

# A Comparison of Methods for Estimating Track-to-Track Assignment Probabilities

Bret Kragel, *Member, IEEE*, Shawn Herman, and Nick Roseveare

## Abstract

In this paper, we use the two-dimensional track-to-track assignment problem to analyze the trade-off between accuracy and runtime performance for three methods of estimating track-to-track assignment probabilities: enumeration of the  $k$ -best data association hypotheses, importance sampling, and Markov chain Monte Carlo estimation. We also characterize the appropriate operating regime for each method.

Keywords: Multitarget tracking, data association, importance sampling, Markov Chain Monte Carlo, Murty's method or algorithm,  $k$ -best enumeration, track-to-track association, track-to-track assignment probabilities.

## I. INTRODUCTION

The central problem in multitarget, multisensor surveillance is that of partitioning the data from multiple sources into tracks, defined as sets of observations that most likely originated from the same truth object, versus observations that most likely were produced by clutter [1], [2]. If the data can be partitioned correctly, modern filtering algorithms, such as the Kalman filter or the interacting multiple model (IMM) [3], can provide useful estimates of the target states. Although modern tracking algorithms [4] are designed to determine the most likely partition of the data (best data association hypothesis) under some assumed statistical model for measurement and modeling errors, the most *likely* data partition may not be the *correct* partition. For situations involving densely-spaced targets, where many alternate hypotheses may be nearly as likely as the best hypothesis, this may lead to an erroneous perception of certainty. Therefore, just as a track state report includes both a state estimate and a corresponding state covariance matrix, it is beneficial to enhance each association decision in the best hypothesis with an assessment of *data association ambiguity*, defined as the probability that an association decision made in the best hypothesis is actually correct.

Unfortunately, as we shall show in Section II, while it is not difficult to determine the best association hypothesis, computation of exact data association ambiguity is a nontrivial problem; indeed, for interesting problems exact ambiguities cannot be computed with reasonable computational effort, necessitating the use of approximate methods. In this paper, focus on methods for estimating data association ambiguity for the two-dimensional track-to-track assignment problem. We present three methods for estimating data association ambiguity in this environment:  $k$ -best

B.D. Kragel and S.M. Herman are with the Numerica Corporation, Loveland, CO, 80538 USA e-mail: {bret.kragel, shawn.herman}@numerica.us.

enumeration, importance sampling (IS), and Markov chain Monte Carlo (MCMC) estimation. We then compare their performance using Monte Carlo simulations for a set of data association scenarios ranging from low-density problems with small numbers of targets to extremely dense problems.

The  $k$ -best enumeration method [5] is a deterministic algorithm that estimates the probability of the association decisions in the “best” hypothesis by computing the  $k$ -best alternate association hypotheses, then summing and normalizing the likelihoods of the solutions that postulate each decision. This method takes advantage of the multiple hypothesis approach popular in modern tracking systems, meaning that ambiguity is extracted without much additional computational cost to the tracking system. In contrast, MCMC and importance sampling are stochastic algorithms that draw samples from the hypothesis space according to a distribution based on a statistical model for the data. The probability of an association decision is determined by the relative frequency with which that decision appears in the samples. The MCMC method has previously been applied to the data association problem in the tracking context by a number of authors [6], [7], [8]. The use of MCMC for estimating the probability of the association decisions was introduced in [9]; however, this report did not provide an assessment of the appropriate operating regimes for the MCMC and  $k$ -best enumeration methods. The investigation of the conditions under which each of the stated estimation methods is most effective is one of the main contributions of this paper. The application of importance sampling to the estimation of data association probabilities is, to our knowledge, novel.

Our results show that, except for very dense problems, the  $k$ -best enumeration method provides excellent estimates of association probabilities while requiring far fewer computational resources than do the stochastic methods. For very dense problems,  $k$ -best enumeration can no longer provide accurate estimates, and the stochastic methods are preferable. Using this characterization of the preferred operating regime for each method, we present a novel hybrid ambiguity estimation approach that chooses the appropriate estimation method based on a measure of problem density. We conduct our numerical analysis within the context of a two-dimensional track-to-track formulation; however, the results for all three estimation approaches apply to any multiframe tracking environment that involves the solution of a series of two-dimensional problems.

The paper is organized as follows. Section II introduces and illustrates the concept of data association ambiguity and shows that the computation of exact ambiguities is difficult because it requires exhaustive inspection of the hypothesis space, which can become enormous. Section III provides a detailed description of three methods for estimating ambiguity in data association problems. In Section IV, we describe the results of a comparative Monte Carlo analysis of the ambiguity estimation methods presented in Section III. Finally, Section V summarizes our most interesting findings.

## II. AMBIGUITY ASSESSMENT

Consider  $n > 0$  objects moving independently within a surveillance region common to two sensors, where neither the number of objects, nor their true positions is known a priori. Each sensor and an associated tracker detects, observes, and builds tracks on a subset of the objects; that is, we consider the possibility of missed detections. Periodically, the sensors report the estimated track states to a central node for fusion. Let  $Z_1 := \{(\hat{x}_i, P_i), i =$

$1, \dots, n_1\}$  and  $Z_2 := \{(\hat{y}_j, Q_j), j = 1, \dots, n_2\}$  denote the sets of track state reports from the sensors, where the estimation errors in the tracks  $Z_1$  from sensor 1 are independent of those in the set  $Z_2$  from sensor 2.<sup>1</sup> For sensor 1,  $\hat{x}_i$  denotes the estimated position for track  $i$ , and  $P_i$  is the corresponding error covariance of the estimate. Likewise, for sensor 2,  $\hat{y}_j$  denotes the estimated position for track  $j$ , and  $Q_j$  is the error covariance of the estimate.

For this two-dimensional problem, the data association hypothesis space  $\mathcal{H}$  can be viewed as the set of all bijective mappings from subsets of the sensor 1 indices  $\{1, \dots, n_1\}$  onto subsets of the sensor 2 indices  $\{1, \dots, n_2\}$ . The association problem for this data can be generally posed as

$$H^* = \operatorname{argmax}_{H \in \mathcal{H}} \left\{ \frac{\Pr(H | \mathcal{Z})}{\Pr(H_0 | \mathcal{Z})} \right\}, \quad (1)$$

where  $H$  denotes a particular partition of the data into tracks,  $H_0$  denotes a degenerate reference partition in which all observations are assumed to emanate from separate objects,  $\mathcal{H}$  denotes the set of all feasible partitions of the data  $\mathcal{Z} \triangleq \{Z_1, Z_2\}$ , and  $H^*$  denotes the optimal partition; thus, we seek the association hypothesis with the maximum *a posteriori* probability (MAP assignment).

Under the assumption that the targets move independently, the objective function in (1) can be expressed in terms of a product

$$\frac{\Pr(H | \mathcal{Z})}{\Pr(H_0 | \mathcal{Z})} = \frac{p(H, \mathcal{Z})}{p(\mathcal{Z})} \frac{p(\mathcal{Z})}{p(H_0, \mathcal{Z})} = \prod_{(i,j) \in H} \Lambda((i,j) | \mathcal{Z}) \quad (2)$$

of track likelihoods  $\Lambda((i,j) | \mathcal{Z})$ , where the pairs  $(i,j) \in H$  denote the *probabilistic data association event* that track  $i$  from the set  $Z_1$  and track  $j$  from  $Z_2$  emanate from a common truth object, or singleton tracks on objects reported on in  $Z_1$  or  $Z_2$  that went undetected in the other set, denoted by  $(i, 0)$  and  $(0, j)$ , respectively. Appropriate specifications for the track likelihoods can be found in the literature [10], [11], [12], [13]. For this problem, we used the dimensionless likelihood

$$\Lambda((i,j) | \mathcal{Z}) = \frac{\exp\left(-\frac{1}{2}(\hat{x}_i - \hat{y}_j)^T (P_i + Q_j)^{-1} (\hat{x}_i - \hat{y}_j)\right)}{\beta_T (1 - P_D^1)(1 - P_D^2) |2\pi (P_i + Q_j)|^{1/2}}, \quad (3)$$

where the pairs  $(i,j) \in H$  denote the association of sensor 1 indices  $i \in \{1, \dots, n_1\}$  with sensor 2 indices  $j \in \{1, \dots, n_2\}$ ,  $\beta_T$  is the object density in the surveillance region, and  $P_D^k$  is the probability that sensor  $k$ ,  $k = 1, 2$ , detects an object in the surveillance region. In this formulation, the tracks in a hypothesis are composed either of pairs of tracks from sensors 1 and 2 that are postulated by the hypothesis to arise from a common object, or singleton tracks on objects detected by sensor 1 or 2 that went undetected by the other sensor. Since the singleton tracks have a track likelihood ratio of unity, they are suppressed in the product (3), which contains explicit multiplicative terms only for objects tracked by both sensors.

Typically, when evaluating (2), it is advantageous to reformulate the problem using the negative log-likelihoods, or *hypothesis costs*

$$c_H = -\ln \left\{ \frac{\Pr(H | \mathcal{Z})}{\Pr(H_0 | \mathcal{Z})} \right\} = \sum_{(i,j) \in H} c_{ij}, \quad (4)$$

<sup>1</sup>The reported track state estimates are presumed to be time-aligned, either because the sensors report all tracks simultaneously, or by propagating the tracks to a common time. We are ignoring common process noise.

where the track-to-track *association costs*

$$c_{ij} \triangleq -\ln \Lambda((i, j) | \mathcal{Z}) \quad (5)$$

can be arranged into a *cost matrix*  $\mathcal{C} \triangleq \{c_{ij}\}$ ,  $i = 1, \dots, n_1$ ,  $j = 1, \dots, n_2$ . The hypothesis probabilities can be recovered from the costs using

$$\Pr(H | \mathcal{Z}) = \frac{\Pr(H | \mathcal{Z}) / \Pr(H_0 | \mathcal{Z})}{\sum_{\bar{H} \in \mathcal{H}} \Pr(\bar{H} | \mathcal{Z}) / \Pr(H_0 | \mathcal{Z})} = \frac{e^{-c_H}}{\sum_{\bar{H} \in \mathcal{H}} e^{-c_{\bar{H}}}}. \quad (6)$$

The optimal solution for (1) can be found by minimizing the linear objective function (4), which can be accomplished by an assignment algorithm, such as the Jonker-Volgenant-Castañón (JVC) Algorithm [14]. In combination with Murty's Method [15], optimized for the data association problem [16], [17], [18], any assignment algorithm for the solution of (1) can be used to generate the  $k$ -best solutions in order of increasing cost.

