Defence Research and Development Canada | Recherche et développement pour la défense Canada

CAN UNCLASSIFIED

DRDC | RDDC
technologysciencetechnologie

Algorithms for the multi-sensor assignment problem in the δ-generalized labeled multi-Bernoulli filter

Jun Ye Yu
Augustin-Alexandru Saucan
Mark Coates
Michael Rabbat
McGill University

Published in: Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2017 IEEE 7th International Workshop on, 10-13 Dec. 2017, Curacao, Netherlands Antilles
DOI: 10.1109/CAMSAP.2017.8313114

Prepared by:
McGill University
Electrical and Computer Engineering
845 Sherbrooke St W
Montreal, Quebec H3A 0G4

PSPC Contract Number: W7707-145675
Technical Authority: Stephane Blouin, Defence Scientist
Contractor's date of publication: December 2017

Defence Research and Development Canada
Contract Report
DRDC-RDDC-2017-C244
May 2018

Canada

IMPORTANT INFORMATIVE STATEMENTS

This document was reviewed for Controlled Goods by Defence Research and Development Canada (DRDC) using the Schedule to the *Defence Production Act*.

Algorithms for the multi-sensor assignment problem in the δ-generalized labeled multi-Bernoulli filter

Jun Ye Yu, Augustin-Alexandru Saucan, Mark Coates, Michael Rabbat

Electrical and Computer Engineering, McGill University, Montreal, Quebec

Email: jun.y.yu, augustin.saucan@mail.mcgill.ca, and mark.coates, michael.rabbat@mcgill.ca

Abstract

Practical implementations of the multi-sensor δ-generalized labeled multi-Bernoulli filter require solving the multi-sensor assignment problem which is NP-hard. In this paper, we present two different algorithms, the combination method and the cross-entropy method, that find T highly likely target-measurement associations without exhaustive enumeration of all possible multi-sensor assignments. Numerical simulations are conducted to evaluate the aforementioned multi-sensor assignment methods together with the standard sequential processing method and a stochastic optimization algorithm based on Gibbs sampling. The combination method is based on an approximate assignment score function which leads to a lower running time, and it also explores a greater portion of the space of assignments compared to other methods. The cross-entropy method does not rely on the approximation and achieves better tracking performance than the sequential method, albeit at a higher computational overhead. The impact of the approximate score function on the algorithms' performance is also studied via simulations and it is shown that the cross-entropy method consistently yields the best tracking performance whereas the combination method has the shortest runtime at high measurement noise level or high clutter rate.

I. Introduction

Multi-target tracking is an important research topic with applications in numerous domains including air traffic control, computer vision, surveillance, and autonomous vehicles (see [1] for a recent survey on the topic). The objective is to estimate the unknown number of targets and their kinematic states; but non-uniform detection probability, measurement origin uncertainty, false detection and target birth/death are all difficult obstacles to solving the problem.

Random finite set (RFS) filters [2] have emerged as a popular paradigm for solving the multi-target tracking problem in the Bayesian framework. Since the exact multi-target Bayes filter is computationally intractable, the *probability hypothesis density* (PHD) filter [3], *cardinalized PHD* (CPHD) [4] filter and *multi-Bernoulli* (MB) filter [5] have been proposed as tractable approximations. All three algorithms have been successfully applied to many tracking problems [6]–[13]; however they only provide target state estimates at individual time instants as opposed to target tracks over time. To overcome this limitation, Vo et al. [14] introduce the notion of labeled RFSes in which unique labels are appended to a RFS to facilitate generating target

This work was supported by PWGSC contract W7707-145675/001/HAL funded by Defence R&D Canada

tracks. They subsequently developed the δ-*generalized multi-Bernoulli* (δ-GLMB) density which is shown to be conjugate to the standard point measurement likelihood function [14], [15].

While all these filters have been initially developed for single-sensor tracking, multi-sensor extensions have also been proposed in literature [16]–[25]. In the iterator-corrector PHD filter [16], [19], each sensor's measurements are processed sequentially and the output from one sensor is used as input for the next sensor. The performance of this approach may depend strongly on the order in which sensors are processed, especially when sensors have different characteristics (e.g., different detection probability). The recent work by Papi [21] presents the multi-sensor extension of the δ-GLMB filter, but the exact implementation also involves iterating through each sensor to generate all multi-sensor assignments. Liu et al. [22] use an extended association table to generate the most likely associations between targets and measurements from all sensors, but no simulations are provided to validate the algorithm's performance. An alternate approach is to have each sensor run a single-sensor filter to process its own measurements. The posterior multi-target densities are then fused based on general cross-covariance intersection [17], [18], [20]. In both approaches, at most one sensor's measurements are processed at once and therefore only single-sensor filters are required.

Both the multi-sensor CPHD filter [23] and the multi-sensor multi-Bernoulli filter [24] process all sensor measurements simultaneously by using a greedy algorithm to select the most likely target-measurement associations. More recently Vo et al. [25] have proposed the use of Gibbs sampling to find a number of likely multi-sensor assignments in the δ-GLMB filter.

The exact implementation of the multi-sensor δ-GLMB filter requires enumerating all multi-sensor assignments to compute the posterior multi-target density. In practice, we look for a number of likely multi-sensor assignments to construct a truncated density. Although the problem of finding the T best single-sensor assignments can be solved efficiently using Murty's algorithm [26], the multi-sensor counterpart is NP-hard [27]. In this paper, we present two approximation algorithms, the combination and the cross entropy methods, that yield a number of likely multi-sensor assignments without exhaustive enumeration. We compare their performance to the Gibbs method [25] and the standard sequential processing method [21]. The combination method first solves the assignment problem locally at each sensor and then combines the locally optimal solutions to form likely multi-sensor assignments. The cross entropy method constructs a distribution on the space of all multi-sensor assignments with higher probability for more likely assignments. Finally, the Gibbs method uses Gibbs sampling to find likely multi-sensor assignments. Both the combination and Gibbs methods rely on an approximate score function to reduce the computational overhead and we investigate the impact of this approximation. All three algorithms' performance are compared via simulation. The cross-entropy method consistently yields the best tracking performance albeit with higher runtime than the combination and the Gibbs methods.

The rest of the paper is organized as follows. We define the tracking problem and present the relevant background in Section II and Section III. We present the δ-GLMB filter in Section IV and the multi-sensor assignment algorithms in Section V. In Section VI, we compare the algorithms' performance while in Section VII we conclude the paper.

II. Problem formulation

Consider a network of S sensors tracking a time-varying number of targets. The measurements from all sensors are transmitted to a fusion center for processing. Our objective is to estimate the number of targets and their kinematic states over time.

Let $x_1, x_2, ...x_{n(k)} \in \mathbb{X}$ denote the targets' kinematic states at time instant k where $\mathbb{X}$ is the single-target state space and $n(k)$ is the number of targets. At time $k+1$, an existing target x disappears with probability $1 - P_s(x)$ or survives with probability $P_s(x)$ independent of other targets. A surviving target transitions to new state x' with probability $f_{k+1|k}(x'|x)$. A new target with state $x_b \in \mathbb{X}$ can also appear with probability $P_b(x_b)$ independent of all existing targets.

A target x is detected by sensor j with probability $P_{D,j}(x)$ and generates a measurement z with likelihood $g_j(z|x)$. The measurements from different sensors are assumed to be conditionally independent given the target state. The sensor receives at most one measurement from each target. Each sensor may also receive clutter measurements which are modeled as a Poisson process with intensity function $\kappa(\cdot)$.

III. The δ-GLMB RFS

Consider the multi-target state $X = \{x_1, x_2, ..., x_n\}$. Define $\mathcal{F}(\mathbb{X})$ as the space of all finite subsets of $\mathbb{X}$. An RFS is a random variable that takes values in $\mathcal{F}(\mathbb{X})$. In other words, an RFS is a finite set with random cardinality and elements, and thus it conveniently captures the two unknown quantities of interest in multi-target tracking problems, the number of targets and their states. Note that an RFS does not follow the standard Euclidean notions of integration; instead we must apply the theory of Random Finite Set Statistics [2].

In addition to inferring the number of targets and their states, multi-target tracking involves the inference of target trajectories or tracks across time. To allow target tracking in the RFS framework, it is necessary to associate estimated target states from different time steps. One solution is to augment each element in an RFS with a unique label, leading to a *labeled RFS* [15]. States at different time steps with the same label thus correspond to the same target. By convention, a label is a 2-element vector where the first entry is the time instant when the element is created and the second entry is an index to distinguish elements born at the same time instant. Each element in a labeled RFS thus has state space $\mathbb{X} \times \mathbb{L}$ where $\mathbb{L}$ is the label space. Furthermore, a labeled RFS is said to have distinct labels if and only if the number of elements is equal to the number of unique labels.

For the rest of the paper, unlabeled single-object states are denoted by lower case letters (e.g., x) and multi-object states (realizations of an RFS) by upper case letters (e.g., $X = \{x_1, ...x_n\}$). Their labeled counterparts are bold letters (e.g., $\mathbf{x}=(x, l)$ and $\mathbf{X}=\{\mathbf{x}_1, ...\mathbf{x}_n\}$). The blackboard bold letters (e.g., $\mathbb{X}$ and $\mathbb{L}$) denote the corresponding state space.

The projection function $\mathcal{L}(\cdot)$ returns the labels of a labeled RFS and is defined as

$$\mathcal{L}(\mathbf{x}) = \mathcal{L}(x, l) = l \tag{1}$$

$$\mathcal{L}(\mathbf{X}) = \{\mathcal{L}(\mathbf{x}) : \mathbf{x} \in \mathbf{X}\}. \tag{2}$$

The distinct label indicator function $\Delta(\cdot)$ is defined as

$$\Delta(\mathbf{X}) = \begin{cases} 1 & |\mathbf{X}| = |\mathcal{L}(\mathbf{X})| \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

where $|\mathbf{X}|$ denotes the cardinality of $\mathbf{X}$.

For a real-valued function $h(x)$ and a set $\mathbf{X}$, the multi-object exponential is defined as

$$[h]^{\mathbf{X}} = \prod_{x \in \mathbf{X}} h(x), \tag{4}$$

with $h^{\emptyset} = 1$ by convention.

For real-valued functions $p(x)$ and $h(x)$, their inner product is defined as

$$\langle p, h \rangle = \int p(x)h(x)dx. \tag{5}$$

A generalization of the indicator function is introduced as

$$1_Y(X) = \begin{cases} 1, & X \subseteq Y \\ 0, & \text{otherwise,} \end{cases} \tag{6}$$

where we will also employ $1_Y(x)$ in place of $1_Y(\{x\})$. Finally, we adopt a generalized Kronecker delta function defined as

$$\delta_Y(X) = \begin{cases} 1, & X = Y \\ 0, & \text{otherwise,} \end{cases} \tag{7}$$

where X and Y can be scalars, vectors, or unordered sets.

The δ-GLMB RFS is a labeled RFS on $\mathbb{X} \times \mathbb{L}$ with the following distribution [14]:

$$\pi(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I,\epsilon) \in \mathcal{F}(\mathbb{L}) \times \Xi} w^{(I,\epsilon)} \delta_I(\mathcal{L}(\mathbf{X}))[p^{(\epsilon)}]^{\mathbf{X}}, \tag{8}$$

where Ξ is a discrete space, and the weights $w^{(I,\epsilon)}$ are normalized such that $\sum_{(I,\epsilon) \in \mathcal{F}(\mathbb{L}) \times \Xi} w^{(I,\epsilon)} = 1$.

The δ-GLMB density (Eq. (8)) can be interpreted as a mixture of exponential terms. Each term consists of the weight $w^{(I,\epsilon)}$ and the multi-target exponential function $\left[p^{(\epsilon)}\right]^{\mathbf{X}}$. Each I represents a set of track labels, and ϵ represents a history of sensor measurements associated with these labels. Each pair (I, ϵ) thus represents the hypothesis that the track labels I have the measurement-association history ϵ and hypothesis weight $w^{(I,\epsilon)}$. The term $\Delta(\mathbf{X})$ ensures that all elements in the RFS $\mathbf{X}$ have unique labels. Finally, $p^{(\epsilon)}(\cdot, l)$ is the single-target density of track label $l \in I$ given the association history ϵ.

The δ-GLMB density has the attractive property of being a conjugate prior for the standard point-measurement likelihood model and for the Chapman-Kolmogorov equation [14]. In other words, if the prior density is a δ-GLMB density, after time-prediction and the Bayes update, the posterior density is also a δ-GLMB density.

