framework of graphical models. Our basic idea is to treat the measurement assignment for each sensor as a random variable, which is in turn represented as a node in an underlying graph. Then we exploit the problem structure and transform the data association problem to that of computing the maximum *a posteriori* (MAP) configuration in a graphical model, to which efficient techniques (e.g., the max-product/min-sum algorithm) can be applied. We use a variant of the usual max-product algorithm, know as the tree-reweighted max-product algorithm,[5] that is guaranteed to either find the MAP configuration, or to acknowledge failure. For acyclic graphs, this message-passing algorithm can solve the data association problem directly and recursively with complexity of $\mathcal{O}\left((n!)^2 N\right)$. On graphs with cycles, which we are more likely to encounter, the algorithm

may require more iterations to converge and need not output an unambiguous assignment. However, for the data association problems, the coupling matrices involved in the computations are inherently of low rank, and experiments show that the algorithm converges very fast and finds the MAP configuration in this case.

This paper is organized in the following way. Section 2 reviews the previous approaches for data association and introduces graphical models, as well as the tree-reweighted max-product algorithm. Section 3 contains the mathematical formulation of the data association problem in our framework, and the transformation of this problem to an inference problem on graphical models. Experimental results are presented in Section 4. Finally, we conclude this paper and propose some future research directions in Section 5.

2. BACKGROUND

2.1. Previous Research in Data Association

The problem of centralized data association has been extensively investigated in the past three decades. An early approach, popular primarily due to its simplicity, was that of joint probabilistic data association (JPDA).[1] JPDA incorporates all observations within a gated region about the predicted target state into the update of that target's state. The contribution of each observation is determined by a probability-based weight. A given observation can also be used to update multiple targets' states. In essence, JPDA averages over the data association hypotheses that have roughly comparable likelihoods and thus suffers from degradation in performance in a dense target environment. The multiple hypothesis tracking (MHT) algorithm[6] is the most successful approach to date. MHT enumerates all possible association hypotheses to form an association hypothesis tree and evaluates the probability of each hypothesis to pick out the most probable one. At each time frame the hypothesis tree is expanded with new report data. The most noticeable advantage is that deferred decisions are possible with this approach. When several ambiguous hypotheses arise, MHT allows a firm decision to be delayed until later data are received to aid the evaluation. However, in the MHT algorithm the number of branches of the hypothesis tree explodes exponentially. An intricate procedure to prune unlikely hypotheses and to combine similar hypotheses has to be developed. Even with these techniques, the computation is still formidable in a dense target environment. Another approach is to treat the data association problem as a multi-dimensional assignment problem and use linear programming (LP) techniques to solve it.[7–9] However, for the assignment problem with more than two dimensions, the problem is NP-complete. Thus for non-trivial problems it is impractical to pursue the optimal solution. Most algorithms with multi-dimensional assignment formulation harness Lagrangian relaxation to reduce the complexity and achieve suboptimal solutions with acceptable accuracy.

JPDA and MHT also have distributed versions,[10,11] which share the same drawbacks of their centralized versions. A discussion of distributed fusion architectures and algorithms can be found in Ref. 12, which also introduces an approach based on information graphs and underlines the challenges associated with the arrival of redundant information in such graphs through multiple paths. This redundant information flow on loopy graphs have served as a motivation for our work in the development of distributed data association algorithms. Our approach involves exploiting the structure that may exist in the overlaps between sensor detection ranges. A convenient way in which to encode this structure is using graphical models, to which we now turn.

2.2. Graphical Models

Graphical models[13] provide a powerful framework for representing the structure of statistical dependencies in a collection of random variables. Accordingly, they are used in many applications, including computer vision,[14] speech recognition,[15] and error-correcting codes.[16] While there are various formalisms for graphical models, the work in this paper involves Markov random fields, which are based on undirected graphs.

Let $\mathcal{G} = (V, E)$ be an undirected graph, where V is a set of nodes or vertices, and E is a set of edges. A *clique* of the graph is a fully connected subset of the vertex set. Associated with each node $s \in V$ is a random variable $\mathbf{q}_s$ that takes values in the discrete set $\mathcal{Q} := \{0, 1, \ldots, m-1\}$, so that the full vector $\mathbf{q} := \{\mathbf{q}_s \mid s \in V\}$ takes value in the Cartesian product space $\mathcal{Q}^N$. For any subset $S \subset V$, we define $\mathbf{q}_S := \{\mathbf{q}_s \mid s \in S\}$. Let D be a vertex cutset in the graph, so that removing it separates the graph into at least two pieces A and B. We say that $\mathbf{q}$ is *Markov* with respect to the graph if $\mathbf{q}_A$ and $\mathbf{q}_B$ are conditionally independent given $\mathbf{q}_D$. In this case, the Hammersley-Clifford theorem[17] guarantees that the distribution $p(\mathbf{q})$ can be factorized as the product

$p(\mathbf{q}) \propto \prod_{C \in \mathcal{C}} \psi_C(\mathbf{q}_C)$, where each function ψ_C depends only on the subvector $\mathbf{q}_C$ of random variables in the clique C and $\mathcal{C}$ is the set of all cliques of $\mathcal{G}$.

Now suppose that we receive a vector of noisy observations $\mathbf{y} = \{\mathbf{y}_s \mid s \in V\}$, where $\mathbf{y}$ is related to $\mathbf{q}$ by the conditional distribution $p(\mathbf{y} \mid \mathbf{q}) = \prod_{s \in V} \psi_s(\mathbf{q}_s, \mathbf{y}_s)$. Applying Bayes' theorem, we obtain

$$p(\mathbf{q}|\mathbf{y}) \propto \prod_{s \in V} \psi_s(\mathbf{q}_s, \mathbf{y}_s) \prod_{C \in \mathcal{C}} \psi_C(\mathbf{q}_C).$$

This conditional density is central to various estimation problems. Of central interest in this paper is the problem of finding the MAP configuration, defined by $\hat{\mathbf{q}}_{MAP} = \arg\max_{\mathbf{q} \in \mathcal{Q}^N} p(\mathbf{q} \mid \mathbf{y})$.

