Markov Chain Monte Carlo Data Association for Multi-Target Tracking

Songhwai Oh, *Member, IEEE*, Stuart Russell, and Shankar Sastry, *Fellow, IEEE*

***Abstract*—This paper presents Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multi-target tracking in a cluttered environment. When the number of targets is fixed, the single-scan version of MCMCDA approximates joint probabilistic data association (JPDA). Although the exact computation of association probabilities in JPDA is NP-hard, we prove that the single-scan MCMCDA algorithm provides a fully polynomial randomized approximation scheme for JPDA. For general multi-target tracking problems, in which unknown numbers of targets appear and disappear at random times, we present a multi-scan MCMCDA algorithm that approximates the optimal Bayesian filter. We also present extensive simulation studies supporting theoretical results in this paper. Our simulation results also show that MCMCDA outperforms multiple hypothesis tracking (MHT) by a significant margin in terms of accuracy and efficiency under extreme conditions, such as a large number of targets in a dense environment, low detection probabilities, and high false alarm rates.**

***Index Terms*—Joint probabilistic data association (JPDA), Markov chain Monte Carlo data association (MCMCDA), multiple hypothesis tracking (MHT).**

I. Introduction

MULTI-TARGET tracking plays an important role in many areas of engineering such as surveillance [1], computer vision [2], [3], network and computer security [4], and sensor networks [5]. In the standard setup, some indistinguishable targets move continuously in a given region, typically independently according to a known, Markovian process. Targets arise at random in space and time, persist for a random length of time, and then cease to exist; the sequence of states that a target follows during its lifetime is called a *track*. The positions, or more generally partial states, of moving targets are measured, either at random intervals or, more typically, in periodic *scans* that measure the positions of all targets simultaneously. The position measurements are noisy and occur with detection probability less than one, and there is a noise background of spurious position reports, i.e., false alarms or clutter.

Manuscript received August 07, 2007; revised June 18, 2008. Current version published March 11, 2009. This work was supported in part by the National Science Foundation (EIA-0122599 and CCF-0424422), and the Defense Advanced Research Projects Agency (F33615-01-C-1895). Recommended by Associate Editor S. Dey.

S. Oh is with the Electrical Engineering and Computer Science, University of California, Merced, CA 95344 USA (e-mail: songhwai.oh@ucmerced.edu).

S. Russell and S. Sastry are with the Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, CA 94720 USA (e-mail: russell@eecs.berkeley.edu; sastry@eecs.berkeley.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TAC.2009.2012975

The essence of the multi-target tracking problem is to find tracks from the noisy measurements. Now, if the sequence of measurements associated with each target is known, multi-target tracking (at least under the assumption of independent motion) reduces to a set of state estimation problems, which, for the purposes of this paper, we assume to be straightforward. Unfortunately, the association between measurements and targets is unknown. The *data association* problem is to work out which measurements were generated by which targets; more precisely, we require a partition of measurements such that each element of a partition is a collection of measurements generated by a single target or clutter [6]. In the general case, uncertainty as to the correct association is unavoidable.

Multi-target tracking algorithms are often categorized according to the objective function that they purport to optimize:

- *Heuristic* approaches typically involve no explicit objective function. For example, the greedy nearest-neighbor filter (NNF) [1] processes the new measurements in some order and associates each with the target whose predicted position is closest, thereby selecting a single association after each scan. Although effective under benign conditions, the NNF gives order-dependent results and breaks down under more difficult circumstances.
- *Bayesian* approaches: (1) *Maximum a posteriori* (MAP) approaches find the most probable association, given the measurements made so far, and estimate tracks given this association. (2) *Bayes estimator* approaches estimate tracks by minimizing the posterior expected value of some risk function. When the mean squared error is used as a risk function, the Bayes estimator is an *minimum mean-square error* (MMSE) estimate. The MMSE estimates of tracks are computed by summing over all possible associations, weighted by their posteriors.

Tracking algorithms can also be categorized by the way in which they process measurements:

- *Single-scan* algorithms estimate the current states of targets based on their previously estimated states and the current scan of measurements. Single-scan algorithms typically use a highly simplified approximate representation of the posterior state estimate. (An exception is the Kalman filter for which the exact posteriors can be computed.)
- *Multi-scan* algorithms estimate the current states of targets based on their previously estimated states, multiple past scans and the current scan of measurements. They may revisit past scans when processing each new scan, and can thereby revise previous estimates in the light of new evidence.

MAP approaches include the well-known *multiple hypothesis tracking* (MHT) algorithm [7]. MHT is a multi-scan tracking

algorithm that maintains multiple hypotheses associating past measurements with targets. When a new set of measurements arrives, a new set of hypotheses is formed from each previous hypothesis. The algorithm returns a hypothesis with the highest posterior as a solution. MHT is categorized as a "deferred logic" method [8] in which the decision about forming a new track or removing an existing track is delayed until enough measurements are collected. MHT is capable of initiating and terminating a varying number of tracks and is suitable for autonomous surveillance applications. The main disadvantage of MHT in its pure form is its computational complexity since the number of hypotheses grows exponentially over time. Various heuristic methods have been developed to control this growth [7], [9], [10]; but these methods sacrifice the MAP property. Other MAP approaches have been tried besides MHT, including 0–1 integer programming [11] and multidimensional assignment [8]. As the latter reference shows, the underlying MAP data association problem is NP-hard, so we do not expect to find efficient, exact algorithms.

Bayes estimator approaches to solve data association problems are even less tractable than the MAP computation. Several "pseudo-Bayesian" methods have been proposed, of which the best-known is the *joint probabilistic data association* (JPDA) filter [1]. JPDA is a suboptimal single-scan approximation to the optimal Bayesian filter; it can also be viewed as an *assumed-density* filter in which the joint posterior distribution is approximated by a product of simpler distributions such as moment-matching Gaussian distributions. 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. JPDA has proved very effective in cluttered environments compared with NNF [1]. The exact calculation of association probabilities $\{\beta_{jk}\}$ in JPDA, which requires the summation over all association event probabilities, is NP-hard [12] since the related problem of finding the permanent of a matrix is #P-complete[1] [14]. Some heuristic approaches to approximate JPDA include a "cheap" JPDA algorithm [15], "suboptimal" JPDA [16] and "near-optimal" JPDA [17]. In [18], a single-scan data association problem is considered and a leave-one-out heuristic is developed to avoid the enumeration of all possible associations. Other Bayes estimator approaches include the probability hypothesis density (PHD) filter [19], [20], probabilistic multi-hypothesis tracking (PMHT) [21], and mixture reduction [22], to name a few.

The main contribution of this paper is the development of a real-time multi-target tracking method called Markov chain Monte Carlo data association (MCMCDA). Unlike MHT and JPDA, MCMCDA is a true approximation scheme for the optimal Bayesian filter; i.e., when run with unlimited resources, it converges to the Bayesian solution. As the name suggests, MCMCDA uses Markov chain Monte Carlo (MCMC) sampling instead of enumerating over all possible associations. MCMC was first used to solve data association problems by Pasula *et al.* [23], [24], who showed it to be effective for multi-camera traffic surveillance problems involving hundreds of vehicles. More recently, in [25], MCMC was used to approximate the association probabilities in JPDA and was shown to outperform Fitzgerald's cheap JPDA.[2] MCMCDA goes beyond these contributions by incorporating missing measurements, false alarms and an ability to initiate and terminate tracks, so that the algorithm can be applied to the full range of data association problems.

The paper has two main technical results. The first is a theorem showing that, when the number of targets is fixed, single-scan MCMCDA is a fully polynomial randomized approximation scheme for JPDA. More specifically, for any $\epsilon > 0$ and any $0 < \eta < 0.5$, the algorithm finds estimates within ratio ϵ with probability at least $1 - \eta$ in time complexity $O(\epsilon^{-2} \log \eta^{-1} N(N \log N + \log(\epsilon^{-1})))$, where N is the number of measurements. The theorem is based on the seminal work of Jerrum and Sinclair [13], who designed an MCMC algorithm for approximating the permanent of a matrix and developed new techniques for analyzing its rate of convergence. As mentioned earlier, the relationship between JPDA and computing the permanent was identified by Collins and Uhlmann [12]; the connection to the polynomial-time approximation theorems of Jerrum and Sinclair was first noticed by Pasula *et al.* [23]. Although our proof has the same structure as that of Jerrum and Sinclair, substantial technical work was required to complete the mapping from computing the permanent to solving JPDA, including the usage of gating conditions that ensure appropriate lower bounds on individual association probabilities. In addition, we present simulation results supporting our theoretical results.

Our second technical result is the complete specification of the transition structure for a multi-scan version of MCMCDA that includes detection failure, false alarms, and track initiation and termination. We prove that the resulting algorithm converges to the full Bayesian solution and present simulation results supporting the convergence result. We also provide the first extensive experimental investigation of MCMCDA's performance on classical data association problems. We demonstrate remarkably effective real-time performance of MCMCDA compared to MHT under extreme conditions, such as a large number of targets in a dense environment, low detection probabilities, and high false alarm rates. We also present an example in which MCMCDA runs 20 times faster than MHT while outperforming MHT.

The remainder of this paper is structured as follows. The multi-target tracking problem and its probability model are described in Section II. In Section III, the Markov chain Monte Carlo (MCMC) method is summarized. The single-scan MCMCDA algorithm is presented in Section IV along with the proof that it approximates JPDA in polynomial time. The multi-scan MCMCDA algorithm is described in Section V.

[1]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 [13].

[2]MCMC has also been used for problems that are roughly isomorphic to the data association problem, including state estimation in the switching Kalman filter [26] and stereo correspondence in computer vision [3].

II. Multi-Target Tracking

A. Problem Formulation

Let $T \in \mathbb{Z}^+$ be the duration of surveillance. Let K be the (unknown) number of objects that appear in the surveillance region $\mathcal{R}$ during the surveillance period. Each object k moves in $\mathcal{R}$ for some unknown duration $[t_i^k, t_{\rm f}^k] \subseteq [1, T]$. Each object arises at a random position in $\mathcal{R}$ at t_i^k, moves independently around $\mathcal{R}$ until $t_{\rm f}^k$ and disappears. At each time, an existing target persists with probability $1 - p_{\rm z}$ and disappears with probability $p_{\rm z}$. The number of objects arising at each time over $\mathcal{R}$ has a Poisson distribution with a parameter $\lambda_{\rm b} V$ where $\lambda_{\rm b}$ is the birth rate of new objects per unit time, per unit volume, and V is the volume of $\mathcal{R}$. The initial position of a new object is uniformly distributed over $\mathcal{R}$.

Let $F^k : \mathbb{R}^{n_x} \to \mathbb{R}^{n_x}$ be the discrete-time dynamics of the object k, where n_x is the dimension of the state variable, and let $x_t^k \in \mathbb{R}^{n_x}$ be the state of the object k at time t.[3] The object k moves according to

$$x_{t+1}^k = F^k(x_t^k) + w_t^k, \qquad \text{for } t = t_i^k, \ldots, t_{\rm f}^k - 1 \tag{1}$$

where $w_t^k \in \mathbb{R}^{n_x}$ are white noise processes.

The noisy observation (or measurement) of the state of the object is measured with a detection probability $p_{\rm d}$. With probability $1 - p_{\rm d}$, the object is not detected and we call this a missing observation. There are also false alarms and the number of false alarms has a Poisson distribution with a parameter $\lambda_{\rm f} V$ where $\lambda_{\rm f}$ is the false alarm rate per unit time, per unit volume. Let n_t be the number of observations at time t, including both noisy observations and false alarms. Let $y_t^j \in \mathbb{R}^{n_y}$ be the $j^{\rm th}$ observation at time t for $j = 1, \ldots, n_t$, where n_y is the dimension of each observation vector. Each object generates an observation at each sampling time if it is detected. Let $H^j : \mathbb{R}^{n_x} \to \mathbb{R}^{n_y}$ be the observation model. Then the observations are generated as follows:

$$y_t^j = \begin{cases} H^j(x_t^k) + v_t^j & \text{if the } j^{\rm th} \text{ observation is from } x_t^k \\ u_t & \text{otherwise} \end{cases} \tag{2}$$

where $v_t^j \in \mathbb{R}^{n_y}$ are white noise processes and $u_t \sim \text{Unif}(\mathcal{R})$ is a random process for false alarms.

The multi-target tracking problem is to estimate K, $\{t_i^k, t_{\rm f}^k\}$ and $\{x_t^k : t_i^k \le t \le t_{\rm f}^k\}$, for $k = 1, \ldots, K$, from observations.

B. Probability Model

In order to perform Bayesian inference on a multi-target tracking problem, we need first to specify the probability model for multi-target tracking. This section describes the probability model and derives a formula for computing the posterior (up to a normalizing constant).

Suppose that ω denotes a set of parameters of interest. ω is not directly observable. Instead, we make a set of measurements Y. The objective of Bayesian inference is to make probability statements about ω given Y. For this objective, we first need a *joint probability distribution* $P(\omega, Y)$. Then the *posterior* $P(\omega|Y)$ can be represented as below using Bayes rule:

$$P(\omega|Y) = \frac{P(\omega, Y)}{\int P(\omega, Y)\, d\omega} = \frac{P(Y|\omega)P(\omega)}{P(Y)} \tag{3}$$

where $P(\omega)$ is the *prior distribution* of ω and $P(Y|\omega)$ is the *likelihood* of Y given ω.

In the probability model for multi-target tracking, ω is an association event, i.e., a partition of measurements such that each element of the partition is a collection of measurements generated by a single target or clutter [6]. Since, in general, there is no closed-form formula for computing $P(Y)$, one can only compute $P(\omega|Y)$ up to a normalizing constant, which requires computation of $P(Y|\omega)$ and $P(\omega)$. For a fixed association event ω, $P(Y|\omega)$ can be computed by solving a set of single-target tracking problems. Hence, we focus our attention to the derivation of $P(\omega)$ in the remainder of this section.