IV. The multi-sensor δ-GLMB filter

The δ-GLMB filter is a Bayesian filter which models the multi-object target states as δ-GLMB RFSes and recursively computes the posterior multi-target density via prediction and update. Filter implementation requires the formation of hypotheses (I, ϵ) for which two frameworks exist. The separate predict and update δ-GLMB filter [15] creates hypotheses at both the predict and update stages while the joint δ-GLMB filter [25] creates posterior hypotheses directly from the previous time posterior by jointly incorporating target death, birth and measurement update without explicitly constructing the predicted hypotheses. Both δ-GLMB filter implementations are detailed in the following.

A. Separate predict-update framework

Under the separate predict-update framework, the δ-GLMB filter is a two-step algorithm. Let $\pi_{k|k}(\mathbf{X})$ denote the multi-target density at time instant k.

Consider first the predict step at time $k+1$. New targets can appear in the tracking area. We model these targets with the labeled RFS density

$$f_{B,k+1}(\mathbf{Y}) = \Delta(\mathbf{Y}) w_{B,k+1}(\mathcal{L}(\mathbf{Y}))[p_{B,k+1}]^{\mathbf{Y}}, \tag{9}$$

two particular examples of which are the labeled Poisson and labeled multi-Bernoulli RFSes [14]. For a labeled multi-Bernoulli birth model with existence probability $r_{B,k+1}(l)$ and single-target density $p_{B,k+1}(x,l)$, we have

$$w_{B,k+1}(\mathcal{L}(\mathbf{Y})) = \prod_{i \in \mathbb{L}_{k+1}} [1 - r_{B,k+1}(i)] \prod_{l \in \mathcal{L}(\mathbf{Y})} \frac{1_{\mathbb{L}_{k+1}}(l) r_{B,k+1}(l)}{1 - r_{B,k+1}(l)}, \tag{10}$$

where $\mathbb{L}_{k+1}$ denotes the label space of targets born at time step $k+1$. We also define $\mathbb{L}_{0:k}$ as the label space of targets that exist from time step 0 to k. Note that $\mathbb{L}_{0:k} \cap \mathbb{L}_{k+1} = \emptyset$ and $\mathbb{L}_{0:k+1} = \mathbb{L}_{0:k} \cup \mathbb{L}_{k+1}$.

If the posterior multi-object distribution at time k is a δ-GLMB density of the form (8), then the predicted multi-object distribution is also a δ-GLMB of the following form [14]:

$$\pi_{k+1|k}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{(I_{k+1},\epsilon) \in \mathcal{F}(\mathbb{L}_{0:k+1}) \times \Xi} w_{k+1|k}^{(I_{k+1},\epsilon)} \delta_{I_{k+1}}(\mathcal{L}(\mathbf{X}))[p_{k+1|k}^{(\epsilon)}]^{\mathbf{X}} \tag{11}$$

$$w_{k+1|k}^{(I_{k+1},\epsilon)} = w_{B,k+1}(I_{k+1} \cap \mathbb{L}_{k+1}) w_S^{(\epsilon)}(I_{k+1} \cap \mathbb{L}_{0:k}) \tag{12}$$

$$p_{k+1|k}^{(\epsilon)}(x,l) = 1_{\mathbb{L}_{0:k}}(l) p_s^{(\epsilon)}(x,l) + 1_{\mathbb{L}_{k+1}}(l) p_{B,k+1}(x,l) \tag{13}$$

$$p_s^{(\epsilon)}(x',l) = \frac{\langle P_s(\cdot,l) f_{k+1|k}(x'|\cdot,l), p_k^{(\epsilon)}(\cdot,l) \rangle}{\eta_s^{(\epsilon)}(l)} \tag{14}$$

$$\eta_s^{(\epsilon)}(l) = \langle P_s(\cdot,l), p_k^{(\epsilon)}(\cdot,l) \rangle \tag{15}$$

$$w_S^{(\epsilon)}(L) = [\eta_s^{(\epsilon)}]^L \sum_{J \supseteq L} [1 - \eta_s^{(\epsilon)}]^{J-L} w_{k|k}^{(J,\epsilon)}. \tag{16}$$

The predicted multi-object density (11) is a weighted mixture of multi-target exponentials. Each mixture component has weight $w_{k+1|k}^{(I_{k+1},\epsilon)}$. The label set I_{k+1} contains both surviving labels from previous time steps and new birth labels. The mixture weight is a product of $w_{B,k+1}$ for the birth labels and w_S for surviving labels. Each target in the mixture component has

either density $p_s^{(\epsilon)}(\cdot)$ for surviving target or $p_{B,k+1}(\cdot)$ for newly born target. For a surviving target, the single-target density is a function of the survival probability $P_s(\cdot)$ and the dynamic model $f_{k+1|k}(\cdot)$.

Exact computation of the predicted δ-GLMB distribution requires exhaustive enumeration of all possible combinations of surviving and new labels (i.e., all elements in $\mathcal{F}(\mathbb{L}_{0:k+1})$), which is computationally intractable in general. In practice, the K-shortest-paths algorithm [28] is used to find the K label sets with the highest mixture weight $w_{k+1|k}^{(I_{k+1},\epsilon)}$ [14].

Consider next the update step at time $k+1$. Let $Z_{k+1,i} = \{z_{k+1,i}^1, ...z_{k+1,i}^{m_{k+1,i}}\}$ denote the $m_{k+1,i}$ measurements from the i-th sensor for time step $k+1$. An association map θ_i is a function that maps a label l to a measurement index, $\theta_i : \mathbb{L}_{0:k+1} \to \{0, 1, \ldots, m_{k+1,i}\}$, with $\theta_i(l) = 0$ for missed detection. An association map is valid if each label is associated with at most one measurement ($\theta_i(l) = j \in \{1, 2, ...m_{k+1,i}\}$) or miss detected and, conversely, a measurement is associated with at most one label. We denote by Θ_i the space of all valid maps θ_i for sensor i. We denote by $\theta_{1:S} = (\theta_1, \ldots, \theta_S)$ the multi-sensor map for all S sensors and $\Theta_{1:S} = \Theta_1 \times \cdots \times \Theta_S$. Hence, for a valid multi-sensor map we have $\theta_{1:S} \in \Theta_{1:S}$. Additionally, we denote by $\Theta_i(I) \subseteq \Theta_i$ and $\Theta_{1:S}(I) \subseteq \Theta_{1:S}$ the space of valid association maps for label set I. The collection of measurement sets at time $k+1$ is denoted by $Z_{k+1} = (Z_{k+1,1}, \ldots, Z_{k+1,S})$. In the following, to simplify the notation we omit the time index $k+1$ for measurements and measurement-related parameters when there is no ambiguity.

If the prior distribution is a δ-GLMB of the form (11), then the multi-sensor posterior distribution is also a δ-GLMB density given by

$$\pi_{k+1|k+1}(\mathbf{X}|Z) = \Delta(\mathbf{X}) \sum_{I_{k+1},\epsilon} \sum_{\theta_{1:s} \in \Theta_{1:S}(I_{k+1})} w_{k+1|k+1}^{(I_{k+1},\epsilon,\theta_{1:S})}(Z) \delta_{I_{k+1}}(\mathcal{L}(\mathbf{X})) \left[p_{k+1|k+1}^{(\epsilon,\theta_{1:s})}(\cdot|Z)\right]^{\mathbf{X}} \tag{17}$$

$$w_{k+1|k+1}^{(I_{k+1},\epsilon,\theta_{1:s})}(Z) \propto w_{k+1|k}^{(I_{k+1},\epsilon)} [\eta_Z^{(\epsilon,\theta_{1:S})}]^{I_{k+1}} \tag{18}$$

$$p_{k+1|k+1}^{(\epsilon,\theta_{1:S})}(x, l|Z) = \frac{p_{k+1|k}^{(\epsilon)}(x, l) \prod_{j=1}^{S} \psi_j(x, l; \theta_j)}{\eta_Z^{(\epsilon,\theta_{1:S})}(l)} \tag{19}$$

$$\eta_Z^{(\epsilon,\theta_{1:S})}(l) = \langle p_{k+1|k}^{(\epsilon)}(\cdot, l), \prod_{j=1}^{S} \psi_j(\cdot, l; \theta_j) \rangle \tag{20}$$

$$\psi_j(x, l; \theta_j) = \begin{cases} 1 - P_{D,j}(x) & \theta_j(l) = 0 \\ \frac{P_{D,j}(x) g_j(z_j^i|x,l)}{\kappa_j(z_j^i)} & \theta_j(l) \in \{1, ..., m_j\} \end{cases} \tag{21}$$

Considering (17), the posterior multi-target density is a weighted mixture of multi-target exponentials. For each predicted hypothesis with label set I_{k+1} and association history ϵ, we generate multi-sensor association maps $\theta_{1:S}$. Each tuple $(I_{k+1}, \epsilon, \theta_{1:S})$ thus forms a new posterior hypothesis with mixture weight $w^{(I_{k+1},\epsilon,\theta_{1:S})}$.

The exact multi-sensor δ-GLMB update (17) requires generating all valid multi-sensor association maps for each predicted hypothesis which is infeasible in all but the simplest scenarios. Practical solutions involve exploring only a limited number of valid associations for each predicted hypothesis, leading to a truncated δ-GLMB posterior. For a given hypothesis (I_{k+1}, ϵ), we select T valid association maps that have high-weight posterior hypotheses according to (18). If we can find the T highest-weight posterior hypotheses, the resulting truncated posterior δ-GLMB density would have the smallest L_1-distance from the exact posterior compared to all other truncated densities containing T updated hypotheses [15, Sec. IV-C].

Note that the exact sequential multi-sensor δ-GLMB filter [21] enumerates multi-assignment maps by recursively applying the single-sensor δ-GLMB update step for each sensor. If no component is pruned after each sensor update, the enumeration is exhaustive and the resulting posterior density is exact. In practical implementations however, truncation is necessary [21].

B. Joint predict-update framework

The original implementation of the δ-GLMB filter in [14] involves the generation of hypotheses at both the predict and update stages. Considering at time k the posterior δ-GLMB density (8) formed with hypothesis (I_k, ϵ), a number of predicted hypotheses (I_{k+1}, ϵ) are formed in order to obtain the predicted density of (11). Subsequently, posterior hypotheses $(I_{k+1}, \epsilon, \theta_{1:S})$ are generated in order to obtain the δ-GLMB posterior at $k+1$ in (17). The number of hypotheses grows exponentially with time, and in practice both the predicted and updated δ-GLMB densities are truncated in order to retain only the highest-weight hypotheses first at predict and subsequently at the update stages.

