A Distributed Deterministic Approximation Algorithm for Data Association

Songhwai Oh
School of Electrical Engineering and Computer Science, ASRI
Seoul National University, Seoul, Korea
songhwai@snu.ac.kr

***Abstract*—The data association problem appears in many applications and is considered as the most challenging problem in intelligent systems. In this paper, we consider the Bayesian formulation of data association problems and present a deterministic polynomial-time approximation algorithm with guaranteed error bounds using correlation decay from statistical physics. We then show that the proposed algorithm naturally partitions a complex problem into a set of local problems and develop a distributed version of the algorithm. The performance of the proposed algorithm is evaluated in simulation.**

I. Introduction

Data association plays an important role in many applications, such as computer vision[1], robotics, sensor networks, information retrieval, and computer network security. In particular, data association is a fundamental problem in wireless sensor networks, as multiple sensors make measurements about simultaneous occurrences of multiple events and uncertainty as to the correct association between events and measurements is unavoidable. The data association problem appears in applications such as multi-target tracking, identity management, and localization of multiple events [1]–[4].

In this paper, we consider the Bayesian formulation of general data association problems based on joint probabilistic data association (JPDA) [1].[2] JPDA has proved very effective in cluttered environments compared to nearest neighbor filters [1]. JPDA is also frequently applied to computer vision [12] and robotics [13] and has been substantially extended [14]–[16]. JPDA assumes a fixed (or bounded) number of targets (or events) and computes Bayes estimates of the states. At each time step, instead of finding a single best association between measurements and tracks, JPDA enumerates all possible associations and computes association probabilities $\{\beta_{jk}\}$, where β_{jk} is the probability that the j-th measurement extends the k-th track. Given an association, the state of a target is estimated by a filtering algorithm and this conditional state estimate is weighted by the association probability. Then the state of a target is estimated by summing over the weighted conditional estimates. The exact calculation of association probabilities $\{\beta_{jk}\}$ in JPDA, which requires the summation over all association event probabilities, is #P-complete [17], [18].[3] Some heuristic approaches have been developed to avoid the enumeration of all possible associations, including "cheap" JPDA [20], "suboptimal" JPDA [21], "near-optimal" JPDA [15], and a leave-one-out heuristic [22]. Later, researchers found that Markov chain Monte Carlo is highly effective for computing approximations [12], [23]–[25]. Belief propagation and particle filters are also used for data association problems [3]. The Fourier transform based approximation for data association has recently been studied in [26]. But there has been no analytical work for these approximation algorithms, i.e., no approximation error bounds; the performance of an algorithm is often justified only in simulation. Although a fully polynomial randomized approximation scheme (FPRAS) for data association is presented in [27], there has been no deterministic approximation algorithm. The goal of this paper is to extend this line of work and present a deterministic approximation algorithm for data association with guaranteed error bounds.

Recently, there has been an exciting new development in computational complexity theory. There has been no known deterministic polynomial time approximation algorithm for #P-complete problems until very recently. But, now, there is a growing class of #P-complete problems for which there is a deterministic fully polynomial time approximation scheme (FPTAS). Examples include problems such as counting the number of independent sets and colorings of a graph [28], [29], counting the number of proper colorings of a graph [30], computing permanent of a 0-1 matrix [31], and counting the number of matchings [32]. Bayati et al. [32] derived a deterministic FPTAS for computing the number of matchings in a bounded degree graph using the *correlation decay* technique originating in statistical physics [33]. They have shown that there is a deterministic FPTAS algorithm for computing the *partition function* with time complexity $O((n/\epsilon)^\delta)$, where n is the number of vertices, $\epsilon > 0$ is the desired approximation ratio, and δ is a constant for a bounded degree graph.

In this paper, we present a deterministic polynomial-time approximation algorithm for data association using correlation decay. A data association problem can be thought of as a generalized version of the matching counting problem with variable edge weights. We extend work by Bayati et al. [32] to data association problems by showing that there is a deterministic FPTAS algorithm for data association problems.

[1] It is also known as *correspondence* in computer vision.

[2] Other notable data association algorithms include multiple hypothesis tracker (MHT) [5]–[7], probabilistic multi-hypothesis tracking [8], mixture reduction [9], multidimensional assignment [10], and probability hypothesis density [11].

[3] A #P-complete problem is computationally equivalent to computing the number of accepting computations of a polynomial-time nondeterministic Turing machine and #P contains NP [19].

To the best of our knowledge, the proposed algorithm is the first deterministic polynomial-time approximation algorithm for Bayesian data association problems. We notice that the proposed algorithm naturally decomposes a complex problem into a set of local problems, thanks to correlation decay. Based on this observation, we develop a distributed version of the algorithm without compromising performance. The distributed algorithm is suitable for sensor networks for its scalability. We have found that the analysis of correlation decay can reveal the effective neighborhood size for a complex problem and we think it will be an important tool for developing near-optimal distributed algorithms for a wide range of complex problems.

The remainder of this paper is structured as follows. We briefly describe data association and correlation decay in Section II and Section III, respectively. The deterministic approximation algorithm for data association is presented in Section IV with analysis of the algorithm. A distributed version of the deterministic approximation data association algorithm is developed for sensor networks in Section V. Simulation results are included in Section VI.

II. Data Association

In this section, we present a Bayesian formulation for general data association problems based on joint probabilistic data association (JPDA) [1].

Suppose there are N measurements and K events. Some measurements are from the K events while other measurements are false measurements (false alarms). It is also possible that there is no measurement for some events (missing detection). Let Y be the set of N measurements, $Y = \{y_1, \ldots, y_N\}$. We assume the availability of likelihood density functions $L(y_j|k)$ for all feasible measurement-event pairs. $L(y_j|k)$ describes the density of having measurement y_j when y_j is a measurement of event k. A measurement of an event is detected with probability p_{d} and, with probability $1-p_{\text{d}}$, there is no measurement about the event. There are also false alarms and the number of false alarms has a Poisson distribution with parameter $\lambda_{\text{f}} V$, where λ_{f} is the false alarm rate per unit volume and V is the volume of the region being monitored.

In general, we do not know precisely which measurement is generated by which event and the goal of data association is to figure out the association between measurements and events.[4] Once this data association problem is solved, the result can be used for joint state estimation of multiple events [1], [2], [12], [23].

Let Ω be the set of all feasible (joint) association events. For each $\omega \in \Omega$, $\omega = \{(j,k)\}$, where (j,k) denotes the event that measurement j is associated with event k. An association event ω is *feasible* when (i) y^j is within the detection range of event k, for each $(j,k) \in \omega$; (ii) a measurement is associated with at most one event; and (iii) an event is associated with at most one measurement. The second and third requirements are the mutual exclusiveness property of data association [1], [5], [26].