We first define $\{\omega\}$ for all possible measurement sizes. Let μ be a nonnegative T-dimensional vector, i.e., $\mu = [\mu_1, \ldots, \mu_T]^T$, representing the possible numbers of measurements from $t = 1$ to $t = T$, where $\mu_t \in \mathbb{Z}^+ \cup \{0\}$. For each value of μ, define a set of measurement indices $\Upsilon_t^\mu = \{(t,1), (t,2), \ldots, (t, \mu_t)\}$ for $\mu_t > 0$, where (t, i) is an index to the $i^{\rm th}$ measurement at time t, and $\Upsilon_t^\mu = \emptyset$ for $\mu_t = 0$. Now let $\Upsilon^\mu = \cup_{t=1}^T \Upsilon_t^\mu$ be an index set to a set of measurements whose size matches μ and let the set $\{\Upsilon^\mu : \mu \in \mathbb{Z}^T\}$ contain all possible index sets.

For each μ, let Ω^μ be a collection of partitions of Υ^μ such that, for $\omega \in \Omega^\mu$, $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$, where τ_0 is a set of indices to false alarms and τ_k is a set of indices to measurements from target k, for $k = 1, \ldots, K$. More formally, $\omega \in \Omega^\mu$ is defined as follows:

1) $\omega = \{\tau_0, \tau_1, \ldots, \tau_K\}$;
2) $\bigcup_{k=0}^K \tau_k = \Upsilon^\mu$ and $\tau_i \cap \tau_j = \emptyset$ for $i \ne j$;
3) τ_0 is a set of indices to false alarms; and
4) $|\tau_k \cap \Upsilon_t^\mu| \le 1$ for $k = 1, \ldots, K$ and $t = 1, \ldots, T$;

Here, $K = K(\omega)$ is the number of tracks for the given partition $\omega \in \Omega^\mu$ and $|S|$ denotes the cardinality of the set S. We call τ_k a track when there is no confusion, although the actual track is a sequence of state estimates computed from the observations indexed by τ_k. (We assume there is a deterministic function that returns a sequence of estimated states given a set of observations, so no distinction is required.) The fourth requirement says that a track can have at most one observation at each time, but, in the case of multiple sensors with overlapping sensing regions, we can easily relax this requirement to allow multiple observations per track. For special cases in which $p_{\rm d} = 1$ or $\lambda_{\rm f} = 0$, the definition of Ω^μ can be adjusted accordingly.

Now let $\widetilde{\Omega} = \{\omega \in \Omega^\mu : \mu \in \mathbb{Z}^T\}$. Notice that $\mu = \mu(\omega)$ is a deterministic function of $\omega \in \widetilde{\Omega}$. In addition, we can compute the following numbers from $\omega \in \widetilde{\Omega}$:

- e_t, the number of targets present at time t with $e_0 = 0$;
- z_t, the number of targets terminated at time t with $z_1 = 0$;
- a_t, the number of new targets at time t;
- d_t, the number of detected targets at time t; and
- f_t, the number of false alarms at time t, $f_t = \mu_t - d_t$.

Since these numbers are deterministic functions of $\omega \in \widetilde{\Omega}$, we have $P(\omega) = P(\omega, \mathcal{N}) = P(\omega|\mathcal{N})P(\mathcal{N})$, where $\mathcal{N} =$

[3] We assume that targets are indistinguishable in this paper, but if observations include target type or attribute information, the state variable can be extended to include target type information.

$\{\mu_t, e_t, z_t, a_t, d_t : 1 \leq t \leq T\}$. Based on the target termination, target detection, new target arrival, and false alarm models described in Section II-A, we have

$$P(\mathcal{N}) = \prod_{t=1}^{T} \left[\binom{e_{t-1}}{z_t} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{e_{t-1}-z_t} \times \binom{e_{t-1}-z_t+a_t}{d_t} p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{e_{t-1}-z_t+a_t-d_t} \times \frac{(\lambda_{\mathrm{b}} V)^{a_t}}{a_t!} \exp(-\lambda_{\mathrm{b}} V) \frac{(\lambda_{\mathrm{f}} V)^{f_t}}{f_t!} \exp(-\lambda_{\mathrm{f}} V) \right]. \quad (4)$$

Since $\omega \in \widetilde{\Omega}$ with the same $\mathcal{N}$ are indistinguishable, i.e., invariant under permutation of target indices, they are exchangeable and we assign a uniform prior on them. Hence,

$$P(\omega|\mathcal{N}) \propto \prod_{t=1}^{T} \left[\binom{e_{t-1}}{z_t} \binom{e_{t-1}-z_t+a_t}{d_t} \binom{\mu_t}{d_t} \binom{d_t}{a_t} (d_t - a_t)! \right]^{-1}. \quad (5)$$

Combining (4) and (5), the prior $P(\omega)$ is

$$P(\omega) \propto \prod_{t=1}^{T} \frac{1}{\mu_t!} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{e_{t-1}-z_t} \times p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{e_{t-1}-z_t+a_t-d_t} \times (\lambda_{\mathrm{b}} V)^{a_t} (\lambda_{\mathrm{f}} V)^{\mu_t - d_t}. \quad (6)$$

We simplify (6) by letting $c_t = e_{t-1} - z_t$ be the number of targets from time $t-1$ that have not terminated at time t, and $g_t = e_{t-1} - z_t + a_t - d_t$ be the number of undetected targets. Then, the prior model (6) becomes

$$P(\omega) \propto \prod_{t=1}^{T} \frac{1}{\mu_t!} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{c_t} p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{g_t} (\lambda_{\mathrm{b}} V)^{a_t} (\lambda_{\mathrm{f}} V)^{f_t}. \quad (7)$$

Let $Y_t = \{y_t^j : j = 1, \ldots, n_t\}$ be all measurements at time t and $Y = \{Y_t : 1 \leq t \leq T\}$ be all measurements from $t = 1$ to $t = T$. Y_t can be considered as a vector with random ordering as indicated by the exchangeability of indices in (5). Applying Bayes rule, the posterior of $\omega \in \widetilde{\Omega}$ becomes:

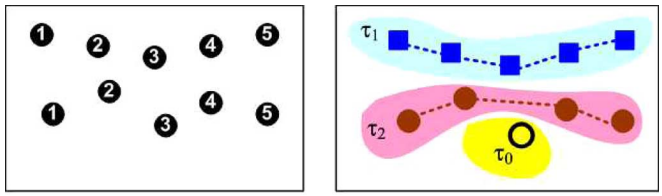
$$P(\omega|Y) \propto P(Y|\omega) P(\omega) \quad (8)$$

with $P(\omega)$ given in (7).

It is important to notice that $P(Y|\omega) = 0$ if $\mu(\omega) \neq n(Y)$, where $n(Y) = [n_1(Y), \ldots, n_T(Y)]^T$ denotes the number of measurements at each time in Y. Hence, we can restrict our attention to those $\omega \in \widetilde{\Omega}$ with $\mu(\omega) = n(Y)$. This crucial observation makes the numerous computations based on (8) practical. The set of all possible associations is now defined as $\Omega := \Omega^{n(Y)} = \{\omega \in \widetilde{\Omega} : \mu(\omega) = n(Y)\}$ and Ω is used instead of $\widetilde{\Omega}$ throughout this paper. Thus, it is convenient to view Ω as a collection of partitions of Y. An example of one such partition is shown in Fig. 1.

The posterior (8) can be further simplified as

$$P(\omega|Y) \propto P(Y|\omega) \times \prod_{t=1}^{T} p_{\mathrm{z}}^{z_t} (1-p_{\mathrm{z}})^{c_t} p_{\mathrm{d}}^{d_t} (1-p_{\mathrm{d}})^{g_t} (\lambda_{\mathrm{b}} V)^{a_t} (\lambda_{\mathrm{f}} V)^{f_t} \quad (9)$$



(a) (b)

Fig. 1. (a) Example of observations Y (each circle represents an observation and numbers represent observation times). (b) An example of a partition ω of Y.

where the term $\prod_{t=1}^{T} V^{a_t+f_t}$ will be canceled out by the matching initial state and false alarm densities in $P(Y|\omega)$. The likelihood $P(Y|\omega)$ can be computed based on the chosen dynamic and measurement models. For example, the computation of $P(Y|\omega)$ for linear dynamic and measurement models can be found in [27].

The posterior $P(\omega|Y)$ can be applied to both MAP and Bayes estimator approaches to solve the multi-target tracking problem. In the MAP approach, we first seek $\hat{\omega}$ such that

$$\hat{\omega} = \arg\max_{\omega \in \Omega} P(\omega|Y). \quad (10)$$

Then the states of the targets are estimated based on $\hat{\omega}$.

In the Bayes estimator approach, we look for Bayesian estimates of parameters. For instance, if we are interested in estimating the state x_t^k of target with the label k, the MMSE estimate of x_t^k is:

$$\hat{x}_t^k = \sum_{\omega \ni \tau_k} \int x_t^k P(dx_t^k|\omega, Y) P(\omega|Y). \quad (11)$$

Notice that it considers the contribution of all ω that contain a target with the label k when computing $\hat{x}_t^k$, whereas the MAP approach uses only $\hat{\omega}$. The method proposed in this paper (Algorithm 3) can be used to find both MAP and Bayes estimators to the multi-target tracking problem.

III. Markov Chain Monte Carlo

Markov chain Monte Carlo (MCMC) plays a significant role in many fields such as physics, statistics, economics, finance, and engineering [28]–[30]. The MCMC method includes algorithms such as Gibbs sampling [31] and the Metropolis-Hastings algorithm [32], [33]. Beichl and Sullivan described the Metropolis-Hastings algorithm as "the most successful and influential of all the members of … the *Monte Carlo Method*" [29]. MCMC techniques have been applied to complex probability distribution integration problems, counting problems, and combinatorial optimization problems [29]. In some cases, MCMC is the only known general algorithm that finds a good approximate solution to a complex problem in polynomial time [13].

MCMC is a general method to generate samples from a distribution π on a space Ω by constructing a Markov chain $\mathcal{M}$ with states $\omega \in \Omega$ and stationary distribution $\pi(\omega)$. We now describe an MCMC algorithm known as the Metropolis-Hastings algorithm. If $\mathcal{M}$ is at state $\omega \in \Omega$, $\omega' \in \Omega$ is proposed following

the proposal distribution $q(\omega, \omega')$. The move is accepted with an acceptance probability $A(\omega, \omega')$ where

$$A(\omega, \omega') = \min\left(1, \frac{\pi(\omega')q(\omega', \omega)}{\pi(\omega)q(\omega, \omega')}\right) \tag{12}$$

otherwise the sampler stays at ω. With this construction, the detailed balance condition is satisfied, i.e., for all $\omega, \omega' \in \Omega$,

$$Q(\omega, \omega') := \pi(\omega)P(\omega, \omega') = \pi(\omega')P(\omega', \omega) \tag{13}$$

where $P(\omega, \omega') = q(\omega, \omega')A(\omega, \omega')$ is the transition probability from ω to ω'. Hence, $\mathcal{M}$ is a reversible Markov chain.

If $\mathcal{M}$ is also irreducible and aperiodic, then $\mathcal{M}$ converges to its stationary distribution by the ergodic theorem [34]. Hence, for any bounded function f, the sample mean $\hat{f} = (1/N)\sum_{n=1}^{N} f(\omega^{(n)})$ converges to $\mathbb{E}_\pi f(\omega)$ as $N \to \infty$, where $\omega^{(n)}$ is the state of $\mathcal{M}$ at the n^{th} MCMC step and $\mathbb{E}_\pi f(\omega)$ is the expected value of $f(\omega)$ with respect to measure π. Notice that (12) requires only the ability to compute the ratio $\pi(\omega')/\pi(\omega)$, avoiding the need to normalize π, and this is why MCMC, especially the Metropolis-Hastings algorithm, can be applied to a wide range of applications.

An ergodic chain $\mathcal{M}$ on state space Ω converges to its stationary distribution asymptotically. But a practical question is how fast $\mathcal{M}$ approaches stationarity. One way to measure the rate of convergence of $\mathcal{M}$ to stationarity is the "mixing time" of the Markov chain. Let P be the transition probabilities of $\mathcal{M}$ and let $P_\omega^{(n)}(\cdot)$ be the distribution of the state at the n^{th} MCMC step given that $\mathcal{M}$ is started from the initial state $\omega \in \Omega$. If π is the stationary distribution of $\mathcal{M}$, then the *total variation distance* at the n^{th} MCMC step with initial state ω is defined as

$$\begin{aligned}\Delta_\omega(n) &= \|P_\omega^{(n)} - \pi\|_{\text{tv}} \\ &= \max_{S \subset \Omega} |P_\omega^{(n)}(S) - \pi(S)| \\ &= \frac{1}{2}\sum_{y \in \Omega} |P_\omega^{(n)}(y) - \pi(y)|. \end{aligned} \tag{14}$$

The rate of convergence of $\mathcal{M}$ to stationarity can be measured by the *mixing time*

$$\tau_\omega(\epsilon) = \min\{n : \Delta_\omega(s) \le \epsilon \text{ for all } s \ge n\}. \tag{15}$$

After the mixing time $\tau_\omega(\epsilon)$, $P_\omega^{(n)}(\cdot)$ for $n \ge \tau_\omega(\epsilon)$ is very close to the stationary distribution π.

One approach to bound $\tau_\omega(\epsilon)$ of a Markov chain with a complex structure is the canonical path method [13]. In this paper, the canonical path method is used to bound $\tau_\omega(\epsilon)$ of the Markov chain simulated by the MCMCDA algorithm given in Section IV. For the remainder of this section, we describe the canonical path method.