The formulation of predicted hypotheses does not take into account the current measurement set Z_{k+1} and hence could potentially create a high number of hypotheses that do not correlate with the current measurements and subsequently lead to low-weight updated hypothesis. In order to avoid the selection of unlikely predicted hypotheses, in [29] and [25] it is proposed to combine the predicted and updated δ-GLMB densities and perform a single operation of hypothesis selection that incorporates the current measurement set into the process of determining target death/birth. The resulting δ-GLMB framework is referred to as the joint predict-update δ-GLMB filter. In order to incorporate target death into the mapping process, we introduce the S-tuple $-1_{1:S} \triangleq (-1, \ldots, -1)$, the space of extended mappings $\bar{\Theta}_{1:S} \triangleq \{\{-1_{1:S}\} \uplus \Theta_{1:S}\}$ ($\uplus$ is the disjoint union operator) and the extended mappings $\bar{\theta}_{1:S} : I \to \bar{\Theta}_{1:S}(I)$. Note the added assignment of $-1_{1:S}$ signaling target death. Similarly with $\theta_{1:S}$, valid multi-assignments $\bar{\theta}_{1:S}$ assign at most one measurement for each target, do not assign the same measurement to two distinct targets, and assignments containing $\bar{\theta}_i(l) = -1$ and $\bar{\theta}_j(l) \neq -1$ are not allowed for $i \neq j$ (i.e., all sensors must agree on the death/survival/birth of a target). By using the extended mappings and combining the predict (11) and update (17) equations, we obtain Eq. (22).

$$\pi_{k+1|k+1}(\mathbf{X}) = \Delta(\mathbf{X}) \sum_{I_k, \epsilon} \sum_{\bar{\theta}_{1:S} \in \bar{\Theta}_{1:S}(I_k \cup \mathbb{L}_{k+1})} \delta_{I_k \cup \mathbb{L}_{k+1} \setminus \{l : \bar{\theta}_{1:S}(l) = -1_{1:S}\}}(\mathcal{L}(\mathbf{X})) w_{k|k}^{(I_k, \epsilon)} \times \left[\bar{\eta}_Z^{(\epsilon, \bar{\theta}_{1:S})}(\cdot)\right]^{I_k \cup \mathbb{L}_{k+1}} \left[p_{k+1|k+1}^{(\epsilon, \bar{\theta}_{1:s})}(\cdot|Z)\right]^{\mathbf{X}}, \tag{22}$$

$$p_{k+1|k+1}^{(\epsilon, \bar{\theta}_{1:s})}(x, l|Z) = p_{k+1|k+1}^{(\epsilon, \theta_{1:s})}(x, l|Z) \text{ if } \bar{\theta}_{1:s}(l) \neq -1_{1:S} \tag{23}$$

$$\bar{\eta}_Z^{(\epsilon, \bar{\theta}_{1:S})}(l) = \begin{cases} 1 - \eta_s^{(\epsilon)}(l) & \text{if } l \in \mathbb{L}_{0:k}, \text{ and } \bar{\theta}_{1:S}(l) = -1_{1:S}, \\ \eta_s^{(\epsilon)}(l) \eta_Z^{(\epsilon, \theta_{1:S})}(l), & \text{if } l \in \mathbb{L}_{0:k}, \text{ and } \bar{\theta}_{1:S}(l) \neq -1_{1:S}, \\ 1 - r_{B,k+1}(l) & \text{if } l \in \mathbb{L}_{k+1}, \text{ and } \bar{\theta}_{1:S}(l) = -1_{1:S}, \\ r_{B,k+1}(l) \eta_Z^{(\epsilon, \theta_{1:S})}(l), & \text{if } l \in \mathbb{L}_{k+1}, \text{ and } \bar{\theta}_{1:S}(l) \neq -1_{1:S}. \end{cases} \tag{24}$$

In (22), starting with (I_k, ϵ), new hypotheses are formed by appending to I_k the birth label set $\mathbb{L}_{k+1}$ and considering all extended mappings on $\bar{\theta}_{1:S} \in \bar{\Theta}_{1:S}(I_k \cup \mathbb{L}_{k+1})$. Note that the mappings $\bar{\theta}_{1:S}$ take into account target death for existing tracks

I_k and non-birthed tracks from $\mathbb{L}_{k+1}$. Additionally, in (22) the term $\delta_{I_k \cup \mathbb{L}_{k+1} \setminus \{l:\bar{\theta}_{1:S}(l)=-1_{1:S}\}}(\mathcal{L}(\mathbf{X}))$ ensures that $\mathbf{X}$ can only contain targets surviving from the previous time I_k and birthed targets from the set $\mathbb{L}_{k+1}$. The weight of (24) accounts for target survival (via the probability of survival $\eta_s^{(\epsilon)}(\cdot)$), target birth (via the probability of target existence of birthed targets $r_{B,k+1}(\cdot)$) and also the correlation between the targets and their assigned measurements (via the modified likelihood $\eta_Z^{\epsilon,\bar{\theta}_{1:s}}$). Regarding the expression (22), a posterior hypothesis can be identified by $(I_k, \epsilon, \bar{\theta}_{1:S})$ or equivalently by $(I_k, \epsilon, I_{k+1}, \theta_{1:S})$, where we explicitly introduced the posterior label set $I_{k+1} = I_k \cup \mathcal{L}_{k+1} \setminus \{l : \bar{\theta}_{1:S}(l) = -1_{1:S}\}$. The corresponding weight of the posterior hypothesis is $w_{k|k}^{(I_k,\epsilon)} \left[\bar{\eta}_Z^{(\epsilon,\bar{\theta}_{1:S})}(\cdot)\right]^{I_k \cup \mathbb{L}_{k+1}}$.

The exact posterior update in Eq. (22) requires exhaustive enumeration of all possible extended association mappings. In practice, the T best extended association mappings are generated for each hypothesis (I_k, ϵ).

V. Implementation of the multi-sensor δ-GLMB filter

An efficient implementation of the δ-GLMB filter requires generating a number of (extended) multi-sensor association mappings that lead to high-weight posterior hypotheses. In the following section, we present three algorithms that generate T likely multi-sensor assignments without exhaustive enumeration in both the separate and joint frameworks.

A. Separate δ-GLMB filter

In the separate predict-update framework, given a valid multi-sensor association map $\theta_{1:S}$ for the predicted hypothesis (I, ϵ) with weight $w_{k+1|k}^{(I,\epsilon)}$, the posterior hypothesis weight is proportional to $w_{k+1|k}^{(I,\epsilon)} \prod_{l \in I} \eta_Z^{(\epsilon,\theta_{1:S})}(l)$. Therefore, maximizing the posterior hypothesis weight for a given predicted hypothesis is equivalent to solving

$$\begin{aligned} \underset{\theta_{1:S}}{\text{maximize}} \quad & C_{\text{exact}}(\epsilon, \theta_{1:S}) = \sum_{l \in I} \log\left(\eta_Z^{(\epsilon,\theta_{1:S})}(l)\right), \\ \text{subject to} \quad & \theta_j \text{ valid association map } \forall\, j = 1, \ldots, S. \end{aligned} \tag{28}$$

Equation (28) is known as the multi-dimensional assignment problem and is NP-hard for $S > 1$ [27]. The term $\log\left(\eta_Z^{(\epsilon,\theta_{1:S})}(l)\right)$ can be interpreted as the score of assigning measurements $\theta_1(l), ...\theta_S(l)$ to label l. The objective is to maximize the total score of $|I|$ assignments without any conflict.

1) Combination method: In the following, we present the combination method that produces T high-scoring multi-sensor associations for the label set I without exhaustive enumeration.

Consider the following optimization problem,

$$\begin{aligned} \underset{\theta_1 \cdots \theta_S}{\text{maximize}} \quad & C_{\text{approx}}(I, \epsilon, \theta_{1:S}) = \sum_{l \in I} \sum_{j=1}^{S} \log\left(\eta_Z^{(\epsilon,\theta_j)}(l)\right) \\ \text{subject to} \quad & \theta_j \text{ valid association map } \forall\, j = 1, \ldots, S. \end{aligned} \tag{29}$$

Eq. (29) differs from Eq. (28) in the score function. The term $\log\left(\eta_Z^{(\epsilon,\theta_{1:S})}(l)\right)$ is approximated by the sum (or log-product) of single sensor score functions $\log\left(\eta_Z^{(\epsilon,\theta_j)}(l)\right)$. Ideally, solving Eq. (29) would yield the same multi-sensor maps as solving

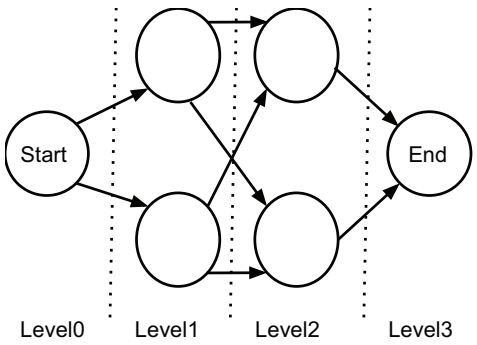

Fig. 1. Graph for combining single-sensor maps from two sensors. Each vertex represents one single-sensor map. All maps from the same sensor form one level. Traversing from Start to End generates a multi-sensor map.

Eq. (28); however this does not hold true in general. We defer discussing the impact of the approximate score function on the algorithm's performance until section VI-B.

Solving (29) is equivalent to solving S independent single-sensor assignment problems using only the sensor's local measurements. Each sensor first generates T best single-sensor association maps using Murty's algorithm. Then these maps are combined by constructing a weighted directed acyclic graph where each single-sensor association map is a vertex in the graph. We also include two dummy vertices, Start and End. All vertices are further divided into different levels. The Start vertex is at level 0. All maps of sensor j are at level j. The End vertex is at level $S+1$. Every vertex at level j has an edge to every vertex at level $j+1$. The weight of each edge is equal to the value of the head vertex (i.e., score of the corresponding single-sensor map). The End vertex has value of 0 by definition. See Fig. 1 for an illustration with two sensors (with two maps at each level).

A path from Start to End traverses through every level exactly once and contains one vertex from each level. Such a path would constitute a valid multi-sensor assignment map. Therefore, our objective is to find the highest-score paths from Start to End. We say that a path reaches level j if it reaches any vertex on that level.

Let $c_{i:j}$ denote the score of a path traveling from level i to level j. Every path from Start to End contains exactly $S+1$ hops and the total score can be decomposed into a sum of $S+1$ terms

$$c_{0:S+1} = c_{0:1} + c_{1:S+1} = c_{0:1} + c_{1:2} + ...c_{S:S+1}. \tag{30}$$

Our algorithm iterates through each level and greedily constructs the highest-score paths to reach that level. Consider the T highest-score paths from level 0 to level j with corresponding scores $\{c^1_{0:j} \geq c^2_{0:j} ... \geq c^T_{0:j}\}$. We also know the T highest-scores paths to travel from level j to $j+1$: $\{c^1_{j:j+1} \geq c^2_{j:j+1} ... \geq c^T_{j:j+1}\}$. We can then simply generate T^2 new paths with scores $\{c^u_{0:j} + c^v_{j:j+1} | 1 \leq u, v \leq T\}$ and select the T best ones. This gives us the best paths to travel from level 0 to level $j+1$. To reduce computational overhead, the T (rather than T^2) paths with highest scores are then extended at the next level. We repeat the same steps until we reach level $S+1$ (the End vertex). As the starting point, the best maps of sensor 1 correspond to the best paths from Start to level 1. Algorithm 1 shows the pseudo-code for the combination method. Lines 2-4 account for single-sensor map generation. Lines 6-10 account for the combination of single-sensor maps.

The combination approach yields the same outputs as the approach by Liu et al [22] which introduces the notion of extended

Algorithm 1 Combination algorithm for the multi-assignment problem

function COMBINED_MS_ASSIGNMENT(T)
  for $s \leftarrow 1$ to S **do**
    Generate T single-sensor maps: $\theta_s^{(1)}, ...\theta_s^{(T)}$=Murty($Z_s$, T)
  end for
  Initialize $\theta_{1:1}^{(t)} = \theta_1^{(t)} \quad t = 1...T$
  for $s \leftarrow 2$ to S **do**
    Generate T^2 new maps: $\theta_{1:s}^{((i-1)T+j)} = [\theta_{1:s-1}^{(i)}, \theta_s^{(j)}] \quad i = 1...T, j = 1...T$
    Compute map score: $c_{1:s}^{((i-1)T+j)} = c_{1:s-1}^{(i)} + c_s^{(j)} \quad i = 1...T, j = 1...T$
    Sort maps in decreasing score: $c_{1:s}^{(1)} \geq ... \geq c_{1:s}^{(T^2)}$
    Propagate T best maps: $\theta_{1:s}^{(1)}, ...\theta_{1:s}^{(T)}$
  end for
  return $\theta_{1:S}^{(1)}, \ldots, \theta_{1:S}^{(T)}$
end function

association maps. In the single-sensor δ-GLMB filter, given label set I and measurement set Z, an $|I| \times |Z|$ association cost matrix is constructed for Murty's algorithm. In [22], the authors construct a block-diagonal extended association cost matrix by concatenating individual sensors' cost matrices diagonally. This large matrix is then used to yield T high-scoring multi-sensor assignments directly. To contrast the two methods, we seek to solve S small assignment problems and combine the results whereas [22] solves one large assignment problem with S times more elements.

The complexity of the combination method is $\mathcal{O}(ST\tilde{m}^3 + ST^2)$ where the first term accounts for running Murty's algorithm ($\tilde{m}$ being the average number of measurements per sensor) and the second term accounts for the combination of single-sensor maps. For comparison, the complexity of the method in [22] is $\mathcal{O}(TS^3\tilde{m}^3)$. For sufficiently large number of sensors and measurements per sensor ($S^2\tilde{m}^3 > T$), our method is expected to have lower complexity.