Let us now specify that  $\mathcal{H} = \{H_1, H_2, \dots, H_{\varpi}\}$  denotes the ranked set of all possible data association hypotheses for (1), such that  $H_1$  is the solution with the highest probability,  $H_2$  the solution with the next highest probability, and so forth, where  $\varpi = |\mathcal{H}|$  denotes the cardinality of the hypothesis space, and

$$\sum_{i=1}^{\varpi} \Pr(H_i | \mathcal{Z}) = 1. \quad (7)$$

As described in [5], the probabilities  $\Pr(H_i | \mathcal{Z})$  of the data association hypotheses are directly related to the probabilities of the data assignments as demonstrated next. As before, let  $(i, j)$  denote the event that track  $i$  from  $Z_1$  and track  $j$  from  $Z_2$  emanate from a common truth object; then, the exact probability  $\Pr((i, j))$  of this event is given by

$$\Pr((i, j)) = \sum_{H \in \mathcal{H}^{ij}} \Pr(H | \mathcal{Z}), \quad (8)$$

where  $\mathcal{H}^{ij} \subseteq \mathcal{H}$  denotes the subset of data association hypotheses in  $\mathcal{H}$  that postulate the event  $(i, j)$ . The assignment probability  $\Pr((i, j))$  serves as a measure of *data association uncertainty* or *ambiguity*: If the association  $(i, j)$  is postulated by the best solution  $H_1$  and  $\Pr((i, j)) < 1$ , the tracking correlator estimates there is a nonzero probability that this association decision is incorrect. However, to compute the assignment probability  $\Pr((i, j))$ , we need to find all data association hypotheses in which the event  $(i, j)$  is postulated. Although current combinatorial algorithms are capable of finding the optimal data association hypothesis with acceptable computational effort, the size of  $\mathcal{H}^{ij}$  can become enormous for complicated scenarios; thus, it may be computationally impractical for any algorithm to explicitly enumerate all “reasonable” hypotheses.<sup>2</sup>

<sup>2</sup>For the two-source problem with missed detections, the number of data association hypotheses is given by

$$\sum_{k=0}^{n_1} \binom{n_1}{k} \frac{n_2!}{(n_2 - k)!}, \text{ for } n_1 \leq n_2.$$

### III. AMBIGUITY ESTIMATION

As discussed in Section II, exact computation of association ambiguities is impractical for difficult data association problems. For this reason, we describe in this section three methods for estimating data association ambiguity:  $k$ -best hypothesis enumeration, Markov chain Monte Carlo methods, and importance sampling.

#### A. Ambiguity Estimation Using $k$ -Best Enumeration

Estimation of association probabilities using  $k$ -best enumeration is straightforward: Let  $\mathcal{H}^k = \{H_1, H_2, \dots, H_k\}$  denote the set of  $k$ -best ranked solutions of (1). Then, as long as  $k$  is sufficiently large, (8) can be approximated using

$$\Pr((i, j)) \approx \sum_{H \in \mathcal{H}^{ij} \cap \mathcal{H}^k} \Pr(H | \mathcal{Z}). \quad (9)$$

It is important to note that the  $k$ -best method is not “compelled” to compute  $k$  full hypotheses; indeed, if no hypothesis with nonnegligible probability remains after the  $\bar{k} < k$  best hypotheses have been generated, the method can halt processing. This capability can sometimes result in significant reduction in computational effort.

#### B. Monte Carlo Ambiguity Estimation

As mentioned at the beginning of Section II, for tracking scenarios with large numbers of densely-spaced objects,  $k$ -best enumeration typically cannot generate a number of hypotheses sufficient to compute accurate association probabilities with reasonable computational resources. For these situations, we consider two stochastic algorithms: Markov chain Monte Carlo sampling and importance sampling. Since both methods are special cases of Monte Carlo estimation, we begin with a short exposition on the use of Monte Carlo methods for ambiguity estimation.

Consider the problem of evaluating the expectation

$$\mathbb{E}[g(X)] = \int_{x \in \mathcal{X}} g(x)p(x)dx, \quad (10)$$

where  $X$  is a random variable defined on the space  $\mathcal{X}$  and distributed according to the probability function  $p$ ,  $\mathbb{E}[\cdot]$  denotes the expectation operator, and  $g$  is some function of interest. Then  $\mathbb{E}[g(X)]$  can be estimated by generating a random sample  $\{X_1, X_2, \dots, X_n\}$  from  $p$  and computing the empirical average [19]

$$\bar{g}_n = \frac{1}{n} \sum_{i=1}^n g(x_i), \quad (11)$$

where the lower case letters denote a realization of the random sample. If  $\mathbb{E}[g(X)] < \infty$ , the Strong Law of Large Numbers guarantees

$$\lim_{n \rightarrow \infty} \bar{g}_n \stackrel{a.e.}{=} \mathbb{E}[g(X)]. \quad (12)$$

Ambiguity assessment can be accommodated within the framework of (10)–(12) by identifying  $\mathcal{X} \leftrightarrow \mathcal{H}$  and  $X \leftrightarrow H$ , choosing  $\Pr(H | \mathcal{Z})$  as the probability function for  $H$ , and setting  $g(H) = \mathbb{I}_{ij}(H)$ , where

$$\mathbb{I}_{ij}(H) = \begin{cases} 1, & \text{if } (i, j) \in H, \\ 0, & \text{otherwise.} \end{cases}$$

Then,

$$\Pr((i, j)) = \mathbb{E}[\mathbb{I}_{ij}(H)] \approx \frac{1}{n} \sum_{k=1}^n \mathbb{I}_{ij}(H_k), \quad (13)$$

where the random sample  $\{H_1, \dots, H_n\}$  is generated from  $\Pr(H | \mathcal{Z})$ .

The difficulty now becomes the problem of generating samples from

$$\Pr(H | \mathcal{Z}) = \frac{p(H, \mathcal{Z})}{p(\mathcal{Z})}, \quad (14)$$

an issue complicated by two factors. First, though  $p(H, \mathcal{Z})$  can be computed [10], the partition function  $p(\mathcal{Z})$  cannot be evaluated directly with reasonable computational resources. Second, even if (14) could be computed, drawing samples from this distribution would remain nontrivial. Nevertheless, there are methods available for solving this problem. Two such methods, Markov chain Monte Carlo sampling and importance sampling, appear to be suitable for the problem of ambiguity estimation. In the following, we describe these methods and assess their performance within the context of ambiguity estimation as expressed in (13).

### C. Markov Chain Monte Carlo Ambiguity Estimation

Recently, some authors [6], [7], [8], [20], [21] have begun experimenting with the use of Markov chain Monte Carlo methods for multitarget tracking. The primary advantage of MCMC is the ability to generate samples from fairly arbitrary probability distributions; however, the samples generated are strongly correlated [22], [23], which significantly complicates convergence analysis [19]. Nevertheless, due to the inherent limitations of ambiguity estimation using  $k$ -best enumeration, we wish to evaluate the potential of MCMC for enhancing ambiguity estimation in difficult tracking scenarios.

1) *Markov Chains*: Given a large (finite) set  $\mathcal{X}$  of combinatorial structures (e.g., the space of all data association hypotheses), a sequence of random variables  $\{X_0, X_1, X_2, \dots\}$  defined on  $\mathcal{X}$  is called a *Markov chain* if

$$\Pr(X_{n+1} = x | X_n = x_n, \dots, X_0 = x_0) = \Pr(X_{n+1} = x | X_n = x_n), \quad x \in \mathcal{X},$$

where  $x_n \in \mathcal{X}$  denotes a realization of  $X_n$ ,  $n = 0, 1, 2, \dots$ ; thus, a Markov chain is characterized by the condition that, given the current state, all future states are conditionally independent of the prior states. Let  $\pi_n(x) = \Pr(X_n = x)$  denote the state distribution at time (step)  $n$ , with initial distribution  $\pi_0(x)$ . If the *transition kernel*  $T_n$  at time  $n$  is defined by  $T_n(x, y) = \Pr(X_{n+1} = y | X_n = x)$ ,  $x, y \in \mathcal{X}$ , then by the total probability theorem,

$$\pi_{n+1}(y) = \sum_{x \in \mathcal{X}} \pi_n(x) T_n(x, y).$$

A Markov chain is called *homogeneous* if the transition probabilities do not depend on time; that is,  $T_n \equiv T$ ,  $\forall n$ . The distribution  $\pi(x)$  is *stationary* with respect to the Markov chain with transition probabilities  $T_n(x, y)$  if for all  $y \in \mathcal{X}$  and all  $n$ ,

$$\pi(y) = \sum_{x \in \mathcal{X}} \pi(x) T_n(x, y).$$

---

**Algorithm 1** Metropolis-Hastings Algorithm
 

---

**Given:**  $\pi, x \in \mathcal{X}, N > 0$

- 1: **for**  $n = 1$  to  $N$  **do**
  - 2:   Select candidate state  $y \in \mathcal{X}$  from the proposal distribution  $q(y|x)$ .
  - 3:   Sample  $u$  from  $U(0, 1]$ .
  - 4:   **if**  $u < a(x, y) = \min \left\{ 1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)} \right\}$  **then**
  - 5:     Set  $x = y$ .
  - 6:   **else**
  - 7:     Set  $x = x$ .
  - 8:   **end if**
  - 9: **end for**
- 

A homogeneous Markov chain is said to be *irreducible* if there is a positive probability of moving from any state  $x$  to any other state  $y$  in a finite number of time steps, i.e.,  $\forall x, y \in \mathcal{X}, \exists n = n(x, y)$ , such that  $T^n(x, y) > 0$ , where  $T^n$  is the  $n$ -step composition of  $T$ . The *period* of a state  $x \in \mathcal{X}$  is defined as  $d(x) = \gcd\{n \geq 1 \mid T^n(x, x) > 0\}$ .<sup>3</sup> The Markov chain is called *aperiodic* if  $d(x) = 1, \forall x$ .

The *detailed balance condition* holds for a homogeneous Markov chain if there is a distribution  $\pi(x)$ , such that for all states,  $x, y \in \mathcal{X}$ ,

$$\pi(y)T(y, x) = \pi(x)T(x, y).$$

The detailed balance condition implies  $\pi(x)$  is a stationary distribution, since

$$\pi(y) = \pi(y) \sum_{x \in \mathcal{X}} T(y, x) = \sum_{x \in \mathcal{X}} \pi(x)T(x, y).$$

2) *The Metropolis-Hastings Algorithm:* MCMC can be used when one desires to sample from a distribution  $\pi$  known only up to a constant factor, i.e.,  $\pi = \pi^*/C$ , where  $\pi^*$  is known, but the constant  $C$  is unknown or difficult to compute. Note that this is the situation in the multitarget tracking problem, where one must evaluate probabilities of the form (14), and the unknown normalizing constant is  $C = p(\mathcal{Z})$ . MCMC operates by constructing an aperiodic and irreducible Markov chain that has the distribution of interest as its unique stationary distribution, then sampling from this chain as shown below.