Any Markov random field can be converted, via the introduction of auxiliary variables, to an equivalent model based on a graph with only pairwise cliques. In this case, the factorization of $p(\mathbf{q} \mid \mathbf{y})$ takes the simpler form

$$p(\mathbf{q}|\mathbf{y}) = \frac{1}{\kappa} \prod_{s \in V} \psi_s(\mathbf{q}_s, \mathbf{y}_s) \prod_{(s,t) \in E} \psi_{st}(\mathbf{q}_s, \mathbf{q}_t), \tag{1}$$

where κ denotes a normalization constant.* We will frequently omit explicit reference to $\mathbf{y}$, since it is a fixed (known) quantity. When the graph is cycle-free (i.e., a tree), then the *max-product or min-sum algorithm*, a non-serial generalization of the Viterbi algorithm, can be applied to efficiently compute the MAP estimate.

One interpretation[18] of the max-product algorithm is as computing the so-called max-marginals $P_s(\mathbf{q}_s) = \kappa \max_{\{\mathbf{q}' \mid \mathbf{q}'_s = \mathbf{q}_s\}} p(\mathbf{q}')$ at each node. If, for each node, the max-marginal P_s over $\mathbf{q}'_s \in \mathcal{Q}$ is attained at a unique value, then it can be seen that the MAP configuration $\{\hat{\mathbf{q}}_s \mid s \in V\}$ is unique, with elements given by $\hat{\mathbf{q}}_s = \arg\max_{\mathbf{q}'_s \in \mathcal{Q}} P_s(\mathbf{q}'_s)$. The max-product algorithm computes the max-marginals efficiently by a parallel set of message-passing operations. At each iteration $n = 0, 1, 2, \ldots$, every node $t \in V$ passes a message, denoted by $M^n_{ts}(\mathbf{q}_s)$, to each of its neighbors $s \in \mathcal{N}(t)$. Observe that each message passed to node s is a function of $\mathbf{q}_s$. The messages are then updated according to the recursion

$$M^{n+1}_{ts}(\mathbf{q}_s) = \kappa \max_{\mathbf{q}'_t} \left\{ \psi_{st}(\mathbf{q}_s, \mathbf{q}'_t) \psi_t(\mathbf{q}'_t) \prod_{u \in \mathcal{N}(t)/s} M^n_{ut}(\mathbf{q}'_t) \right\}, \tag{2}$$

where $\mathcal{N}(t)$ is the set of neighbors of node t in the graph $\mathcal{G}$. For any tree-structured graph, the message update equation (2) converges to a unique fixed point $\mathbf{M}^* = \{M^*_{st}\}$ after a finite number of iterations. The converged values of the messages $\mathbf{M}^*$ define the max-marginal at node s via

$$P_s(\mathbf{q}_s) \quad = \quad \kappa \, \psi_s(\mathbf{q}_s) \prod_{u \in \mathcal{N}(s)} M^*_{us}(\mathbf{q}_s).$$

For tree-structured problems, the max-product algorithms produces exact solutions with complexity $\mathcal{O}(m^2 N)$, in which m is the number of states per node. Notice that the message-passing update (2) is inherently a distributed algorithm and can be realized on physically distributed processors in parallel. The max-product algorithm is also applied frequently to graphs with cycles, as an approximate method. In the presence of cycles, the algorithm may not converge, and need not compute the MAP solution; however, see Ref. 18 for analysis of the quality of the approximate MAP solutions.

2.3. The Tree-Reweighted Max-Product Algorithm

In this section, we describe a modified version of the max-product algorithm that is guaranteed to give the correct answer, or to acknowledge failure. One way to describe this tree-reweighted max-product (TRMP) algorithm[5] is as a sequence of updates on trees of the graph, using the ordinary max-product algorithm as a subroutine. The basic idea is to represent the original problem on the graph with cycles as a convex combination of tree-structured problems. It can be shown[5] that whenever the tree problems all share an optimal configuration in common, this

*We use this notation throughout the paper, where the value of κ may change from line to line.

configuration must be the MAP configuration for the original problem. Based on this idea, the goal of the TRMP algorithm is to find a convex combination of tree-structured problems that share a common optimum.

Let $\vec{\mu} = \{\mu(\mathcal{T})\}$ be a probability distribution over a set of spanning trees $\{\mathcal{T}^i \mid i = 1, \ldots, L\}$ of the graph. For each edge $(s,t) \in E$, let $\mu_{st} = \Pr_{\vec{\mu}}[(s,t) \in \mathcal{T}]$ be the probability that edge (s,t) appears in a tree $\mathcal{T}$ chosen randomly under $\vec{\mu}$. We require a choice of $\vec{\mu}$ such that $\mu_{st} > 0$ for all edges of the graph. With this notation, the updates take the following form:

1. For each spanning tree $\mathcal{T}^i, i = 1, \ldots, L$, specify a set $\psi^i = \{\psi^i_s, \psi^i_{st}\}$ of compatibility functions for the tree via:
$$\psi^i_s = \psi_s \quad \forall s \in V, \qquad \psi^i_{st} = \left[\psi_{st}\right]^{\frac{1}{\mu_{st}}} \quad \forall (s,t) \in E(\mathcal{T}^i).$$
2. For each tree $\mathcal{T}^i, i = 1, \ldots, L$, run the max-product algorithm until convergence to obtain a set of tree max-marginals $\{P^i_s \mid s \in V\}$ and messages $\{M^i_{st} \mid (s,t) \in E(\mathcal{T}^i)\}$.
3. Check if all trees agree on the assignments produced by the max-product algorithm. If yes, output the assignment and stop. If not, update the compatibility functions by
$$\psi^{new}_s = \exp\left[\sum_i \mu(\mathcal{T}^i) \log P^i_s\right] \quad \forall s \in V, \qquad \psi^{new}_{st} = \exp\left[\sum_{\mathcal{T}^i} \mu(\mathcal{T}^i) \log \frac{\psi^i_{st}}{M^i_{st} M^i_{ts}}\right],$$
and return to step (1) with $\psi \equiv \psi^{new}$.