We encode a data association problem in a bipartite graph $G = (V_L, V_R, E)$, where $V_L = \{y^j : 1 \leq j \leq N\}$ is a vertex set of measurements, $V_R = \{k : 1 \leq k \leq K\}$ is a vertex set of events, and $(u,v) \in E$ if measurement u is within the detection range of event v. We call this graph a *data association graph* and an example is given in Figure 1. A feasible association event is a *matching* in G, i.e., a subset $M \subseteq E$ such that no two edges in M share a vertex. The set of all feasible association events Ω can be represented as $\Omega = \mathcal{M}_0(G) \cup \mathcal{M}_1(G) \cup \cdots \cup \mathcal{M}_K(G)$, where $M_k(G)$ is the set of all k-matchings[5] in G. Some examples of matchings or feasible association events are shown in Figure 1(c) and 1(d).

Under the Bayesian framework, in order to estimate the states of events $x_1, \ldots, x_K$, where x_k is the state of event k, we need to compute the posteriors $P(x_k|Y)$. Although $P(x_k|Y)$ cannot be computed directly since measurements are correlated, we can apply the total probability theorem and compute

$$\begin{aligned} P(x_k|Y) &= \sum_{\omega \in \Omega} P(x_k|\omega, Y) P(\omega|Y) \\ &= \sum_{j=0}^{N} \beta_{jk} P(x_k|\omega_{jk}, Y), \end{aligned} \tag{1}$$

where ω_{jk} denotes the event $\{\omega \in \Omega : (j,k) \in \omega\}$, ω_{0k} denotes the event that no observation is associated with event k, and β_{jk} is an *association probability*, such that

$$\beta_{jk} = P(\omega_{jk}|Y) = \sum_{\omega : (j,k) \in \omega} P(\omega|Y). \tag{2}$$

Hence, for the objective of state estimation, the data association problem reduces to the computation of $N \times K$ association probabilities. But the exact computation of association probabilities is NP-hard [17] and we present the first deterministic approximation algorithm for computing association probabilities in Section IV.

III. Correlation Decay

Bayati et al. [32] presented a deterministic fully polynomial time approximation scheme (FPTAS) for computing the total number of matchings in a bounded degree graph using *correlation decay*.

Given a graph $G = (V, E)$, where V and E denote the vertex set and the edge set of the graph, respectively, let $\mathcal{M} = \mathcal{M}(G)$ be the set of all matchings of G. The probability distribution defined on $\mathcal{M}$ can be represented as

$$P_G(M) = \frac{\lambda^{|M|}}{Z(G)}, \tag{3}$$

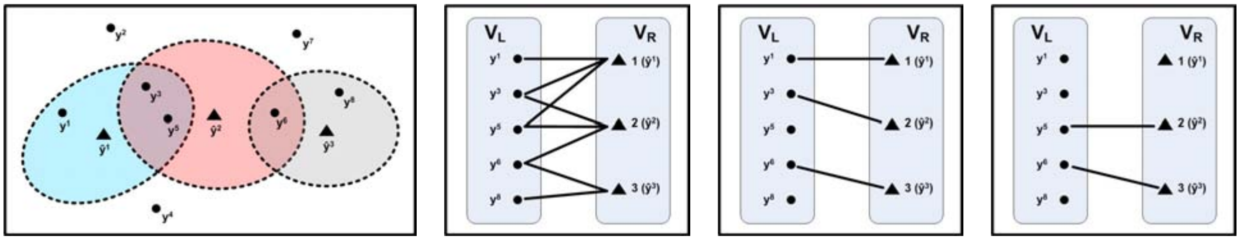
where λ is a fixed parameter called the activity, $Z(G)$ is the normalizing constant called the *partition function*, and $|M|$ denotes the number of edges in M. The partition function $Z(G)$ can be computed as

$$Z(G) = \sum_{M \in \mathcal{M}} \lambda^{|M|}. \tag{4}$$

Definition 1: An approximation algorithm $\mathcal{A}$ is defined to be a fully polynomial time approximation scheme (FPTAS)

[4] The association of measurement-event pairs is analogous to track-identity pairs in identity management, where $N = K$ [2], [26].

[5] A k-matching is a matching of cardinality k.



(a) (b) (c) (d)

Fig. 1. Graph theoretic interpretation of data association: an example. (a) In this 2D example, there are three events, $\hat{y}^1$, $\hat{y}^2$, and $\hat{y}^3$ (shown as a solid triangle). Measurements $\{y^j : j = 1, 2, \ldots, 8\}$ are shown as black dots. A measurement is within the detection range of an event if it is inside the shaded region centered at $\hat{y}^k$. (b) Data association graph $G = (V_L, V_R, E)$, which encodes the data association problem as a bipartite graph. An edge between $y^j \in V_L$ and $k \in V_R$ indicates that measurement y^j is within the detection range of k and $(y^j, k) \in E$. (c,d) Examples of matchings in G. A matching in G is an association event in the original data association problem. For example, the matching in (d) encodes an association event, in which no observation is associated with event 1, y^5 is associated with event 2, and y^6 is associated with event 3.

for computing $Z(G)$ if, given arbitrary $\epsilon > 0$, it produces a value $\hat{Z}$ satisfying

$$\exp(-\epsilon) \leq \frac{\hat{Z}}{Z(G)} \leq \exp(\epsilon),$$

in time which is polynomial in n, the number of vertices in G, and $1/\epsilon$.

The computation of $Z(G)$ is #P-complete and Bayati et al. showed that $Z(G)$ can be approximated deterministically in polynomial time by showing that $Z(G)$ can be computed using $P_G(v \notin \mathbf{M})$, the probability that a random matching chosen according to $P_G(M)$ does not contain vertex v (see Lemma 1), and this probability can be approximated from recursive computation of

$$\Phi_{\hat{G}}(v, t+1) = \frac{1}{1 + \lambda \sum_{u \in N(v, \hat{G})} \Phi_{\hat{G} \setminus \{v\}}(u, t)} \qquad (5)$$

for every subgraph $\hat{G}$ of G and for every vertex $v \in V(\hat{G})$, where $N(v, \hat{G})$ is the set of neighbors of v in $\hat{G}$ and $\hat{G} \setminus \{v\}$ is a subgraph of $\hat{G}$ induced by vertices $V \setminus \{v\}$. The concept of correlation decay was the main ingredient in their proof.