B. Joint δ-GLMB filter

In this section, we present two stochastic algorithms that provide high-weight posterior hypotheses for the joint δ-GLMB filter of Sec. IV-B. In order to provide a low L_1 error approximation to (22), one needs to retain only hypotheses $(I_k, \epsilon, \bar{\theta}_{1:S})$ with high weights $w_{k|k}^{(I_k,\epsilon)} \left[\bar{\eta}_Z^{(\epsilon,\bar{\theta}_{1:S})}(\cdot)\right]^{I_k \cup \mathbb{L}_{k+1}}$.

In a stochastic optimization framework, the objective is to sample multi-sensor assignments according to a probability distribution that leads to high-weight hypotheses. As shown in [25], this can be achieved in two steps. First, we sample tuples (I_k, ϵ) with weights $w_{k|k}^{(I_k,\epsilon)}$, i.e., according to the previous time-step hypotheses weights. Second, for each sampled (I_k, ϵ) we form the set $I = I_k \cup \mathbb{L}_{k+1}$ and sample for extended mappings $\bar{\theta}_{1:S}(I)$ with high weights $\left[\bar{\eta}_Z^{(\epsilon,\bar{\theta}_{1:S})}(\cdot)\right]^{I_k \cup \mathbb{L}_{k+1}}$. Once a mapping $\bar{\theta}_{1:S}(I)$ is selected, we can form the posterior label set I_{k+1} as $I_{k+1} = I_k \cup \mathbb{L}_{k+1} \setminus \{l : \bar{\theta}_{1:S}(l) = -1_{1:S}\}$.

In Sec. V-B1, we review the Gibbs sampling algorithm of [25] for sampling multi-assignments. The main drawback of the Gibbs sampling method is the high computational cost when employing the exact form of the weights $\bar{\eta}_Z^{(\epsilon,\bar{\theta}_{1:S})}(\cdot)$ of (24). A drastic reduction in complexity can be achieved by employing an approximation for $\bar{\eta}_Z^{(\epsilon,\bar{\theta}_{1:S})}(\cdot)$. In Sec. V-B2, we propose a cross-entropy method that employs the exact weights of (24) but with a computational complexity of the same order as the Gibbs method with approximate weights.

1) Gibbs sampling: To facilitate the presentation of the Gibbs sampler for multi-sensor assignments, we resort to a matrix representation of multi-assignments. For consistency with [25], we employ the following matrix notation for the multi-sensor assignments of the label set $I = \{l^1, \dots, l^n\}$:

$$\gamma \triangleq \begin{bmatrix} \bar{\theta}_1(l^1) & \bar{\theta}_2(l^1) & \dots & \bar{\theta}_S(l^1) \\ \bar{\theta}_1(l^2) & \bar{\theta}_2(l^2) & \dots & \bar{\theta}_S(l^2) \\ \vdots & \vdots & \ddots & \vdots \\ \bar{\theta}_1(l^n) & \bar{\theta}_2(l^n) & \dots & \bar{\theta}_S(l^n) \end{bmatrix} \in \Omega^n, \tag{31}$$

where $\Omega \triangleq (-1_{1:S} \uplus \{0, \dots, m_1\} \times \cdots \times \{0, \dots, m_S\})$ and Ω^n is the space of multi-sensor assignment matrices for label set I. We denote by γ_i the i-th row of γ and by $\gamma_{i,j}$ the i, j element of γ.

Note the relationship between γ and the extended multi-assignment mapping $\bar{\theta}_{1:S}$: $\gamma_{i,s} = \bar{\theta}_s(l^i)$. If $\gamma_{i,s} = -1$ for some sensor s, then $\gamma_{i,s} = -1$ for all sensors $s = \{1, \dots, S\}$. Additionally, valid multi-assignment mappings $\bar{\theta}_{1:S}$ lead to valid multi-assignment matrices γ. We denote by $\Gamma \subset \Omega^n$ the space of all valid matrices γ.

By a slight abuse of notation, we employ the same notation of (24) for the weights of multi-assignment vectors γ_i

$$\bar{\eta}_i(\gamma_i) \triangleq \bar{\eta}_Z^{(\epsilon, \bar{\theta}_{1:S})}(l^i) \quad \text{with } \bar{\theta}_j(l^i) = \gamma_{ij}, \forall j = 1, \dots, S. \tag{32}$$

As shown in [25], sampling multi-assignment matrices γ becomes equivalent to sampling from

$$\pi(\gamma) \propto 1_\Gamma(\gamma) \prod_{i=1}^{n} \bar{\eta}_i(\gamma_i). \tag{33}$$

Sampling matrices γ from the distribution (33) is made difficult due to the function $1_\Gamma(\gamma)$ that ensures that all multi-sensor assignments are valid, i.e., no two targets are assigned the same measurement and that no target is assigned to two different measurements. However, it is relatively easy to sample the assignment for a target, i.e., to sample γ_i, when the assignments for targets $I \setminus l^i$ are known. Therefore sampling from the target distribution of (33) can be achieved via Gibbs sampling

$$\pi(\gamma'|\gamma) = \prod_{i=1}^{n} \pi_i(\gamma_i'|\gamma_{1:i-1}', \gamma_{i+1:n}), \tag{34}$$

where γ' and γ are the current and previous samples. As shown in [25], the conditionals have the following form

$$\begin{aligned} &\pi_i(\gamma_i|\gamma_{1:i-1}, \gamma_{i+1:n}) \propto \\ &\quad \bar{\eta}_i(\gamma_i) \prod_{s=1}^{S} \prod_{\substack{j=1 \\ j \neq i}}^{n} \left[1 - 1_{\{1,\dots,m_s\}}(\gamma_{i,s}) \delta_{\gamma_{i,s}}(\gamma_{j,s})\right]. \end{aligned} \tag{35}$$

Sampling assignments for target l^i from (35) amounts to sampling from a categorical distribution (with $1 + \prod_{s=1}^{S}(1 + m_s)$ categories), where the conditioning on the remaining targets $I \setminus l^i$ acts to prohibit the assignments (categories) that contain measurements already assigned to a target from $I \setminus l^i$. Indeed, the probability of the multi-assignment γ_i is set to 0 whenever targets l^i and $l^j \in I \setminus l^i$ share at least one assignment (i.e., $\delta_{\gamma_{i,s}}(\gamma_{j,s}) = 1$) unless it is a miss-detection (i.e., $1_{\{1,\dots,m_s\}}(\gamma_{i,s}) = 0$). Categorical sampling has linear complexity with the number of categories [30] and hence sampling from (35) has a complexity

of $\mathcal{O}(\prod_{s=1}^{S} m_s)$. Sampling a number of T multi-assignments for the label set I and according to [25, Algorithm 1] has a computational complexity of $\mathcal{O}(Tn^2 \prod_{s=1}^{S} m_s)$. Note that the T multi-sensor assignment maps are not guaranteed to be distinct. Additionally, in this algorithm the weights $\bar{\eta}_i(\gamma_i)$ are assumed known and available for all $l^i \in I$ and all possible multi-assignments $\gamma_i \in \Omega$. Computing the multi-sensor assignment weights $\bar{\eta}_i(\gamma_i)$ for the n targets has a computational complexity of $\mathcal{O}(n \prod_{s=1}^{S} m_s)$. Note the exponential complexity with respect to S of both [25, Algorithm 1] and the associated weight pre-computation. However, the weight pre-computation has a larger complexity constant than the Gibbs sampling algorithm since the computation of a single weight $\bar{\eta}_i(\gamma_i)$ involves the evaluation of the integral (20).

A significant reduction of complexity can be achieved by approximating the multi-sensor assignment weights as the product of single-sensor assignment weights. More precisely, for $\gamma_i \neq -1_{1:S}$ it is assumed in [25, Eqs. (59-67)] that the multi-sensor weights $\bar{\eta}_i(\gamma_i)$ are approximated as

$$\bar{\eta}_i(\gamma_i) \approx \prod_{s=1}^{S} \bar{\eta}_{i,s}(\gamma_{i,s}) \triangleq \prod_{s=1}^{S} \langle p_{k+1|k}^{(\epsilon)}(\cdot, l^i), \psi_s(\cdot, l^i; \gamma_{i,s}) \rangle. \tag{36}$$

By using (36), sampling from (35) can be achieved in two steps. First one samples for target death/survival, and then in the case of target survival, one samples measurement assignments independently for each sensor s [25, Eqs. (59-67)]. The resulting Gibbs algorithm achieves a computational complexity of $\mathcal{O}(Tn^2 \sum_{s=1}^{S} m_s)$ for sampling T maps. The required pre-computations incur an additional computational complexity of $\mathcal{O}(n \sum_{s=1}^{S} m_s)$. Note in this case the linear complexity with the number of sensors for both Gibbs sampling and the necessary pre-computations.

2) Cross-entropy for multi-sensor assignment selection: The Cross Entropy (CE) method was originally designed as an adaptive importance sampling scheme for the estimation of rare-event probabilities [31]. In [32], CE was extended for solving both continuous and combinatorial optimization problems. The main drawback of the Gibbs sampling method of Sec. V-B1 is the high computational load when using the exact multi-sensor assignment weights. Indeed, the Gibbs method requires the evaluation of the exact multi-sensor assignment weights of (24) for all targets and all possible multi-assignments. In contrast the CE method, as it will be shown further on, only requires the evaluation of the weights (24) for the sampled multi-sensor assignments. Since the number of distinct assignments sampled by CE is in general much smaller than the total number of possible multi-sensor assignments (i.e., $|\Omega^n|$), the CE method can attain a much smaller computational overhead than the Gibbs method when using (24).

CE methods rely on the construction of a random sequence of solutions which converge probabilistically to the optimal solution. This procedure involves iterating the following two steps. First, given a sampling mechanism we generate samples that represent potential solutions to the optimization problem. Second, the drawn samples are used to update the parameters of the sampling mechanism (typically pdf parameters) in order to obtain better solutions at the next iteration with respect to the optimization functional or some suitably chosen reward function. The updated parameters are obtained by minimizing the Kullback-Leibler distance (or cross-entropy) between the optimal importance distribution and the parametric family of distributions used for sampling. A comprehensive review of CE methods is given in [33].

Based on [33, Ch. 2.4], in [27] a CE method was proposed for the generic multi-dimensional assignment problem. In the following, we adapt the CE method of [27] for sampling multi-sensor assignments in the joint δ-GLMB framework of

Sec. IV-B.

In order to achieve a simple mechanism for sampling multi-sensor assignments, we construct marginal assignment pdfs for each sensor. Through the iterative process of CE, we aim to adaptively select marginal assignment pdfs that converge toward the exact marginal assignment distributions. Consider label set $I = I_k \cup \mathbb{L}_{k+1}$ with n labels and sensor j with measurements $\{z_j^1, ...z_j^{m_j}\}$. We define the space $\Xi_j = \{-1, 0, 1, \dots, m_j\}$ and the row-stochastic probability matrix Π_j

$$\Pi_j = \begin{bmatrix} p_j(1|-1) & p_j(1|0) & ... & p_j(1|m_j) \\ & ... & & \\ p_j(n|-1) & p_j(n|0) & ... & p_j(n|m_j) \end{bmatrix} \tag{37}$$

where $p_j(i|-1)$ is the probability of death for target i; $p_j(i|0)$ is the miss-detection probability and $p_j(i|r)$ for $r > 0$ is the probability of assigning the measurement z_j^r of sensor j to the i-th target. Sampling an assignment map $\xi_j : I \to \Xi_j$ is achieved by performing categorical sampling for each row of Π_j, amounting to sampling assignments for each target in I for sensor j. Let $\Pi_{1:S} = (\Pi_1, \Pi_2, ..., \Pi_S)$ denote the collection of probability matrices for all S sensors. Additionally, we denote with $\xi_{1:S} = (\xi_1, \dots, \xi_S)$ the sequence of mappings drawn according to Π.

The probability $p(\xi_{1:S}; \Pi_{1:S})$ of drawing $\xi_{1:S}$ according to $\Pi_{1:S}$ is

$$p(\xi_{1:S}; \Pi_{1:S}) = \prod_{j=1}^{S} \prod_{i=1}^{n} p_j(i|\xi_j(i)). \tag{38}$$

The resulting maps ξ_j are not guaranteed to be valid, i.e., elements of $\bar{\theta}_{1:S}(I)$. In order to obtain high-weight $\bar{\eta}_Z^{(\epsilon,\xi_{1:S})}(\cdot)$ assignments and guarantee their validity, we introduce the modified score function

$$C(\xi_{1:S}) = \begin{cases} \prod_{l \in I} \bar{\eta}_Z^{(\epsilon,\xi_{1:S})}(l) & \text{if } \xi_{1:S} \in \bar{\Theta}_{1:S}(I), \\ 0 & \text{otherwise.} \end{cases} \tag{39}$$

In effect, scoring samples $\xi_{1:S}$ via $C(\xi_{1:S})$ assigns a score of 0 for non-valid assignment maps and employs the weight of (24) otherwise.