It can be shown that this algorithm always has a fixed point for positive compatibilities. The algorithm either outputs the correct MAP assignment, or it returns a set of pseudo-max-marginals for the graph that do not uniquely specify any configuration. More details on TRMP and its link to a tree-based linear programming relaxation can be found in Ref. 5.

3. PROBLEM STATEMENT AND SOLUTION

In this section we first outline the problem setup in our framework and our approach for solving the problem. Then we give a detailed explanation on how to construct graphical models for data association problems in Subsection 3.2. The general case with false alarms and missed detections is covered in Subsection 3.3. We conclude this section with a discussion on the complexity of our algorithm.

3.1. Outline of Problem Setup and Solution

We consider a planar region where N homogeneous sensors $\{s_1, s_2, \ldots, s_N\}$ are deployed to monitor the surveillance area. Each sensor has limited detection range and can only generate measurements for the targets falling into its range. In this paper we assume the measurements are 2-D Cartesian coordinates of targets' positions, however the ideas of our algorithm can be applied to other measurement models as well. In the surveillance area, there is a set of randomly distributed targets which are independent of each other, and the exact positions of these targets are unknown. It is clear that some targets will be observed by two or more sensors due to the overlapping coverage. Since the tracks have been established during most of the tracking process except for the first few frames of track initiation stage, it is reasonable to assume we already know which sensor is maintaining which targets' tracks, and thus assume that the coverage configuration of the targets relative to the sensors is known. Fig. 2 shows an example of such a configuration. We use $X_i = \{x_{ij} \mid j = 1, 2, \ldots, n_i\}$ to denote the set of n_i targets located in the range of sensor i. Each sensor i gives out a list of measurements $Z_i = \{z_{ij} \mid j = 1, 2, \ldots, m_i\}$ for the targets in X_i. The number of elements m_i in Z_i is not necessarily equal to n_i, due to possible false alarms and missed detections. The union of all the Z_i's composes the whole measurement set Z. Our task is to figure out the most probable way to assign each measurement in Z_i to at most one of the targets in X_i and guarantee each target x_{ij} gets no more than one measurement in Z_i. To satisfy this mutually exclusive assignment constraint, we fix the order of the targets in X_i and permute the measurements in Z_i to enumerate all the possible association configurations for each sensor i. Let $\mathcal{Q}_i$ be the ensemble of all such association configurations; $\mathbf{q}_i$ be the discrete random variable taking values in $\mathcal{Q}_i$; and $\mathbf{q}_{ij}$ be the j_{th}

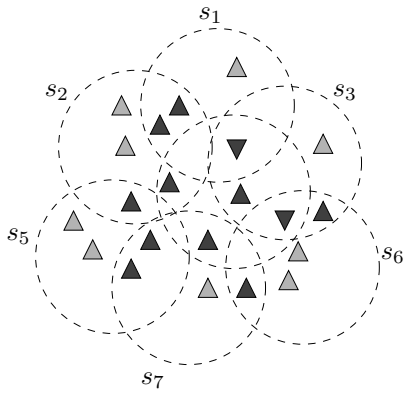

Figure 2. A coverage configuration of multitarget-multisensor data association. There are seven sensors (s_4 is the inner sensor unlabeled) with detection ranges shown as dashed circles. Targets are shown as $\triangle$ and $\triangledown$. Each of the gray $\triangle$ targets falls into the range of exactly one sensor. Black $\triangle$ targets are shared by 2 sensors and the two black $\triangledown$ targets are shared by 3 sensors. We assume this sensor-target coverage configuration is known.

element of the permutation. Let q_i be a realization of $\mathbf{q}_i$; it represents the association configuration in which we assign the measurement q_{ij} to target x_{ij} for $j = 1, 2, \ldots, n_j$. Let $\mathbf{q}$ be the vector formed by all $\mathbf{q}_i$; then $\mathbf{q}$ serves as the association operator: each state of $\mathbf{q}$ represents a valid association combination in all the sensors. Mathematically, the data association problem is to get the MAP estimate of $\mathbf{q}$ based on the observation $\mathbf{Z} = Z$:

$$\hat{\mathbf{q}}_{MAP} = \arg \max_{\mathbf{q} \in \mathcal{Q}^N} p(\mathbf{q} \mid \mathbf{Z} = Z).$$

Now it is clear that the TRMP algorithm can be applied to this problem if we can construct the underlying graph describing the structure of the joint distribution $p(\mathbf{q})$ and the pairwise compatibility functions. To construct the graph, we build one node for each association variable $\mathbf{q}_i$. If sensors i and j share some targets in their coverage, it turns out $\mathbf{q}_i$ and $\mathbf{q}_j$ are dependent and should be connected by an edge. Otherwise, if there are no targets shared by sensors i and j, $\mathbf{q}_i$ and $\mathbf{q}_j$ are conditionally independent given the association at a set of sensors that separate them (Markov property), as it will be explained in Section 3.2. In this way, we can convert the coverage configuration to an undirected graph. We will show how to define compatibility functions for the graph based on the likelihoods of the measurements Z in each association configuration and the prior distribution of the association variable $\mathbf{q}$. With this setup, we can then apply the TRMP algorithm to the associated MAP estimation problem.