The most popular method in the tracking literature for generating a Markov chain with the desired characteristics is the Metropolis-Hastings algorithm [24], [20], [19], described in Algorithm 1. Given the current state  $x \in \mathcal{X}$ , a *candidate state*  $y \in \mathcal{X}$  is chosen according to a *proposal distribution*  $q(y|x)$ . The proposed state  $y$  is accepted only if the value of the *acceptance function*

$$a(x, y) = \min \left\{ 1, \frac{\pi(y)q(x|y)}{\pi(x)q(y|x)} \right\} \quad (15)$$

<sup>3</sup> $\gcd \triangleq$  greatest common divisor

is greater than or equal to a number  $u$  drawn uniformly from the interval  $(0, 1]$ . If  $y$  is rejected, then the chain remains in the current state  $x$ . In this manner, every proposed state with  $\pi(y)q(x|y) \geq \pi(x)q(y|x)$  is accepted, while any other proposal has probability  $\frac{\pi(y)q(x|y)}{\pi(x)q(y|x)}$  of being accepted. This second condition is designed to keep the chain from getting “stuck” at a local optimum. For  $y \neq x$ , the resulting transition kernel, computed by multiplying the probability of proposal with the probability of acceptance, is given by

$$T(x, y) = q(y|x)a(x, y) = \min \left\{ q(y|x), \frac{\pi(y)}{\pi(x)}q(x|y) \right\}.$$

Thus, as can be seen by

$$\pi(y)T(y, x) = \pi(y) \min \left\{ q(x|y), \frac{\pi(x)}{\pi(y)}q(y|x) \right\} = \pi(x) \min \left\{ \frac{\pi(y)}{\pi(x)}q(x|y), q(y|x) \right\} = \pi(x)T(x, y),$$

the inclusion of  $q(y|x)$  in (15) ensures that the detailed balance condition holds,<sup>4</sup> meaning that  $\pi(x)$  is stationary for the Markov chain. Since aperiodicity is guaranteed by the acceptance function, if the proposal distribution can be constructed in such a way that the Metropolis-Hastings Markov chain is also irreducible, then as noted by Robert and Casella [19], the satisfaction of the aperiodicity and irreducibility conditions implies that a sequence produced by the Metropolis Hastings algorithm can be employed as an independent identically distributed sample from  $\pi(x)$ , with convergence of the empirical average of the chain guaranteed by the ergodic theorem [19], [25]; that is,

$$\lim_{n \rightarrow \infty} \frac{1}{n} \sum_{i=1}^n g(x_i) \underset{\text{a.e.}}{=} \text{E}[g(X)], \quad (16)$$

for the sequence  $\{x_1, x_2, \dots, x_n\}$  of samples produced by the Metropolis-Hastings Markov chain.

3) *Estimating Association Probabilities Using MCMC*: If we can use the Metropolis-Hastings method to construct a Markov chain that converges to  $\Pr(H | \mathcal{Z})$ , then we can draw sample hypotheses  $\{H_1, H_2, H_3, \dots\}$  from this chain and, according to (13), estimate the individual association ambiguities using

$$\Pr((i, j)) \approx \frac{N_{ij}}{N_{MC}}, \quad (17)$$

where  $N_{MC}$  is the total number of samples drawn, and  $N_{ij}$  is the number of sampled hypotheses that postulate the event  $(i, j)$ . It is important to note that  $n$  indexes the *sampled* hypotheses here; thus, in contrast to the situation of Section III-A, the hypotheses drawn from the simulated Markov chain are not ordered; that is,  $c_{H_n} < c_{H_{n+1}}$  does not hold in general for any  $n$ .

For the two-source problem described in Section II, we employ an MCMC proposal distribution  $q(\tilde{H}|H)$  that constructs a proposal hypothesis  $\tilde{H}$  from the current hypothesis  $H$  by altering the set of associations in  $H$  in one of three ways: addition, deletion, or swap. The proposal distribution  $q(\tilde{H}|H)$  is described in detail in Algorithm 2. The algorithm begins by selecting a feasible pair of track indices at random from  $Z_1$  and  $Z_2$ . If the selected indices are already associated in  $H$ , the proposed hypothesis  $\tilde{H}$  is generated by removing this association from  $H$  in Line 4. If the indices are not associated, but *both* indices are currently assigned to other indices, a swap is attempted. If the test in Line 7 fails, a swap among the selected associations is not possible, and  $\tilde{H} = H$ . If neither, or exactly

<sup>4</sup>The detailed balance condition always holds for the transition from  $x$  to itself.

---

**Algorithm 2** Proposal Distribution for MCMC Testing
 

---

**Given:** Current hypothesis  $H$ .

- 1: Select an index  $i$  uniformly at random from  $1, \dots, n_1$ .
- 2: Select an index  $j$  uniformly at random from

$$\{j \mid j \in 1, \dots, n_2, c_{ij} < \infty\},$$

where  $c_{ij}$  denotes the cost of associating track  $i$  from  $Z_1$  with track  $j$  from  $Z_2$ .

- 3: **if**  $H(i) = j$  **then**
  - 4:     Set  $H(i) = 0$  (deletion).
  - 5: **else**
  - 6:     **if**  $H(i) = j'$  and  $H(i') = j$  for some  $i' \neq i$  and  $j' \neq j$  **then**
  - 7:         **if**  $c_{i'j'} < \infty$  **then**
  - 8:             Set  $H(i') = j'$  and  $H(i) = j$  (swap).
  - 9:         **end if**
  - 10:     **else**
  - 11:         **if**  $H(i) = j'$  for some  $j' \neq j$  **then**
  - 12:             Set  $H(i) = 0$  (deletion).
  - 13:         **end if**
  - 14:         **if**  $H(i') = j$  for some  $i' \neq i$  **then**
  - 15:             Set  $H(i') = 0$  (deletion).
  - 16:         **end if**
  - 17:         Set  $H(i) = j$  (addition).
  - 18:     **end if**
  - 19: **end if**
- 

one, of the selected indices is currently associated with another index, an addition is generated in Line 17. In the latter case, the current association must first be deleted in Line 12 or Line 15 to maintain a feasible hypothesis.

4) *On the Difficulty of Constructing Appropriate Proposal Distributions:* Although (16) guarantees convergence in theory for irreducible, aperiodic Markov chains, this does not guarantee effective sampling in practice. Consider the cost matrix

$$\mathcal{C} = \begin{bmatrix} \infty & \infty & -20 \\ \infty & -35 & \infty \\ -15 & \infty & -35 \end{bmatrix}, \quad (18)$$

Hypothesis	Assignments	Cost	Probability
$H_1$	(3, 2, 1)	-70	$\sim 0.5$
$H_2$	(0, 2, 3)	-70	$\sim 0.5$
$H_3$	(0, 2, 1)	-55	$1.53 \times 10^{-7}$
$H_4$	(3, 2, 0)	-50	$1.03 \times 10^{-9}$

TABLE I: Assignments for the 4-best hypotheses corresponding to the cost matrix (18). Tracks from the set  $Z_1$  are indexed by values in the assignment vector, while tracks from the set  $Z_2$  are indexed by position in the assignment vector. For example, the assignment (3, 2, 1) indicates that track 1 from  $Z_2$  was assigned to track 3 from  $Z_1$ , track 2 from  $Z_2$  to track 2 from  $Z_1$ , and track 3 from  $Z_2$  to track 1 from  $Z_1$ . A ‘0’ indicates that the corresponding track from  $Z_2$  was not assigned to a track from  $Z_1$ .

which might occur after gating for two sets of track reports,  $Z_1$  and  $Z_2$  in a typical tracking scenario, where the entry at row  $i$  and column  $j$ ,  $i = 1, \dots, 3$ ,  $j = 1, \dots, 3$ , is the cost, computed according to (5), of assigning track  $i$  from  $Z_1$  to track  $j$  from  $Z_2$ . The cost of not assigning a track is zero (not shown in matrix). There are 10 hypotheses with cost less than infinity for this problem, with the best two solutions consuming nearly all of the probability mass. The hypotheses with probability exceeding machine epsilon are listed in Table I. The  $k$ -best algorithm can easily enumerate all solutions, resulting in the exact ambiguity matrix

$$\mathcal{A} = \begin{bmatrix} 0.0 & 0.0 & 0.5 \\ 0.0 & 1.0 & 0.0 \\ 0.5 & 0.0 & 0.5 \end{bmatrix},$$

where  $\mathcal{A}_{ij} = \Pr((i, j))$  is the probability of assigning track  $i$  from  $Z_1$  to track  $j$  from  $Z_2$ . The entries of the ambiguity matrix are computed from the costs using (6). The ambiguities do not sum to unity across some of the rows and columns due to positive probability that the row or column remains unassigned.

In contrast, despite using  $10^6$  Monte Carlo samples during repeated experiments, the ambiguity matrix returned by the MCMC sampler of Section III-C3 reflects exactly one of the two best hypotheses depending on which hypothesis,  $H_1$  or  $H_2$ , is visited first by the MCMC routine:

$$\mathcal{A}_1 = \begin{bmatrix} 0.0 & 0.0 & 1.0 \\ 0.0 & 1.0 & 0.0 \\ 1.0 & 0.0 & 0.0 \end{bmatrix} \quad \text{or} \quad \mathcal{A}_2 = \begin{bmatrix} 0.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix},$$

where  $\mathcal{A}_1$  corresponds to  $H_1$ , and  $\mathcal{A}_2$  corresponds to  $H_2$ . Upon examination of the probable hypotheses and the proposal distribution from Algorithm 2, it becomes clear why this is so. Once the chain has reached either  $H_1$  or  $H_2$ , the chance of leaving that hypothesis becomes remote. Indeed, none of the simple proposals of Algorithm 2 allows a direct transition between hypotheses  $H_1$  and  $H_2$ ; thus, a transition between these hypotheses requires an intermediate transition through a hypothesis with much smaller probability. Since the maximum probability of

the remaining hypotheses is on the order of  $10^{-7}$ , we would on average need  $10^7$  Monte Carlo samples before completing a transition between  $H_1$  and  $H_2$ .

The problem described in this simple example may be circumvented by employing a more sophisticated proposal distribution [22], [24], in this case by allowing for a concurrent swap and deletion at Line 7 of Algorithm 2; however, there is no guarantee that this would solve similar difficulties that could arise involving multiple correlation swaps. Moreover, every modification of the proposal procedure generally results in additional computational effort and, since both the proposal distribution  $q(\tilde{H}|H)$  and its inverse  $q(H|\tilde{H})$  must be computed, increased design complexity. Another possibility is to shift the costs so that hypotheses that postulate unassigned columns become more probable. This approach would promote Markov chain convergence by increasing the acceptance rate for difficult transitions, but the chain would converge to a distribution different from the target distribution.

Based on these considerations, we expect that the MCMC method may not always perform well for problems involving a small number of competing hypotheses that effectively partition the hypothesis space into regions of significant probability mass that are separated by areas of low mass. In contrast, for problems in which only one solution has significant probability mass, or all solutions have nearly equal probability, we expect the MCMC method to perform better.

#### D. Ambiguity Estimation Using Importance Sampling

While Markov chain Monte Carlo methods provide a general framework for sampling from a (nearly) arbitrary probability distribution, the estimation of quantities computed from the sample may suffer from slow convergence due to the inherent correlation between MCMC samples. A somewhat less general method that utilizes independent samples is importance sampling [19], [26].

Consider again the problem of evaluating the expectation (10), where the probability distribution  $p(x)$  is too complex to be sampled directly but can be computed to within a multiplicative constant; that is,  $p(x) = p^*(x)/C_p$ , where  $p^*(x)$  can be computed. Assume now the existence of another probability function,  $q(x)$ , from which independent samples can be generated easily, and which can also be evaluated to within a multiplicative constant; that is  $q(x) = q^*(x)/C_q$ , where  $q^*(x)$  can be computed. If the support of  $p$  is contained in the support of  $q$ , then

$$\mathbb{E}[g(X)] = \int_{x \in \mathcal{X}} g(x) \frac{p(x)}{q(x)} q(x) dx, \quad (19)$$

and  $\mathbb{E}[g(X)]$  can be estimated by drawing a random sample from  $q(X)$  and computing

$$\hat{g}_n = \frac{1}{\bar{w}_n} \sum_{i=1}^n g(x_i) w_i, \quad (20)$$

with weights  $w_i = p^*(x_i)/q^*(x_i)$ , and  $\bar{w}_n = \sum_{i=1}^n w_i$ . Similar to (11), the empirical average (20) converges to  $\mathbb{E}[g(X)]$  by the Strong Law of Large Numbers [19]. In principle, any distribution that meets these conditions can be used as a sampling distribution, but not all distributions will be equal in terms of convergence rate. Generally, distributions that are similar to the target distribution but have “fatter tails” are preferable [19].