For a finite, reversible and ergodic Markov chain $\mathcal{M}$ with state space Ω, consider an undirected graph $G = (V, E)$ where $V = \Omega$ and $E = \{(x, y) : Q(x, y) > 0\}$ (recall the definition of $Q(\cdot, \cdot)$ from (13)). So an edge $(x, y) \in E$ indicates that the Markov chain $\mathcal{M}$ can make a transition from x to y or from y to x in a single step. For each ordered pair $(x, y) \in \Omega^2$, the canonical path γ_{xy} is defined as a simple path[4] from x to y in G. In terms of $\mathcal{M}$, the canonical path γ_{xy} is a sequence of legal transitions from x to y in $\mathcal{M}$. Let $\Gamma = \{\gamma_{xy} : x, y \in \Omega\}$ be the set of all canonical paths. Now the mixing time of the chain is related to the *maximum edge loading*:

$$\bar{\rho} = \bar{\rho}(\Gamma) = \max_e \frac{1}{Q(e)} \sum_{\gamma_{xy} \ni e} \pi(x)\pi(y)|\gamma_{xy}| \tag{16}$$

where $|\gamma_{xy}|$ denotes the length of the path γ_{xy}. If $\bar{\rho}$ is not so big, i.e., no single edge is overloaded, then the Markov chain can mix rapidly. The main result for the canonical path method is as follows [13], [35]:

Theorem 1: Let $\mathcal{M}$ be a finite, reversible, ergodic Markov chain with loop probabilities $P(x, x) \ge 1/2$ for all states x. Let Γ be a set of canonical paths with maximum edge loading $\bar{\rho}$. Then the mixing time of $\mathcal{M}$ satisfies $\tau_\omega(\epsilon) \le \bar{\rho}(\log \pi(x)^{-1} + \log \epsilon^{-1})$, for any choice of initial state ω.

IV. Single-Scan MCMCDA

In this section, we consider a special case of the multi-target tracking problem described in Section II, in which the number of targets K is known. We begin by specifying the optimal single-scan Bayesian filter, noting that it is infeasible to implement the optimal filter in practice due to its computational complexity. We then define the *assumed-density* single-scan Bayesian filter, which is frequently used in practice as an approximation to the optimal filter. The well-known JPDA filter [1] is an example of this approach. Although JPDA is less computationally demanding than the optimal single-scan Bayesian filter, the data association aspect remains intractable. We show, nevertheless, that the single-scan MCMCDA algorithm finds an ϵ-*good* approximate solution to JPDA in polynomial time.

A. Optimal Single-Scan Bayesian Filter

The (optimal) single-scan Bayesian filter for multi-target tracking is a recursive filtering algorithm in which each measurement is processed in turn and the posterior distribution of the current state is computed based on the current measurements and the posterior distribution computed at the previous scan. It resembles an ordinary recursive filter (e.g., a Kalman filter) for single-target tracking, but there are two major differences: first, the filter must compute the joint posterior of all target states, whose complexity grows without bound over time; second, the likelihood model for observations factors *conditional* on the (unknown) association variable, and so the exact update step must sum over exponentially many possible associations.

Let $X_t = (X_t^1, \ldots, X_t^K)$ be the joint state of all targets at time t. We assume the availability of the prior distribution $P(X_0)$. Now suppose that we are at time t and the optimal single-scan Bayesian filter has computed the posterior distribution $P(X_{t-1}|y_{1:t-1})$ from the previous scan time $t-1$, where $y_{1:t-1} = \{y_1, \ldots, y_{t-1}\}$ is a set of all past measurements. The optimal single-scan Bayesian filter computes $P(X_t|y_{1:t})$ from the new measurements y_t and the previous posterior distribution

[4] A simple path in a graph is a path with no repeated vertices.

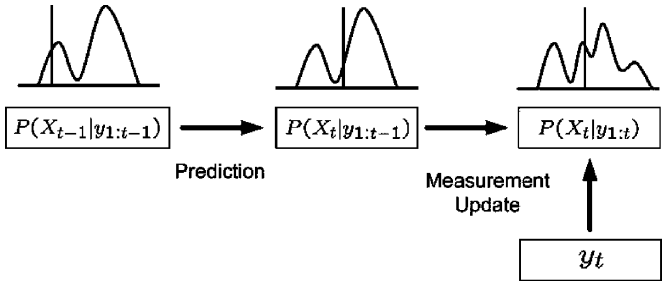


Fig. 2. Graphical illustration of the optimal single-scan Bayesian filter.

$P(X_{t-1}|y_{1:t-1})$ using the *prediction* and *measurement update* steps. For a graphical illustration, see Fig. 2.

1) Step 1 (Prediction): Using $P(x_{t-1}|y_{1:t-1})$, compute the distribution

$$P(X_t|y_{1:t-1}) = \int P(X_t|x_{t-1}, y_{1:t-1})P(x_{t-1}|y_{1:t-1})dx_{t-1} = \int P(X_t|x_{t-1})P(x_{t-1}|y_{1:t-1})dx_{t-1}, \quad (17)$$

where the Markovian assumption is used in the second equality and $P(X_t|x_{t-1})$ is determined by the dynamics model (1).

2) Step 2 (Measurement Update): In general, Bayes rule can be applied to compute the desired posterior

$$P(X_t|y_{1:t}) = \frac{P(y_t|X_t, y_{1:t-1})P(X_t|y_{1:t-1})}{\int P(y_t|x_t, y_{1:t-1})P(x_t|y_{1:t-1})dx_t}.$$

But, since we do not know which measurement is originated from which target, we cannot compute $P(y_t|X_t, y_{1:t-1})$ directly. Instead, we introduce a latent variable ω_t to represent a possible association between n_t measurements and K targets and we let $\Omega_t = \{\omega_t\}$ be a set of all possible association events at time t. The formal definition of Ω_t resembles the definition in Section II-B and is given in Section IV-B.

Using the total probability theorem, we can compute

$$P(X_t|y_{1:t}) = \sum_{\omega_t \in \Omega_t} P(X_t|\omega_t, y_{1:t})P(\omega_t|y_{1:t}) \quad (18)$$

where

$$P(X_t|\omega_t, y_{1:t}) = \frac{P(y_t|X_t, \omega_t, y_{1:t-1})P(X_t|y_{1:t-1})}{\int P(y_t|X_t, \omega_t, y_{1:t-1})P(X_t|y_{1:t-1})dx_t}.$$

Because ω_t specifies the association between measurements and targets, the term $P(y_t|X_t, \omega_t, y_{1:t-1})$ in (18) can be computed easily—according to the standard observation model, it simplifies to a product of independent likelihood factors, one per target/observation pair. Thus, we can compute $P(X_t|y_{1:t})$ in (18); unfortunately, the summation over ω_t has exponentially many terms. Not only is this summation intractable, but as a consequence the posterior representation becomes exponentially more complex at each time step.

B. Assumed-Density Single-Scan Bayesian Filter

Complexity in filtering problems is often addressed by a generic approach often called *assumed-density filtering* [36], in which the posterior state distribution is assumed to belong to a fixed family of density functions. Each exact update step typically takes the posterior outside this family, but a projection step then finds the best approximation within the family. Thus, an assumed-density single-scan Bayesian filter for multi-target tracking uses a simplified approximate representation for the posterior over the joint state of the targets. For example, one can make an (incorrect) independence assumption about the joint posterior, approximating it by a product of marginal distributions:

$$P(X_t|y_{1:t}) = P(X_t^1, \ldots, X_t^K|y_{1:t}) \approx \prod_{k=1}^{K} P(X_t^k|y_{1:t}). \quad (19)$$

This assumption is made by the JPDA filter, which has been traditionally used with the Kalman filter, assuming linear-Gaussian models, i.e., linear dynamic and measurement models and white Gaussian noise processes [1].[5] In fact, the assumed-density method with a factored posterior can be used with the general dynamics and measurement models defined in Section II. Furthermore, whereas JPDA sums over the association hypotheses in (18), the single-scan MCMCDA filter described in Section IV-C approximates this sum efficiently using MCMC.

The independence assumption made in (19) makes it possible to apply a standard recursive filtering update to each target separately, but only if conditioned on the (unknown) association between targets and observations. Thus, the measurement update step has to consider possible associations. This update step has two phases: first, *measurement validation* (another approximation) identifies the mappings between observations and targets that are considered plausible under some threshold; then *state update* computes the new approximate posterior distribution.

We begin with an (approximate) posterior distribution $\hat{P}(X_{t-1}^k|y_{1:t-1})$ computed from the previous time $t-1$, for each target k, where $\hat{P}(X_t^k|y_{1:t})$ approximates $P(X_t^k|y_{1:t})$ according to the independence assumption and the measurement validation step described below. At time t, the following three steps show how the assumed-density single-scan Bayesian filter computes $\hat{P}(X_t^k|y_{1:t})$ from the new measurements y_t and the previous posterior distribution $\hat{P}(X_{t-1}^k|y_{1:t-1})$.

1) Step 1 (Prediction): Similar to the prediction step (17) of the optimal single-scan Bayesian filter, for each k, we compute

$$\hat{P}(X_t^k|y_{1:t-1}) := \int P(X_t^k|x_{t-1}^k, y_{1:t-1})\hat{P}(x_{t-1}^k|y_{1:t-1})dx_{t-1}^k = \int P(X_t^k|x_{t-1}^k)\hat{P}(x_{t-1}^k|y_{1:t-1})dx_{t-1}^k. \quad (20)$$

[5]Recently, JPDA has also been applied to nonlinear problems using a particle filter [37]. Notice that when the dynamics and measurement model are nonlinear or the noise processes are non-Gaussian, the posterior distribution can be approximated using techniques such as linearization, unscented filtering [38], interacting multiple models [39], particle filters [40], or other numerical methods.

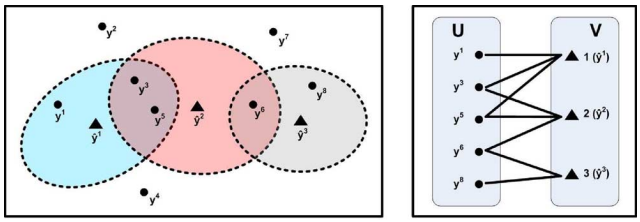


Fig. 3. (a) An example of measurement validation. For this 2D example, $\hat{P}^k(y|y_{1:t-1})$ is a Gaussian density with mean $\hat{y}^k$ for $k = 1, 2, 3$ (shown as a solid triangle). Measurements $\{y^j : j = 1, 2, \ldots, 8\}$ are shown as disks. A measurement is validated for target k if it is inside the shaded region centered at $\hat{y}^k$. (b) Measurement validation encoded as a bipartite graph $G = (U, V, E)$. An edge between $y^j \in U$ and $k \in V$ indicates that measurement y^j is validated for target k and $(y^j, k) \in E$. (The subscript t is omitted.).

2) Step 2 (Measurement Validation): Since there can be a large number of measurements and some measurements are very unlikely to have originated from a particular target, the state of each target is estimated from a subset of measurements. The process of selecting a subset of measurements for state estimation is called *measurement validation*. Let $\hat{P}^k(Y_t^j|y_{1:t-1})$ be the probability density of having observation Y_t^j given $y_{1:t-1}$, when Y_t^j is a measurement originated from target k. For each k and j, compute the distribution

$$\begin{aligned}\hat{P}^k(Y_t^j|y_{1:t-1}) &:= \int P(Y_t^j|x_t^k, y_{1:t-1})\hat{P}(x_t^k|y_{1:t-1})dx_t^k \\ &= \int P(Y_t^j|x_t^k)\hat{P}(x_t^k|y_{1:t-1})dx_t^k, \end{aligned} \tag{21}$$

where the second equality uses the fact the current observation is independent of previous observations given the current state and $P(Y_t^j|x_t^k)$ is determined by the measurement model (2). For linear-Gaussian models, $\hat{P}^k(Y_t^j|y_{1:t-1})$ is a Gaussian distribution and completely determined by its mean and variance. The measurement y_t^j is validated for target k if and only if

$$\hat{P}^k(y_t^j|y_{1:t-1}) \geq \delta^k \tag{22}$$

where δ^k are appropriate thresholds. An example of measurement validation is shown in Fig. 3(a).

3) Step 3 (State Estimation): As in the measurement update step in the optimal single-scan Bayesian filter, we introduce a latent variable ω to represent a feasible association between n_t measurements and K targets and we let $\Omega_t = \{\omega\}$ be a set of all feasible (joint) association events at time t. For each $\omega \in \Omega_t$, $\omega = \{(j, k)\}$, where (j, k) denotes the event that observation j is associated with target k. An association event ω is *feasible* when (i) for each $(j, k) \in \omega$, y_t^j is validated for target k; (ii) an observation is associated with at most one target; and (iii) a target is associated with at most one observation.

Let $N \leq n_t$ be the number of validated observations. We encode the feasible association events in a bipartite graph $G = (U, V, E)$, where $U = \{y_t^j : 1 \leq j \leq N\}$ is a vertex set of validated observations, $V = \{k : 1 \leq k \leq K\}$ is a vertex set of target indices, and $E = \{(u, v) : u \in U, v \in V, \hat{P}^v(u|y_{1:t-1}) \geq \delta^v\}$. An edge $(u, v) \in E$ indicates that observation u is validated for target v according to (22). An example of measurement validation encoded as a bipartite graph is shown in Fig. 3(b). A feasible association event is a *matching* in G, i.e., a subset $M \subset E$ such that no two edges in M share a vertex. The set of all feasible association events Ω_t can be represented as $\Omega_t = M_0(G) \cup \cdots \cup M_K(G)$, where $M_k(G)$ is the set of k-matchings in G. Some examples of matchings or feasible association events are shown in Fig. 4.

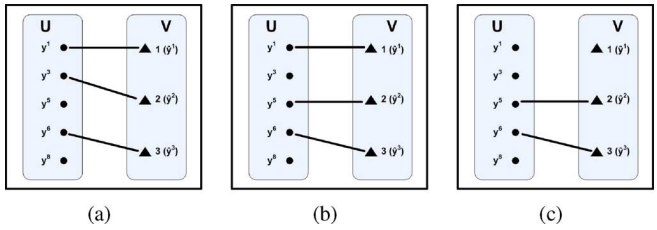


Fig. 4. Examples of matchings (feasible association events) based on the measurement validation example given in Fig. 3.

Now using the total probability theorem, we can compute the approximate distribution as:

$$\begin{aligned}\hat{P}(X_t^k|y_{1:t}) &:= \sum_{\omega \in \Omega_t} \hat{P}(X_t^k|\omega, y_{1:t})\hat{P}(\omega|y_{1:t}) \\ &= \sum_{j=0}^{N} \beta_{jk}\hat{P}(X_t^k|\omega_{jk}, y_{1:t}), \end{aligned} \tag{23}$$

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

$$\beta_{jk} = \hat{P}(\omega_{jk}|y_{1:t}) = \sum_{\omega:(j,k)\in\omega} \hat{P}(\omega|y_{1:t}). \tag{24}$$

$\hat{P}(X_t^k|\omega_{jk}, y_{1:t})$ in (23) can be easily computed by considering it as a single-target estimation problem with a single observation. On the other hand, the computation of β_{jk} requires a summation over exponentially many association events.