3.2. Graphical Models for Data Association

In this section, we explain in detail how we construct the graph and the compatibility functions. To simplify our discussion and notation, we assume throughout this subsection that there are no false alarms and no missed detections. Without false alarms and missed detections, the measurements generated and the targets covered by each sensor should be in one-to-one correspondence, hence $n_i = m_i$ for all i. The association variable $\mathbf{q}_i$ for each sensor i has $n_i!$ different states. Each of these states can be acquired by a particular permutation of the elements in Z_i on X_i. In the following, we first work on the scenario where no targets are shared by more than 2 sensors, in which case we can get pairwise compatibility functions easily. In Subsection 3.2.2 we discuss how to deal with the case where some targets are shared by more than 2 sensors. We end this section with a short summary for our modeling approach in Subsection 3.2.3.

3.2.1. No targets shared by more than two sensors

In this section we assume a target can be seen by at most two sensors. We use X_i^j to denote the set of targets shared by sensor i and sensor j and use X_i^0 to denote the set of targets that can only be seen by sensor i, *i.e.*,

$$X_i^j = X_i \cap X_j, \quad \forall i, j \in \{1, 2, \ldots, N\} \text{ and } i \neq j,$$

$$X_i^0 = X_i \Big\backslash \bigcup_{\substack{j=1 \\ j \neq i}}^{N} X_i^j \ ,$$

Note that $X_i^j = X_j^i$. For each target $x_{ij} \in X_i$ lying in the range of sensor i, we have a random variable $\mathbf{x}_{ij}$ for its true position. All the $\mathbf{x}_{ij}$ form the prediction set $\mathbf{X}$. We assume each $\mathbf{x}_{ij}$ is independent and Gaussian distributed with mean x_{ij} and covariance matrix Σ_{ij}. The parameters x_{ij} and Σ_{ij} are known from the prediction step. Without loss of generality, we assume that the target positions all have the same covariance matix (*i.e.*, $\Sigma_{ij} = \Sigma$ for all (i, j)), then $\mathbf{x}_{ij} \sim \mathcal{N}(x_{ij}, \Sigma)$. Note that we use the same notation x_{ij} for targets and their mean positions. From now on we do not discriminate between the concept of a target and its mean position. For any z_{ij} in Z_i that actually comes from some target x_{ik} in X_i, z_{ij} is generated by the measurement model $\mathbf{z}_{ij} = \mathbf{x}_{ik} + \mathbf{v}_{ij}$, in which we assume all $\mathbf{v}_{ij}$ are independent identically distributed Gaussian noise with zero mean and covariance matrix Λ_{ij}, and all $\mathbf{v}_{ij}$ are independent of $\mathbf{x}_{ij}$. Again, we assume $\Lambda_{ij} = \Lambda$ for all i, j. Thus the likelihood that the observation $\mathbf{z}_{ij} = z_{ij}$ comes from target x_{ik} is

$$p(\mathbf{z}_{ij} = z_{ij} \mid \mathbf{z}_{ij} \text{ is associated with } x_{ik}) = \mathcal{N}(z_{ij}; x_{ik}, \Sigma + \Lambda). \tag{3}$$

Please note that any target in the set X_i^j produces two measurements, z_{ik} and z_{jl}, one at each of sensor i and sensor j. In this case, these two measurements are dependent, and the joint likelihood for these two measurements given the target's predicted mean position x_0 should be

$$p(z_{ik}, z_{jl} \mid \mathbf{z}_{ik}, \mathbf{z}_{jl} \text{ associated with } x_0) = \mathcal{N}\left(\begin{bmatrix} z_{ik} \\ z_{jl} \end{bmatrix}; \begin{bmatrix} x_0 \\ x_0 \end{bmatrix}, \begin{bmatrix} \Sigma + \Lambda & \Sigma \\ \Sigma & \Sigma + \Lambda \end{bmatrix} \right). \tag{4}$$

Recall each realization of the random permutation $\mathbf{q}_i$ is the list of measurements in Z_i in a certain order. Let Z_i^0 denote the set of measurements assigned to targets in X_i^0, *i.e.*, $Z_i^0 = \{\mathbf{q}_{ij} \mid \text{ any } j \text{ such that } 1 \leq j \leq m_i \text{ and } x_{ij} \in X_i^0\}$. Let Z_i^j denote the set of measurements assigned to targets in X_i^j, $Z_i^j = \{\mathbf{q}_{ik} \mid \text{ any } k \text{ such that } 1 \leq k \leq m_i \text{ and } x_{ik} \in X_i^j\}$. Note that Z_i^0 and Z_i^j are random sets. They have different elements for different realizations of $\mathbf{q}_i$, and $Z_i^j \neq Z_j^i$. With the values of each $\mathbf{q}_i$ uniformly distributed *a priori* (*i.e.*, before we observe Z_i), the MAP estimate is the same as the maximum likelihood estimate $\arg\max_q p(Z \mid \mathbf{q} = q)$. Note that when $\mathbf{q}$ is fixed

$$\begin{aligned} p(Z \mid \mathbf{q} = q) &= p(Z_1, Z_2, \ldots, Z_N \mid q_1, q_2, \ldots, q_N) \\ &= \prod_{i=1}^{N} p(Z_i^0 \mid q_i) \prod_{\substack{i=1, \\ j>i}}^{N} p(Z_i^j, Z_j^i \mid q_i, q_j) \\ &= \prod_{i=1}^{N} \left(\prod_{\substack{k \text{ s.t.} \\ z_{ik} \in Z_i^0}} p(z_{ik} \mid q_i) \right) \prod_{\substack{i=1, \\ j>i}}^{N} \left(\prod_{\substack{k,l \text{ s.t.} \\ z_{ik} \in Z_i^j, \ z_{jl} \in Z_j^i}} p(z_{ik}, z_{jl} \mid q_i, q_j) \right), \end{aligned} \tag{5}$$