Based on the preceding discussion, a good sampling distribution for the two-source track-to-track association problem analyzed in Section IV will prefer hypotheses with lower costs, but should generate with positive probability any hypothesis with finite cost. Intuitively then, any method for constructing samples should be based on the cost matrix itself. Perhaps the easiest such approach is to create a likelihood matrix by transforming the entries of the cost matrix into likelihood space by negation and exponentiation,<sup>5</sup> then generating samples by considering each column in turn, assigning a row based on a random draw according to the relative likelihoods in the column. The assigned row would then be removed from consideration, and the next column processed.<sup>6</sup> Unfortunately, naive application of this approach leads to ineffective sampling. Consider following cost and likelihood matrices:

$$\mathcal{C} = \begin{bmatrix} -15 & -15 \\ 0 & 0 \end{bmatrix} \quad \text{and} \quad \mathcal{L} \approx \begin{bmatrix} 1 - \epsilon & 1 - \epsilon \\ \epsilon & \epsilon \end{bmatrix},$$

where  $\epsilon \approx 3.06e^{-7}$ . The bottom row in each matrix corresponds to missed detections and we have normalized the columns of the likelihood matrix so that the entries of each column comprise a probability distribution over the row indices. According to the cost matrix, each column should have an equal chance of being assigned to the first row; however, since the first column is always processed prior to the second column, it will (nearly) always be assigned to the first row based on the entries in the first column of the likelihood matrix, leaving the second column unassigned. This is indeed a poor sampling distribution. One could attempt to circumvent this difficulty by shuffling the column order between samples, but this would mean that the same sample could be drawn in different ways. When computing the sample likelihood  $q^*(x)$ , these alternatives would have to be accounted for, a problem that becomes enormously complex even for small matrices.

One heuristic for mitigating this issue is to normalize across the rows of the likelihood matrix to form a probability distribution across each row, making it less likely that a column processed early in the sampling procedure will take a row if columns processed later also have a positive probability for assignment to the row. We found this could be accomplished without excessive computational effort by partial matrix balancing; that is, by successive normalization of the columns and rows of the likelihood matrix. For the simple case above, one cycle of normalization applied to the likelihood matrix  $\mathcal{L}$  results in the matrix with 0.5 in all entries, which is the correct ambiguity matrix. Our importance sampling procedure is described in detail in Algorithm 3. The input to the algorithm consists of the cost matrix, the number of samples  $N_s$ , and the number of balancing cycles  $N_b$ . The importance weights in Line 16 are generated and stored in cost space. This is necessary to maintain numeric stability.

Fig. 1 depicts a more sophisticated example of the transformation of a cost matrix into a sampling matrix. The cost matrix corresponds to a situation in which one sensor has reported four tracks, while the second has reported only three. A row has been added to the bottom of the cost matrix to account for missed detections. The cost matrix of Fig. 1(a) is transformed into the likelihood matrix of Fig. 1(b) according to Lines 1 — 2 of Algorithm 3. In

<sup>5</sup>Recall that the costs are simply negative log-likelihoods.

<sup>6</sup>We assume without loss of generality that the cost matrix has more rows than columns.

---

**Algorithm 3** Importance Sampling Procedure
 

---

**Given:** Cost matrix  $\mathcal{C} \triangleq \{c_{ij}\}_{i=1,\dots,m, j=1,\dots,n}$ ,  $n \leq m$ , balancing steps  $N_b \geq 0$ , samples  $N_s \geq 1$ .

- 1: Compute the scaling factor  $\sigma = \min_{i,j} \{c_{ij}\}$ .
  - 2: Compute the likelihood matrix  $\mathcal{L} \triangleq \exp(-\mathcal{C} + \sigma)$ .
  - 3: **for**  $b = 1$  to  $N_b$  **do** {Matrix balancing.}
  - 4:   Normalize columns of  $\mathcal{L}$ .
  - 5:   Normalize rows of  $\mathcal{L}$ .
  - 6: **end for**
  - 7: **for**  $s = 1$  to  $N_s$  **do**
  - 8:   Initialize sampling matrix  $\mathcal{S} = \mathcal{L}$ .
  - 9:   Set  $\mathcal{I} = \{1, \dots, m\}$ .
  - 10:   **for**  $j = 1$  to  $n$  **do**
  - 11:     Normalize column  $j$  of  $\mathcal{S}$  over row indices  $i \in \mathcal{I}$ .
  - 12:     Sample  $u$  from  $U(0, 1]$ .
  - 13:     Set  $r_j = \max \{i \mid \sum_{k=1}^i s_{kj} \leq u, i \in \mathcal{I}\}$ , where  $s_{ij}$  is the element in row  $i$  and column  $j$  of  $\mathcal{S}$ .
  - 14:     Remove index  $r_j$  from  $\mathcal{I}$ .
  - 15:   **end for**
  - 16:   Compute and store sample weight in cost space:  $w_s^c = \sum_{j=1}^n (c_{r_j j} + \ln(s_{r_j j}))$ .
  - 17: **end for**
  - 18: Set  $\omega = \min_{s=1,\dots,N_s} \{w_s^c\}$ . {Compute cost shift for numeric stabilization.}
  - 19: **for**  $s = 1$  to  $N_s$  **do**
  - 20:   Set  $w_s = \exp(-w_s^c + \omega)$ . {Compute sample likelihoods.}
  - 21: **end for**
  - 22: Use sample likelihoods to estimate ambiguity according to (20) on Page 11.
- 

principle, one could use the likelihood matrix as a sampling matrix, but order effects render this approach inefficient. The more efficient sampling matrix of Fig. 1(c) is derived from the likelihood matrix using one step of the matrix balancing procedure of Algorithm 3, Lines 3 — 6; that is  $N_b = 1$ .

An example of sample selection using the sampling matrix from Fig. 1(c) is depicted in Fig. 2. Prior to sample selection, each column of the sampling matrix corresponds to a sampling distribution over the rows. In Step (a), row 3 was assigned to column 1 with probability 0.14. This row will no longer be available for assignment during draws for subsequent columns. In Step (b), the second column is renormalized to account for the assignment of the third row to the first column, and row 4 is assigned to column 2 with probability 0.59. This row will no longer be available for assignment during draws for subsequent columns. Finally, the third column is renormalized in Step (c) to account for previous assignments, and row 1 is assigned to column 3 with probability 0.74. The likelihood for

the sample is obtained by multiplying the probabilities of the column draws, approximately 0.06 for this sample.

$$\begin{array}{ccc}
 \begin{bmatrix} -13.68 & -14.29 & -13.62 \\ -13.60 & -13.86 & -12.08 \\ -12.73 & -13.64 & -14.08 \\ -12.12 & -14.92 & -11.88 \\ 0 & 0 & 0 \end{bmatrix} & 
 \begin{bmatrix} 0.29 & 0.53 & 0.27 \\ 0.27 & 0.35 & 0.06 \\ 0.11 & 0.28 & 0.43 \\ 0.06 & 1.00 & 0.05 \\ 3.31\text{e-}7 & 3.31\text{e-}7 & 3.31\text{e-}7 \end{bmatrix} & 
 \begin{bmatrix} 0.30 & 0.17 & 0.28 \\ 0.46 & 0.19 & 0.10 \\ 0.14 & 0.11 & 0.54 \\ 0.10 & 0.53 & 0.08 \\ 3.39\text{e-}7 & 1.06\text{e-}7 & 3.37\text{e-}7 \end{bmatrix} \\
 \text{(a) Cost matrix.} & \text{(b) Likelihood matrix.} & \text{(c) Sampling matrix.}
 \end{array}$$

Fig. 1: Example construction of importance sampling matrix.

$$\begin{array}{ccc}
 \begin{bmatrix} 0.30 & 0.17 & 0.28 \\ 0.46 & 0.19 & 0.10 \\ \textcircled{0.14} & 0.11 & 0.54 \\ 0.10 & 0.53 & 0.08 \\ 3.39\text{e-}7 & 1.06\text{e-}7 & 3.37\text{e-}7 \end{bmatrix} & 
 \begin{bmatrix} 0.30 & 0.20 & 0.61 \\ 0.46 & 0.21 & 0.22 \\ 0.14 & 0.11 & 0.54 \\ 0.10 & \textcircled{0.59} & 0.17 \\ 3.39\text{e-}7 & 1.20\text{e-}7 & 7.29\text{e-}7 \end{bmatrix} & 
 \begin{bmatrix} 0.30 & 0.20 & \textcircled{0.74} \\ 0.46 & 0.21 & 0.26 \\ 0.14 & 0.11 & 0.54 \\ 0.10 & 0.59 & 0.17 \\ 3.39\text{e-}7 & 1.20\text{e-}7 & 8.82\text{e-}7 \end{bmatrix} \\
 \text{(a) A random draw from the first column.} & \text{(b) A random draw from the second column.} & \text{(c) A random draw from the third column.}
 \end{array}$$

Fig. 2: An example of sample selection using the sampling matrix from Fig. 1(c).

The effectiveness of this method will depend critically on the relationship between the target and sampling distributions. Since the sampling distribution is derived from the cost matrix, the convergence rate may vary depending upon the target configuration.

#### E. Range Accuracy and Resolution: Practical Constraints

Before proceeding with an analysis of ambiguity estimation methods, it is useful to consider the range of operating regimes, or target densities, that can be expected to arise in practice. This analysis is critical for two reasons: First, it helps in the development of a test environment that is simultaneously realistic enough to be useful, and robust enough to thoroughly test the estimation procedures throughout the entire range of expected operating conditions. Second, it will later promote deeper understanding of the strengths and weaknesses of each ambiguity estimation algorithm, and guide the development of procedures for switching between algorithms based on current operating conditions. For purposes of ambiguity estimation, the target density is determined by the number of targets, their estimated separation, and the estimation error. The number of targets determines the number of possible association configurations; i.e., the size of the hypothesis space, while the separation and estimation error determine the fraction of hypotheses with nonnegligible likelihood. The following exposition by Slocumb [27] uses the Cramer-Rao lower bound (CRLB) on the variance of the range estimate to derive a bound on target densities that may arise in practice for monopulse radars.