The main result of [32] is that the algorithm based on recursive computation of (5) is an FPTAS for estimating $Z(G)$ with complexity $O\left(\left(\frac{n}{\epsilon}\right)^{\kappa \log \Delta + 1}\right)$, where Δ is the maximum degree and $\kappa = -2/\log\left(1 - \frac{2}{\sqrt{1+\lambda\Delta}+1}\right)$.

IV. A Deterministic Approximation Algorithm for Data Association

In this section, we present an algorithm for deterministically approximating association probabilities in polynomial time using correlation decay. As shown in Section II, a feasible association event is a matching in a data association graph (see Figure 1). However, while every edge has the same contribution of λ in the computation of the partition function, the contribution of each edge is different when computing association probabilities due to the likelihood terms $L(y_j|k)$. Hence, it is unclear that the method proposed in [32] can be directly applied to estimate association probabilities. We show that it is possible and present a deterministic approximation algorithm for computing association probabilities using correlation decay.

Consider the data association graph $G = (V_L, V_R, E)$ for a data association problem described in Section II. Ω is a set of all feasible association events in G and $\Omega = \mathcal{M}(G)$, i.e., a set of all matchings in G. In order to compute the association probability in (2), we need to compute $P(\omega|Y)$. Based on the model given in Section II, it can be shown that [27]

$$P(\omega) \propto (\lambda_{\mathrm{f}} V)^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1 - p_{\mathrm{d}})^{K-|\omega|} \qquad (6)$$

and

$$\begin{aligned} P(\omega|Y) &= \frac{1}{Z_1} P(\omega) P(Y|\omega) \qquad (7) \\ &= \frac{1}{Z_2} \lambda_{\mathrm{f}}^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1 - p_{\mathrm{d}})^{K-|\omega|} \prod_{(u,v) \in \omega} L_{uv}, \end{aligned}$$

where Z_1 and Z_2 are normalizing constants and L_{uv} is the likelihood density $L(y_j|k)$ when $v = y_j$ and $u = k$.[6] We also let $L_{vu} := L_{uv}$ for convenience. We assume that the likelihood density function is bounded and there exist $L_{\max} < \infty$ and $L_{\min} > 0$, such that $L_{\min} \leq L(y|k) \leq L_{\max}$, for any y and k. This is a reasonable assumption with many density functions and a measurement validation method such as gating [1] (for example, see [34]). Since N and K are fixed, we can simplify (7) further and obtain

$$P(\omega|Y) = \frac{1}{Z(G)} \left(\lambda_{\mathrm{f}}^{-1} p_{\mathrm{d}} (1 - p_{\mathrm{d}})^{-1}\right)^{|\omega|} \prod_{(u,v) \in \omega} L_{uv}. \qquad (8)$$

Let $\alpha = \lambda_{\mathrm{f}}^{-1} p_{\mathrm{d}} (1 - p_{\mathrm{d}})^{-1}$. Then the normalizing constant $Z(G)$ can be computed as

$$Z(G) = \sum_{\omega \in \Omega} \alpha^{|\omega|} \prod_{(u,v) \in \omega} L_{uv}. \qquad (9)$$

Comparing to (4), $Z(G)$ has a similar form except the product of likelihood terms for the edges in ω.

Association probabilities β_{jk} that we seek in this paper can be computed from $Z(G)$ as shown in the following theorem.

Theorem 1: For an edge $(u, v) \in E$ of the data association graph $G = (V_L, V_R, E)$, we have

$$\beta_{uv} = \frac{Z(G \setminus \{u, v\})}{Z(G)} \alpha L_{uv}. \qquad (10)$$

[6] Recall p_{d} and λ_{f} from Section II.

Proof:

$$\begin{aligned}
\beta_{uv} &= \sum_{\omega\in\Omega:(u,v)\in\omega} P(\omega|Y) \\
&= \frac{1}{Z(G)} \sum_{\omega\in\Omega:(u,v)\in\omega} \alpha^{|\omega|} \prod_{(p,q)\in\omega} L_{pq} \\
&= \frac{1}{Z(G)} \left(\sum_{\omega'\in\Omega(G\setminus\{u,v\})} \alpha^{|\omega'|} \prod_{(u',v')\in\omega'} L_{u'v'} \right) \alpha L_{uv} \\
&= \frac{Z(G\setminus\{u,v\})}{Z(G)} \alpha L_{uv},
\end{aligned}$$

where the third equality is due to the fact that, for a matching ω' of $G\setminus\{u,v\}$, $\omega=\omega'\cup\{(u,v)\}$ is a matching in G, and vice versa. ∎

Let $P_G(v\notin\omega)$ be the probability that a randomly chosen association event ω does not contain a vertex $v\in V := V_L\cup V_R$. Let $G_k = G\setminus\{v_1,\dots,v_{k-1}\}$ for $k=1,2,\dots$ and $G_1=G$. Then, $Z(G)$ can be computed using a recursive set of $P_{G_k}(v_k\notin\omega)$ using the following well-known identity [19], [32].

Lemma 1: The following identity holds

$$Z(G) = \frac{1}{\prod_{1\le k\le|V|} P_{G_k}(v_k\notin\omega)}. \tag{11}$$

Hence, we can focus on approximating $P_{G_k}(v_k\notin\omega)$ for the computation of $Z(G)$. The following lemma shows how $P_{G_k}(v_k\notin\omega)$ can be computed.

Lemma 2: For every vertex $u\in V$, we have

$$P_G(u\notin\omega) = \frac{1}{1+\alpha\sum_{v\in N(u)} P_{G\setminus\{u\}}(v\notin\omega)L_{uv}}$$

and

$$P_G(u\notin\omega) \ge \frac{1}{1+\alpha\Delta L_{\max}},$$

where Δ is the maximum degree of the graph.

Proof: Since $P_G(u\notin\omega)$ has the same probability of randomly choosing $\omega\in\mathcal{M}(G\setminus\{u\})$ from $\Omega=\mathcal{M}(G)$,

$$P_G(u\notin\omega) = \frac{Z(G\setminus\{u\})}{Z(G)}. \tag{12}$$

For a vertex $u\in V$, we can decompose $Z(G)$ into two parts, where one part is a set of matchings in which a vertex u is not contained and the other part is a set of matchings containing the vertex u. Hence,

$$\begin{aligned}
Z(G) &= \sum_{\omega:u\notin\omega} \alpha^{|\omega|} \prod_{(u',v')\in\omega} L_{u'v'} \\
&+ \sum_{v\in N(u)} \sum_{\omega:(u,v)\in\omega} \alpha^{|\omega|} \prod_{(u',v')\in\omega} L_{u'v'} \\
&= Z(G\setminus\{u\}) + \sum_{v\in N(u)} Z(G\setminus\{u,v\})\alpha L_{uv},
\end{aligned}$$

since a matching containing the vertex u is a matching containing an edge (u,v) for some $v\in N(u)$ and $\omega\cup\{(u,v)\}$ is a matching in G containing the edge (u,v) for every matching ω in $G\setminus\{u,v\}$. Now we can divide both sides by $Z(G\setminus\{u\})$ and take the inverse to obtain the result using (12).