Notice that even when linear-Gaussian models are assumed, the posterior (23) is no longer a Gaussian distribution. As explained earlier, the posterior (23) becomes a mixture of posteriors from the previous time step. Hence, the complexity of the posterior distribution grows exponentially in multi-target tracking due to uncertainty in measurement-to-target associations. To combat this growth, an assumed-density filter projects back to a fixed family at each step. In particular, JPDA approximates the posterior for each target using a moment-matching Gaussian distribution, and assumes that the targets are all independent.

The exact calculation of $\{\beta_{jk}\}$ in JPDA is NP-hard [12] and this is the major drawback of JPDA. In the following sections, we describe the single-scan MCMCDA filter, which approximates the association probabilities $\{\beta_{jk}\}$, and prove that the running time of the algorithm is polynomial in the size of the problem.

C. Single-Scan MCMCDA Filter

The single-scan MCMCDA filter follows the same filtering steps as the assumed-density single-scan Bayesian filter

described in Section IV-B, except that the association probabilities $\{\beta_{jk}\}$ in (24) are approximated using MCMC. Since the filtering steps are already described in Section IV-B, this section describes only the method of approximation.

Based on the parametric false alarm model described in Section II-A and a derivation similar to that of (7), for each $\omega \in \Omega_t$, the prior $P(\omega)$ can be written as

$$P(\omega) \propto (\lambda_{\mathrm{f}} V)^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1-p_{\mathrm{d}})^{K-|\omega|}. \tag{25}$$

Then, the posterior of $\omega \in \Omega_t$ can be written as

$$\begin{aligned} P(\omega|y_{1:t}) &= \frac{1}{Z_0} P(\omega) P(y_t|\omega, y_{1:t-1}) \\ &\approx \frac{1}{Z} \lambda_{\mathrm{f}}^{N-|\omega|} p_{\mathrm{d}}^{|\omega|} (1-p_{\mathrm{d}})^{K-|\omega|} \prod_{(u,v)\in\omega} \hat{P}^v(u|y_{1:t-1}) \\ &=: \hat{P}(\omega|y_{1:t}) \end{aligned} \tag{26}$$

where Z_0 and Z are normalizing constants.

Algorithm 1 Single-scan-MCMCDA

1: INPUT $G = (U, V, E), n_{\mathrm{mc}}, n_{\mathrm{bi}}, \theta$.
2: OUTPUT $\{\hat{\beta}_{jk}\}$
3: $\hat{\beta}_{jk} = 0$ for all j and k
4: choose $\omega^{(0)}$ randomly from Ω_t
5: **for** $n = 1$ to n_{mc} **do**
6: $\quad \omega^{(n)} =$ Single-scan-MCMCDA.single-step
7: $\qquad (G, \omega^{(n-1)}, \theta)$ (see Algorithm 2)
8: $\quad$ **if** $n > n_{\mathrm{bi}}$ **then**
9: $\qquad$ **for each** $(y^j, k) \in \omega^{(n)}$ **do**
10: $\qquad\quad \hat{\beta}_{jk} = \hat{\beta}_{jk} + 1/(n_{\mathrm{mc}} - n_{\mathrm{bi}})$
11: $\qquad$ **end for**
12: $\quad$ **end if**
13: **end for**

The MCMC data association (MCMCDA) algorithm is an MCMC algorithm whose state space is the set of all feasible association events Ω_t and whose stationary distribution is the posterior $\hat{P}(\omega|y_{1:t})$ (26). The single-scan MCMCDA algorithm is shown in Algorithm 1, where $\theta = \{\{\hat{P}^v(u|y_{1:t-1})\}, \lambda_{\mathrm{f}}, p_{\mathrm{d}}, K, N\}$ along with its MCMC step described in Algorithm 2. The inputs to Algorithm 1 are the graph G, the number of samples n_{mc}, the number of burn-in samples n_{bi}, and θ. The input θ contains likelihoods $\{\hat{P}^v(u|y_{1:t-1})\}$ and model parameters λ_{f}, p_{d}, K, and N. Algorithm 1 computes the approximate association probabilities $\{\hat{\beta}_{jk}\}$, which can be used in (23) to compute the approximate posterior distribution $\hat{P}(X_t^k|y_{1:t})$. Algorithm 2 uses the MCMC transition rules from [13] and it describes how the MCMCDA algorithm updates its states. For the example given in Fig. 3, the move from Fig. 4(a) to Fig. 4(b) is a *switch move*. The move from Fig. 4(c) to Fig. 4(b) is an *addition move* while its reverse is a *deletion move*. However, the move from Fig. 4(a) to Fig. 4(c) is not a legal single-step move according to Algorithm 2. Since we have a uniform proposal distribution, $A(\omega, \omega') = \min(1, \pi(\omega')/\pi(\omega))$, where $\pi(\omega) = \hat{P}(\omega|y_{1:t})$ from (26).

Notice that, in line 2 of Algorithm 2, a self-loop transition probability of 1/2 is introduced to make the analysis easier (see [41, p.18] for more detail). In practice, however, the self-loop transition probability in line 2 can be set close to 0 for faster convergence.

Algorithm 2 Single-scan-MCMCDA.single-step

1: INPUT $G = (U, V, E), \omega, \theta$
2: OUTPUT ω
3: sample Z from $\mathrm{Unif}[0, 1]$
4: **if** $Z < 1/2$ **then**
5: $\quad \omega' = \omega$
6: **else**
7: $\quad$ choose $e = (u, v) \in E$ uniformly at random
8: $\quad$ **if** $e \in \omega$ **then**
9: $\qquad \omega' = \omega - e$ (deletion move)
10: $\quad$ **else if** both u and v are unmatched in ω **then**
11: $\qquad \omega' = \omega + e$ (addition move)
12: $\quad$ **else if** exactly one of u and v is matched in ω and e' is the matching edge **then**
13: $\qquad \omega' = \omega + e - e'$ (switch move)
14: $\quad$ **else**
15: $\qquad \omega' = \omega$
16: $\quad$ **end if**
17: **end if**
18: $\omega = \omega'$ with probability $A(\omega, \omega')$

D. Analysis

Let $\mathcal{M}$ be the Markov chain simulated by Algorithm 2. Since the self-loop probability is nonzero, $\mathcal{M}$ is aperiodic. It can be easily seen that $\mathcal{M}$ is irreducible, i.e., all states communicate, for example via the empty matching. In addition, since Algorithm 2 uses the Metropolis-Hastings kernel, the detailed balance condition (13) is satisfied and $\mathcal{M}$ is reversible. Hence, by the ergodic theorem, the chain converges to its stationary distribution [34].

We assume that each likelihood term in (26) can be bounded as $\underline{L} \le \hat{P}^v(u|y_{1:t-1}) \le \bar{L}$, for all $(u, v) \in E$. The lower bound $\underline{L} = \min \delta^k$ is guaranteed by measurement validation. In JPDA, measurement validation is used to reduce the number of feasible association events. However, we will see

that it is also required in the proof of polynomial-time approximation. The upper bound $\bar{L}$ can be precomputed based on $\hat{P}^v(u|y_{1:t-1})$. Here, we are making the reasonable assumption that $\hat{P}^v(u|y_{1:t-1}) \leq \bar{L} < \infty$ for all $(u,v) \in E$. (An example of $\bar{L}$ for linear-Gaussian models can be found in [42].)

The following theorems show that the single-scan MCMCDA algorithm provides a fully polynomial randomized approximation scheme for JPDA. The proof can be found in the longer version [43] of this paper.

Theorem 2: Suppose that $\lambda_{\rm f} > 0$ and $0 < p_{\rm d} < 1$. Then the mixing time of the Markov chain $\mathcal{M}$ is bounded by

$$\tau_\omega(\epsilon) \leq 4R^4K^2N(m_0(K,N) + \log \epsilon^{-1})$$

for all $\omega \in \Omega_t$, where

$$\begin{aligned}
R &= \max\left\{1, \frac{p_{\rm d}\bar{L}}{\lambda_{\rm f}(1-p_{\rm d})}, \frac{\lambda_{\rm f}(1-p_{\rm d})}{\underline{L}p_{\rm d}}\right\} \\
m_0(K,N) &= K\log\frac{m_1}{m_2} + \log\frac{m_3(K,N)}{m_4(K,N)} \\
&\quad + \sum_{k=1}^{K+1}\log k + \sum_{n=1}^{N}\log n \\
m_1 &= \max\{1, \bar{L}\} \\
m_2 &= \min\{1, \underline{L}\} \\
m_3(K,N) &= \max_{0\leq k\leq K}\{\lambda_{\rm f}^{N-k}p_{\rm d}^k(1-p_{\rm d})^{K-k}\} \\
m_4(K,N) &= \min_{0\leq k\leq K}\{\lambda_{\rm f}^{N-k}p_{\rm d}^k(1-p_{\rm d})^{K-k}\}
\end{aligned}$$

Remark 1: If $0.5 < p_{\rm d} < 1$ and $\lambda_{\rm f} < 1 - p_{\rm d}$, then $m_3(K,N) = \lambda_{\rm f}^{N-K}p_{\rm d}^K$ and $m_4(K,N) = \lambda_{\rm f}^N(1-p_{\rm d})^K$. So $m_3(K,N)/m_4(K,N) = (p_{\rm d}/\lambda_{\rm f}(1-p_{\rm d}))^K$ and K is the only remaining exponent.

Remark 2: Let $\bar{\tau}(\epsilon)$ be the upper bound found in Theorem 2. $\bar{\tau}(\epsilon)$ is polynomial in K and N. If $m_3(K,N)/m_4(K,N)$ does not grow fast, e.g., Remark 1, $\bar{\tau}(\epsilon) = O(K^2N(K\log K + N\log N + \log\epsilon^{-1}))$. If K is fixed, $\bar{\tau}(\epsilon) = O(N(N\log N + \log\epsilon^{-1}))$.

Let $p(\omega)$ be the distribution of the states of $\mathcal{M}$ after simulating Algorithm 2 for at least $\bar{\tau}(\epsilon)$ steps. Then the total variation distance satisfies $\|p - \pi\|_{\rm tv} \leq \epsilon$. So we can sample from p to estimate $\{\beta_{jk}\}$. However, there is a small bias in our estimates since we are not sampling directly from π. The following theorem gives an upper bound on the number of samples needed for finding good estimates. The proof can be found in [43].

Theorem 3: Let $0 < \epsilon_1, \epsilon_2 \leq 1$ and $0 < \eta < 0.5$. Suppose that $\|p - \pi\|_{\rm tv} \leq \epsilon$ for $\epsilon \leq \epsilon_1\epsilon_2/8$. Then, with a total of $504\epsilon_1^{-2}\epsilon_2^{-1}\lceil\log\eta^{-1}\rceil$ samples from p, we can find estimates $\hat{\beta}_{jk}$ for β_{jk} with probability at least $1-\eta$, such that, for $\beta_{jk} \geq \epsilon_2$, $\hat{\beta}_{jk}$ estimates β_{jk} within ratio $1+\epsilon_1$, i.e., $(1-\epsilon_1)\beta_{jk} \leq \hat{\beta}_{jk} \leq (1+\epsilon_1)\beta_{jk}$, and, for $\beta_{jk} < \epsilon_2$, $|\hat{\beta}_{jk} - \beta_{jk}| \leq (1+\epsilon_1)\epsilon_2$.

Remark 3: Following Remark 2, for fixed K, $\bar{\tau}(\epsilon) = O(N(N\log N + \log\epsilon^{-1}))$. Combining this fact with Theorem 3, the time complexity of the overall procedure is

$$\tilde{n}_{\rm mc} = O(\epsilon_1^{-2}\epsilon_2^{-1}\log\eta^{-1}N(N\log N + \log(\epsilon_1^{-1}\epsilon_2^{-1}))).$$

Hence, with a total of $\tilde{n}_{\rm mc}$ samples, Algorithm 2 finds estimates $\hat{\beta}_{jk}$ for β_{jk} with probability at least $1-\eta$, such that, for $\beta_{jk} \geq \epsilon_2$, $\hat{\beta}_{jk}$ estimates β_{jk} within ratio $1+\epsilon_1$, and, for $\beta_{jk} < \epsilon_2$, $|\hat{\beta}_{jk} - \beta_{jk}| \leq (1+\epsilon_1)\epsilon_2$. We can simplify further by letting $\epsilon_0 = \epsilon_1\epsilon_2$. Then the time complexity is $O(\epsilon_0^{-2}\log\eta^{-1}N(N\log N + \log(\epsilon_0^{-1})))$.

E. Simulation Results

To demonstrate Theorem 3, we use a scenario in which there are five predicted observations and 16 actual observations over a 4×4 two-dimensional region [see Fig. 5(a)]. The predicted observations are $\{\left[\begin{smallmatrix}0\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\1\end{smallmatrix}\right], \left[\begin{smallmatrix}0\\-1\end{smallmatrix}\right], \left[\begin{smallmatrix}1\\0\end{smallmatrix}\right], \left[\begin{smallmatrix}-1\\0\end{smallmatrix}\right]\}$ and the actual observations are

$$\begin{bmatrix}0.33\\0.07\end{bmatrix}, \begin{bmatrix}-1.50\\0.58\end{bmatrix}, \begin{bmatrix}-0.02\\-0.77\end{bmatrix}, \begin{bmatrix}-0.01\\-0.66\end{bmatrix}$$
$$\begin{bmatrix}-0.44\\-1.19\end{bmatrix}, \begin{bmatrix}1.45\\-1.30\end{bmatrix}, \begin{bmatrix}-0.87\\-0.11\end{bmatrix}, \begin{bmatrix}1.49\\-1.05\end{bmatrix}$$
$$\begin{bmatrix}-0.69\\0.89\end{bmatrix}, \begin{bmatrix}-1.16\\0.81\end{bmatrix}, \begin{bmatrix}-0.43\\0.63\end{bmatrix}, \begin{bmatrix}-0.97\\-0.30\end{bmatrix}$$
$$\begin{bmatrix}0.13\\-0.26\end{bmatrix}, \begin{bmatrix}-1.43\\0.95\end{bmatrix}, \begin{bmatrix}0.45\\-0.40\end{bmatrix}, \begin{bmatrix}1.37\\0.74\end{bmatrix}.$$

$\hat{P}^k(Y_t^j|y_{1:t-1})$ has a Gaussian distribution with zero mean and covariance $B^k = \mathrm{diag}(1,1)$ for all k. The other parameters used in this simulation are: $\lambda_{\rm f} = 0.5$, $p_{\rm d} = 0.8$, and $\delta^k = P((y_t^j - \hat{y}^k)^T(B^k)^{-1}(y_t^j - \hat{y}^k) = 4)$ for all k.