The CRLB is a lower bound on the variance of any unbiased estimator [28]. For a linear frequency modulation (LFM) waveform with bandwidth  $\Delta f$ , the CRLB for the range estimate [29, Page 376] is

$$\text{Var}(\hat{R}) \triangleq \sigma_R^2 \geq \frac{3c^2}{4\pi^2(2SNR)(\Delta f)^2}, \quad (21)$$

where  $\Delta f$  denotes the bandwidth,  $c$  is the speed of light, and  $SNR$  is the signal-to-noise ratio in decibels, while the range resolution is given by

$$\Delta R = \frac{c}{2\Delta f}. \quad (22)$$

Equations (21) and (22) imply that range resolution and accuracy are coupled. To see this coupling, we combine (21) with (22) to obtain

$$\text{Var}(\hat{R}) \triangleq \sigma_R^2 \geq \frac{3c^2}{4\pi^2(2SNR)(c/2\Delta R)^2},$$

which yields the linear relationship

$$\sigma_R \geq \left( \frac{1}{(\pi/\sqrt{3})\sqrt{2SNR}} \right) \Delta R,$$

or, equivalently,

$$\Delta R \leq \pi \sqrt{\frac{2}{3}SNR} \sigma_R. \quad (23)$$

If the range variance meets the CRLB, then for a given value of signal-to-noise ratio, the radar resolution is equal to the right-hand side of (23), and only those targets with separation exceeding this value can be resolved. Since the range variance should be close to the CRLB for a well-designed radar, this effectively implies an upper limit on measurement uncertainty for a given resolution.

Because covariance matrices are often inflated for residual bias or time propagation (i.e., non-overlapping coverage), we consider problems that are denser than the bound implied by (23) as part of our numerical analysis in Section IV. However, we use the bound derived for monopulse radars in this section to provide a center point for the total range of densities we consider. This provides a range of operating conditions that covers those that can be expected in practice.

#### IV. COMPARATIVE ANALYSIS OF AMBIGUITY ESTIMATION METHODS

In this section, we perform a comparative analysis of the three methods of ambiguity estimation presented in the preceding sections, namely  $k$ -best enumeration, importance sampling, and MCMC, using the two-dimensional track-to-track assignment problem. Missed detections were considered, but false targets were not simulated. Specifically, we considered a problem in which two sensors track targets within a three-dimensional cubic surveillance region  $\mathcal{V}$  of volume  $V = l_V^3$ , where  $l_V$  denotes the length of one side of the cube. Each of the two sensors detects a subset of the objects within the surveillance region, and, after establishing tracks, the sensors simultaneously transmit track state estimates of the object positions to a central node for fusion. The track-to-track association costs for this problem are given by [10]

$$c_{ij} = \frac{1}{2} (\hat{x}_i - \hat{y}_j)^T (P_i + Q_j)^{-1} (\hat{x}_i - \hat{y}_j) + \frac{1}{2} \ln |2\pi (P_i + Q_j)| - \ln \beta_T (1 - P_D^1)(1 - P_D^2), \quad (24)$$

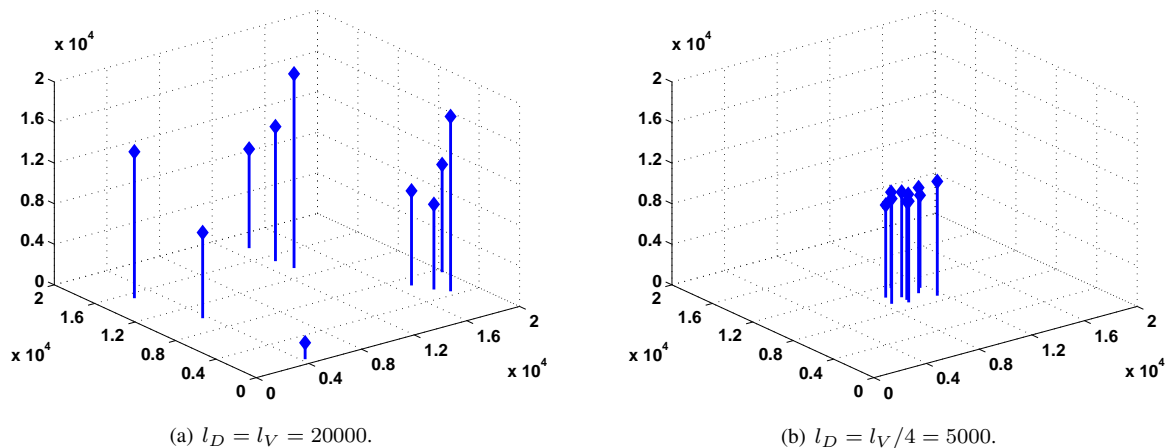


Fig. 3: Example target distributions. The targets in plot (a) were distributed uniformly at random throughout the entire surveillance region, while the targets in plot (b) were generated uniformly at random in a small subset of the surveillance region, resulting in a much denser problem.

where  $\beta_T$  is the object density in the sensor surveillance region, and  $P_D^k$  is the probability that sensor  $k$ ,  $k = 1, 2$ , detects an object in the surveillance region.

The difficulty (i.e., the number of “plausible” hypotheses) of this problem is determined by the target density, which in turn is a function of the number of targets and their spacing. While the number of targets determines the absolute cardinality of the hypothesis space, it is the inter-target spacing that determines the number of hypotheses with significant probability mass. When using this scenario for simulation purposes, the target density for a fixed number of targets is controlled by size of region in which the targets are distributed. Specifically, targets are distributed in cubic subsets  $\mathcal{V}_D \subset \mathcal{V}$  of the surveillance region with volumes  $V_D = l_D^3$ , where  $l_D$  denotes the length of one side of  $\mathcal{V}_D$ . This is illustrated in the plots of Fig. 3, where the target density is much higher in the situation depicted in Fig. 3(b).

We simulated scenarios of various densities by generating targets uniformly at random in 3-dimensional surveillance regions of various sizes according to Algorithm 4, where  $\mathcal{A} = \{k\text{-best, IS, MCMC}\}$  denotes the set of ambiguity estimation algorithms,  $\mathcal{T} = \{5, 20, 100\}$  is the set of target counts used,  $\mathcal{S}$  denotes 30 uniformly distributed surveillance region sizes, and  $N_m$  is the number of Monte Carlo simulations for each configuration. The setting  $N_m = 2000$  was used for the 5-target scenarios, while  $N_m = 500$  was used for the 20-target and 100-target scenarios, with a smaller number of samples used in the latter cases to keep total runtime manageable. For the purpose of simulating a large range of possible operating conditions, the smallest surveillance region measured 100 meters on a side, while the largest measured 20000 meters on a side. The probability of detection was set at  $P_D = 0.90$  for both sensors, and the reported error covariances were assumed to be normally distributed with a

---

**Algorithm 4** Metrics Generation for Monte Carlo Testing
 

---

```

1: for  $n \in \mathcal{T}$  do
2:   for  $V \in \mathcal{S}$  do
3:     for  $m = 1$  to  $N_m$  do
4:       Generate a set of  $n$  targets uniformly at random within a surveillance region of volume  $V$ .
5:       Generate noisy estimates by sensor 1 and sensor 2 of the target states.
6:       Detect and generate track state reports for  $n_1 \leq n$  sensor 1 targets and  $n_2 \leq n$  sensor 2 targets.
7:       Determine the optimal track-to-track assignment hypothesis  $H_1$  using the  $k$ -best algorithm.
8:       Compute  $P_E^{m,V} = \frac{n_c}{n_1}$ , where  $n_c$  denotes the number of correct sensor 1 associations in the optimal hypothesis.
9:       for  $A \in \mathcal{A}$  do
10:        Compute the assignment probabilities  $\Pr((i, j))$  using algorithm  $A$ .
11:        Compute  $P_A^{m,V} = \frac{1}{n_1} \sum_{(i,j) \in H_1} \Pr((i, j))$  for algorithm  $A$ .
12:       end for
13:     end for
14:     Compute the empirical probability of correct association as  $P_E^{V,n} = \frac{1}{N_m} \sum_{m=1}^{N_m} P_E^{m,V}$ .
15:     for  $A \in \mathcal{A}$  do
16:       Compute the predicted probability of correct association as  $P_A^{V,n} = \frac{1}{N_m} \sum_{m=1}^{N_m} P_A^{m,V}$ .
17:     end for
18:   end for
19: end for

```

---

standard deviation of  $\sigma = 100$  meters in each dimension; that is,  $P_i = Q_j = \sigma^2 \cdot I_3$ ,  $\forall i, j$ , where  $I_3$  denotes the  $3 \times 3$  identity matrix. The dimensions of the smallest surveillance regions considered were equal to the noise level, representing very dense scenarios. *The selected simulation parameters do not reflect any specific scenario, sensor characteristics, or multitarget sensor performance, but were chosen for comparison of the proposed ambiguity estimation methods under controlled conditions.*

We also varied the number of samples used by our estimation algorithms for each of the test configurations presented above. Since the computational effort required by each method to generate one candidate hypothesis varies considerably, we divided our simulations into two sets: The first set tested accuracy versus problem density for a given number of hypotheses or samples, while the second set tested accuracy versus runtime for a given density. Nevertheless, we achieved reasonably comparable runtimes for the first set of tests by adjusting the number of samples used by each method for a given test configuration. Generally, for a given setting of  $k = \{100, 1000, 10000\}$

in the  $k$ -best approach,  $10k$  importance samples, and  $100k$  MCMC proposals were generated.

### A. Simulation Metrics

Since the number of nonnegligible hypotheses can be expected to increase exponentially, it is necessary to devise a measure of scenario density that is comparable across tests as the target density increases, and allows consideration of the resolution constraints discussed in Section III-E. We chose to use the average Mahalanobis distance (normalized by the dimension of the state space) between each truth object and its nearest neighbor. This *normalized target spacing* (NTS) was computed according to

$$\zeta \triangleq \frac{1}{3n} \sum_{i=1}^n \min_{j \neq i} \{(x_i - x_j)^T (P_i + Q_j)^{-1} (x_i - x_j)\},$$

where  $n$  is the number of targets, and  $x_i$  denotes the true position of Target  $i$ ,  $i = 1, \dots, n$ .

If we make the conservative assumption that the range resolution considerations of Section III-E apply equally to all dimensions of our target position vectors, then normalized target spacing is related to resolution by

$$\begin{aligned} \zeta &= \frac{1}{3n} \sum_{i=1}^n \min_{j \neq i} \{(x_i - x_j)^T (P_i + Q_j)^{-1} (x_i - x_j)\} \\ &\approx \frac{1}{3n} \sum_{i=1}^n \Delta x \mathbf{e}_3^T \left( \frac{1}{2} \sigma^{-2} I_3 \right) \Delta x \mathbf{e}_3 \\ &= \frac{1}{3n} \frac{1}{2} \sum_{i=1}^n 3 \frac{\Delta x^2}{\sigma^2} \\ &= \frac{1}{2} \frac{\Delta x^2}{\sigma^2}, \end{aligned} \tag{25}$$

where  $\mathbf{e}_3 = (1, 1, 1)^T$ , and  $\Delta x$  is defined as the average of the component differences; i.e.,

$$\Delta x \triangleq \frac{1}{3n} \sum_{i=1}^n \left\| \min_{j \neq i} \{(x_i - x_j)\} \right\|_1.$$

Now, by reformulating (23) as

$$\frac{1}{2} \left( \frac{\Delta R}{\sigma_R} \right)^2 \leq \frac{\pi^2}{3} SNR, \tag{26}$$

we see the correspondence between (23) and (25). Extrapolating from the discussion at the end of Section III-E, we expect on average that all targets are resolvable for  $\zeta > \bar{\zeta} \triangleq \pi^2 SNR / 3$ , while ever fewer targets can be resolved as  $\zeta$  decreases below this value. A conservative value of  $SNR = 10$  yields a bound of  $\bar{\zeta} \approx 10^{1.5}$ .