The second result of the lemma is a simple consequence of the first, noting that $|N(u)|\le\Delta$ for all $u\in V$. ∎

We now define $\Phi_{\hat{G}}(u,t)$ for every subgraph $\hat{G}$ of G and for every vertex $u\in V(\hat{G})$ and for $t\ge 0$ to approximate $P_{\hat{G}}(u\notin\omega)$. With $\Phi_{\hat{G}}(u,0)=1$ for all $\hat{G}$ and u, for $t\ge 1$, we define

$$\Phi_{\hat{G}}(u,t+1) = \frac{1}{1+\alpha\sum_{v\in N(u)}\Phi_{\hat{G}\setminus\{u\}}(v,t)L_{uv}}. \tag{13}$$

The idea is not to enumerate all exponentially many subgraphs of G but to compute only a small number of Φ to estimate $P_{\hat{G}}(u\notin\omega)$ using the parameter t to control the number of recursion levels. The algorithm for computing Φ is shown in Algorithm 1. If $\Phi_G(u,t)$ is a good approximation to $P_{\hat{G}}(u\notin\omega)$, we can apply Theorem 1 and Lemma 1 to devise an algorithm for computing association probabilities of a data association problem. The deterministic approximation algorithm for data association is shown in Algorithm 2, where $\hat{\beta}_{uv}$ and $\hat{Z}(G)$ are approximations to β_{uv} and $Z(G)$, respectively.

Algorithm 1 Correlation Decay

```
Input: $G$, $u$, $t$
Output: $\Phi_G(u,t)$
if $t=0$ or $|G|=0$ then
    Return $\Phi_G(u,t)=1$
end if
for all $v\in N(u)$ do
    if $t=1$ or $|G\setminus\{u\}|=0$ then
        $\Phi_s(v)=L_{uv}$
    else
        Compute $\Phi_{G\setminus\{u\}}(v,t-1)$ (Algorithm 1)
        $\Phi_s(v)=L_{uv}\Phi_{G\setminus\{u\}}(v,t-1)$
    end if
end for
$\Phi_G(u,t)=\left(1+\alpha\sum_{v\in N(u)}\Phi_s(v)\right)^{-1}$
```

Algorithm 2 Deterministic Approximation Data Association

```
Input: $t$
Output: $\{\hat{\beta}_{uv}\}$
for $k=1$ to $|V|$ do
    Compute $\Phi_{G_k}(v_k,t)$ (Algorithm 1)
end for
$\hat{Z}(G)=\left(\prod_{1\le k\le|V|}\Phi_{G_k}(v_k,t)\right)^{-1}$
for all $(u,v)\in E$ do
    $G'=G\setminus\{u,v\}$
    for $k=1$ to $|V(G')|$ do
        Compute $\Phi_{G'_k}(v'_k,t)$ (Algorithm 1)
    end for
    $\hat{Z}(G\setminus\{u,v\})=\left(\prod_{1\le k\le|V(G')|}\Phi_{G'_k}(v'_k,t)\right)^{-1}$
    $\hat{\beta}_{uv}=\frac{\hat{Z}(G\setminus\{u,v\})}{\hat{Z}(G)}\alpha L_{uv}$
end for
```

We now show that Algorithm 2 is an FPTAS for computing association probabilities. First, we bound the gap between

$P_G(v \notin \omega)$ and $\Phi_G(v,t)$ using correlation decay in the following theorem. The proof is given in Appendix IX-A.

Theorem 2: For every vertex $v \in V$ and every even $t > 0$, the following bound holds.

$$\left|\log P_G(v \notin \omega) - \log \Phi_G(v,t)\right| \\ \leq \left(1 - \frac{1}{1+\frac{L_{\text{max}}}{L_{\text{min}}}\sqrt{1+\alpha\Delta L_{\text{max}}}}\right)^{t/2} \log(1+\alpha\Delta L_{\text{max}}).$$

We now present the approximation error bounds on association probabilities. The proof is given in Appendix IX-B.

Theorem 3: Let $\delta = -\log\left(1 - \frac{1}{1+\frac{L_{\text{max}}}{L_{\text{min}}}\sqrt{\rho}}\right)$, where $\rho = 1+\alpha\Delta L_{\text{max}}$. If $t = 2\lceil \frac{\log \frac{2(N+K)\log\rho}{\epsilon}}{\delta} \rceil$ for $\epsilon > 0$, then

$$\exp(-\epsilon) \leq \frac{\hat{\beta}_{uv}}{\beta_{uv}} \leq \exp(\epsilon),$$

for any $(u,v) \in E$.

Theorem 3 gives us the desired error bound for being an FPTAS. Since the computational complexity of Algorithm 1 is $O(\Delta^t)$ and $t = O(\log(N+K))$ from Theorem 3, we can compute $\hat{\beta}_{uv}$ in polynomial time in $(N+K)$ and $1/\epsilon$ when $\Delta = O(1)$ or the degree of the graph is bounded. This fact is summarized in the next theorem.

Theorem 4: Algorithm 2 is an FPTAS when the degree of the data association graph is bounded.

V. A Distributed Data Association Algorithm for Sensor Networks

Consider a sensor network deployed over a bounded region. Let N_{s} be the number of sensors. For each sensor s_i, let R_i be its sensing region. Let $G_{\text{s}} = (V,E)$ be a *sensing graph*, where $V = \{s_i : 1 \leq i \leq N_{\text{s}}\}$ is a set of sensor nodes and $E = \{(s_i,s_j) : R_i \cap R_j \neq \emptyset\}$ is a set of sensor pairs with overlapping sensing regions. For a pair of nodes s_i and s_j, if $(s_i,s_j) \in E$, we assume that s_i and s_j can communicate with each other through direct or multi-hop communication.