We draw N samples $\xi_{1:S}^1, \dots, \xi_{1:S}^N$ and compute their costs using (39). For a given scoring threshold ϑ, the CE approach calls for updating the parameters of $\Pi_{1:S}$ through the following program [27]

$$\underset{\Pi_{1:S}}{\text{maximize}} \ \frac{1}{N} \sum_{i=1}^{N} 1_{[\vartheta,+\infty)}(C(\xi_{1:S}^i)) \ln\left(p(\xi_{1:S}^i; \Pi_{1:S})\right). \tag{40}$$

The threshold ϑ can be adaptively estimated as the $(1-\tau)$ quantile of the sample set $C(\xi_{1:S}^1), \dots, C(\xi_{1:S}^N)$. In practice, the sample maps are sorted $C(\xi_{1:S}^{\sigma(1)}) \leq \cdots \leq C(\xi_{1:S}^{\sigma(N)})$, with σ a permutation of $\{1, \dots, N\}$, and the $H = \lfloor \tau N \rfloor$ highest scoring mappings are retained. Let $\tilde{\xi}_{1:S}^1, \dots, \tilde{\xi}_{1:S}^H$ denote the resulting set of highest scoring H samples.

Employing the probabilities of (38), the program (40) becomes

$$\underset{p_j(\cdot|\cdot)}{\text{maximize}} \ \sum_{h=1}^{H} \ln\left(\prod_{j=1}^{S} \prod_{i=1}^{n} p_j(l^i|\tilde{\xi}_j^h(i))\right) \tag{41}$$

$$\text{subject to} \quad \sum_{r\in\Xi_j} p_j(i|r) = 1, 1 \leq i \leq n, 1 \leq j \leq S$$
$$p_j(i|r) \geq 0 \ , 1 \leq i \leq n, 1 \leq j \leq S, \ r \in \Xi_j.$$

Solving the program (41) is equivalent to solving a program for each sensor $j = 1, \ldots, S$, yielding the following update rules for the probability matrices Π_j [27]

$$p_j(i|r) = \frac{|\{h \in \{1, 2, ..., H\} : \tilde{\xi}_j^h(i) = r\}|}{H}. \tag{42}$$

The numerator in (42) simply counts the number of samples in which sensor j selects assignment $r \in \Xi_j$ for label l^i whereas the denominator acts as a normalization constant.

In [27], the authors suggest initializing $\Pi_{1:S}$ using uniform distributions. In our simulations, we find that a poor initialization leads to nearly all drawn samples having very low score and slow convergence to the optimal distribution. We instead initialize Π_j^0 with the probabilities

$$p_j^0(i|r) \propto \begin{cases} 1 - \eta_s^\epsilon(l^i) \\ \qquad \text{if } l^i \in \mathbb{L}_{0:k}, \text{ and } r = -1, \\ \eta_s^\epsilon(l^i)\langle p_{k+1|k}^{(\epsilon)}(\cdot, l^i), \psi_j(\cdot, l^i; r)\rangle \\ \qquad \text{if } l^i \in \mathbb{L}_{0:k}, \text{ and } r \geq 0, \\ 1 - r_{B,k+1}(l^i) \\ \qquad \text{if } l^i \in \mathbb{L}_{k+1}, \text{ and } r = -1, \\ r_{B,k+1}(l^i)\langle p_{k+1|k}^{(\epsilon)}(\cdot, l^i), \psi_j(\cdot, l^i; r)\rangle \\ \qquad \text{if } l^i \in \mathbb{L}_{k+1}, \text{ and } r \geq 0. \end{cases} \tag{43}$$

This initialization scheme involves the same $\mathcal{O}(n\sum_{s=1}^S m_s)$ pre-computations as the Gibbs method when employing the approximation (36).

In Algorithm 2, we present the algorithm for solving the multi-assignment problem with CE. The algorithm iterates the main steps of sampling and updating the marginal probabilities $\Pi_{1:S}$ until convergence. Convergence implies a small change in the updated marginals. Let $\Pi'_{1:S}$ be the updated marginals for a given iteration and $\Pi_{1:S}$ the marginals from the previous iteration. For a small $\nu \in \mathbb{R}_+$, we adopt the convergence criterion $\left\|\Pi'_j - \Pi_j\right\|_F \leq \nu \ \forall \ j \in \{1, \ldots, S\}$, where $\|\cdot\|_F$ denotes the Frobenius matrix norm.

Algorithm 2 CE method for the multi-sensor assignment problem

1: Initialize $\Pi_{1:S} = \Pi_{1:S}^0$.
2: Draw N samples $\xi_{1:S}^1, \ldots, \xi_{1:S}^N$ according to $\Pi_{1:S}$.
3: Compute $C(\xi_{1:S}^i)$ for $i = 1, \ldots, N$.
4: Sort the sequence $\{C(\xi_{1:S}^i)\}_{i=1}^N$ and choose the $H = \lfloor \tau N \rfloor$ best samples $\{\tilde{\xi}_{1:S}^h\}_{h=1}^H$.
5: Update $\Pi_{1:S}$ with (42).
6: Iterate steps 2-5 until convergence.
7: Return T-best distinct and valid maps $\{\xi_{1:S}^i, C(\xi_{1:S}^i)\}_{i=1}^T$ from all samples and all iterations.

Step 2 of Algorithm 2 involves sampling of multi-assignments $\xi_{1:S}$. Sampling directly from (37) is inefficient since many sampled mappings $\xi_{1:S}$ may not be valid. These mappings will not contribute to the updated $\Pi_{1:S}$ due to having a zero score (39). To avoid generating irrelevant mappings, we employ the sampling procedure described in Algorithm 3, where $\mathcal{U}(0,1)$ denotes a uniform probability distribution over $[0,1]$ and Categorical(c,q) is the categorical probability distribution where each category c_i has probability q_i. The sample $\xi_{1:S}$ drawn using the mechanism in Algorithm 3 represents a valid

Algorithm 3 Generation of a mapping $\xi_{1:S} \in \bar{\Theta}_{1:S}(I)$, i.e., 1-1 positive.

Set $\hat{\Pi}_{1:S} = \Pi_{1:S}$.
for $i \leftarrow 1$ to n **do**
  Sample $u \sim \mathcal{U}(0,1)$.
  if $u \leq p_1(i|-1)$ **then**
    Target death: $\xi_j(i) = -1 \ \forall \ j \in \{1,\ldots,S\}$.
  else
    for $j \leftarrow 1$ to S **do**
      Set $q(r) = \hat{p}_j(i|r)$ for $r = \{0,\ldots,m_j\}$.
      Normalize $q(\cdot)$.
      Sample $\xi_j(i) \sim$ Categorical$(0:m_j, q)$.
      if $\xi_j(i) \neq 0$ **then**
        Set $\hat{p}_j(k|\xi_j(i)) = 0$ for $k \in \{i+1,\ldots,n\}$.
      end if
    end for
  end if
end for

multi-assignment in the joint δ-GLMB framework. The computational complexity of Algorithm 3 is $\mathcal{O}(n\sum_{s=1}^{S} m_s + n^2 S)$. The computational complexity of evaluating the score $C(\xi_{1:S})$ of a multi-assignment is $\mathcal{O}(nS)$. Hence, Step 3 of Algorithm 2 has a complexity of $\mathcal{O}(NnS)$. The complexities of the sorting and updating of $\Pi_{1:S}$, in Algorithm 2, are $\mathcal{O}(N\ln(N))$ and $\mathcal{O}(nSH)$. For d iterations, the computational complexity of Algorithm 2 is $\mathcal{O}(dNn\sum_{s=1}^{S} m_s + dNn^2 S + dN\ln(dN))$. Note the linear complexity with the number of sensors and quadratic with the number of targets. This represents a drastic reduction when compared with $\mathcal{O}(Tn^2\prod_{s=1}^{S} m_s)$ of the Gibbs method when using the exact multi-assignment weights (24).

VI. Performance evaluation

In this section, we evaluate and compare the performance of all multi-sensor assignment algorithms presented in Sec. V. In addition, we also include the sequential multi-sensor δ-GLMB filter of [21] as a baseline.

We consider a network of $S = 4$ sensors. Each sensor detects target(s) with constant probability $P_D = 0.9$. We consider a linear measurement model (i.e., sensors measure target positions directly) so sensor positions do not affect the algorithms' performance. All measurements are corrupted by white Gaussian noise with covariance matrix $\sigma_{xy}^2 I_2$ where $\sigma_{xy} = 3$m and I_2 is the 2×2 identity matrix. The number of clutter measurements per sensor is Poisson distributed with rate λ_c. Clutter measurements are uniformly distributed over a 100m $\times$ 100m tracking area.

For the combination and sequential methods, each sensor generates a maximum of T single-sensor maps. For the cross entropy method, at each sampling iteration, a maximum of T (and a minimum of 10) multi-sensor maps are generated, and the τT best (highest-score) sampled maps are used to update the sampling distribution parameters ($\tau = 0.3$). We repeat the iterations until convergence (i.e., the L_2 distance between the new and old marginal sampling distributions is below the

threshold $\nu = 1e-3$) or after running ten iterations. The best T distinct samples from all iterations are output as the multi-sensor assignment maps. Finally, for the Gibbs sampling method, we use every drawn sample as a valid multi-sensor map. All algorithms are required to generate a maximum of T multi-sensor assignment maps to ensure a fair comparison.

A. Single time-step test

In the first test, we focus on a single time step and a single hypothesis. The objective is to evaluate and compare the multi-sensor assignment algorithms in a controlled environment where all inputs are equal and known. Fig. 2 shows the targets' positions and sensor measurements. We omit the sensors' positions since they have no impact on the algorithm's performance.

In this test, we generate one label set based on true target states and apply the association algorithms to this hypothesis to generate a maximum of T assignment maps. Recall that, for the two sampling methods, cross entropy and Gibbs, the generated maps are not necessarily distinct. Duplicate maps are removed and only the distinct maps are retained.

We run a series of simulations where we vary one parameter of interest and investigate its impact on the algorithms' performance. The parameters we vary are the measurement noise standard deviation σ_{xy}, the Poisson rate of clutter measurements per sensor λ_c, and the maximum number of association maps T. In all simulations, the targets' positions remain unchanged. Each simulation consists of 200 random Monte Carlo trials. Although the Gibbs method is fast, it tends to return many fewer than T distinct maps when T steps of Gibbs sampling are executed. To better illustrate the trade-off between runtime and number of distinct maps returned, we also run the Gibbs method for $10T$ steps (referred to as Gibbs-10 below). If more than T distinct maps are generated, the T maps with the highest approximate score are returned.

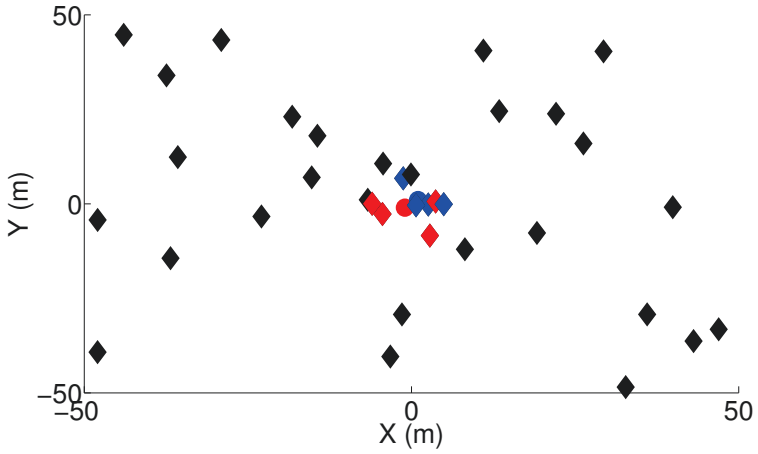

Fig. 2. Targets (blue and red dots) and sensor measurements. Clutter measurements are black diamonds. Target-originated measurements are diamonds with matching colors.