The main goal of our analysis is to assess the relative performance versus runtime of the algorithms under consideration. This comparison was based on two metrics: average maximum distance from the baseline ambiguity matrix versus runtime, and predicted probability of correct association versus scenario density. The first of these metrics, distance from the baseline ambiguity matrix, was computed simply as<sup>7</sup>

$$\frac{1}{N_m} \sum_{m=1}^{N_m} \max_{i,j} \{ |\mathcal{A}_*^m - \mathcal{A}_{est}^m| \}, \tag{27}$$

<sup>7</sup>As in Section III-C4, the ambiguity matrix  $\mathcal{A}$  is defined by  $\mathcal{A}_{ij} = \Pr((i, j))$ ,  $i = 1, \dots, n_1$ ,  $j = 1, \dots, n_2$ , where  $n_k$  denotes the number of reports from sensor  $k$ ,  $k = 1, 2$ .

where for Monte Carlo run  $m \in \{1, \dots, N_m\}$ ,  $\mathcal{A}_*^m$  denotes the exact, or *baseline*, ambiguity matrix,  $\mathcal{A}_{est}^m$  is the *estimated* ambiguity matrix, and the maximum is taken over all entries of the absolute difference of the matrices. For the second metric, the predicted probability of correct association (computed in Line 16 of Algorithm 4) was compared with the empirical probability of correct association (computed in Line 14 of Algorithm 4). For a sufficient number of Monte Carlo runs, the empirical probability of correct association should yield a reasonably accurate estimate of the true average probability that an association is correct, while the predicted probability will be a function of the accuracy of the association probability estimates derived from the  $k$ -best, importance sampling, and MCMC methods.

### B. Simulation Results

Using the simulation environment described above, we compared the performance of  $k$ -best enumeration, importance sampling, and MCMC for estimation of association probabilities (i.e., ambiguity). The results are plotted in Fig. 4. The normalized target spacing is plotted increasing from left to right along the  $x$ -axis, meaning that the problem density decreases from left to right. The average maximum deviation is plotted increasing from bottom to top along the  $y$ -axis.

Figs. 4(a) and 4(b) display the average maximum deviation from the baseline ambiguity matrix for the 5-target case. The cardinality of the hypothesis space for this case was 1546, so the baseline ambiguity matrix could be computed exactly using the  $k$ -best approach with  $k = 1546$ . The  $k$ -best enumeration method (with  $k \in \{100, 1000\}$ ) outperforms the Monte Carlo methods for this problem because the number of hypotheses with significant probability seldom exceeds  $k$ . The importance sampling and MCMC methods appear competitive with each other for this scenario.

The average maximum deviation for the 20-target case is depicted in Figs. 4(c) and 4(d). The cardinality of the hypothesis space for this case was well beyond the capacity of the  $k$ -best method, so the baseline ambiguity matrix was generated by MCMC using 10 million samples. These plots are more interesting with respect to the  $k$ -best method. For both values of  $k$ , the performance of  $k$ -best enumeration equals or exceeds the Monte Carlo methods until the NTS is well below unity. Note also that a ten-fold increase in  $k$  does not significantly change the curve for the  $k$ -best method; thus, while a costly increase in  $k$  is not likely to improve the performance of the  $k$ -best method for dense scenarios, the  $k$ -best method is more accurate down to problem densities slightly below unity. For the Monte Carlo methods, the importance sampler and MCMC are again competitive.

Figs. 4(e) and 4(f) display the average maximum deviation for the enormous 100-target test case, where the baseline was again generated using 10 million MCMC samples. Compared to the 20-target plots, we see that the average maximum errors are greater for all ambiguity estimation methods. This is due both to the inherent higher density of the problem, and to the fact that the maximum in each Monte Carlo run is taken over a larger number of matrix elements—10000 instead of 400. The  $k$ -best estimation method works better than the Monte Carlo methods down to a density of about unity. Comparing the change in the error curves between Fig. 4(e) and Fig. 4(f) for NTS

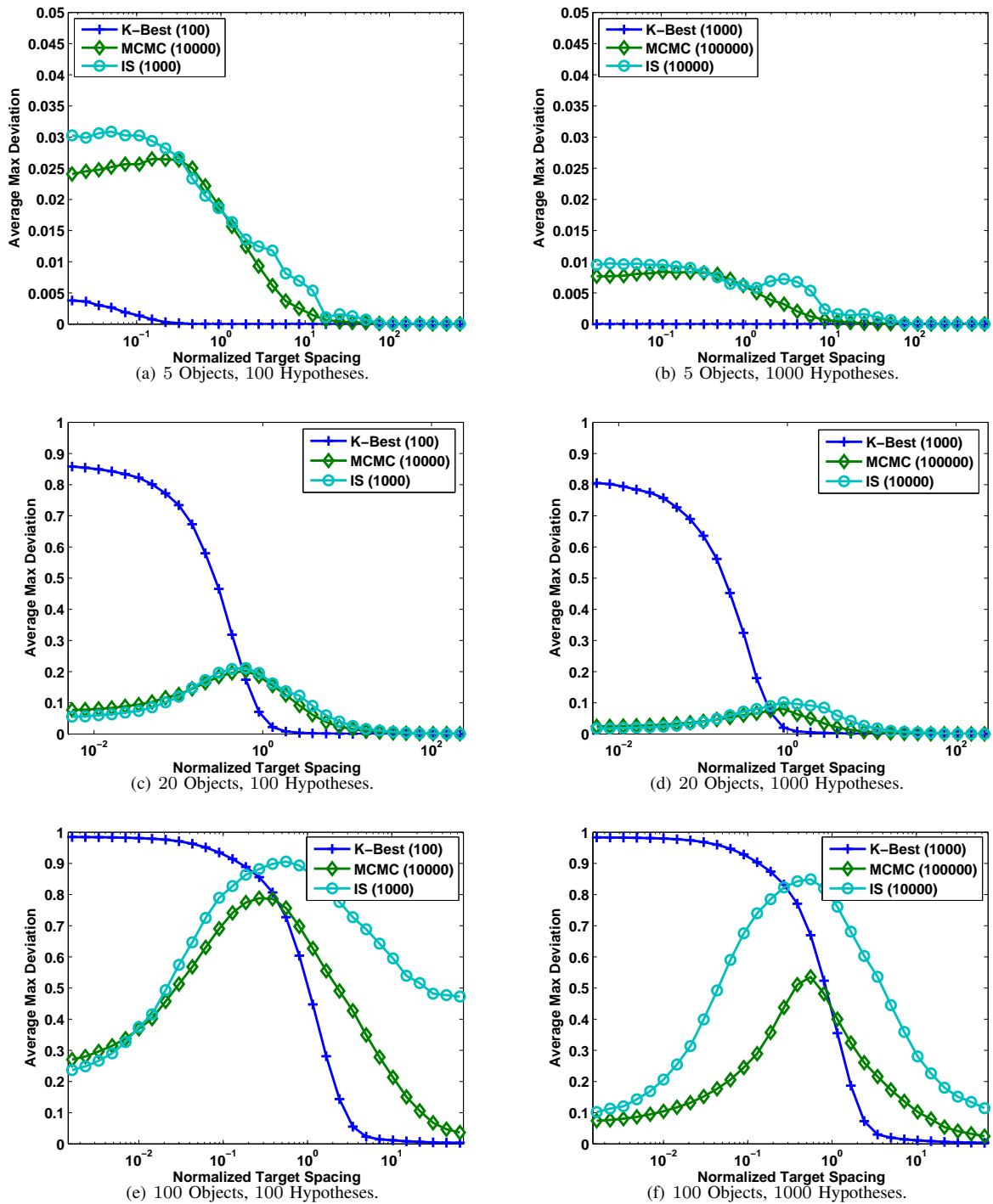


Fig. 4: Average maximum deviation.

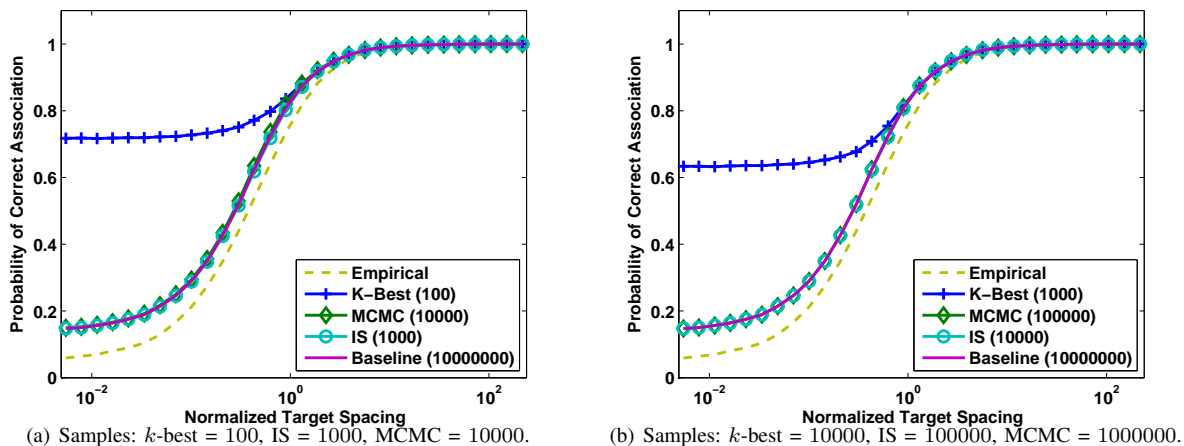


Fig. 5: Probability of correct association: 20 targets.

values below unity, it appears that MCMC scales better than the competing estimation methods for this problem. A ten-fold increase in the number of samples results in a significant improvement in performance for MCMC, while the improvements for  $k$ -best and importance sampling are not as encouraging. The transition point at which the Monte Carlo methods yield better estimates than  $k$ -best has shifted slightly to the right in the 100-target example as compared to the 20-target case; however, the size of the errors in the transition region are considerably higher for all methods. Based on these results, it appears that accurate correlation probability estimates for dense scenarios with large numbers of targets may be achievable only if significant computational resources are available.

It is noteworthy that for all test scenarios, the Monte Carlo methods demonstrate their best performance at the extremes of the range of tested densities. For low densities, this happens because the only hypothesis with non-negligible probability is the correct hypothesis, making it the only hypothesis selected by the sampling procedure. At the other extreme, as the size of the surveillance region approaches the noise level, the uncertainty in target position becomes much higher than the average target separation, so that no hypothesis is any more likely than another, and the hypothesis probabilities approach a uniform distribution, as can be seen by examining the empirical curves of Fig. 5. Since the importance sampling distribution and MCMC acceptance function are driven by relative likelihoods, these sampling methods will, in effect, mimic a uniform sampling distribution, which is appropriate for this situation. Finally, the performance of the Monte Carlo methods deteriorates toward the center of each plot for the reasons discussed in Section III-C4. At densities in the range around unity, the probability space may contain a number of competing hypotheses that partition the hypothesis space into regions that are difficult for the MCMC chain to adequately explore in the allotted number of samples. For the importance sampler, it is in this region of the density range that the greatest deviation between the sample and target distributions occurs, leading to slower average convergence.