The true values of $\{\beta_{jk}\}$ are computed using JPDA. In order to study the convergence of the single-scan MCMCDA algorithm, we ran 100 independent runs with initial states randomly chosen from Ω_t. For each run, two types of estimates are made at each MCMC step: (type $r=1$) $\hat{\beta}_{jk}^1$, which are computed after $\bar{\tau}(\epsilon)$ burn-in samples; and (type $r=2$) $\hat{\beta}_{jk}^2$, which are computed after 10,000 burn-in samples. Let $\hat{\beta}_{jk}^r(m,n)$ be the estimate made at the $n^{\rm th}$ MCMC step for the $m^{\rm th}$ run, for type $r \in \{1,2\}$. Using $\epsilon_1 = 0.1$, $\epsilon_2 = 0.05$, $\eta = 0.05$, and $\epsilon = \epsilon_1\epsilon_2/8$, we have $\bar{\tau}(\epsilon) = 8.4\times 10^6$.

Based on Theorem 3, $\tilde{n}_{\rm mc} = 11.5\times 10^6$ samples suffice to ensure that the estimate using $\bar{\tau}(\epsilon)$ burn-in samples approximates the true value with ratio less than ϵ_1 with probability at least $1-\eta$. In order to show the progressive improvement of estimation ratios, we show the worst estimation ratios over all test cases as a function of the number of samples in Fig. 5(b). Hence, if the worst estimation error ratios are within ratio less than ϵ_1 after $\tilde{n}_{\rm mc}$ samples, i.e., the desired estimation ratio is satisfied for all test cases, it is certain that the statement of Theorem 3 is satisfied. Since there are upper and lower estimation ratio bounds in Theorem 3, we compute $\bar{R}^r(n)$, the largest estimation ratio, and $\underline{R}^r(n)$, the smallest estimation ratio over all test cases as a function of the number of samples and they are defined below.

Fig. 5(b) shows a pair of envelopes, one for each type of estimate. The top curve of an envelope plots the largest estimation ratio over all (j,k) pairs and all 100 runs

$$\bar{R}^r(n) = \max_{m=1,\ldots,100}\max_{jk}\frac{\hat{\beta}_{jk}^r(m,n)}{\beta_{jk}}$$

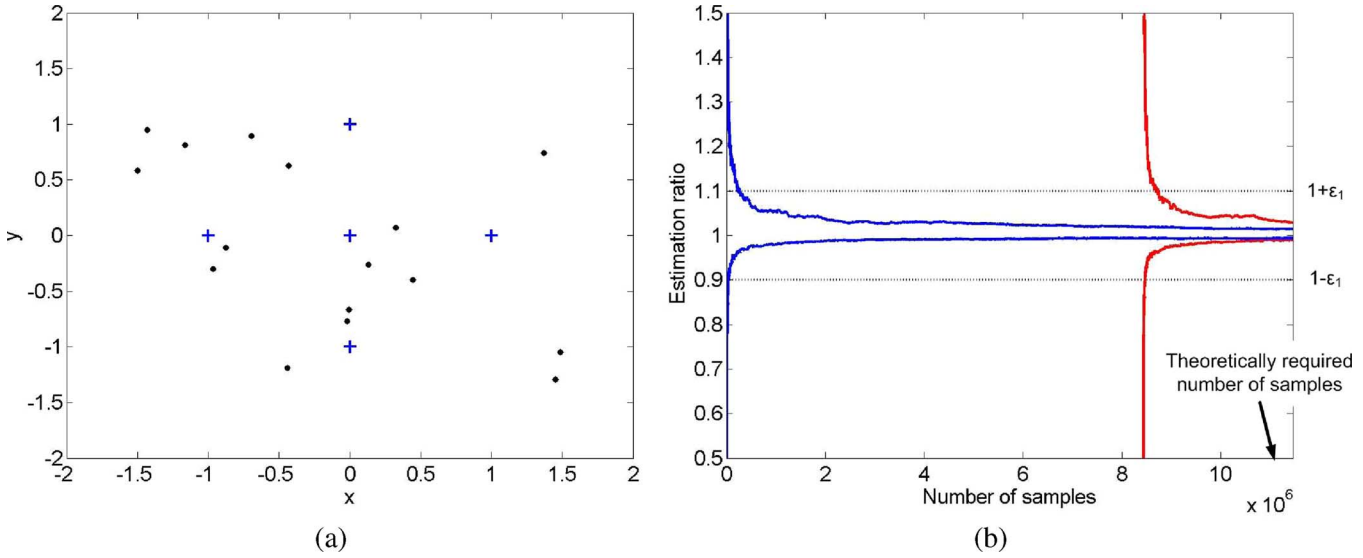


Fig. 5. (a) Predicted observations (crosses) and actual observations (dots). (b) The largest and smallest estimation ratios $(\underline{R}^r, \bar{R}^r)$ for two types of estimates ($r \in \{1,2\}$) each computed from 100 independent single-scan MCMCDA runs. The estimation ratios for $r = 1$ start from $\bar{\tau}(\epsilon) = 8.4 \times 10^6$ and the estimation ratios for $r = 2$ start from 10,000. The dotted ϵ_1-tube centered at 1 represents the goal estimation ratio ϵ_1. If $(\underline{R}^r, \bar{R}^r)$ is completely contained in $(1-\epsilon_1, 1+\epsilon_1)$, we have achieved our goal estimation ratio. Theoretically, $\bar{n}_{\mathrm{mc}} = 11.5 \times 10^6$ samples suffice to ensure that the estimate using $\bar{\tau}(\epsilon)$ burn-in samples approximates the true value with ratio less than ϵ_1 with probability at least $1-\eta$. But both estimates achieve the estimation ratio of ϵ_1 much faster than $\bar{n}_{\mathrm{mc}}$.

and the bottom curve plots the smallest estimation ratio over all (j,k) pairs and all 100 runs

$$\underline{R}^r(n) = \min_{m=1,\ldots,100} \min_{jk} \frac{\hat{\beta}^r_{jk}(m,n)}{\beta_{jk}}$$

for $r \in \{1,2\}$, where n is the number of MCMC samples. The envelope for type $r = 1$ starts from $n = \bar{\tau}(\epsilon)$ and type $r = 2$ starts from $n = 10{,}000$.

A couple of observations can be made from Fig. 5(b). The envelope $(\underline{R}^1(n), \bar{R}^1(n))$ is completely contained in $(1-\epsilon_1, 1+\epsilon_1)$ for all $n > \tilde{n}_{\mathrm{mc}}$, hence, MCMCDA approximates the true value with ratio less than ϵ_1 with $\tilde{n}_{\mathrm{mc}} = 11.5 \times 10^6$ samples. Since the estimation ratios are satisfied for all test cases, they are certainly satisfied with probability at least $1-\eta$ and Theorem 3 is verified. Both types of estimates ($r = 1$ and $r = 2$) achieve the estimation ratio of ϵ_1 much faster than $\tilde{n}_{\mathrm{mc}}$. As frequently observed in many practical applications of MCMC, we see that the algorithm requires a significantly smaller number of burn-in samples than the theorem requires. This is not especially surprising since the theorem is based on a worst-case analysis. For this example, 10,000 burn-in samples were enough, i.e., 800 times less than $\bar{\tau}(\epsilon)$.

In this example, JPDA took 134.1 seconds. With 10,000 burn-in samples, 95% of MCMCDA runs are within the goal approximation ratio of ϵ_1 in 33.9 seconds. Both algorithms are implemented in MATLAB on a PC with a 2.6-GHz Intel processor.

V. Multi-Scan MCMCDA

The single-scan MCMCDA algorithm described in Section IV assumes a fixed, known number of targets. This assumption leads to a simple filtering scheme, but in most situations of interest the number of targets is unknown and

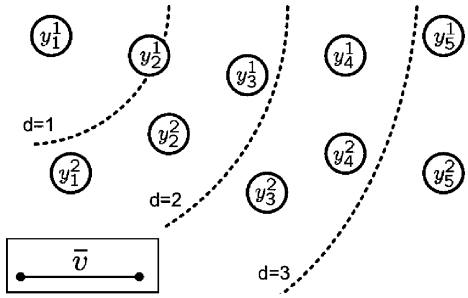


Fig. 6. An example to illustrate (A1) and (A2). Circles represent measurements (positions of targets in 2D). y_t^j is the j^{th} measurement made at time t. The maximum directional speed $\bar{v}$ is shown in the bottom left corner. The sampling period is 1. Suppose that the measurement y_1^1 is generated from a target. A measurement of this target at time $t = 2$ must be in the range denoted by $d = 1$ centered at y_1^1. A measurement of this target at time $t = 3$ must be in the range denoted by $d = 2$ centered at y_1^1. A measurement of this target at time $t = 4$ must be in the range denoted by $d = 3$ centered at y_1^1. If $\bar{d} = 3$, there cannot be a track whose measurements are only y_1^1 and y_5^1 (or y_1^1 and y_5^2) since such track contains 4 consecutive missing measurements.

changes over time. Furthermore, a single-scan algorithm that makes approximations (such as measurement validation and independence) to avoid complexity may end up being unable to maintain tracks over long periods because it cannot revisit previous, possibly incorrect, association decisions in the light of new evidence. For these reasons, methods for solving the general multi-target tracking problem described in Section II often adopt a multi-scan design, maintaining state in the form of both the posterior approximation and the observation history. This section describes a multi-scan MCMCDA algorithm that can handle unknown numbers of targets. The solution space Ω for this algorithm contains association *histories* over multiple time steps, as well as considering all possible numbers of targets at each step, and is therefore much larger than the solution space considered by a single-scan algorithm. The multi-scan

MCMCDA algorithm features efficient mechanisms to search over this large solution space in addition to birth and death moves to add or remove tracks. The multi-scan MCMCDA algorithm is presented in Section V-A and its recursive online version is described in Section V-B. In Section V-C, we compare the performance of MCMCDA against MHT and multi-scan NNF.

A. Multi-Scan MCMCDA Algorithm

The multi-scan MCMCDA algorithm is described in Algorithm 3. It is an MCMC algorithm whose state space is Ω as defined in Section II-B and whose stationary distribution is the posterior (9). The proposal distribution for MCMCDA consists of eight moves grouped into five types as follows: (1) birth/death move pair; (2) split/merge move pair; (3) extension/reduction move pair; (4) track update move; and (5) track switch move. (See Fig. 7.) We index each move by an integer such that $m = 1$ for a birth move, $m = 2$ for a death move and so on. The move m is chosen randomly from the distribution $\xi_K(m)$ where K is the number of tracks of the current partition ω. When there is no track, we can only propose a birth move, so we set $\xi_0(m = 1) = 1$ and 0 for all other moves. When there is only a single target, we cannot propose a merge or track switch move, so $\xi_1(m = 4) = \xi_1(m = 8) = 0$. For other values of K and m, we assume $\xi_K(m) > 0$. The inputs for MCMCDA are the set of all observations Y, the number of samples n_{mc}, the initial state ω_{init}, and the model parameters p_{z}, p_{d}, and λ_{b}. When we want to estimate $\mathbb{E}_\pi f$ of a bounded function $f : \Omega \to \mathbb{R}^n$, MCMCDA can also take the function f as an input. At each step of the algorithm, ω is the current state of the Markov chain. The acceptance probability $A(\omega, \omega')$ is defined in (12) where $\pi(\omega) = P(\omega|Y)$ from (9). Notice that MCMCDA can provide both Bayes estimator and MAP solutions to the multi-target tracking problem: the output $\hat{f}$ approximates the Bayesian posterior expectation $\mathbb{E}_\pi f$ and $\hat{\omega}$ approximates the MAP estimate $\arg\max_{\omega\in\Omega} P(\omega|Y)$. The computation of $\hat{\omega}$ can be viewed as simulated annealing [44] at a constant temperature.