and each $p(z_{ik} \mid q_i)$, $p(z_{ik}, z_{jl} \mid q_i, q_j)$ can be evaluated based on (3) and (4) respectively. Equation (5) also specifies the structure of the graphical models for the data association problem. We build one node for each $\mathbf{q}_i$ to represent the association in sensor i, and the state space of the node $\mathbf{q}_i$ is all the permutations of the measurements in Z_i on the targets in X_i. Two nodes are connected if the two corresponding sensors can observe a common target. In this way an undirected graph is constructed. Comparing (5) with (1), the node compatibility functions are defined based on the likelihoods of the measurements assigned to the targets only visible to this sensor; compatibility functions on the edges are defined based on the likelihoods of the measurements assigned to the targets visible to both sensors, *i.e.*,

$$\psi_i(q_i) = p(Z_i^0 \mid q_i), \quad \psi_{ij}(q_i, q_j) = p(Z_i^j, Z_j^i \mid q_i, q_j).$$

Then we can use the TRMP algorithm to infer the most probable state of each association variable $\mathbf{q}_i$.

3.2.2. Some targets covered by more than two sensors

The modeling approach described in Section 3.2.1 no longer applies if there is any target shared by more than two sensors. We illustrate the problem by a simple example in Fig. 3(a), in which two targets x_{c_1} and x_{c_2} are shared by all the three sensors. (However, the approach established in this section can handle the more complicated scenarios as well.) The previous modeling approach gives the model shown in Fig. 3(b), where the three nodes form a clique. In a certain association configuration q_1, q_2 and q_3, we have

$$p(Z_1, Z_2, Z_3 \mid q_1, q_2, q_3) = p(Z_1^0 \mid q_1)p(Z_2^0 \mid q_2)p(Z_3^0 \mid q_3)p(Z_1^C, Z_2^C, Z_3^C \mid q_1, q_2, q_3)$$

in which Z_i^C is the set of measurements assigned to targets x_{c_1} and x_{c_2} by the sensor i. The measurements assigned to x_{c_i} from each of the sensors are dependent, with the joint distribution:

$$\begin{bmatrix} z_{1c_i} \\ z_{2c_i} \\ z_{3c_i} \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} x_{c_i} \\ x_{c_i} \\ x_{c_i} \end{bmatrix}, \begin{bmatrix} \Sigma + \Lambda & \Sigma & \Sigma \\ \Sigma & \Sigma + \Lambda & \Sigma \\ \Sigma & \Sigma & \Sigma + \Lambda \end{bmatrix} \right). \tag{6}$$

Then we cannot factorize $p(Z_1^C, Z_2^C, Z_3^C \mid q_1, q_2, q_3)$ in terms of compatibility functions associated with only pairs of sensors. Instead, the probability density function (6) must contain a compatibility function for the whole clique. Unfortunately, the message passing implementation for the standard max-product algorithm requires pairwise compatibility functions, as in (1). Although there are extensions of the max-product algorithm that operate directly over higher-order cliques,[18] it is more convenient here to develop a description in terms of pairwise compatibilities. The basic idea is to introduce auxiliary random variables to describe the higher order interactions. Notice that

$$\begin{aligned} p(Z_1^C, Z_2^C, Z_3^C \mid q_1, q_2, q_3) &= \prod_{j=1,2} p(z_{1c_j}, z_{2c_j}, z_{3c_j} \mid q_1, q_2, q_3) \\ &= \prod_{j=1,2} p(q_{1c_j}, q_{2c_j}, q_{3c_j}). \end{aligned} \tag{7}$$

in which q_{ic_j} is the measurement assigned to target x_{c_j} by sensor i in the association q_i. If we define an association variable $\mathbf{q}_{c_i}$ by the relationship $\mathbf{q}_{c_i} = (\mathbf{q}_{1c_i}, \mathbf{q}_{2c_i}, \mathbf{q}_{3c_i})$, *i.e.*, each sample of $\mathbf{q}_{c_i}$ is a tuple of three measurements, which are the same as those measurements assigned to target x_{c_i} from Z_1, Z_2 and Z_3 in a particular combination of configurations $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$, then equation (7) suggests a way to factorize the clique compatibility functions. We can introduce two special nodes for $\mathbf{q}_{c_1}$ and $\mathbf{q}_{c_2}$ and make the two factors in equation (7) the compatibility functions of these two nodes respectively. Then we connect $\mathbf{q}_{c_i}$ with all $\mathbf{q}_j$ corresponding to the sensors that share target x_{c_i}, as shown in Fig. 3(c). Notice that for any specific combination q_1, q_2 and q_3, q_{c_i} is determined. This means any certain state of $\mathbf{q}_{c_i}$ is only consistent with some particular state combinations of $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$. For example, if $\mathbf{q}_{c_i}$ is on state q_{c_i} with a measurement z_{jk} from sensor j, then $\mathbf{q}_j$ cannot be on any states such that z_{jk} is assigned to any target other than x_{c_i}. Such an interaction should be reflected by the compatibility function on the edge between $\mathbf{q}_{c_i}$ and $\mathbf{q}_j$. So with the model in Fig. 3(c), the node compatibility functions are defined as

$$\psi_i(q_i) = p(Z_i^0 \mid q_1), \quad \psi_{c_i}(q_{c_i}) = p(Z_{c_i} \mid q_{c_i}),$$

and the edge compatibilities are 0-1 matrices such that

$$\psi_{jc_i}(q_j, q_{c_i}) = \begin{cases} 1 & \text{if } q_{jc_i} \in q_{c_i} \\ 0 & \text{otherwise.} \end{cases}$$

3.2.3. Modeling Approach Summary

In summary, we construct the graphical model for multitarget-multisensor data association as follows. We first build one node for each sensor to represent the association configurations of this sensor. Two nodes are connected by an edge if they share any target, and no third sensor covers this target. If there is any target shared by more than 2 sensors, then we build one node for each such target and connect it with all the nodes that represent the sensors which cover that particular target. As an example, Fig. 4 shows the graphical model for the scenario in Fig. 2. Although the graph contains cycles, the TRMP algorithm can be applied, and under certain conditions will yield the exact MAP estimate.