The probability of correct association is displayed in Fig. 5 for the 20-target case. The desired performance for

Object Count	Per-Sample Runtime	
	IS	MCMC
5	$2.58 \times 10^{-6}$	$1.30 \times 10^{-6}$
20	$12.17 \times 10^{-6}$	$1.57 \times 10^{-6}$
100	$91.50 \times 10^{-6}$	$2.27 \times 10^{-6}$

TABLE II: Average per-sample runtimes (seconds) for Monte Carlo ambiguity estimation methods.

this metric is not tied to the absolute values along the  $y$ -axis, but is instead demonstrated by conformity with the empirical curve. Estimated values for the probability of correct correlation that lie above the empirical line are too optimistic, while those lying below the line are too pessimistic.

The baseline curve was again computed using 10 million MCMC proposals. Similar to Fig. 4, Fig. 5(a) corresponds to a setting of  $k = 100$ ; however, Fig. 5(b) uses  $k = 10000$ , rather than  $k = 1000$ . Note also that this metric aggregates the data more than the average maximum deviation metric does. In the left-hand plot, we see that both the importance sampling and MCMC methods are very close to the baseline curve, while the baseline, importance sampling, and MCMC curves are superimposed in the right-hand plot. The  $k$ -best curve also shows good agreement with the baseline curve until the problem density overwhelms this method. This indicates that these three methods are indeed converging to the curve corresponding to the exact ambiguity; i.e. the baseline curve.

The attentive reader will have noticed that the empirical curve deviates from the predicted curves in Figs. 5(a) and 5(b) for NTS values below approximately two. This deviation occurs because the assumptions on initial target distribution and estimation errors used to derive the likelihood formulation from which the costs (24) were computed no longer hold for such closely-spaced targets. As the density continues to increase (decreasing NTS), this violation becomes greater, resulting in increasing deviations between the empirical and predicted curves. Failure of the  $k$ -best method does not occur until after the empirical curve has begun to deviate from the baseline curve, indicating that the  $k$ -best method is viable except for densities that begin to violate the assumptions underlying the derivation of the track likelihoods.

We now turn our attention to runtime analysis of the three ambiguity estimation algorithms. Table II displays the average time per-sample required to generate importance samples and MCMC proposals for both the 5-target and 20-target cases. While the average time required by the importance sampler increased rapidly as the size of the problem increases, the average time required to generate an MCMC proposal remains nearly flat. This occurs because our MCMC proposal generator alters at most a single association for each new hypothesis, whereas the independent samples generated by the importance sampler require reevaluation of all associations for each new hypothesis.

Figs. 6 and 7 plot average runtime against average maximum deviation for a fixed density. In both figures, the left-hand plot corresponds to a normalized target spacing of 0.1, while the spacing for the right-hand plot is unity.

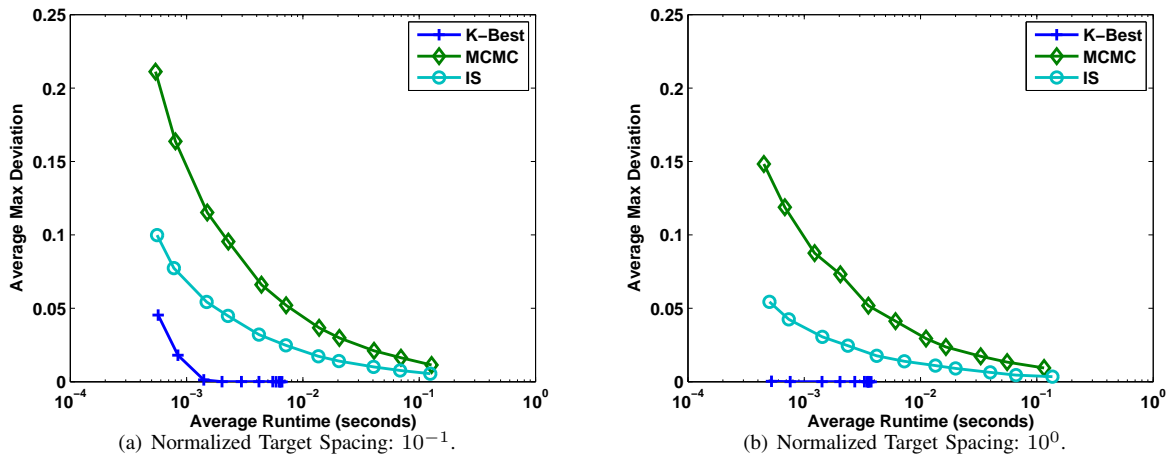


Fig. 6: Average maximum deviation vs. runtime: 5 targets.

Different runtime and error combinations were generated by sweeping the  $k$ -best settings over

$$k \in \{30, 50, 100, 150, 300, 500, 1000, 1500, 3000, 5000, 10000\}.$$

The importance samples and MCMC proposals sweeps were chosen to align the left sides of the plot curves as well as possible. The results are clear for the 5-target case in Fig. 6: Regardless of target spacing,  $k$ -best enumeration exhibits superior performance, while the importance sampler is slightly better than MCMC. The 20-target results in Fig. 7 are more interesting. In the right-hand plot, we again see that  $k$ -best enumeration maintains a solid advantage, while the importance sampler and MCMC are competitive, though the importance sampler exhibits better performance at lower runtimes. In the left-hand plot, the  $k$ -best method can no longer enumerate a sufficient number of hypotheses, resulting in significant performance degradation. In contrast, the importance sampling and MCMC methods remain effective, though the results of Table II and Fig. 7 indicate that MCMC will be more appropriate for the most difficult problems because it scales better than importance sampling.

In summary, our results show that the  $k$ -best method is uniformly superior to the Monte Carlo methods down to a normalized target spacing of slightly less than unity. Outside of these conditions, either importance sampling or MCMC can be used for small to medium numbers of targets. For dense problems with normalized target spacing less than unity and large numbers of targets, the performance of the MCMC method is superior to that of  $k$ -best and importance sampling.

### C. Improvements to $k$ -Best Ambiguity Assessment

The results of Section IV-B showed clearly that the  $k$ -best method of ambiguity estimation is more effective than the Monte Carlo algorithms for values of normalized target spacing above unity, while the stochastic algorithms performed better for normalized target spacing below  $10^{-1}$ . Values of normalized target spacing between  $10^{-1}$

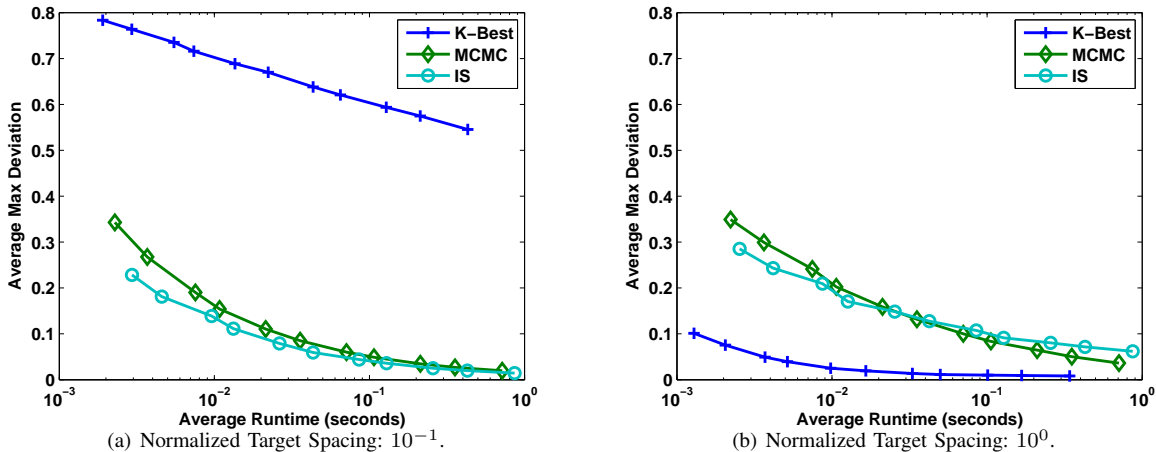


Fig. 7: Average maximum deviation vs. runtime: 20 targets.

and unity represented a transition region. Based on this analysis, we present in this section a hybrid ambiguity estimation procedure that chooses the appropriate estimation method by monitoring average problem density.

1) *Hybrid Ambiguity Assessment*: In Section IV, we compared the performance of the  $k$ -best and MCMC algorithms with respect to ambiguity estimation and found that although the  $k$ -best is superior for scenarios with normalized target spacing above unity, the MCMC algorithm exhibits improved performance for extraordinarily dense scenarios. With these results in mind, we devised an adaptive ambiguity estimation method that uses a heuristic to identify cases in which the MCMC method is preferable.

Based on the results of Section IV, we would like to use the MCMC method for ambiguity estimation whenever the probability mass in the hypotheses returned by the  $k$ -best solver does not dominate the mass in the unexplored portion of the hypothesis space. Since the probabilities of the hypotheses can be estimated using their approximate likelihoods according to (6), a test comparing the sum of likelihoods of the explored and unexplored hypothesis subspaces is both appealing and sensible. For the hypotheses returned by the  $k$ -best solver, this sum is given by

$$P_E = \sum_{i=1}^k e^{-cH_i};$$

however, the sum for the unexplored portion of the hypothesis space cannot be computed in this manner because we do not have the costs for those hypotheses. Nevertheless, since we are interested only in knowing when the mass of the unexplored hypothesis space is significant relative to that of the hypotheses returned by the solver, it is sufficient to use an upper bound for the unknown value. There are many heuristics that could be used to make the necessary comparison; we chose to use MCMC for ambiguity estimation when the maximum increase in likelihood,

$$P_U = \bar{k} \cdot e^{-cH_k},$$

where  $\bar{k} \triangleq \min(k, |\mathcal{H}| - k)$ , that could be achieved by doubling the number of hypotheses returned by the  $k$ -best

Number of Objects	$k$ -Best Hypotheses	MCMC Proposals	Detection Probability
5	100, 1000	$100 \cdot k$	0.9
10	100, 1000	$100 \cdot k$	0.9
20	100, 1000	$100 \cdot k$	0.9

TABLE III: Testing parameters for MCMC Ambiguity Assessment vs. Hybrid Ambiguity Assessment.

solver, exceeds some user-determined fraction  $\vartheta$  of  $P_E$ :

$$P_U \geq \vartheta P_E. \quad (28)$$

2) *Numerical Evaluation of Hybrid Ambiguity Assessment*: To assess the performance of the hybrid ambiguity estimation method introduced in the previous section, we ran a set of simulations with the configuration parameters listed in Table III. As with the previous simulations, the minimum and maximum surveillance region dimensions were 100 meters and 20000 meters, respectively, and the standard deviation was held constant at 100 meters. We chose  $\vartheta = 0.1$  in (28), ensuring that the MCMC method was used whenever the maximum possible additional increase in likelihood that could be achieved by doubling the setting of  $k$  exceeds 10% of the total likelihood carried in the  $k$ -best hypotheses.