Algorithm 3 Multi-scan-MCMCDA

1: INPUT $Y, n_{\mathrm{mc}}, \omega_{\mathrm{init}}, p_{\mathrm{z}}, p_{\mathrm{d}}, \lambda_{\mathrm{b}}, f : \Omega \to \mathbb{R}^n$

2: OUTPUT $\hat{\omega}, \hat{f}$

3: $\omega = \omega_{\mathrm{init}}$; $\hat{\omega} = \omega_{\mathrm{init}}$; $\hat{f} = 0$

4: **for** $n = 1$ to n_{mc} **do**

5: propose ω' based on ω (see **Sections V-A.I** to **V-A.V**)

6: sample u from $\mathrm{Unif}[0, 1]$

7: $\omega = \omega'$ if $u < A(\omega, \omega')$

8: $\hat{\omega} = \omega$ if $P(\omega|Y)/P(\hat{\omega}|Y) > 1$

9: $\hat{f} = ((n-1)/n)\hat{f} + (1/n)f(\omega)$

10: **end for**

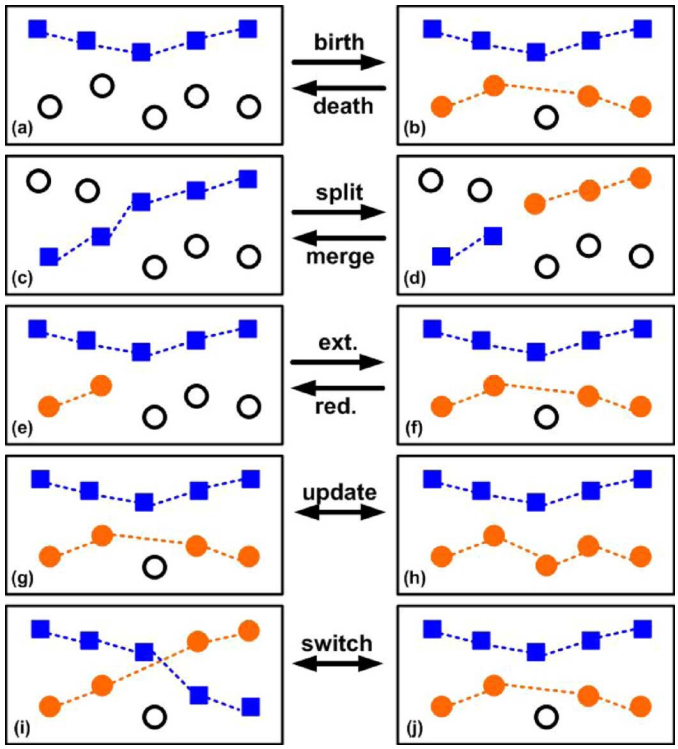


Fig. 7. Graphical illustration of MCMCDA moves (associations are indicated by dotted lines and hollow circles are false alarms). Each move proposes a new joint association event ω' that is a modification of the current joint association event ω. The birth move proposes ω' by forming a new track from the set of false alarms ((a) → (b)). The death move proposes ω' by combining one of the existing tracks into the set of false alarms ((b) → (a)). The split move splits a track from ω into two tracks ((c) → (d)) while the merge move combines two tracks in ω into a single track ((d) → (c)). The extension move extends an existing track in ω ((e) → (f)) and the reduction move reduces an existing track in ω ((f) → (e)). The track update move chooses a track in ω and assigns different measurements from the set of false alarms ((g) ↔ (h)). The track switch move chooses two track from ω and switches some measurement-to-track associations ((i) ↔ (j)).

An MCMC algorithm can be specialized and made more efficient by incorporating domain-specific knowledge into the proposal distribution $q(\omega, \omega')$. For example, the MCMC algorithm by Pasula *et al.* [45] for citation matching incorporates a clustering method by McCallum *et al.* [46] to improve the performance of the algorithm. This method precomputes overlapping sets of "possibly matching" records and restricts the MCMC proposal distribution to consider associating record pairs only from these sets. This method prevents proposals that are certain to be rejected as impossible.

In multi-target tracking, we can make two assumptions: (A1) the maximum directional speed of any target in $\mathcal{R}$ is less than some $\bar{v}$; and (A2) the number of consecutive missing observations of any track is less than some $\bar{d}$. The first assumption (A1) is reasonable in a target-tracking scenario since, in many cases, the maximum speed of a vehicle is generally known based on the vehicle type and terrain conditions. We assume that the value of $\bar{v}$ is chosen large enough such that it accommodates measurement noise (e.g., adding a multiple of the standard deviation of the measurement noise). The second assumption (A2) is a user-defined parameter. Let $p_{\mathrm{dt}}(s) = 1-(1-p_{\mathrm{d}})^s$ be the probability that an object is observed at least once out of s measurement

times. Then, for given $\bar{p}_{\rm dt}$, we set $\bar{d} \geq \log(1-\bar{p}_{\rm dt})/\log(1-p_{\rm d})$ to detect a track with probability at least $\bar{p}_{\rm dt}$. For example, given $p_{\rm d} = 0.7$ and $\bar{p}_{\rm dt} = 0.99$, a track is detected with probability larger than 0.99 for $\bar{d} \geq 4$. We will now assume that these two new conditions (A1–2) are added to the definition of Ω so each element $\omega \in \Omega$ satisfies these two additional assumptions. In the following description of the multi-scan MCMCDA algorithm, we also assume that a track contains at least two measurements for computational efficiency.

According to (A1–2), we can determine whether two measurements at different times can be generated from the same target or not. For example, see Fig. 6. Let's consider y_1^1 in Fig. 6, a measurement made at $t = 1$. If y_1^1 is a measurement from a target, then a measurement of this target at time $t = 2$ must be in the range denoted by $d = 1$ centered at y_1^1. Hence, y_2^1 can be a measurement from this target while y_2^2 cannot be a measurement from this target. We formalize this concept into the following data structure. Let $L(y,t,d)$ be a set of all measurements at time $t+d$ that can be associated with a measurement $y \in \mathbb{R}^{n_y}$, i.e.,

$$L(y,t,d) = \{y_{t+{\rm d}}^k \in y_{t+{\rm d}} : \varphi(y, y_{t+{\rm d}}^k) \leq d \cdot \bar{v}\} \tag{27}$$

where $d \in \{1,\ldots,\bar{d}\}$, and $\varphi : \mathbb{R}^{n_y} \times \mathbb{R}^{n_y} \to \mathbb{R}$ is an appropriate metric, e.g., for a Cartesian coordinate system, φ is induced by the Euclidean norm.

For our example shown in Fig. 6, $L(y_1^1, t=1, d=1) = \{y_2^1\}$, $L(y_1^1, t=1, d=2) = \{y_3^1\}$, and $L(y_1^1, t=1, d=3) = \{y_4^1, y_4^2\}$. The use of $L(y,t,d)$ makes the algorithm more scalable since distant observations will be considered separately and makes the computation of the proposal distribution easier. It is similar to the gating technique used in MHT but $L(y,t,d)$ in MCMCDA is fixed for a given set of observations. We fix $L(y,t,d)$ so that the proposal distribution $q(\omega,\omega')$ can be computed consistently.

We now describe each move of the sampler in detail. First, let $\zeta(d)$ be a distribution of a random variable d taking values from $\{1,2,\ldots,\bar{d}\}$. We assume the current state of the chain is $\omega = \omega^0 \cup \omega^1 \in \Omega$, where $\omega^0 = \{\tau_0\}$ and $\omega^1 = \{\tau_1,\ldots,\tau_K\}$. The proposed partition is denoted by $\omega' = \omega'^0 \cup \omega'^1 \in \Omega$. The proposal distribution $q(\omega,\omega')$ can be computed by keeping track of how ω is modified to form ω'. Note the abuse of notation below with indexing of time, i.e., when we say $\tau(t_i)$, t_i means the time at which a target corresponding to the track τ is observed i times.

1) Birth and Death Moves (Fig. 7, $a \leftrightarrow b$): For a birth move, we increase the number of tracks from K to $K' = K+1$ and select t_1 uniformly at random (u.a.r.) from $\{1,\ldots,T-1\}$ as the appearance time of a new track. Let $\tau_{K'}$ be the track of this new object. Then we choose d_1 from the distribution ζ. Let

$$L^{\rm B}(t_1,d_1) = \{y_{t_1}^j \in y_{t_1} : L(y_{t_1}^j, t_1, d_1) \neq \emptyset, y_{t_1}^j \notin \tau_k(t_1), k = 1,\ldots,K\}.$$

$L^{\rm B}(t_1,d_1)$ is a set of observations at t_1 such that $y \in L^{\rm B}(t_1,d_1)$ does not belong to other tracks and $L(y,t_1,d_1)$ is not empty. We choose $\tau_{K'}(t_1)$ u.a.r. from $L^{\rm B}(t_1,d_1)$. If $L^{\rm B}(t_1,d_1)$ is empty, the move is rejected since the move is not reversible. Once the initial observation is chosen, we then choose the subsequent observations for the track $\tau_{K'}$. For $i = 2,3,\ldots$, we choose d_i from ζ and choose $\tau_{K'}(t_i)$ u.a.r. from $L(\tau_{K'}(t_{i-1}), t_{i-1}, d_i) \setminus \{\tau_k(t_{i-1}+d_i) : k = 1,\ldots,K\}$ unless this set is empty. But, for $i = 3,4,\ldots$, the process of adding observations to $\tau_{K'}$ terminates with probability γ, where $0 < \gamma < 1$. If $|\tau_{K'}| \leq 1$, the move is rejected. We then propose this modified partition where $\omega'^1 = \omega^1 \cup \{\tau_{K'}\}$ and $\omega'^0 = \{\tau_0 \setminus \tau_{K'}\}$. For a death move, we simply choose k u.a.r. from $\{1,\ldots,K\}$ and delete the $k^{\rm th}$ track and propose a new partition where $\omega'^1 = \omega^1 \setminus \{\tau_k\}$ and $\omega'^0 = \{\tau_0 \cup \tau_k\}$.

2) Split and Merge Moves (Fig. 7, $c \leftrightarrow d$): For a split move, we select $\tau_s(t_r)$ u.a.r. from $\{\tau_k(t_i) : |\tau_k| \geq 4, i = 2,\ldots,|\tau_k|-2, k = 1,\ldots,K\}$. Then we split the track τ_s into τ_{s_1} and τ_{s_2} such that $\tau_{s_1} = \{\tau_s(t_i) : i = 1,\ldots,r\}$ and $\tau_{s_2} = \{\tau_s(t_i) : i = r+1,\ldots,|\tau_s|\}$. The modified track partition becomes $\omega'^1 = (\omega^1 \setminus \{\tau_s\}) \cup \{\tau_{s_1}\} \cup \{\tau_{s_2}\}$ and $\omega'^0 = \omega^0$. For a merge move, we consider the following set of possible merge move pairs:

$$M = \{(\tau_{k_1}(t_{\rm f}), \tau_{k_2}(t_1)) : \tau_{k_2}(t_1) \in L(\tau_{k_1}(t_{\rm f}), t_{\rm f}, t_1 - t_{\rm f}), f = |\tau_{k_1}| \text{ for } k_1 \neq k_2, 1 \leq k_1, k_2 \leq K\}.$$

We select a pair $(\tau_{s_1}(t_{\rm f}), \tau_{s_2}(t_1))$ u.a.r. from M. The tracks are combined into a single track $\tau_s = \tau_{s_1} \cup \tau_{s_2}$. Then we propose a new partition where $\omega'^1 = (\omega^1 \setminus (\{\tau_{s_1}\} \cup \{\tau_{s_2}\})) \cup \{\tau_s\}$ and $\omega'^0 = \omega^0$.

3) Extension and Reduction Moves (Fig. 7, $e \leftrightarrow f$): In a track extension move, we select a track τ u.a.r. from K available tracks in ω. We reassign observations for τ after the disappearance time $t_{|\tau|}$ as done in the track birth move. For a track reduction move, we select a track τ u.a.r. from K available tracks in ω and r u.a.r. from $\{2,\ldots,|\tau|-1\}$. We shorten the track τ to $\{\tau(t_1),\ldots,\tau(t_r)\}$ by removing the observations assigned to τ after the time t_{r+1}.

4) Track Update Move (Fig. 7, $g \leftrightarrow h$): In a track update move, we select a track τ u.a.r. from K available tracks in ω. Then we pick r u.a.r. from $\{1,2,\ldots,|\tau|\}$ and reassign observations for τ after the time t_r as done in the track birth move.

5) Track Switch Move (Fig. 7, $i \leftrightarrow j$): For a track switch move, we select a pair of observations $(\tau_{k_1}(t_p), \tau_{k_2}(t_q))$ from two different tracks such that, $\tau_{k_1}(t_{p+1}) \in L(\tau_{k_2}(t_q), t_q, d)$ and $\tau_{k_2}(t_{q+1}) \in L(\tau_{k_1}(t_p), t_p, d')$, where $d = t_{p+1} - t_q$, $d' = t_{q+1} - t_p$ and $0 < d, d' \leq \bar{d}$. Then we let

$$\tau_{k_1} = \{\tau_{k_1}(t_1),\ldots,\tau_{k_1}(t_p), \tau_{k_2}(t_{q+1}),\ldots,\tau_{k_2}(t_{|\tau_{k_2}|})\}$$
$$\tau_{k_2} = \{\tau_{k_2}(t_1),\ldots,\tau_{k_2}(t_q), \tau_{k_1}(t_{p+1}),\ldots,\tau_{k_1}(t_{|\tau_{k_1}|})\}.$$

The main result of this section is that MCMCDA is an optimal Bayesian filter in the limit. Let $\mathcal{M}$ be the Markov chain specified by Algorithm 3. Then we have:

Theorem 4: Suppose that $0 < p_{\rm z}, p_{\rm d} < 1$ and $\lambda_{\rm b}, \lambda_{\rm f} > 0$. If $\zeta(d) > 0$ for all $d \in \{1,\ldots,\bar{d}\}$, then the Markov chain $\mathcal{M}$ is ergodic and $\hat{f} \to \mathbb{E}_\pi f$ as $n_{\rm mc} \to \infty$.

For the proof and a numerical demonstration of the theorem, see [43].

B. Online MCMCDA

The MCMCDA algorithm described in previous section is a batch algorithm and its computational complexity grows as

more measurements are collected. In filtering, since recent measurements are more relevant to the current states, good estimates of the current states can still be found from recent measurements [47]. Based on this idea, we propose an online MCMCDA algorithm whose estimates are based on measurements from a window of time $[t_{\rm cur} - t_{\rm win} + 1, \ldots, t_{\rm cur}]$, where $t_{\rm cur}$ is the current time and $t_{\rm win}$ is the size of a window. Hence, at all times, only finitely many measurements are kept by the algorithm. This online implementation of MCMCDA, shown in Algorithm 4, is suboptimal because it considers only a subset of past measurements.