The deterministic approximation data association algorithm presented in Section IV naturally decomposes a complex problem into a set of local problems. For the desired approximation ratio, we can choose an appropriate value for the parameter t in Algorithm 1. When estimating Φ, we only consider vertices that are at most t hops away on the data association graph. Hence, for a node to perform Algorithm 1, it only needs measurements from at most $\frac{t}{2}$-hop neighboring sensors in the sensing graph G_{s}, for even t. This idea is used to develop the distributed algorithm described below.

For node s_i, let $Y^i = \{y^i_1, \ldots, y^i_{j_i}\}$ be the measurements collected by s_i and $K^i = \{k^i_1, \ldots, k^i_{m_i}\}$ be the events that can be associated with a measurement in Y^i. If $s_{\text{id}}(y) = s_i$, then y is a measurement from sensor node s_i. The distributed deterministic approximation algorithm for data association is given in Algorithm 3. We assume t is an even number. Each node first broadcasts its measurements and event information (Y^i, K^i) to its $\frac{t}{2}$-hop neighbors of the sensing graph G_{s} and receives incoming measurements and event information from its $\frac{t}{2}$-hop neighbors. With information from $\frac{t}{2}$-hop neighbors, Algorithm 1 can be performed within each node. In order to compute $\hat{Z}$, it is required to compute factors Φ on a sequence of subgraphs, $G_1, G_2, \ldots, G_{|V|}$. For this, Algorithm 4 is used by passing the computation of the factor Φ on a subgraph to the corresponding sensor node. Although we have stated the computation of $\Phi_{G\setminus U}(v,t)$ in Algorithm 4, we do not need to know the overall structure of G since $\Phi_{G\setminus U}(v,t)$ can be computed with a subgraph of radius $\frac{t}{2}$ centered at s_i. One interesting consequence of this algorithm is that when the association graph is disconnected, each partition will be computed separately, making the algorithm more efficient and scalable.

Algorithm 3 Distributed Deterministic Approximation Algorithm for Data Association

```
Input: $t$
Output: $\{\hat{\beta}_{uv} : u \in Y^i, v \in K^i, (u,v) \in E\}$
$i$ = my node id
Send measurements to $\frac{t}{2}$-hop neighbors in $G_{\text{s}}$
Receive measurements from $\frac{t}{2}$-hop neighbors in $G_{\text{s}}$
$\hat{Z}(G) = \Phi_{\text{top}}(s_i, \emptyset, t)^{-1}$ (Algorithm 4)
for all $\{(u,v) \in E : u \in Y^i, v \in K^i\}$ do
    $\hat{Z}(G \setminus \{u,v\}) = \Phi_{\text{top}}(s_i, \{u,v\}, t)^{-1}$ (Algorithm 4)
    $\hat{\beta}_{uv} = \frac{\hat{Z}(G\setminus\{u,v\})}{\hat{Z}(G)} \alpha L_{uv}$
end for
```

Algorithm 4 Factor Computation

```
Input: $s_j$, $U$, $t$
Output: $\Phi_{\text{top}}(s_j, U, t)$
$i$ = my node id
$A = (Y^i \cup K^i) \setminus U$
if $A = \emptyset$ then
    $\Phi_{\text{top}}(s_j, U, t) = 1$
else
    Choose $v \in A$
    Compute $\Phi_{G\setminus U}(v,t)$ (Algorithm 1)
    if $v$ is a measurement then
        Send a request to $s_{\text{id}}(v)$ to compute $\Phi_{\text{top}}(s_i, U \cup \{v\}, t)$
    else
        Compute $\Phi_{\text{top}}(s_i, U \cup \{v\}, t)$
    end if
    $\Phi_{\text{top}}(s_j, U, t) = \Phi_{G\setminus U}(v,t) \cdot \Phi_{\text{top}}(s_i, U \cup \{v\}, t)$
end if
Return $\Phi_{\text{top}}(s_j, U, t)$ to $s_j$
```

VI. Simulation Results

We first study the performance of Algorithm 2. In order to compare its estimates against the exact values of β_{jk}, a small 2D localization problem is chosen. There are 6 events and 24 measurements (see Figure 2(a)). The predicted values of measurements of the events are

$$\{(0,0), (0,1), (0,-1), (1,0), (-1,0), (1,1)\}.$$

The likelihood density function $L(y_j|k) = \mathcal{N}(y_j|\hat{y}_k, \Sigma)$ is used for all measurement-event pairs, where $\mathcal{N}(\cdot|\hat{y}_k, \Sigma)$ is the

Gaussian density function with mean $\hat{y}_k$ and covariance Σ. A measurement y is validated for event k, i.e., the measurement-event pair (y, k) is feasible, if $(y - \hat{y}_k)^T \Sigma^{-1} (y - \hat{y}_k)^T < 4$. For this simulation, we used $p_{\mathrm{d}} = 0.8$, $\lambda_{\mathrm{f}} = 0.5$, and $\Sigma = \begin{bmatrix} 0.5 & 0 \\ 0 & 0.5 \end{bmatrix}$.

For this example, there are 8,517,625 association events. The exact values of β_{jk} are computed by enumerating all association events and it took over 40,000 seconds, i.e., over 11 hours. It was the largest example for which we could compute the exact association probabilities. A L_∞ error between β_{jk} and $\hat{\beta}_{jk}$ is used to measure the approximation error. Figure 2(b) shows the approximation error as a function of the parameter t. After $t = 9$, the approximation error is less than 0.005 and its running time is about 26% of the exact algorithm. If the desired error bound is at most 0.05, then $t = 6$ is good enough and the corresponding running time is 0.7% of the exact algorithm. This was a simple example and we expect the performance gap in terms of the running time will become widened for a larger size problem.

Next, we consider a sensor network of 100 nodes randomly placed over $[0, 1] \times [0, 1]$. There are ten events and 40 measurements. A total of ten cases are randomly generated to measure the average performance. An example is shown in Figure 3(a). For simplicity, we assumed that the sensing region of a sensor is a disk with radius 0.08. Note that the algorithm works for any shape of sensing regions as long as we know pairs of nodes with overlapping sensing regions. The likelihood density function used in this case is the same as the previous case except $\Sigma = \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix}$. We used $p_{\mathrm{d}} = 0.8$, $\lambda_{\mathrm{f}} = 0.1$, and measurement y is validated for event k if and only if $(y - \hat{y}_k)^T \Sigma^{-1} (y - \hat{y}_k)^T < 4$. The approximation error using L_∞ error is shown in Figure 3(b). For these examples, approximations at different neighborhood sizes are compared against the values estimated by Algorithm 2 using all sensors.