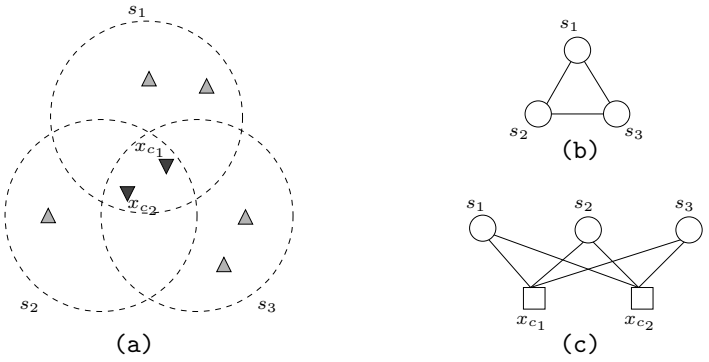

Figure 3. (a) A scenario where targets x_{c_1} and x_{c_2} are shared by three sensors. (b) Graphical model obtained using the method in Section 3.2.1. We cannot factorize the clique compatibility functions to pairwise compatibility functions. (c) The model with auxiliary random variables. The square nodes represent the association variables for x_{c_1} and x_{c_2}.

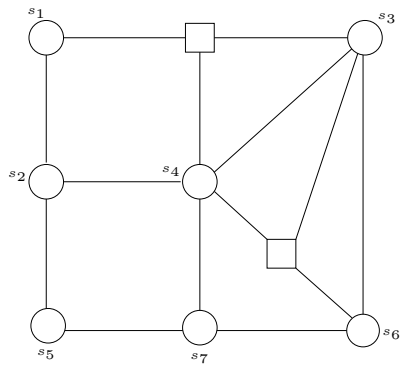

Figure 4. The graphical model for the example shown in Fig. 2. Circle nodes correspond to the sensors as labeled, square nodes are for the two $\bigtriangledown$ targets observed by more than 2 sensors.

3.3. False Alarms and Missed Detections

In this subsection we describe how to extend our approach to deal with false alarms and missed detections. The approach given here applies more generally; however, for the sake of notational simplicity, we limit the development here to the case where each target is observed by no more than two sensors. We assume that all targets are equally likely to be detected by the sensor covering them and the detection rate is P_D. Let the random variables $\mathbf{d}_i$ and $\mathbf{f}_i$ be the number of detected targets and false alarms for sensor i respectively. It is obvious that $0 \leq \mathbf{d}_i \leq n_i$, $0 \leq \mathbf{f}_i \leq m_i$ and $\mathbf{f}_i + \mathbf{d}_i = m_i$. We assume the probability mass function $P(\mathbf{f}_i)$ of $\mathbf{f}_i$ is known. For any $z_{ij} \in Z_i$ that comes from some target $x_{ik} \in X_i$, z_{ij} is generated by the measurement model in Equation (3). Otherwise, if z_{ij} is a false alarm, then we assume that it is generated by a uniform distribution over the detection range of sensor i, *i.e.*, $p(\mathbf{z}_{ij} = z_{ij} \mid \mathbf{z}_{ij}$ is a false alarm$) = U(1/A)$, where A is the area of the detection range of sensor i. In this case, we permute the measurements in Z_i on targets in X_i as follows. For each possible value of $\mathbf{f}_i = f_i$, take f_i measurements from Z_i as false alarms, and permute the remaining measurements on targets in X_i. The targets which get no measurement assigned are missed in this particular association configuration. Different association configurations may result in different numbers of false alarms and missed detections. Let us assume all configurations with same number of false alarms (and thus same number of missed detections) are equally likely *a priori* and let d_{q_i} and f_{q_i} be the number of detected targets and false alarms in a certain association configuration q_i. Then the prior distribution of $\mathbf{q}_i = q_i$ should be proportional to

$P(f_{q_i})P_D^{d_{q_i}}(1-P_D)^{n_i-d_{q_i}}$, and the posterior distribution of an association configuration q should be

$$\begin{aligned} p(\mathbf{q}=q \mid Z) &= \kappa\, p(Z_1, Z_2, \ldots, Z_N \mid q_1, q_2, \ldots, q_N)\, p(q) \\ &= \kappa \prod_{i=1}^{N} p(Z_i^0 \mid q_i) p(Z_i^F \mid q_i) P(f_{q_i}) P_D^{d_{q_i}} (1-P_D)^{(n_i-d_{q_i})} \prod_{\substack{i=1,\\ j>i}}^{N} p(Z_i^j, Z_j^i \mid q_i, q_j), \end{aligned} \tag{8}$$

where Z_i^F is the set of measurements treated as false alarms in association configuration q_i. Therefore, with false alarms and missed detections, the same modeling technique still applies, but the node compatibility functions should be updated with a prior term generated by false alarms and missed detections.

3.4. Algorithmic Complexity

We discuss the algorithmic complexity for the perfect detection case (no false alarms or missed detection). Assuming there are n targets in the range of each sensor, the node for each sensor will have $n!$ possible states. If a target is observed by $k > 2$ sensors, then the node for this target will have n^k states. If the graph is a tree with N nodes, the TRMP algorithm is the same as the max-product algorithm and has complexity $\mathcal{O}\left((n!)^2 N\right)$. On a graph with cycles, the complexity of the TRMP algorithm is $\mathcal{O}\left((n!)^2 N\right)$ per iteration. Unlike the tree case, the algorithm is not finitely convergent, and there are currently no upper bounds on the number of iterations required. However, for the data association problem, the compatibility matrices are inherently of low rank because they are generated by permutation and have some special structure. For example, for two sensors that share a single target in the coverage, the coupling matrix on the associated edge is $n!$ by $n!$; however, it can be shown that the rank of this matrix is only n, which limits the strength of the coupling. Based on experimental trials, the TRMP algorithm converges quickly when the compatibility matrices are of low rank.