Algorithm 4 Online-MCMCDA (at time $t_{\rm cur}$)

1: INPUT: $\hat{\omega}(t_{\rm cur}-1), Y_{\rm w}(t_{\rm cur}-1), Y_{\rm new}(t_{\rm cur}), n_{\rm mc}, p_{\rm z}, p_{\rm d}, \lambda_{\rm b}, f : \Omega \to \mathbb{R}^n$

2: OUTPUT: $\hat{\omega}(t_{\rm cur})$, $Y_{\rm w}(t_{\rm cur})$, $\hat{f}(t_{\rm cur})$

3: $Y_{\rm w}(t_{\rm cur}) = \{y_t^j \in Y_{\rm w}(t_{\rm cur}-1) : t_{\rm cur} - t_{\rm win} + 1 \le t \le t_{\rm cur}\}$

4: add new measurements $Y_{\rm new}(t_{\rm cur})$ into $Y_{\rm w}(t_{\rm cur})$

5: $\omega_{\rm init} = \{\tau(t) \in \hat{\omega}(t_{\rm cur}-1) : t_{\rm cur} - t_{\rm win} + 1 \le t \le t_{\rm cur}\}$

6: $[\hat{\omega}(t_{\rm cur}), \hat{f}(t_{\rm cur})]$ = Multi-scan-MCMCDA ($Y_{\rm w}(t_{\rm cur})$, $n_{\rm mc}$, $\omega_{\rm init}$, $p_{\rm z}$, $p_{\rm d}$, $\lambda_{\rm b}$, f) (see Algorithm 3)

At each time step, we use the previous MAP estimate to initialize MCMCDA and run MCMCDA on the measurements $Y_{\rm w}(t_{\rm cur}) = \{y_t^j : 1 \le j \le n_t, t_{\rm cur} - t_{\rm win} + 1 \le t \le t_{\rm cur}\}$ belonging to the current window. At time $t_{\rm cur}$, the measurements at time $t_{\rm cur} - t_{\rm win}$ are removed from $Y_{\rm w}$ and a set of newly arrived measurements $Y_{\rm new}(t_{\rm cur})$ is appended to $Y_{\rm w}(t_{\rm cur})$. Any delayed measurements are inserted into the appropriate slots. Then, we initialize the Markov chain with the previously estimated tracks and execute Algorithm 3 on $Y_{\rm w}(t_{\rm cur})$. The algorithm is summarized in Algorithm 4. The inputs for online MCMCDA at time $t_{\rm cur}$ are the previous MAP estimate $\hat{\omega}(t_{\rm cur} - 1)$, the existing set of measurements $Y_{\rm w}(t_{\rm cur} - 1)$, and the set of new measurements $Y_{\rm new}(t_{\rm cur})$. The other inputs are the same as Algorithm 3. Other estimates such as state estimates can be computed using the function f or $\hat{\omega}(t_{\rm cur})$. Simulation results from online MCMCDA can be found in Section V-C.IV.

C. Simulation Results

In this section, the performance of multi-scan MCMCDA is evaluated and compared against MHT [48] and multi-scan NNF [49]. We consider surveillance over a rectangular region on a plane, $\mathcal{R} = [0, 1000] \times [0, 1000]$. The state vector is $x = [x, y, \dot{x}, \dot{y}]^T$ where (x, y) is a position on $\mathcal{R}$ along the usual x and y axes and $(\dot{x}, \dot{y})$ is a velocity vector. Linear dynamics and a linear measurement model are used:

$$x_{t+1}^k = Ax_t^k + Gw_t^k \qquad y_t^j = Cx_t^k + v_t^j \tag{28}$$

where

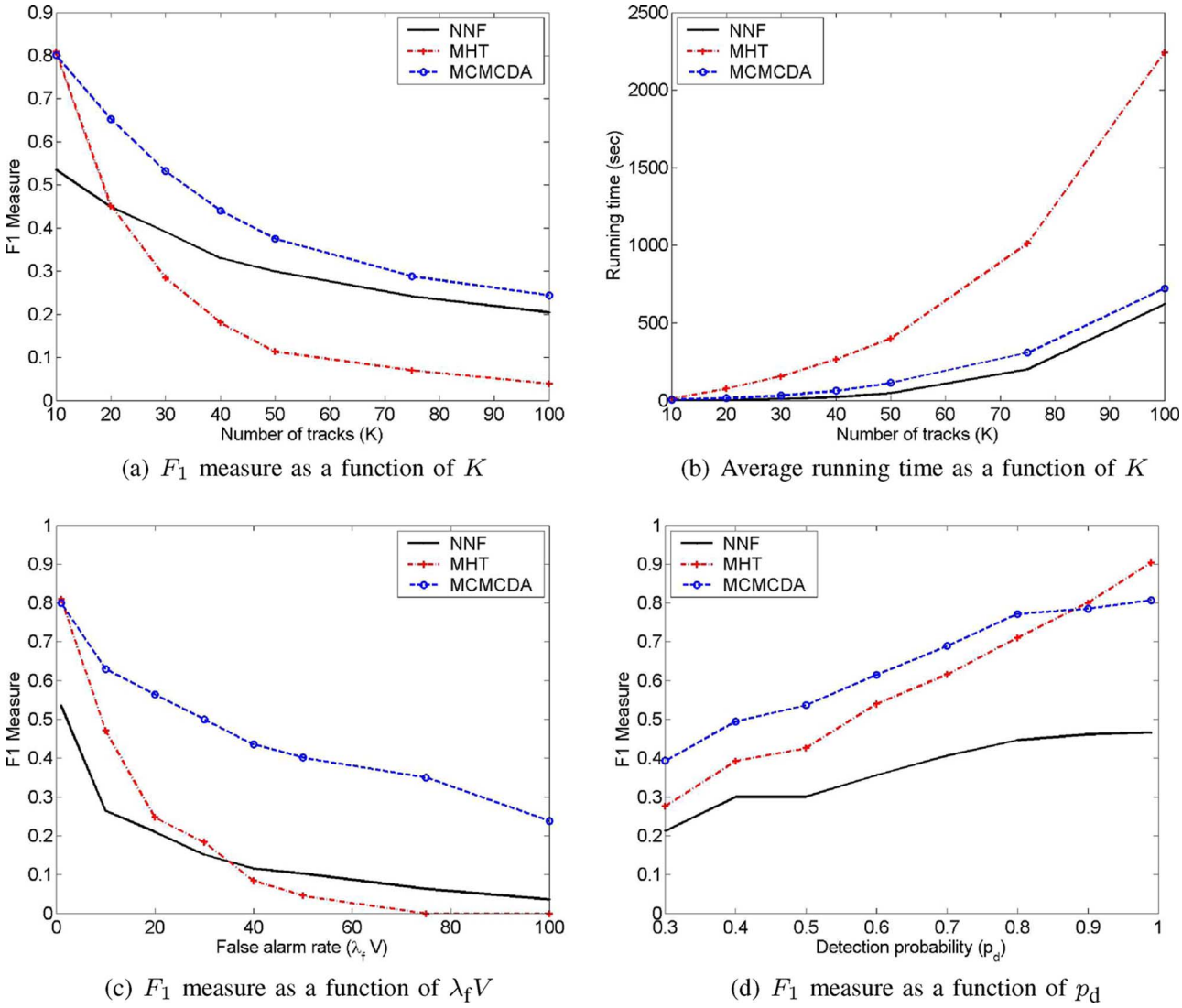
$$A = \begin{bmatrix} 1 & 0 & T_s & 0 \\ 0 & 1 & 0 & T_s \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \qquad G = \begin{bmatrix} \frac{T_s^2}{2} & 0 \\ 0 & \frac{T_s^2}{2} \\ T_s & 0 \\ 0 & T_s \end{bmatrix}$$

$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix}$, T_s is the sampling period, w_t^k is a zero-mean Gaussian process with covariance $Q = \mathrm{diag}(100, 100)$, and v_t^j is a zero-mean Gaussian process with covariance $R = \mathrm{diag}(100, 100)$.

The complexity of multi-target tracking problems can be measured by several metrics: (1) the intensity of the false alarm rate $\lambda_{\rm f}$; (2) the detection probability $p_{\rm d}$; and (3) the density of tracks. The problem gets more challenging with increasing $\lambda_{\rm f}$, decreasing $p_{\rm d}$, and increasing density of tracks. The number of tracks *per se* may not make the problem more difficult if they are scattered apart; the difficulty arises when there are many tracks crossing and moving close to each other; this is when the ambiguity of data association is greatest. Hence, we consider only situations in which tracks move very closely so we can control the density of tracks by the number of tracks.

We study the performance of the MCMCDA algorithm against multi-scan NNF and MHT by varying the parameters listed above. To make the comparison easier, we take the MAP approach, in which the states of targets are estimated from $\hat{\omega}$ computed from Algorithm 3. The multi-scan NNF algorithm is a batch-mode, nearest-neighbor, multi-target tracking algorithm. Initially, all observations are unmarked. Unmarked observations are considered false alarms. The algorithm first picks two unmarked observations at different times to estimate an initial state. Then it forms a candidate track by picking the unmarked observations for the subsequent time step that are closest to the predicted states. The candidate track is validated as a track and observations associated to the candidate track are marked if the marginal of the candidate track exceeds a threshold. The process is repeated until no more tracks can be found. For a more detailed description of the multi-scan NNF algorithm, see [49].

Based on our model described above and in Section II, we have generated a variety of scenarios. In particular, in all cases, except for the online tracking case, half of the new objects appear from the bottom left quadrant of $\mathcal{R}$ and the other half appear from the the bottom right quadrant. (The actual initial positions are chosen randomly from a 200×200 region in each quadrant.) They all move diagonally so that each group of tracks crosses the other group in the middle of $\mathcal{R}$. The targets also move very close to each other and there are crossovers within each group. All targets are present from $t = 1$ to $t = T$. For the test case ($K = 100$) used in Section V-C.I, at $t = 5$, the mean number of neighboring targets with distance less than 50 is 13 and some targets have 25 neighboring targets. For distance less than 100, the mean number of neighboring targets is 42 and some targets have 73 neighboring targets. Based on the dynamics and measurement noise models (28), the effective overall standard deviation in two dimension is 28.3. Hence, the distance of 100 is 3.5 times the effective standard deviation.



(a) F_1 measure as a function of K

(b) Average running time as a function of K

(c) F_1 measure as a function of $\lambda_{\mathrm{f}}V$

(d) F_1 measure as a function of p_{d}

Fig. 8. Simulation results. Multi-scan MCMCDA is compared against the multi-scan NNF [49] and MHT [48].

Since the number of targets is not fixed, it is difficult to compare algorithms using a standard criterion such as the mean squared error. Instead, we use a performance measure, called F_1, which is frequently used in the information retrieval literature [50] and has been used in particular for evaluating data-association-like methods in record matching [51]. The F_1 measure is defined in terms of recall and precision. Recall is the ratio of correct associations made by an algorithm divided by the total number of correct associations. Precision is the ratio of correct associations made by an algorithm divided by the total number of associations made by the algorithm. The F_1 measure is a harmonic mean between recall (r) and precision (p) with an equal weight and defined as $F_1(r,p) = 2rp/(r+p)$. Recall and precision measure the *effectiveness* of an algorithm [50]; the higher the value of the F_1 measure, the more effective the algorithm is.

Both MCMCDA and multi-scan NNF algorithms are written in C++ with MATLAB interfaces. We have used the C++ implementation of MHT [48], which implements pruning, gating, clustering, N-scan-back logic and k-best hypotheses. The parameters for MHT are fine-tuned so that it gives similar performance to that of MCMCDA when there are 10 targets: the maximum number of hypotheses in a group is 1,000, the maximum track tree depth is 5, and the maximum Mahalanobis distance is 11.8. All simulations are run on a PC with a 2.6-GHz Intel processor.

1) Number of Tracks: In this experiment, we vary K from 10 to 100. The other parameters are held fixed: $\mathcal{R} = [0,1000] \times [0,1000]$, $T = 10$, $p_{\mathrm{d}} = 0.9$, $\lambda_{\mathrm{f}}V = 1$, $\lambda_{\mathrm{b}}V = K/T$, $p_{\mathrm{z}} = .0001$, $\bar{d} = 5$, $\bar{v} = 100$ unit lengths per unit time. A uniform mass function is used for each $\xi_k(\cdot)$ and $\zeta(d)$ is computed based on p_{d}. For each value of K, we randomly generated 10 test cases. The initial state of MCMCDA is computed using the multi-scan NNF algorithm and 50,000 samples are used in MCMCDA. For each K, the average F_1 measure and running time are computed from the 10 test cases (for MCMCDA, we also average over 10 runs per test case). The average F_1 measure computed at each value of K is shown in Fig. 8(a). The average running times of the three algorithms are shown in Fig. 8(b) (the running time of MCMCDA includes the initialization step). Although the maximum number of hypotheses of 1,000 per group

is a large number, with increasing numbers of tracks, the performance of MHT deteriorates due to pruning. The F_1 measure for multi-scan NNF is just above 0.5 even when there are only ten targets. MCMCDA shows superior performance against both MHT and multi-scan NNF. In this example, the multi-scan NNF performs better than MHT at higher numbers of targets. This is due to the fact that all targets are present from $t = 1$ to $t = T$. Another striking difference is that the running times of both multi-scan NNF and MCMCDA are significantly less than that of MHT.

2) False Alarms: Now the settings are the same as Section V-C.I but we vary the false alarm rate while the number of tracks is fixed at $K = 10$. The false alarm rate is varied from $\lambda_{\mathrm{f}}V = 1$ to $\lambda_{\mathrm{f}}V = 100$ with an increment of 10. For each value of $\lambda_{\mathrm{f}}V$, we randomly generated 10 test cases and MCMCDA is run 10 times per test case. Again, 50,000 samples are used for MCMCDA. The average F_1 measures for the three algorithms at different false alarm rates are given in Fig. 8(c). The results show that MCMCDA performs well at high false alarm rates. The multi-scan NNF algorithm suffers because it finds too many spurious tracks (poor precision) and MHT becomes hopelessly confused, finding no correct associations at $\lambda_{\mathrm{f}}V \geq 80$.

3) Detection Probability: The detection probability p_{d} is varied from 0.3 to 0.99 with an increment of 0.1, except the last increment which is 0.09, while keeping the other parameters as in the previous experiments except $K = 10$ and $\lambda_{\mathrm{f}}V = 1$. For each value of p_{d}, we randomly generated 10 test cases and MCMCDA is run 10 times per test case. Again, 50,000 samples are used for MCMCDA. The average F_1 measures for three different algorithms at different detection probabilities are shown in Fig. 8(d). The overall performance of MCMCDA is better than that of MHT. MHT only performs slightly better than MCMCDA at $p_{\mathrm{d}} = 0.99$. The running times of both MCMCDA and MHT are comparable in this case (MCMCDA was about 1 second faster than MHT for all cases except when $p_{\mathrm{d}} = 0.99$). The multi-scan NNF algorithm performed very poorly.