The results of our tests are displayed in Fig. 8. The hybrid and MCMC curves match for the denser scenarios, indicating that our hybrid method has successfully switched to the MCMC method when necessary. On the other hand, the runtimes for the hybrid method were significantly lower for all three simulations in Table III, which indicates that the hybrid method stuck with the faster  $k$ -best ambiguity estimation when appropriate. The average runtimes in seconds for the 5 target scenario were  $2.2 \times 10^{-2}$  for the hybrid method and  $4.1 \times 10^{-1}$  for the pure MCMC. The corresponding runtimes were  $5.1 \times 10^{-2}$  and  $4.3 \times 10^{-1}$ , respectively for the 10 target case, and  $1.6 \times 10^{-1}$  and  $3.5 \times 10^{-1}$  for the 20 target case.

## V. SUMMARY

We evaluated three different techniques for online estimation of association probabilities in multitarget tracking:  $k$ -best enumeration, Markov chain Monte Carlo, and importance sampling. We used a variety of object count and density configurations, with the more difficult configurations allowing us to evaluate the sensitivity of the algorithms to various operating conditions. Despite its inherent combinatorial limitations,  $k$ -best enumeration had the best runtime and accuracy performance down to a normalized target spacing of slightly less than unity. For very difficult problems, the Monte Carlo techniques were superior to  $k$ -best enumeration; both the importance sampler and MCMC were able to achieve arbitrarily accurate association probabilities with reasonable computational effort for problems in the 5 to 20-target range. In our tests, the importance sampler required generation of fewer samples

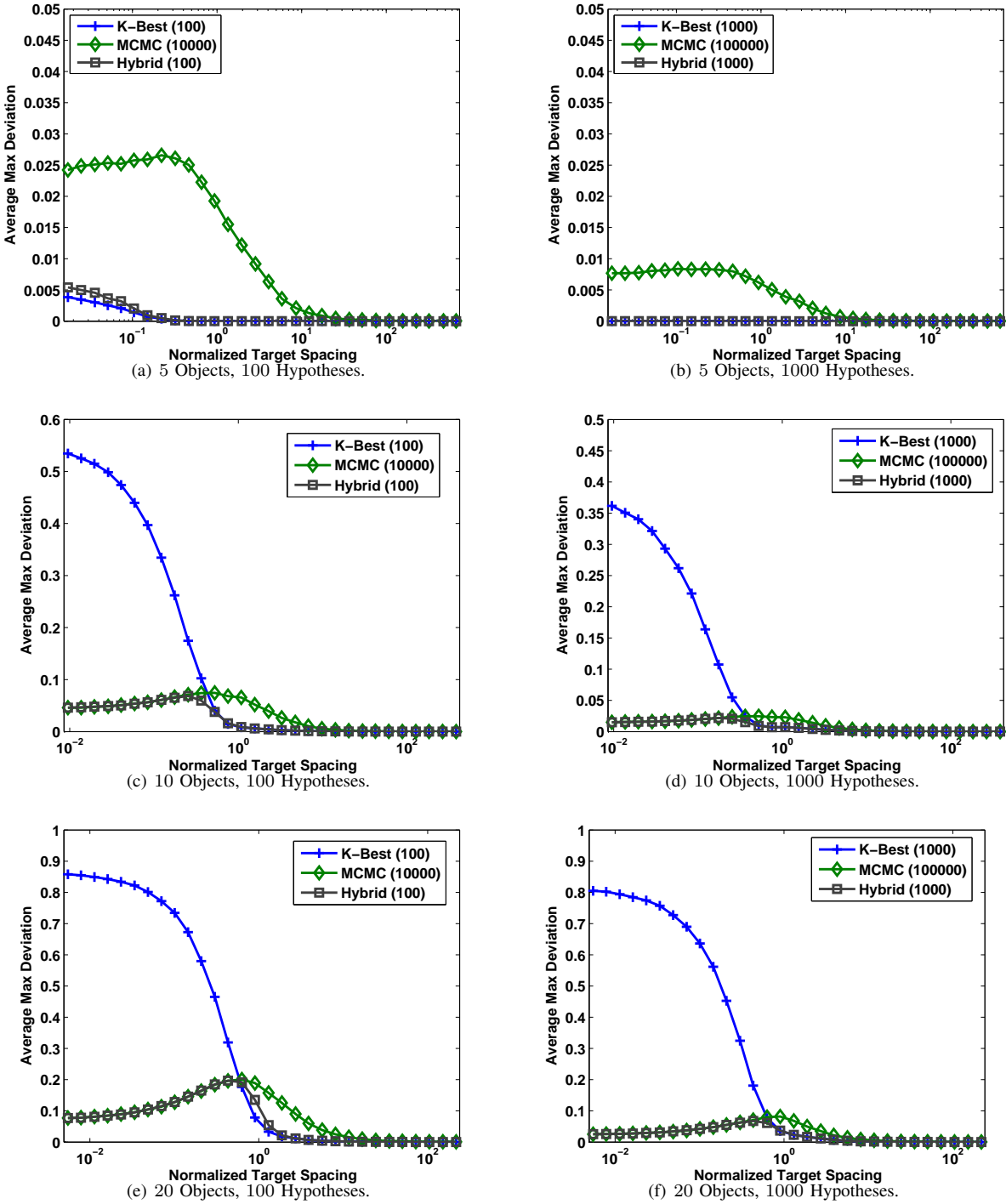


Fig. 8: Comparison of Hybrid and MCMC methods for ambiguity estimation.

than the MCMC approach, although the per-sample runtime was smaller for MCMC and did not increase appreciably with problem size. As the number of samples required for a given task increases, MCMC performance eventually exceeds that of the importance sampler. These results suggest that, depending on desired accuracy, MCMC is appropriate for problems involving large numbers of densely-spaced targets, while either MCMC or importance sampling may be used for moderate numbers of densely-spaced targets.

We also presented a hybrid ambiguity estimation method that adaptively chose either  $k$ -best enumeration or MCMC approximation based on the estimated target density. Our tests indicated that this method provides superior performance with essentially no additional computational effort. In order to better decide when to swap estimation methods in the hybrid approach, additional research is needed to better understand the point at which stochastic methods begin outperforming  $k$ -best enumeration.

## REFERENCES

- [1] Y. Bar-Shalom and T. E. Fortmann, *Tracking and Data Association*. Academic Press, 1988.
- [2] S. Blackman, *Multiple Target Tracking with Radar Application*. Norwood, MA: Artech House, 1986.
- [3] Y. Bar-Shalom, X. R. Li, and T. Kirubarajan, *Estimation with Applications to Tracking and Navigation*. New York: John Wiley & Sons, Inc., 2001.
- [4] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.
- [5] S. Gadaleta, S. Herman, S. Miller, F. Obermeyer, B. J. Slocumb, A. B. Poore, and M. Levedahl, "Short-term ambiguity assessment to augment tracking data association information," in *Eighth International Conference on Information Fusion, Philadelphia, PA*, vol. 1, 2005, p. 8.
- [6] N. Bergman and A. Doucet, "Markov chain Monte Carlo data association for target tracking," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 2, August 2000, pp. II705–II708.
- [7] S. Cong, L. Hong, and D. Wicker, "Markov-chain Monte-Carlo approach for association probability evaluation," in *IEE Proceedings on Control Theory and Applications*, vol. 151, no. 2, March 2004, pp. 185–193.
- [8] S. Oh, S. Russell, and S. Sastry, "Markov chain Monte Carlo data association for general multiple-target tracking problems," in *Proc. of the 43rd IEEE Conference on Decision and Control, Nassau, Bahamas*, vol. 1, December 2004, pp. 735–742.
- [9] M. Horsley, "A comparison of two methods for computing track-to-track assignment probabilities," MIT Lincoln Laboratory, Project Report PH-35, April 2008.
- [10] B. D. Kragel, S. Danford, and A. B. Poore, "Concurrent map data association and absolute bias estimation with an arbitrary number of sensors," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, O. E. Drummond, Ed., vol. 6969, no. 50, Orlando, Florida, March 2008.
- [11] S. Mori and C.-Y. Chong, "Track-to-track association metric I.I.D.-non-poisson cases," in *Proceedings of the Sixth International Conference of Information Fusion*, vol. 1, Cairns, Queensland, Australia, July 2003, pp. 553–559.
- [12] L. D. Stone, M. L. Williams, and T. M. Tran, "Track-to-track association and bias removal," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, O. E. Drummond, Ed., vol. 4278, August 2002, pp. 315–329.
- [13] J. P. Ferry, "Exact association probability for data with bias and features," *Journal of Advances in Information Fusion*, vol. 5, no. 1, pp. 41–67, June 2010.
- [14] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325–340, March 1987.
- [15] K. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, pp. 682–687, 1968.
- [16] I. J. Cox and M. L. Miller, "On finding ranked assignments with application to multitarget tracking and motion correspondence," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 31, no. 1, pp. 486–489, January 1995.
- [17] I. Cox, M. L. Miller, and R. Danchick, "A comparison of two algorithms for determining ranked assignments with application to multitarget tracking and motion correspondence," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 33, no. 1, pp. 295–301, January 1997.

- [18] R. Popp, K. Pattipati, and Y. Bar-Shalom, "Dynamically adaptable m-best 2-D assignment algorithm and multi-level parallelization," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 35, no. 4, pp. 1145–1160, October 1999.
- [19] C. P. Robert and G. Casella, *Monte Carlo Statistical Methods*. Springer, 2004.
- [20] H. Pasula, S. J. Russell, M. Ostland, and Y. Ritov, "Tracking many objects with many sensors," in *Proceedings of the International Joint Conferences on Artificial Intelligence (IJCAI-99)*, vol. 2, 1999, pp. 1160–1171.
- [21] S. Mori and C. Chong, "Evaluation of a posteriori probabilities of multi-frame data association hypotheses," in *Proceedings of SPIE Conference on Signal and Data Processing of Small Targets*, O. E. Drummond, Ed., vol. 6699, no. 20. San Diego: SPIE, August 2007.
- [22] M. Jerrum and A. Sinclair, "The Markov chain Monte Carlo method: An approach to approximate counting and integration," in *Approximations for NP-Hard Problems*, D. Hochbaum, Ed. PWS Publishing, 1996, pp. 482–520.
- [23] R. M. Neal, "Probabilistic inference using Markov chain Monte Carlo methods," Dept. of Computer Science, University of Toronto, Technical Report CRG-TR-93-1, September 1993.
- [24] S. Oh and S. Sastry, "A polynomial-time approximation algorithm for joint probabilistic data association," in *American Control Conference (ACC)*, Portland, OR, June 2005.
- [25] L. Tierney, "Introduction to general state-space Markov theory," in *Markov Chain Monte Carlo in Practice*, W. Gilks, S. Richardson, and D. Spiegelhalter, Eds. Chapman & Hall / CRC, 1996, ch. 4, pp. 59–74.
- [26] D. J. MacKay, *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.
- [27] B. Slocumb, "Radar analysis of MIT/LL CaT data," Numerica Corporation Internal Report, June 2004.
- [28] A. Papoulis, *Probability, Random Variables, and Stochastic Processes*, 3rd ed. McGraw-Hill, Inc., 1991.
- [29] W. D. Blair and B. M. Keel, "Radar systems modeling for tracking," in *Multitarget-Multisensor Tracking: Applications and Advances*, Y. Bar-Shalom and W. D. Blair, Eds. Artech House, 2001, vol. 3, ch. 7, pp. 321–393.