For each random case we considered, the sensing graph is connected and the maximum diameter of the graph is about 16 hops. Figure 3(b) shows how correlation decays as the number of hops is increased. It shows that the approximation error is close to zero after 6 hops, showing that we can make a good approximation using only local information from 6-hop neighbors. We think that the analysis of correlation decay for a complex problem can reveal how an efficient distributed approximation algorithm can be constructed for the problem.

VII. Conclusions

In this paper, we have presented the first deterministic approximation algorithm for a Bayesian data association problem. The problem is known to be #P-complete and there has been no known deterministic approximation algorithm with guaranteed error bounds. Applying the correlation decay technique, we have shown that the proposed algorithm is an FPTAS when the maximum degree of the association graph is bounded. This new formulation leads to natural partitioning of a complex problem into a set of local problems and a distributed algorithm is developed based on this property. We strongly believe that many interesting but difficult problems in wireless sensor networks can be reformulated using correlation decay to derive efficient distributed approximation algorithms.

VIII. Acknowledgments

This work has been supported in part by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education, Science and Technology (No. 2010-0027155).

IX. Appendix

A. Proof of Theorem 2

Our proof is based on the proof of Theorem 3.2 of [32] but it treats a more general case, where each edge is allowed to have a different weight αL_{uv}.

For a vertex $v \in V(G)$, denote by $N(v, G) = \{u_1, \ldots, u_m\}$ the neighborhood of v and $N(u_i, G \setminus \{v\}) = \{w_1^i, \ldots, w_{m_i}^i\}$ the neighborhood of u_i. We introduce the following variables to make our presentation compact. Let $x = \log P_G(v \notin \omega)$ and $y = \log \Phi_G(v, t)$. For $i = 1, \ldots, m$, let $x_i = \log P_{G\setminus\{v\}}(u_i \notin \omega)$ and $y_i = \log \Phi_{G\setminus\{v\}}(u_i, t - 1)$. Finally, for $i = 1, \ldots, m$ and $j = 1, \ldots, m_i$, let $x_i^j = \log P_{G\setminus\{v,u_i\}}(w_j^i \notin \omega)$ and $y_i^j = \log \Phi_{G\setminus\{v,u_i\}}(w_j^i, t - 2)$. Here, x's are the true probabilities and y's are the corresponding approximations.

Let $\mathbf{z} = (z_1^1, \ldots, z_{m_1}^1, \ldots, z_1^m, \ldots, z_{m_m}^m)^T$ and define $\mathbf{x}$ and $\mathbf{y}$ similarly. Define a functional f to capture two steps of recursions of Algorithm 1:

$$f(\mathbf{z}) = \log \left(1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{1 + \alpha \sum_{j=1}^{m_i} e^{z_j^i} L_{u_i w_j^i}} \right).$$

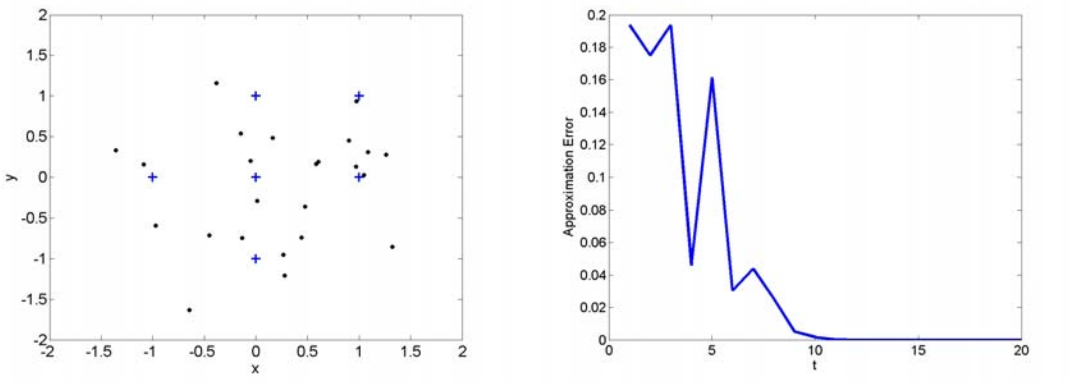
Then we have $x = -f(\mathbf{x})$ and $y = -f(\mathbf{y})$. Let $g(\lambda) = f(\lambda \mathbf{x} + (1 - \lambda)\mathbf{y})$ for $\lambda \in [0, 1]$. Using the mean value theorem, there exists λ such that, for $\mathbf{z}_\lambda = \lambda \mathbf{x} + (1 - \lambda)\mathbf{y}$,

$$\begin{aligned} |x - y| &= |\nabla f(\mathbf{z}_\lambda) \cdot (\mathbf{x} - \mathbf{y})| \\ &\leq \|\nabla f(\mathbf{z}_\lambda)\|_{L_1} \cdot \|\mathbf{x} - \mathbf{y}\|_{L_\infty}, \end{aligned} \tag{14}$$

where the inequality is due to Hölder's inequality. With $A_i = 1 + \alpha \sum_{j=1}^{m_i} e^{z_j^i} L_{ij}$ and $L_{ij} = L_{u_i w_j^i}$, we can write the L_1-norm as

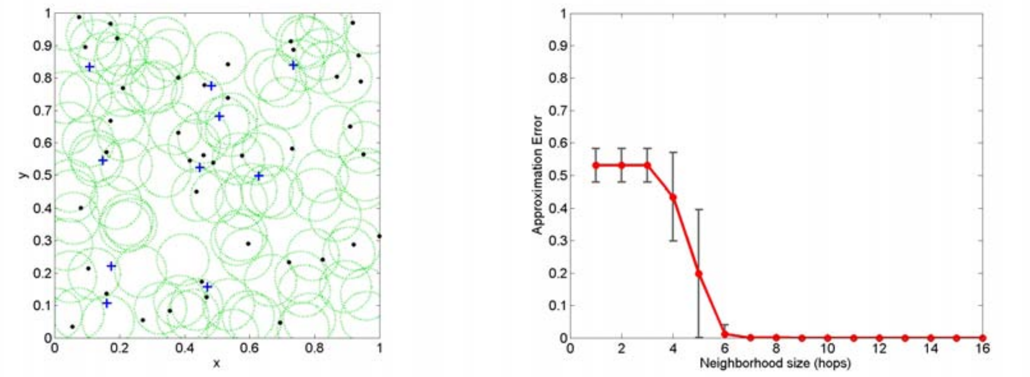
$$\begin{aligned} \|\nabla f(\mathbf{z}_\lambda)\|_{L_1} &= \frac{\alpha \sum_{i=1}^{m} L_{vu_i} \left(\frac{1}{A_i}\right)^2 \alpha \sum_{j=1}^{m_i} e^{z_j^i} L_{ij}}{1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i}} \\ &= \frac{\alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i^2} (A_i - 1)}{1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i}} \\ &= 1 - \frac{1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i^2}}{1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i}}. \end{aligned}$$

With some algebra, one can easily verify that $C := \frac{1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i^2}}{1 + \alpha \sum_{i=1}^{m} \frac{L_{vu_i}}{A_i}}$ is minimized for $0 \leq 1/A_i \leq \infty$ when $1/A_i = \frac{\sqrt{1 + \alpha \sum_{j=1}^{m_i} L_{ij}} - 1}{\alpha \sum_{j=1}^{m_i} L_{ij}}$.