Although, in theory, MHT gives an optimal solution in the sense of MAP, it performs poorly in practice when the detection probability is low or the false alarm rate is high. This is due to the heuristics, such as pruning and N-scan-back techniques, that are required to limit complexity. They work well when a few hypotheses carry most of the weight. (This is why MHT performed slightly better than MCMCDA when $p_{\mathrm{d}} = 0.99$.) When the detection probability is low or the false alarm rate is high, however, there are many hypotheses with appreciable weights and there is no small set of dominating hypotheses, so MHT cannot perform well. A major advantage of the MCMCDA algorithm is that its running time can be regulated by the number of samples and the number of observations but the running time of MHT depends on the complexity of the problem instance, which is not predictable in advance. In addition, the memory required by MCMCDA is significantly less than the memory required by MHT, since MCMCDA is only required to store one association event at a time.

4) Online MCMCDA: An example of tracking multiple targets in a densely cluttered environment is used to demonstrate online MCMCDA from Section V-B. For this example, the surveillance duration is $T = 100$ and the scenario is generated according to the model for multi-target tracking described in Section II-A. The surveillance region is $\mathcal{R} = [0, 100] \times [0, 100]$ and the model parameters are: $\lambda_{\mathrm{b}}V = 5$, $p_{\mathrm{z}} = 1/20$, $p_{\mathrm{d}} = 0.7$, and $\lambda_{\mathrm{f}}V = 30$. There are a total of 380 targets. The linear model (28) is used, but with different covariance matrices: $Q = \mathrm{diag}(0.031, 0.031)$ and $R = \mathrm{diag}(0.031, 0.031)$. The size of the sliding window is $t_{\mathrm{win}} = 14$ for the online MCMCDA algorithm while $\bar{d} = 5$ and $\bar{v} = 3$ unit lengths per unit time.

For this example, MHT took 6,995 seconds while online MCMCDA took only 343 seconds, i.e., a 20-fold reduction in computation time. On the F_1 measure, MHT scored 0.85 and MCMCDA scored 0.91. In addition, MHT found 494 targets but MCMCDA detected 335 targets which is close to the actual number of 380 targets. The tracks estimated by MHT and MCMCDA are available in [43], along with the actual trajectories of targets. In summary, this example shows that MCMCDA is very effective in a dense environment and achieves superior performance with a fraction of the computation time required by MHT.

VI. Conclusion

In this paper, we have presented Markov chain Monte Carlo data association (MCMCDA) for solving data association problems arising in multi-target tracking in a cluttered environment. Instead of enumerating the entire space of associations, MCMCDA randomly samples the region where the posterior is concentrated.

For the case of a fixed number of targets, we have shown that a single-scan MCMCDA algorithm provides a fully polynomial randomized approximation scheme for the JPDA calculation, which is known to be NP-hard and is infeasible in practice for large problems. One can also consider a combined approach, where JPDA is used for small subgraphs created after some of the edges in the bipartite measurement validation graph have been broken by MCMC sampling. The precise division of labor will depend on the specific application and available computing resources.

For the general multi-target tracking problem, in which unknown numbers of targets appear and disappear at random times, we have presented a multi-scan MCMCDA algorithm that is capable of initiating and terminating tracks. Our simulation results show the remarkable performance of the MCMCDA algorithm under extreme conditions such as a large number of targets in a dense environment, low detection probabilities, and high false alarm rates. The MCMCDA algorithm is flexible and can easily incorporate domain specific knowledge to make it more efficient. The efficiency of MCMCDA has been demonstrated as part of of the real-time control system developed for solving multi-agent pursuit-evasion game using a large-scale outdoor wireless sensor network [5].

Acknowledgment

The authors wish to thank the anonymous reviewers for their helpful comments and suggestions.

REFERENCES

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

[2] I. Cox, "A review of statistical data association techniques for motion correspondence," *Int. J. Comp. Vision*, vol. 10, no. 1, pp. 53–66, 1993.

[3] 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.

[4] G. Cybenko, V. Berk, V. Crespi, R. Gray, and G. Jiang, "An overview of process query systems," in *Proc. SPIE, Sensors, Command, Control, Commun., Intell. (C3I) Technol. Homeland Secur. Homeland Def. III*, Orlando, FL, Apr. 2004, vol. 5403, pp. 183–197.

[5] S. Oh, L. Schenato, P. Chen, and S. Sastry, "Tracking and coordination of multiple agents using sensor networks: System design, algorithms and experiments," *Proc. IEEE*, vol. 95, no. 1, pp. 234–254, Jan. 2007.

[6] R. Sittler, "An optimal data association problem on surveillance theory," *IEEE Trans. Military Electron.*, vol. MIL-8, no. 2, pp. 125–139, Apr. 1964.

[7] D. Reid, "An algorithm for tracking multiple targets," *IEEE Trans. Automat. Control*, vol. AC-24, no. 6, pp. 843–854, Dec. 1979.

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

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

[10] 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 Anal. Machine Intell.*, vol. 18, no. 2, pp. 138–150, Feb. 1996.

[11] C. L. Morefield, "Application of 0–1 integer programming to multitarget tracking problems," *IEEE Trans. Automat. Control*, vol. AC-22, no. 3, pp. 302–312, Jun. 1971.

[12] J. Collins and J. Uhlmann, "Efficient gating in data association with multivariate distributed states," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 28, no. 3, pp. 909–916, Jul. 1992.

[13] 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. Boston, MA: PWS Publishing, 1996.

[14] L. Valiant, "The complexity of computing the permanent," *Theor. Comp. Sci.*, vol. 8, pp. 189–201, 1979.

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

[16] J. Roecker and G. Phillis, "Suboptimal joint probabilistic data association," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 29, no. 2, pp. 510–517, Apr. 1993.

[17] J. Roecker, "A class of near optimal JPDA algorithms," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 30, no. 2, pp. 504–510, Apr. 1994.

[18] T. Huang and S. J. Russell, "Object identification in a Bayesian context," in *Proc. 15th Int. Joint Conf. Artif. Intell.*, Nagoya, Japan, Aug. 1997.

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

[20] R. Mahler, "Multi-target Bayes filtering via first-order multi-target moments," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 39, no. 4, pp. 1152–1178, Oct. 2003.

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

[22] L. Pao, "Multisensor multitarget mixture reduction algorithms for tracking," *J. Guid., Control, Dyn.*, vol. 17, no. 6, pp. 1205–1211, Nov. 1994.

[23] H. Pasula, S. J. Russell, M. Ostland, and Y. Ritov, "Tracking many objects with many sensors," in *Proc. Int. Joint Conf. Artif. Intell.*, Stockholm, Sweden, 1999, pp. 1160–1171.

[24] H. Pasula, "Identity Uncertainty," Ph.D. dissertation, Comp. Sci. Div., Univ. California: Computer, Berkeley, CA, 2003.

[25] S. Cong, L. Hong, and D. Wicker, "Markov-chain Monte-Carlo approach for association probability evaluation," *Proc. Inst. Elect. Eng.*, vol. 151, no. 2, pp. 185–193, Mar. 2004.

[26] N. Bergman and A. Doucet, "Markov chain Monte Carlo data association for target tracking," in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, Istanbul, Turkey, Jun. 2000, pp. 11705–11708.

[27] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *Proc. IEEE Conf. Decision Control*, Paradise Island, Bahamas, Dec. 2004, pp. 735–742.

[28] W. Gilks, S. Richardson, and D. Spiegelhalter, *Markov Chain Monte Carlo in Practice*. London, U.K.: Chapman and Hall, 1996.

[29] I. Beichl and F. Sullivan, "The Metropolis algorithm," *Comp. Sci. Eng.*, vol. 2, no. 1, pp. 65–69, 2000.

[30] B. Eraker, "MCMC analysis of diffusion models with application to finance," *J. Bus. Econom. Statist.*, vol. 19, no. 2, pp. 177–191, 2001.

[31] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, no. 6, pp. 721–741, Nov. 1984.

[32] N. Metropolis, A. Rosenbluth, M. Rosenbluth, A. Teller, and E. Teller, "Equation of state calculation by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.

[33] W. Hastings, "Monte Carlo sampling methods using Markov chains and their applications," *Biometrika*, vol. 57, pp. 97–109, 1970.

[34] G. Roberts, "Markov chain concepts related to sampling algorithms," in *Markov Chain Monte Carlo in Practice*, W. Gilks, S. Richardson, and D. Spiegelhalter, Eds. London, U.K.: Chapman and Hall, 1996.

[35] P. Diaconis and D. Stroock, "Geometric bounds for eigenvalues of Markov chains," *Annal. Appl. Prob.*, vol. 1, pp. 36–61, 1991.

[36] H. Kushner, "Approximations to optimal nonlinear filters," *IEEE Trans. Automat. Control*, vol. AC-12, no. 5, pp. 546–556, Oct. 1967.

[37] 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. IEEE Int. Conf. Robot. Automat.*, 2001, pp. 1665–1670.

[38] S. Julier and J. Uhlmann, "Unscented filtering and nonlinear estimation," *Proc. IEEE*, vol. 92, pp. 401–422, 2004.

[39] H. Blom and Y. Bar-Shalom, "The interacting multiple model algorithm for systems with Markovian switching coefficients," *IEEE Trans. Automat. Control*, vol. AC-33, no. 8, pp. 780–783, Aug. 1988.

[40] A. Doucet, J. de Freitas, and N. Gordon, *Sequential Monte Carlo Methods In Practice*. New York: Springer Verlag, 2001.

[41] M. Jerrum and A. Sinclair, "Polynomial-time approximation algorithms for the Ising model," *SIAM J. Comp.*, vol. 22, pp. 1087–1116, 1993.

[42] S. Oh and S. Sastry, "A polynomial-time approximation algorithm for joint probabilistic data association," in *Proc. Amer. Control Conf.*, Portland, OR, Jun. 2005, vol. 2, pp. 1283–1288.

[43] S. Oh, S. Russell, and S. Sastry, Markov Chain Monte Carlo Data Association for Multi-Target Tracking Univ. of California, Merced, CA, Tech. Rep. SoE TR 2008–001, 2008.

[44] S. Kirkpatrick, C. Gelatt, and M. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.

[45] H. Pasula, B. Marthi, B. Milch, S. Russell, and I. Shpitser, "Identity uncertainty and citation matching," in *Advances in Neural Information Processing Systems*. Boston, MA: MIT Press, 2003, vol. 15.

[46] A. McCallum, K. Nigam, and L. Ungar, "Efficient clustering of high-dimensional data sets with application to reference matching," in *Proc. 6th ACM SIGKDD Int. Conf. Knowledge Discovery Data Mining (KDD)*, 2000, pp. 169–178.

[47] B. Marthi, H. Pasula, S. Russell, and Y. Peres, "Decayed MCMC filtering," in *Proc. UAI'02*, Edmonton, AB, Canada, 2002, pp. 319–326.

[48] I. Cox, Multiple Hypothesis Tracking Code Tech. Rep. [Online]. Available: http://www.ee.ucl.ac.uk/~icox/

[49] S. Oh, Multiple Target Tracking for Surveillance Univ. of California, Berkeley, CA, Tech. Rep. UCB/ERL MO3/54, 2003.

[50] C. van Rijsbergen, *Information Retrieval*. London, U.K.: Butterworths, 1979.

[51] M. Bilenko and R. Mooney, "On evaluation and training-set construction for duplicate detection," in *Proc. KDD'03 Workshop Data Cleaning, Record Linkage, Object Consolidation*, Washington, DC, 2003, pp. 7–12.

Songhwai Oh (S'04–M'07) received the B.S., M.S., and Ph.D. degrees in electrical engineering and computer sciences (EECS) from the University of California, Berkeley (UC Berkeley), in 1995, 2003, and 2006, respectively.

He is an Assistant Professor of electrical engineering and computer science in the School of Engineering, University of California, Merced. In 2007, he was a postdoctoral researcher in EECS, UC Berkeley. Before his Ph.D. studies, he worked as a Senior Software Engineer at Synopsys, Inc. and a Microprocessor Design Engineer at Intel Corporation. His research interests include wireless sensor networks, robotics, networked control systems, estimation and learning, and computer vision.

Stuart Russell received the B.A. (with first-class honors) degree in physics from Oxford University, Oxford, U.K., in 1982 and the Ph.D. degree in computer science from Stanford University, Stanford, CA, in 1986.

He then joined the faculty of the University of California at Berkeley, where he is Professor of Computer Science, Chair of the EECS Department, and holds the Smith–Zadeh Chair in Engineering. He is the author of over 150 papers and the book *The Use of Knowledge in Analogy and Induction*; co-author of *Do the Right Thing: Studies in Limited Rationality*, and co-author of *Artificial Intelligence: A Modern Approach*.

Dr. Russell is a Fellow of AAAI and ACM. He received the NSF Presidential Young Investigator Award, the Computers and Thought Award, and the ACM Karl Karlstrom Outstanding Educator Award.

Shankar Sastry (S'79–M'80–SM'95–F'95) received the B.Tech. degree from the Indian Institute of Technology, Bombay, in 1977, and the M.S. degree in electrical engineering and computer science (EECS), the M.A. degree in mathematics, and the Ph.D. degree in EECS from UC Berkeley, in 1979, 1980, and 1981, respectively.

He is currently Dean of the College of Engineering. He was formerly the Director of the Center for Information Technology Research in the Interest of Society (CITRIS) and the Banatao Institute @ CITRIS Berkeley. He served as Chair of the EECS Department from January, 2001 through June 2004. In 2000, he served as Director of the Information Technology Office at DARPA. From 1996 to 1999, he was the Director of the Electronics Research Laboratory at Berkeley (an organized research unit on the Berkeley campus conducting research in computer sciences and all aspects of electrical engineering). He is the NEC Distinguished Professor of Electrical Engineering and Computer Sciences and holds faculty appointments in the Departments of Bioengineering, EECS and Mechanical Engineering. Prior to joining the EECS faculty in 1983 he was a Professor with the Massachusetts Institute of Technology (MIT), Cambridge.