The first performance metric we consider is the total runtime to generate the T multi-sensor association maps. The second performance metric is the ratio of aggregate map score. Recall the posterior hypothesis weight: $w^{(I_{k+1},\epsilon,\theta_{1:S})}(Z) \propto w_{k|k}^{(I_k,\epsilon)}[\eta_Z^{(\epsilon,\theta_{1:S})}]^{I_{k+1}}$ (in separate predict-update framework) and $w^{(I_{k+1},\epsilon,\theta_{1:S})}(Z) \propto w_{k|k}^{(I_k,\epsilon)} w_Z^{(I_k,\epsilon,I_{k+1},\theta_{1:s})}$ (in joint predict-update framework). The term $w^{(I_{k+1},\epsilon,\theta_{1:S})}(Z)$ can thus be interpreted as the score of multi-sensor map $\theta_{1:S}$ for hypothesis (I_{k+1},ϵ). Note that the map score functions in the separate and joint predict-update frameworks are equivalent if no new label can appear and the target survival probability is 1. The four algorithms produce a combined total of up to $4T$ association

TABLE I

AVERAGE RUNTIME (IN SECONDS) FOR MAP GENERATION. THE NUMBER IN BRACKETS IS THE NUMBER OF DISTINCT MAPS. THE LAST NUMBER IS THE RATIO OF AGGREGATE MAP SCORE. THE PARAMETERS VALUES FOR THE BASELINE ARE $P_D = 0.9$, $\lambda_c = 6$, $T = 500$ AND $\sigma_{xy} = 3$. THE SHORTEST RUNTIME FOR EACH SCENARIO IS MARKED IN RED. THE HIGHEST RATIO IS MARKED IN BLUE.

	Combination	Cross-entropy	Gibbs (T)	Gibbs-10 ($10T$)	Sequential
Baseline	0.048 (500) / **0.987**	0.293 (155) / 0.898	**0.037** (146) / 0.802	0.343 (420) / 0.974	0.293 (424) / 0.952
$\sigma_{xy} = 1$	0.053 (500) / **0.995**	0.319 (144) / 0.966	**0.040** (114) / 0.780	0.376 (344) / 0.979	0.249 (247) / 0.984
$\sigma_{xy} = 2$	0.047 (500) / **0.988**	0.291 (138) / 0.936	**0.037** (122) / 0.805	0.341 (367) / 0.975	0.262 (358) / 0.965
$\sigma_{xy} = 4$	0.052 (500) / **0.968**	0.348 (212) / 0.857	**0.040** (191) / 0.762	0.382 (466) / 0.957	0.340 (435) / 0.930
$\sigma_{xy} = 5$	0.050 (500) / **0.958**	0.349 (243) / 0.823	**0.039** (228) / 0.735	0.368 (484) / 0.949	0.351 (466) / 0.917
$\lambda_c = 2$	**0.031** (500) / **0.999**	0.272 (65) / 0.953	0.036 (48) / 0.619	0.367 (172) / 0.986	0.089 (183) / 0.976
$\lambda_c = 4$	0.041 (500) / **0.992**	0.301 (110) / 0.919	**0.039** (99) / 0.757	0.382 (311) / 0.977	0.194 (299) / 0.960
$\lambda_c = 8$	0.067 (500) / **0.971**	0.364 (202) / 0.875	**0.044** (184) / 0.775	0.389 (458) / 0.959	0.478 (466) / 0.946
$\lambda_c = 10$	0.078 (500) / **0.948**	0.381 (259) / 0.847	**0.044** (232) / 0.744	0.382 (485) / 0.940	0.601 (481) / 0.934
$T = 250$	0.037 (250) / **0.965**	0.215 (113) / 0.882	**0.025** (92) / 0.650	0.193 (236) / 0.952	0.271 (231) / 0.936
$T = 750$	0.070 (750) / **0.989**	0.482 (198) / 0.912	**0.058** (190) / 0.853	0.618 (532) / 0.976	0.370 (550) / 0.958

maps, some of which may be duplicates. Let $\theta^{(1)}, ...\theta^{(T')}$ denote the distinct maps ($T' \leq 4T$). We compute the score for all T' maps. Let $\theta^{(1:T)}_{\text{optimal}}$ denote the T maps with highest score and let C^{T}_{optimal} denote their aggregate score. For each algorithm, we compute the aggregate score of all generated maps and report the ratio of the algorithm's aggregate map score to C^{T}_{optimal}.

Table I shows the simulation results. The measurement noise standard deviation σ_{xy} has no impact on the total runtime for both the combination and the Gibbs method. In the cross entropy method, to avoid redundant computation, we store all distinct sampled maps and their scores. When σ_{xy} increases, the sampling distribution is more dispersed, more distinct maps are sampled through the sampling iterations, and the additional score computation increases the running time. In the sequential method, we iterate through each sensor s and incrementally build the multi-sensor maps $\theta_{1:s}$ from $\theta_{1:s-1}$. Higher σ_{xy} leads to more partial maps $\theta_{1:s}$ and by extension more maps $\theta_{1:S}$ at the last sensor S and longer running time. Finally, more clutter measurements and more association maps translate to higher workload and consequently higher running time as expected.

In nearly all tested scenarios, the Gibbs method has the shortest running time, followed by the combination method. The cross entropy method is often the slowest. The combination and sequential methods consistently generate the highest number of distinct maps with the combination method yielding T maps for all scenarios. The two sampling methods, cross-entropy and Gibbs, generate fewer distinct maps as expected. When the number of clutter measurements is low, the distribution of the Gibbs sampler is far from uniform (i.e., most weight falls on one single measurement) and as a result the Gibbs methods can only yield a very small number of distinct maps. Finally, higher σ_{xy} and λ_c lead to more distinct maps for the cross entropy, Gibbs and sequential methods. In the case of Gibbs-10, the runtime is generally higher than that of the cross-entropy method. At low σ_{xy} and λ_c, the number of maps increases almost by a factor of 3 compared to the Gibbs method.

Next, we consider the ratio of aggregate map score. In all scenarios, the combination method has the highest ratio. This can be attributed to the fact that this method yields T maps for all scenario. Conversely, the Gibbs method generates only a small number of distinct maps which leads to a low aggregate map score (especially for $\lambda_c = 2$ and $T = 250$). We do note that, at $\lambda_c = 2$, even though the Gibbs method only generates 46 maps (compared to 500 maps for the combination method), its score ratio still exceeds 0.5. This suggests that the map scores are far from uniform. Therefore, even though the Gibbs method generates many fewer distinct maps, these maps have very high score. In comparison, the Gibbs-10 method is able

to generate more distinct maps which translates to higher aggregate map score and consistently outperforms the cross-entropy method. These results suggest that, for the Gibbs method, one can select the number of Gibbs samples for a trade-off between computational overhead and performance.

Consider now the individual parameter's impact. Fewer clutter measurements and/or lower measurement noise lead to less challenging scenarios and higher aggregate map scores. The Gibbs method does not follow these trends since the low number of generated maps at low σ_{xy} and λ_c has a bigger impact on the aggregate map score. Finally, when more maps are generated, the algorithms are more likely to retrieve more high-scoring maps, leading to higher aggregate score.

Note that generating association maps with higher aggregate score for a given hypothesis does not necessarily imply better tracking performance in a tracking scenario spanning multiple time steps. One important caveat to keep in mind is that the key benefit of the joint predict-update framework (i.e., avoiding the selection of less likely hypotheses) is not applicable in this single hypothesis test.

B. Approximate multi-sensor assignment score function

In this section, we seek to experimentally determine the conditions under which the score function $\sum_{j=1}^{S} \log\left(\eta_Z^{(\epsilon,\theta_j)}(l)\right)$ is a suitable substitute for the exact multi-sensor assignment score function $\log\left(\eta_Z^{(\epsilon,\theta_{1:S})}(l)\right)$. Recall that both the combination and Gibbs methods rely on this approximation to reduce computational overhead.

Before proceeding further, we must define a proper performance metric. Recall that our goal is to generate high-scoring maps. Therefore, we wish to determine whether the use of the approximate score function can yield the same maps as we would get with the exact score function.

Let $\theta^{(1)}, ..., \theta^{(T)}$ denote the T distinct multi-sensor maps with $C_{\text{approx}}^{(i)} \geq C_{\text{approx}}^{(j)}, \quad \forall i \geq j$ (i.e., maps are sorted and indexed in decreasing order of approximate score). Let $[\alpha(1), ...\alpha(T)]$ denote the ordering of these T maps under the exact score (i.e., $C_{\text{exact}}^{(\alpha(i))} \geq C_{\text{exact}}^{(\alpha(j))}, \quad \forall \alpha(i) \geq \alpha(j)$).

Consider what happens when we generate the $\lceil \gamma T \rceil$ best maps. Under the approximate score functions, we would obtain $\theta^{(1)}, ..., \theta^{(\lceil \gamma T \rceil)}$. Under the exact score function, we would obtain $\theta^{(\alpha(1))}, ..., \theta^{(\alpha(\lceil \gamma T \rceil))}$. In the ideal situation, we would obtain the same $\lceil \gamma T \rceil$ maps and have $\delta_{[1,2,...,\lceil \gamma T \rceil]}\left([\alpha(1), \alpha(2), ..., \alpha(\lceil \gamma T \rceil)]\right) = 1$.

A natural choice of performance metric is the Jaccard coefficient which measures the similarity between the sets $[1, 2, ..\lceil \gamma T \rceil]$ and $[\alpha(1), \alpha(2), ..\alpha(\lceil \gamma T \rceil)]$. A coefficient of 1 means that the two sets are identical and a coefficient 0 means no overlap between the two sets. The Jaccard coefficient for two sets A and B is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{44}$$

We adopt the same single time step set-up in Sec. VI.1. We generate a total of $T = 2000$ maps using the combination method, compute their scores using both the exact and approximate score functions, and compute the Jaccard coefficient. We run a series of simulations where we vary one parameter of interest. Table II shows the results.

TABLE II
AVERAGE JACCARD SIMILARITY COEFFICIENT BETWEEN THE MAP SETS UNDER EXACT AND APPROXIMATE SCORE FUNCTIONS. ALL RESULTS ARE AVERAGED OVER 200 MONTE-CARLO TRIALS. FOR THE BASELINE, $P_D = 0.97$, $\lambda_c = 7$, AND $\sigma_{xy} = 6$.

Parameter	$\lceil \gamma T \rceil = 250$	$\lceil \gamma T \rceil = 500$	$\lceil \gamma T \rceil = 750$
Baseline	0.7238	0.7482	0.7722
$P_D = 0.5$	0.7563	0.7715	0.7848
$P_D = 0.6$	0.7388	0.7578	0.7801
$P_D = 0.8$	0.7010	0.7236	0.7553
$P_D = 0.9$	0.7009	0.7230	0.7543
$\sigma_{xy} = 2$	0.4824	0.5200	0.6002
$\sigma_{xy} = 4$	0.6548	0.6858	0.7118
$\sigma_{xy} = 8$	0.7425	0.7605	0.7849
$\sigma_{xy} = 10$	0.7846	0.7958	0.8138
$\lambda_c = 1$	0.8675	0.8780	0.8924
$\lambda_c = 3$	0.7869	0.7994	0.8137
$\lambda_c = 7$	0.6834	0.7097	0.7404
$\lambda_c = 9$	0.6444	0.6721	0.7098

Consider the exact cost function:

$$\eta_Z^{(\epsilon,\theta_{1:S})}(l) = \int p_{k+1|k}^{(\epsilon)}(\cdot, l) \prod_{j=1}^{S} \psi_j(\cdot, l; \theta_j) dx \tag{45}$$

$$\psi_j(x, l; \theta_j) = \begin{cases} 1 - P_D & \theta_j(l) = 0 \\ \frac{P_{D,j}(x) g_j(z_j^i | x, l)}{\kappa_j(z_j^i)} & \theta_j(l) \in \{1, ..., m_j\} \end{cases} \tag{46}$$

When P_D is low, then each label is more likely to be miss-detected. In this case, the term $\psi_j(x, l; \theta_j)$ has a constant value $1 - P_D$. In the case when none of sensors detects the target, the exact and approximate score values are equal. In other words, at lower P_D, the approximate score function is close to the exact cost function. The Jaccard coefficients decrease with increasing P_D in Table II.