(a) (b)

Fig. 2. A simple example of a data association problem with 6 events and 24 measurements. (a) The predicted values of measurements of events are shown as crosses and measurements are shown as dots. (b) The L_∞ approximation error as a function of the parameter t.



(a) (b)

Fig. 3. (a) An example of 100 sensor node deployment. Green dotted circles represent sensing regions of sensors, blue crosses are the predicted values of measurements of events, and black dots are measurements. (b) Approximation errors at different neighborhood sizes (in terms of the number of hops in the sensing graph). The solid line shows the average approximation error over 10 random cases and the error bar indicates one standard deviation.

What follows below is different from the proof of [32]. Let $B_i = \sum_{j=1}^{m_i} L_{ij}$. Then $\Delta L_{\text{min}} \le B_i \le \Delta L_{\text{max}}$. We also have the following inequalities for $1/A_i$.

$$\frac{\sqrt{1+\alpha\Delta L_{\text{min}}}-1}{\alpha\Delta L_{\text{max}}} \le \frac{1}{A_i} = \frac{\sqrt{1+\alpha B_i}-1}{\alpha B_i} \le \frac{\sqrt{1+\alpha\Delta L_{\text{max}}}}{\alpha\Delta L_{\text{min}}}.$$

Now we can apply these bounds to C.

$$\begin{aligned}
C &\ge \frac{1+\alpha\Delta L_{\text{min}}\left(\frac{\sqrt{1+\alpha\Delta L_{\text{min}}}-1}{\alpha\Delta L_{\text{max}}}\right)^2}{1+\alpha\Delta L_{\text{max}}\left(\frac{\sqrt{1+\alpha\Delta L_{\text{max}}}}{\alpha\Delta L_{\text{min}}}\right)} \\
&\ge \frac{\alpha\Delta L_{\text{min}}\left((\alpha\Delta L_{\text{max}})^2+\alpha\Delta L_{\text{min}}\left(\sqrt{1+\alpha\Delta L_{\text{min}}}-1\right)^2\right)}{(\alpha\Delta L_{\text{max}})^2\left(\alpha\Delta L_{\text{min}}+\alpha\Delta L_{\text{max}}\sqrt{1+\alpha\Delta L_{\text{max}}}\right)} \\
&\ge \frac{L_{\text{min}}L_{\text{max}}^2}{L_{\text{max}}^2\left(L_{\text{min}}+L_{\text{max}}\sqrt{1+\alpha\Delta L_{\text{max}}}\right)} \\
&= \frac{1}{1+\frac{L_{\text{max}}}{L_{\text{min}}}\sqrt{1+\alpha\Delta L_{\text{max}}}} =: D
\end{aligned}$$

We note that $D \in (0,1)$. Using this result, we can bound the L_1-norm as below

$$\|\nabla f(\mathbf{z}_\lambda)\|_{L_1} \le 1-D,$$

which is always less than one. We now apply this to (14) and get

$$\begin{aligned}
|x-y| &= \left|\log\frac{P_G(v\notin\omega)}{\Phi_G(v,t)}\right| \\
&\le (1-D)\max_{i,j}\left|\log\frac{P_{G\setminus\{v,u_i\}}(w_j^i\notin\omega)}{\Phi_{G\setminus\{v,u_i\}}(w_j^i,t-2)}\right|.
\end{aligned}$$

After iterating $t/2$ times, we obtain the following bound

$$\left|\log\frac{P_G(v\notin\omega)}{\Phi_G(v,t)}\right| \le (1-D)^{t/2}\max_{\hat{G},u}\left|\log P_{\hat{G}}(u\notin\omega)-\log\Phi_{\hat{G}}(u,0)\right|,$$

where the maximum is over all subgraph-vertex pairs, i.e., $\hat{G} \subset G$ and $u \in V(\hat{G})$. Since $\log\Phi_{\hat{G}}(u,0)=0$, using the second result of Lemma 2, we find that

$$\max_{\hat{G},u}\left|\log P_{\hat{G}}(u\notin\omega)-\log\Phi_{\hat{G}}(u,0)\right| \le \log(1+\alpha\Delta L_{\text{max}})$$

and this completes the proof.

B. *Proof of Theorem 3*

Using Theorem 2, we get

$$\begin{aligned}
&\left|\log \tfrac{\Phi_G(v,t)}{P_G(v\notin\omega)}\right| \\
&\quad \leq \log\rho \left(1 - \tfrac{1}{1+\frac{L_{\max}}{L_{\min}}\sqrt{\rho}}\right)^{\frac{\log \frac{2(N+K)\log\rho}{\epsilon}}{\delta}} \\
&\quad = \log\rho \exp\left(\log \tfrac{\epsilon}{2(N+K)\log\rho}\right) = \tfrac{\epsilon}{2(N+K)}.
\end{aligned}$$

Now using the fact that there are $N+K$ vertices in the data association graph and Lemma 1, we have

$$\exp\left(-\frac{\epsilon}{2}\right) \leq \frac{\hat{Z}(G)}{Z(G)} \leq \exp\left(\frac{\epsilon}{2}\right)$$

and, for any $(u,v) \in E$,

$$\exp\left(-\frac{\epsilon}{2}\right) \leq \frac{\hat{Z}(G\setminus\{u,v\})}{Z(G\setminus\{u,v\})} \leq \exp\left(\frac{\epsilon}{2}\right).$$

Using Theorem 1,

$$\frac{\hat{\beta}_{uv}}{\beta_{uv}} = \frac{\frac{\hat{Z}(G\setminus\{u,v\})}{\hat{Z}(G)}\alpha L_{uv}}{\frac{Z(G\setminus\{u,v\})}{Z(G)}\alpha L_{uv}} = \frac{\hat{Z}(G\setminus\{u,v\})}{Z(G\setminus\{u,v\})}\frac{Z(G)}{\hat{Z}(G)}.$$

Hence, we can conclude that

$$\exp(-\epsilon) \leq \frac{\hat{\beta}_{uv}}{\beta_{uv}} \leq \exp(\epsilon),$$

for any $(u,v) \in E$.