4. EXPERIMENTAL RESULTS

We have tested the algorithm on simulated data. We use 25 sensors whose locations form a 5 by 5 grid on the 2-D plane. Each sensor's detection range overlaps with the range of its eight neighboring sensors. All sensors have the same detection range with radius $r = 5$. We assume the predicted target position has covariance matrix $\Sigma = \sigma^2 I$ and the measurement noise has covariance matrix $\Lambda = \lambda^2 I$. All sensors have roughly the same number of targets covered. We generate the predicted target positions and measurements as follows. In the range of each sensor, we randomly choose n points as the predicted mean positions of the targets. For each point, we generate a Gaussian random vector with the point as the mean and covariance matrix Σ, and use the vector generated as the true target position. Then for each sensor that can cover the target's mean position, we generate the measurement by a Gaussian random vector around the true target position with noise of zero mean and covariance Λ. We have conducted experiments with various levels of target density, measurement noise, and prediction error covariance in order to test the performance of our algorithm. Fig. 5 shows the results for the perfect detection case. Every data point is the average result of 50 trials. The results show that the measurement association error rate increases as the target density increases and the measurement noise increases. Increasing prediction error will also increase the association error in the high target density case but not for low target density. This is due to the fact that when the target density is low, most of the targets will be covered by 2 or more sensors. Even with large uncertainty on the position of the target, we can still recover the association for the target by combining its observations from several sensors. This result also exemplifies the importance of using multiple sensors to provide overlapping coverage. For the experiments with false alarms and missed detections, we fix the detection rate $P_D = 0.8$, and we constrain the number of false alarms generated by each sensor to be no more than one by assuming $P(\mathbf{f}_i)$ is a Bernoulli distribution with parameter $p = 0.15$. Fig. 6 shows the results with 50 trials. The curves have the same tendency as the perfect detection case, but the association error rate is noticeably higher. The TRMP algorithm converges quickly with around 10 iterations to the unique solution in all the experiments we have run.

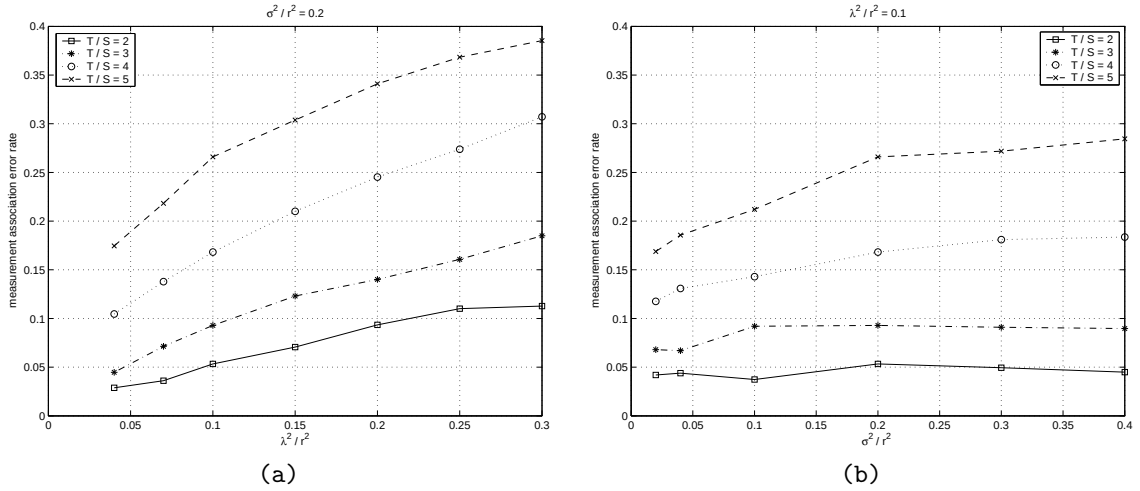

Figure 5. Measurement association error rate in the case of perfect detections. (a) Association error rate *vs.* the relative observation noise variance. The prediction variance is fixed at $\sigma^2 = 5$. (b) Association error rate *vs.* the relative prediction variance. The measurement variance is fixed at $\lambda^2 = 2.5$. In both plots, each of the four curves corresponds to a different average number of targets per sensor (T / S).