At higher σ_{xy}, the measurement likelihood function $g(z|x, l)$ is more spread out with respect to the target density $p_{k+1|k}^{(\epsilon)}(\cdot)$. At sufficiently high σ_{xy}, the likelihood function may appear non-informative with respect to the prior density $p_{k+1|k}^{(\epsilon)}(\cdot)$. In this case we have $\int g(z_1|x, l) g(z_2|x, l) p_{k+1|k}^{(\epsilon)}(x, l) dx \approx g(z_1|\mu, l) g(z_2|\mu, l) = \left[\int g(z_1|x, l) p_{k+1|k}^{(\epsilon)}(x, l) dx\right] \left[\int g(z_2|x, l) p_{k+1|k}^{(\epsilon)}(x, l) dx\right]$ with μ the mean of $p_{k+1|k}^{(\epsilon)}(\cdot)$ and the approximate score function is close to the exact cost function. Conversely, at lower σ_{xy}, the measurements are highly informative and we have $\int g(z_1|x, l) g(z_2|x, l) p_{k+1|k}^{(\epsilon)}(x, l) dx \approx g(z_1|\mu, l) g(z_2|\mu', l)$ where $\mu' \neq \mu$ and by extension $g(z_2|\mu', l) \neq g(z_2|\mu, l)$. When S updates are applied in sequence, the resulting exact score can differ significantly from the approximate score.

At higher clutter rate, a sensor is more likely to generate single-sensor maps containing clutter-target associations. In contrast, the exact score function can more reliably discriminate true target-originated measurements from clutter by incorporating the information from all sensors. As a result, higher clutter rate leads to higher discrepancy (lower similarity coefficient) between the two scoring functions.

Finally, increasing the value of $\lceil \gamma T \rceil$ increases the Jaccard coefficient in all cases. Since there is only a large but finite number of valid maps, retaining more maps is certain to increase the overlap between the two sets.

C. Full tracking test

In our second test, we run a full tracking simulation using each algorithm to solve the multi-sensor assignment problem in the update stage. Fig. 3 shows the target tracks and sensor placements. The simulation spans 30 time steps and contains a total of five targets. Again, we run a series of scenarios to study how certain parameters impact the tracking performance.

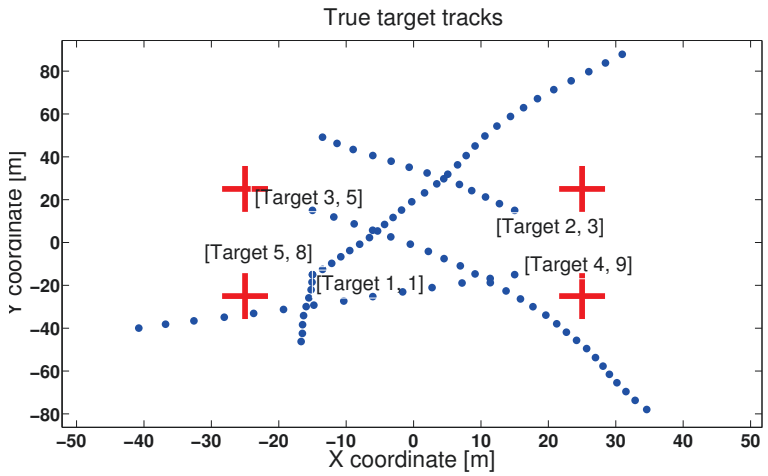

Fig. 3. Target tracks (blue lines) and sensor positions (red crosses). Text labels indicate targets' birth positions and birth time step

All targets follow the constant velocity dynamic model with dynamic noise covariance

$$Q = \sigma_v^2 \begin{bmatrix} \frac{I_2}{3} & \frac{I_2}{2} \\ \frac{I_2}{2} & I_2 \end{bmatrix} \tag{47}$$

where σ_v is the standard deviation of process noise, I_n is the $n \times n$ identity matrix, and we assume a sampling interval of 1 minute. We use $\sigma_v = 0.25$ for the track generation and $\sigma_v = 2.5$ for the target tracking. At each time step, one target may appear at each of the four spawn points: $p_B(x) = \mathcal{N}(x; \mu_B, P_B)$ with $\mu_B = [\pm 15, \pm 15, 0, 0]$, $P_B = \text{diag}([25, 25, 50, 50])$ and birth probability $r_B = 0.4$. At each update step, a maximum of T multi-sensor association maps are generated. If there is more than one predicted hypothesis, we sample T hypotheses with replacement based on the predicted hypothesis weight. After each update step, we prune all posterior hypotheses below a weight threshold set to 10^{-10} and we propagate only the $T/4$ posterior hypotheses with highest weight to the next time step.

We compute the estimated target cardinality distribution and target states as follows:

$$p(n) = \sum_{|I|=n} w^{(I,\epsilon,\theta_{1:S})} \tag{48}$$

$$\hat{n} = \arg\max \quad p(n) \tag{49}$$

$$(\hat{I}, \epsilon) = \arg\max \quad w^{(I,\epsilon,\theta_{1:S})} \delta_{\hat{n}}(|I|)) \tag{50}$$

$$\hat{\mathbf{X}} = \left\{ (\hat{x}, \hat{l}) : \hat{l} \in \hat{I}, \hat{x} = \int y p(y, l) dy \right\} \tag{51}$$

We measure the performance using two metrics: total running time for the entire tracking process (including predict, update and target state extraction) and *optimal sub-pattern assignment* (OSPA) (with cardinality penalty factor of 20 and power factor

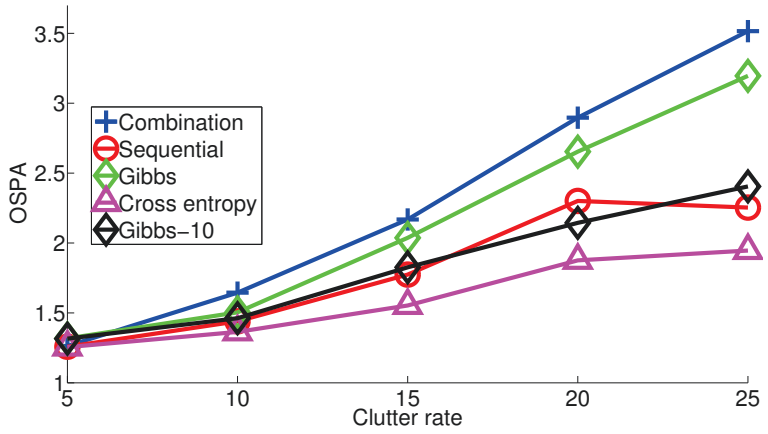

Fig. 4. Average OSPA with respect to λ_c. All results are averaged over 100 random trials. The other parameters are $\sigma_{xy} = 1.5$m and $T = 500$.

of 1) [34].

Fig. 4 shows the average OSPA with respect to the clutter rate for $\sigma_{xy} = 1.5$m and $T = 500$. The Gibbs-10 is also included for comparison. At low clutter rate ($\lambda_c \leq 10$), all algorithms yield similar performance. As the clutter rate increases, the OSPA increases for all algorithms as expected. At higher clutter rate, the cross-entropy and sequential methods outperform the combination and Gibbs methods. This performance gap is to be expected since the latter two methods use an approximate score function which performs more poorly at higher clutter rate. However, the Gibbs-10 method yields similar performance as the sequential method and consistently outperforms the Gibbs method. Finally, the cross-entropy method has consistently the lowest OSPA.

Fig. 5 shows the average runtime for the entire simulation with respect to the clutter rate for $\sigma_{xy} = 1.5$m and $T = 500$. Note that the different algorithms generate a different number of hypotheses at each time step. This in turn can have a major impact on the total runtime. For all methods, the runtime increases with increasing λ_c as expected. For the combination method, more clutter measurements lead to longer runtime for Murty's algorithm at individual sensors, but otherwise have limited impact on the overall runtime since the number of generated multi-sensor maps (and by extension the number of hypotheses) does not change. For the other three methods, more clutter measurements actually lead to more maps and thus have a much larger impact on the total runtime. The Gibbs-10 method has consistently the highest runtime.In the full tracking scenario, the Gibbs-10 method constructs more hypotheses and by extension more labeled targets than the cross-entropy method at each time step. This results in a higher computational load and longer runtime which accumulates over the 30 time steps. Finally, with the exception of the lower clutter rate case ($\lambda_c \leq 10$), the combination method has the shortest runtime.

Fig. 6 shows the average OSPA with respect to σ_{xy} at $\lambda_c = 15$ and $T = 500$. As σ_{xy} increases, the OSPA of all algorithms increases as expected. The cross-entropy method has consistently the lowest OSPA, followed by the sequential and Gibbs-10 methods.

Fig. 7 shows the average runtime for the whole simulation with respect to σ_{xy} at $\lambda_c = 15$ and $T = 500$. Increasing σ_{xy} has the least impact on the runtime of the combination and sequential methods. For the two sampling methods, higher σ_{xy} lead to more maps being generated and processed and subsequently higher runtime. The Gibbs-10 method again has the highest runtime by far. Finally, the combination method has the lowest runtime with the exception of the low noise scenario ($\sigma_{xy} = 1$).

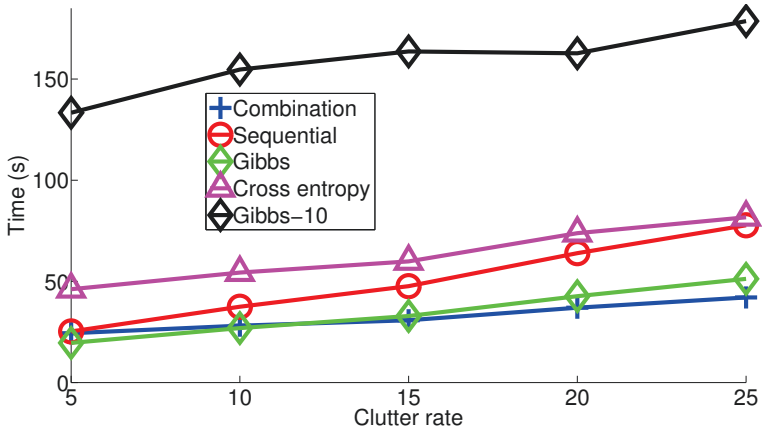

Fig. 5. Average runtime with respect to λ_c. All results are averaged over 100 random trials. The other parameters are $\sigma_{xy} = 1.5$m, $T = 500$.

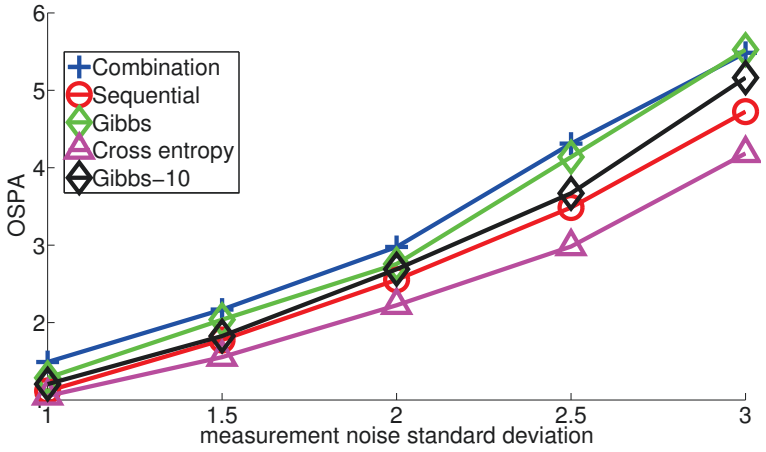

Fig. 6. Average OSPA with respect to σ_{xy}. All results are averaged over 100 random trials. The other parameters are $\lambda_c = 15$, $T = 500$.

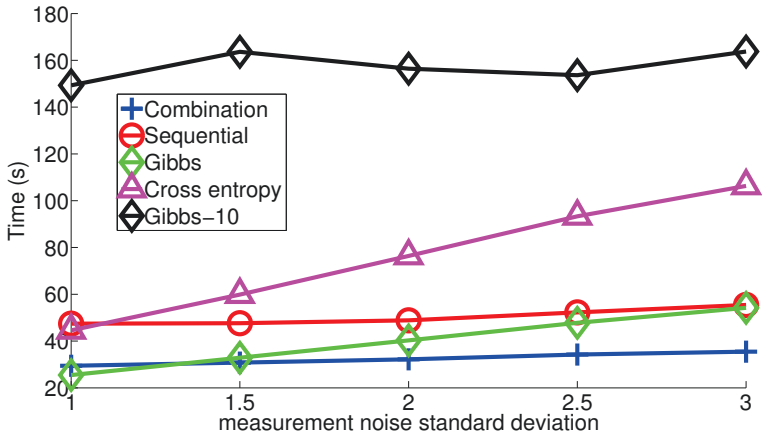

Fig. 7. Average runtime with respect to σ_{xy}. All results are averaged over 100 random trials. The other parameters are $\lambda_c = 15$, $T = 500$.