References

[1] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association*. San Diego, CA: Academic Press, 1988.

[2] J. Shin, L. Guibas, and F. Zhao, "A distributed algorithm for managing multi-target identities in wireless ad-hoc sensor networks," in *Proc. of the 2nd workshop on Information Processing in Sensor Networks*, April 2003, pp. 223–238.

[3] M. Chu, S. Mitter, and F. Zhao, "Distributed multiple target tracking and data association in ad hoc sensor networks," in *Proc. of the 6th International Conference on Information Fusion*, July 2004, pp. 447–454.

[4] R. R. Brooks, D. Friedlander, J. Koch, and S. Phoha, "Tracking multiple targets with self-organizing distributed ground sensors," *J. Parallel Distrib. Comput.*, vol. 64, pp. 874–884, 2004.

[5] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automatic Control*, vol. 24, no. 6, pp. 843–854, December 1979.

[6] T. Kurien, "Issues in the design of practical multitarget tracking algorithms," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Artech House, Norwood, MA, 1990.

[7] I. Cox and S. Hingorani, "An efficient implementation of Reid's multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 18, no. 2, pp. 138–150, 1996.

[8] R. Streit and T. Luginbuhl, "Maximum likelihood method for probabilistic multi-hypothesis tracking," in *Proc. SPIE*, vol. 2235, April 1994, pp. 394–405.

[9] L. Pao, "Multisensor multitarget mixture reduction algorithms for tracking," *Journal of Guidance, Control, and Dynamics*, vol. 17, no. 6, pp. 1205–1211, Nov. 1994.

[10] A. Poore, "Multidimensional assignment and multitarget tracking," in *Partitioning Data Sets*, I. J. Cox, P. Hansen, and B. Julesz, Eds. American Mathematical Society, 1995, pp. 169–196.

[11] R. Mahler, "A theoretical foundation for the stein-winter probability hypothesis density (phd) multi-target tracking approach," in *Proc. of the 2002 MSS Nat. Symp. Sensor Data Fusion*, San Antonio, TX, 2000.

[12] F. Dellaert, S. Seitz, C. Thorpe, and S. Thrun, "EM, MCMC, and chain flipping for structure from motion with unknown correspondence," *Machine Learning*, vol. 50, pp. 45–71, 2003.

[13] D. Schulz, W. Burgard, D. Fox, and A. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association," in *Proc. of the IEEE International Conference on Robotics and Automation*, 2001.

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

[15] J. Roecker, "A class of near optimal JPDA algorithms," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-30, 2, pp. 504–510, April 1994.

[16] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automatic Control*, vol. 33, pp. 780–783, 1988.

[17] J. Collins and J. Uhlmann, "Efficient gating in data association with multivariate distributed states," *IEEE Trans. Aerospace and Electronic Systems*, vol. 28, no. 3, pp. 909–916, July 1992.

[18] L. Valiant, "The complexity of computing the permanent," *Theoretical Computer Science*, vol. 8, pp. 189–201, 1979.

[19] M. Jerrum and A. Sinclair, "The Markov chain Monte Carlo method: An approach to approximate counting and integration," in *Approximations for NP-hard Problems*, D. Hochbaum, Ed. PWS Publishing, Boston, MA, 1996.

[20] R. Fitzgerald, "Development of practical PDA logic for multitarget tracking by microprocessor," in *Multitarget-Multisensor Tracking: Advanced Applications*, Y. Bar-Shalom, Ed. Artech House: Norwood, MA, 1990.

[21] J. Roecker and G. Phillis, "Suboptimal joint probabilistic data association," *IEEE Trans. Aerospace and Electronic Systems*, vol. AES-29, 2, pp. 510–517, April 1993.

[22] T. Huang and S. Russell, "Object identification in a Bayesian context," in *Proc. of the Fifteenth International Joint Conference on Artificial Intelligence*, Nagoya, Japan, Aug. 1997.

[23] H. Pasula, S. Russell, M. Ostland, and Y. Ritov, "Tracking many objects with many sensors," in *Proc. of the International Joint Conference on Artificial Intelligence*, Stockholm, 1999.

[24] N. Bergman and A. Doucet, "Markov chain Monte Carlo data association for target tracking," in *Proc. of the IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, Istanbul, Turkey, June 2000.

[25] S. Cong, L. Hong, and D. Wicker, "Markov-chain Monte-Carlo approach for association probability evaluation," *IEE Proceedings of Control, Theory and Applications*, vol. 151, no. 2, pp. 185–193, March 2004.

[26] J. Huang, C. Guestrin, and L. Guibas, "Fourier theoretic probabilistic inference over permutations," *Journal of Machine Learning (JMLR)*, vol. 10, pp. 997–1070, May 2009.

[27] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for multi-target tracking," *IEEE Trans. Automatic Control*, vol. 54, no. 3, pp. 481–497, March 2009.

[28] D. Weitz, "Counting independent sets up to the tree threshold," in *Proc. of the ACM Symposium on Theory of Computing*, Seattle, WA, May 2006.

[29] A. Bandyopadhyay and D. Gamarnik, "Counting without sampling: New algorithms for enumeration problems using statistical physics," in *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*, Miami, FL, 2006.

[30] D. Gamarnik and D. Katz, "Correlation decay and deterministic fptas for counting list-colorings of a graph," in *Proc. of the ACM-SIAM Symposium on Discrete Algorithms*, New Orleans, LA, 2007.

[31] ——, "A deterministic approximation algorithm for computing a permanent of a 0,1 matrix," Preprint, 2007. [Online]. Available: http://arxiv.org/abs/math.CO/0702039

[32] M. Bayati, D. Gamarnik, D. Katz, C. Nair, and P. Tetali, "Simple deterministic approximation algorithms for counting matchings," in *Proc. of the ACM Symposium on Theory of Computing*, San Diego, CA, June 2007.

[33] R. Dobrushin, "Prescribing a system of random variables by conditional distributions," *Theory of Probability and its Applications*, vol. 15, pp. 469–497, 1970.

[34] S. Oh and S. Sastry, "A polynomial-time approximation algorithm for joint probabilistic data association," in *Proc. of the American Control Conference*, Portland, OR, June 2005.