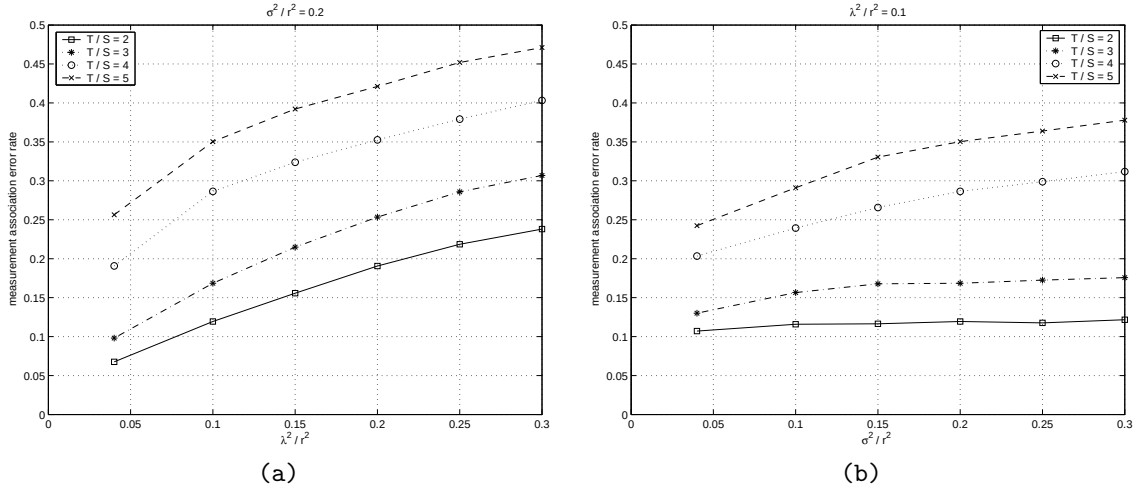

Figure 6. Measurement association error rate in the presence of missed detections and false alarms. $P_D = 0.8$, $P(\mathbf{f}_i) \sim Bernoulli(0.15)$. (a) Association error rate *vs.* the relative observation noise variance. The prediction variance is fixed at $\sigma^2 = 5$. (b) Association error rate *vs.* the relative prediction variance. The measurement variance is fixed at $\lambda^2 = 2.5$.

5. CONCLUSION

Multitarget-multisensor data association is a challenging problem, especially when the density of targets and sensors is high. The complexity of the problem grows exponentially in both the number of local targets and the number of sensors. In this paper, we proposed a distributed algorithm to obtain the globally optimal solution (in the MAP sense) for the data association problem. The work adopts the framework of graphical models and provides a perspective different from the traditional approaches. Our algorithm is distributed, in that it involves only passing messages from nodes adjacent in the network structure. With a complexity per iteration that grows linearly in the number of sensors, our techniques scale well in application to large sensor networks. An important direction for future research is extending the work described here to the dynamic setting. One challenge in a dynamic setting is that of incorporating deferred decisions into the framework of our algorithm. The research on

obtaining the m-best configurations in a graphical model[19] could be a viable solution. Finally, there are many environments in which globally optimal solutions are not possible (e.g., due to communication constraints), or an approximate solution is acceptable. It would also be interesting to develop efficient approaches for obtaining suboptimal solutions with specified performance guarantees for such cases.

ACKNOWLEDGEMENTS

This research was supported in part by the Army Research Office under Grant DAAD19-00-1-0466 and in part by the Air Force Office of Scientific Research under Grant F49620-01-1-0161 through subaward No. Z835101 from the Univ. of Maryland. The authors would like to thank Ayres Fan and Dmitry Malioutov for their contributions to this paper.

REFERENCES

1. Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*, Academic Press, Orlando, 1988.
2. S. S. Blackman, *Multiple-Target Tracking with Radar Applications*, Artech House, Dedham, MA, 1986.
3. S. Mori, K. C. Chang, and C. Y. Chong, "Performance analysis of optimal data association with applications to multiple target tracking," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., **2**, pp. 183–236, Artech House, Norwood, MA, 1992.
4. S. S. Blackman, "Association and fusion of multiple sensor data," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., **1**, pp. 187–218, Artech House, Norwood, MA, 1990.
5. M. J. Wainwright, T. Jaakkola, and A. S. Willsky, "MAP estimation via agreement on (hyper)trees: message-passing and linear programming approaches," in *Proc. Allerton Conference on Communication, Control and Computing*, (Monticello, IL), Oct. 2002.
6. D. B. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. on Automatic Control* **AC-24**, pp. 843–854, 1979.
7. A. Poore and N. Rijavec, "A Lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking," *SIAM Journal of Optimizations* **3**, pp. 544–563, Aug. 1993.
8. K. R. Pattipati, S. Deb, Y. Bar-Shalom, and R. B. Washburn, "Passive multisensor data association using a new relaxation algorithm," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., **1**, pp. 187–218, Artech House, Norwood, MA, 1990.
9. S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation," *IEEE Trans. on Aerospace and Electronic Systems* **33**, pp. 523–538, April, 1997.
10. K. C. Chang, C. Y. Chong, and Y. Bar-Shalom, "Joint probabilistic data association in distributed sensor networks," *IEEE Trans. on Automatic Control* **AC-31**, pp. 889–897, Oct. 1986.
11. C. Y. Chong, S. Mori, and K. C. Chang, "Distributed multitarget multisensor tracking," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, ed., **1**, pp. 247–295, Artech House, Norwood, MA, 1990.
12. M. E. Liggins, C. Y. Chong, I. Kadar, M. G. Alford, V. Vannicola, and S. Thomopoulos, "Distributed fusion architectures and algorithms for target tracking," *Proc. of IEEE* **85**, pp. 95–107, 1997.
13. M. I. Jordan, ed., *Learning in Graphical Models*, MIT Press, Cambridge MA, 1999.
14. W. T. Freeman, E. C. Pasztor, and O. T. Carmichael, "Learning low-level vision," *International Journal of Computer Vision* **40**, pp. 24–57, October 2000.
15. L. Rabiner and B. H. Juang, "A tutorial on hidden Markov models," *Proc. IEEE* **77**, pp. 257–286, 1989.
16. R. J. McEliece, D. J. Mckay, and J. F. Cheng, "Turbo decoding as an instance of Pearl's belief propagation algorithm," *IEEE Jour. Sel. Communication* **16**, pp. 140–152, February 1998.
17. P. Brémaud, *Markov Chains, Gibbs Fields, Monte Carlo Simulations, and Queues*, Springer-Verlag, 1991.
18. M. J. Wainwright, T. Jaakkola, and A. S. Willsky, "Tree consistency and bounds on the performance of the max-product algorithm and its generalizations," Tech. Rep. LIDS P-2554, Laboratory for Information and Decision Systems, MIT, Cambridge, MA, July 2002.
19. R. G. Cowell, A. P. Dawid, S. L. Lauritzen, and D. J. Spiegelhalter, *Probabilistic Networks and Expert Systems*, Springer-Verlag, 1999.