Overall, the cross-entropy method consistently has the lowest OSPA in all our test scenarios. This can be attributed to the use of the exact score function for map selection and the application of the joint predict-update framework to minimize the selection of unlikely hypotheses (i.e., hypotheses that are uncorroborated by the sensor measurements). The sequential method also uses the exact score function but caps the total number of hypotheses after each sensor which may lead to loss of high-weight hypotheses. At low σ_{xy} or low λ_c, the Gibbs method is the fastest and all algorithms yield similar tracking performance. At high σ_{xy} or λ_c, the combination method is the fastest. Our results suggest that the combination method is a suitable choice for solving the multi-sensor assignment problem if the runtime is the primary selection criterion. On the other hand, if we seek the best tracking performance, then the cross-entropy method should be used. We may also fine-tune the number of samples in the Gibbs method to achieve a suitable trade-off between runtime and tracking performance.

VII. Conclusion

In this paper, we present two novel algorithms, the combination and cross-entropy methods, for solving the multi-sensor measurement association problem in the multi-sensor δ-GLMB filter and compare their performance against the Gibbs and sequential methods.

The cross-entropy method has consistently the best tracking performance (measured using the OSPA distance). When the measurement noise level or the clutter rate is low, all four algorithms yield similar tracking performance with the Gibbs method being the fastest. Otherwise, the combination method has the shortest runtime.

The combination and Gibbs methods approximate the exact multi-sensor association score function as a product of single-sensor association scores. Simulations show that this approximation reduces computational overhead but leads to worse tracking performance at low measurement noise level and high clutter rate.

For future work, we will focus on developing distributed implementations of the presented algorithms in order to develop distributed multi-sensor δ-GLMB filters.

References

[1] W. F. Luo, J. L. Xing, X. Q. Zhang, X. W. Zhao, and T.-K. K., "Multiple object tracking: A literature reivew," Sep. 2015, preprint available at https://arxiv.org/abs/1409.7618.

[2] R. Mahler, *Statistical Multisource-Multitarget Information Fusion*. Norwood, MA, USA: Artech House, 2007.

[3] ——, "PHD filters of higher order in target number," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 4, pp. 1523–1543, 2007.

[4] B.-T. Vo, B.-N. Vo, and A. Cantoni, "Analytic implementation of the cardinalized probability hypothesis density filter," *IEEE Trans. Signal Processing*, vol. 55, no. 7, pp. 3553–3567, Jul. 2007.

[5] ——, "The cardinality balanced multi-target multi-Bernoulli filter and its implementations," *IEEE Trans. Signal Processing*, vol. 57, no. 2, pp. 409–423, 2009.

[6] D. Clark, I. T. Ruiz, Y. Petillot, and J. Bell, "Particle PHD filter multiple target tracking in sonar image," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 43, no. 1, pp. 409–416, Jan. 2007.

[7] M. Ulmke, O. Erdinc, and P. Willett, "Gaussian mixture cardinalized PHD filter for ground moving target tracking," in *Int. Conf. Inform. Fusion.*, Jul. 2007, pp. 1–8.

[8] K. Granstrom, C. Lundquist, and O. Orguner, "Extended target tracking using a Gaussian-mixture PHD filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 48, no. 4, pp. 3268–3286, Oct. 2012.

[9] C. Lundquist, K. Granstrom, and U. Orguner, "An extended target CPHD filter and a gamma Gaussian inverse Wishart implementation," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 3, pp. 472–483, Jun. 2013.

[10] L. Lamard, R. Chapuis, and J. P. Boyer, "Multi target tracking with CPHD filter based on asynchronous sensors," in *Proc. Int. Conf. Inform. Fusion*, Jul. 2013, pp. 892–898.

[11] N. T. Pham, W. M. Huang, and S. H. Ong, "Tracking multiple speakers using CPHD filter," in *Proc. ACM Int. Conf. Multimedia*, Augsburg, Germany, Sep. 2007, pp. 529–532.

[12] R. Hoseinnezhad, B. N. Vo, and B. T. Vo, "Visual tracking in background subtracted image sequences via multi-Bernoulli filtering," *IEEE Trans. Signal Proc.*, vol. 61, no. 2, pp. 392–397, Jan. 2013.

[13] D. Dunne and T. Kirubarajan, "Multiple model multi-Bernoulli filters for manoeuvering targets," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 4, pp. 2679–2692, Oct. 2013.

[14] B. T. Vo and B. N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Trans. Signal Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.

[15] B.-N. Vo, B.-T. Vo, and D. Phung, "Labeled random finite sets and the Bayes multi-target tracking filter," *IEEE Trans. Signal Processing*, vol. 62, no. 24, pp. 6554–6567, Dec. 2014.

[16] D. Clark and S. Nagappa, "On the ordering of the sensors in the iterated-corrector probability hypothesis density PHD filter," in *Proc. SPIE Signal Proc., Sensor Fusion, Target Recognition*, vol. 8050, Orlandao, Florida, Apr. 2011, pp. 80 500M–80 500M–6.

[17] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and A. Graziano, "Consensus CPHD filter for distributed multitarget tracking," *IEEE J. Selected Topics in Signal Processing*, vol. 7, no. 3, pp. 508–520, Jun. 2013.

[18] G. Battistelli, L. Chisci, C. Fantacci, A. Farina, and B.-N. Vo, "Average Kullback-Leibler divergence for random finite sets," in *Proc. Intl. Conf. Information Fusion*, Washintgon D.C., USA, Jul. 2015.

[19] R. Mahler, "The multisensor PHD filter: I, general solution via multitarget calculus," in *Proc. SPIE*, vol. 7336, 2009, pp. 73 360E–73 360E–12.

[20] B. Wang, W. Yi, S. Li, L. Kong, and X. Yang, "Distributed fusion with multi-Bernoulli filter based on generaized covariance intersection," in *IEEE Radar Conf.*, May 2015, pp. 958–962.

[21] F. Papi, "Multi-sensor δ-GLMB filter for multi-target tracking using Doppler only measurements," in *Europen Intell. Security Inform. Conf.*, Sept. 2015, pp. 83–89.

[22] W. F. Liu, B. S. Wei, and S. J. Zhu, "A multi-sensor generalized labeled multi-Bernoulli filter via extended association map," in *Proc. Intl. Conf. Control, Automation, Inform. Sci.*, Oct. 2015, pp. 225–230.

[23] S. Nannuru, S. Blouin, M. Coates, and M. Rabbat, "Multisensor CPHD filter," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 52, no. 4, pp. 1834–1854, Aug. 2016.

[24] A.-A. Saucan, M. Coates, and M. Rabbat, "A multi-sensor multi-Bernoulli filter," *IEEE Trans. Signal Process.*, no. 99, pp. 1–1, 2016.

[25] B. N. Vo and B. Vo, "Multi-sensor multi-object tracking with the generalized labeled multi-Bernoulli filter," `arXiv:1702.08849 [stat.CO]`.

[26] M. Miller, H. Stone, and I. Cox, "Optimizing Murty's ranked assignment method," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 33, no. 3, pp. 851–862, Jul. 1997.

[27] D. M. Nguyen, H. A. L. Thi, and T. P. Dinh, "Solving the multidimensional assignment problem by a cross-entropy method," *J. Combinatorial Optimization*, vol. 27, no. 4, pp. 808–823, May 2014.

[28] D. Eppstein, "Finding the K shortest paths," *SIAM J. Comput.*, vol. 28, no. 2, pp. 652–673, Feb. 1999.

[29] B. N. Vo, B. T. Vo, and H. G. Hoang, "An efficient implementation of the generalized labeled multi-Bernoulli filter," *IEEE Trans. on Signal Processing*, vol. 65, no. 8, pp. 1975–1987, Apr. 2017.

[30] L. Devroye, *Non-Uniform Random Variate Generation*. Springer-Verlag, 1986.

[31] R. Y. Rubinstein, "Optimization of computer simulation models with rare events," *European Journal of Operations Research*, vol. 99, pp. 89–112, 1996.

[32] R. Rubinstein, "The cross-entropy method for combinatorial and continuous optimization," *Methodology And Computing In Applied Probability*, vol. 1, no. 2, pp. 127–190, Sep 1999. [Online]. Available: http://dx.doi.org/10.1023/A:1010091220143

[33] R. Y. Rubinstein and D. P. Kroese, *The Cross-Entropy Method: A Unified Approach to Combinatorial Optimization, Monte-Carlo Simulation and Machine Learning*. Springer-Verlag New York, 2004.

[34] T. Vu and R. Evans, "A new performance metric for multiple target tracking based on optimal subpattern assignment," in *17th Int. Conf. Informa. Fusion*, Salamanca, Spain, Jul., pp. 1–8.

DOCUMENT CONTROL DATA

*Security markings for the title, authors, abstract and keywords must be entered when the document is sensitive

1. ORIGINATOR (Name and address of the organization preparing the document. A DRDC Centre sponsoring a contractor's report, or tasking agency, is entered in Section 8.)

McGill University
Electrical and Computer Engineering
845 Sherbrooke St W
Montreal, Quebec H3A 0G4

2a. SECURITY MARKING (Overall security marking of the document including special supplemental markings if applicable.)

CAN UNCLASSIFIED

2b. CONTROLLED GOODS

NON-CONTROLLED GOODS
DMC A

3. TITLE (The document title and sub-title as indicated on the title page.)

Algorithms for the multi-sensor assignment problem in the δ-generalized labeled multi-Bernoulli filter

4. AUTHORS (Last name, followed by initials – ranks, titles, etc., not to be used)

Yu, J.Y.; Saucan, A.-A.; Coates, M.; Rabbat, M.

5. DATE OF PUBLICATION (Month and year of publication of document.)	6a. NO. OF PAGES (Total pages, including Annexes, excluding DCD, covering and verso pages.)	6b. NO. OF REFS (Total references cited.)
December 2017	24	34

7. DOCUMENT CATEGORY (e.g., Scientific Report, Contract Report, Scientific Letter.)

Contract Report

8. SPONSORING CENTRE (The name and address of the department project office or laboratory sponsoring the research and development.)

DRDC - Atlantic Research Centre
Defence Research and Development Canada
9 Grove Street
P.O. Box 1012
Dartmouth, Nova Scotia B2Y 3Z7
Canada

9a. PROJECT OR GRANT NO. (If appropriate, the applicable research and development project or grant number under which the document was written. Please specify whether project or grant.)	9b. CONTRACT NO. (If appropriate, the applicable number under which the document was written.)
W7707-145675	01-CA

10a. DRDC PUBLICATION NUMBER (The official document number by which the document is identified by the originating activity. This number must be unique to this document.)	10b. OTHER DOCUMENT NO(s). (Any other numbers which may be assigned this document either by the originator or by the sponsor.)
DRDC-RDDC-2017-C244	

11a. FUTURE DISTRIBUTION WITHIN CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

Public release

11b. FUTURE DISTRIBUTION OUTSIDE CANADA (Approval for further dissemination of the document. Security classification must also be considered.)

12. KEYWORDS, DESCRIPTORS or IDENTIFIERS (Use semi-colon as a delimiter.)

Target tracking; Multi-sensor; Filter

13. ABSTRACT/RÉSUMÉ (When available in the document, the French version of the abstract must be included here.)

Practical implementations of the multi-sensor -generalized labeled multi-Bernoulli filter require solving the multi-sensor assignment problem which is NP-hard. In this paper, we present two different algorithms, the combination method and the cross-entropy method, that find T highly likely target-measurement associations without exhaustive enumeration of all possible multi-sensor assignments. Numerical simulations are conducted to evaluate the aforementioned multi-sensor assignment methods together with the standard sequential processing method and a stochastic optimization algorithm based on Gibbs sampling. The combination method is based on an approximate assignment score function which leads to a lower running time, and it also explores a greater portion of the space of assignments compared to other methods. The cross-entropy method does not rely on the approximation and achieves better tracking performance than the sequential method, albeit at a higher computational overhead. The impact of the approximate score function on the algorithms' performance is also studied via simulations and it is shown that the cross-entropy method consistently yields the best tracking performance whereas the combination method has the shortest runtime at high measurement noise level or high clutter rate.