Track-to-track Association based on Deterministic Sampling using Herding

Laura M. Wolf and Marcus Baum
Institute of Computer Science
University of Göttingen
Göttingen, Germany
Email: {laura.wolf, marcus.baum}@cs.uni-goettingen.de

***Abstract*—Multi-sensor multi-object tracking in a track-to-track fusion framework involves the grouping of tracks (from different sensors) that belong to the same perceived object. In particular for collective perception scenarios in large-scale traffic systems the number of sensors and objects can be huge, as a large number of vehicles can be equipped with multiple sensors. In order to cope with the intractable number of possible associations, recently a stochastic optimization approach for track-to-track association was proposed. The key idea is to successively improve an initial association by means of performing random modifications, i.e., actions, on the current association. In this work, we develop a novel deterministic version of the algorithm, which employs herding in order to deterministically choose the next action. Simulations demonstrate that the deterministic version of stochastic optimization provides comparable results to the stochastic version with a significantly lower variance.**

***Index Terms*—Multi-sensor fusion, track-to-track association, stochastic optimization, deterministic sampling, herding.**

I. Introduction

The detection and tracking of moving objects in the environment is typically performed with multiple sensors in order to increase the field of view or exploit heterogeneous sensor types. One important application is the collective perception of the traffic by multiple intelligent vehicles that share information about their local environments with each other or with infrastructure modules.

Data fusion can be performed on the measurement level, where all measurements are gathered before tracking is carried out, or on the track level, where tracks are generated for each sensor and then the tracks are combined. Fusing on the measurement level contains more information whereas track fusion requires less bandwidth and is more sensor independent. Here, we focus on track fusion, where tracking and potentially local fusion of data from multiple sensors are performed in each intelligent vehicle and the global fusion is carried out in a fusion center as opposed to sensor-to-sensor fusion [1], [2]. In a collective perception scenario the fusion center could be located in an infrastructure module such as a Road-Side Unit (RSU) [3], [4].

This work was funded by the Lower Saxony Ministry of Science and Culture under grant number ZN3493 within the Lower Saxony "Vorab" of the Volkswagen Foundation and supported by the Center for Digital Innovations (ZDIN).

Several approaches to Track-to-Track Fusion (T2TF) [5], [6] have been investigated, such as covariance intersection [7] or information matrix fusion [2], [8].

The Track-to-Track Association (T2TA) needs to take place before the fusion in order to determine which estimates should be fused together. Various association methods have been proposed, including greedy approaches [9], fuzzy sets [10] and GRASP [11], which combines a greedy randomized approach with local optimization. Furthermore, some methods reduce the multidimensional problem by iteratively solving twodimensional subproblems using optimization [12]–[14] or hypothesis testing [15]. Other approaches are based on clustering [16], [17]. In [18] clustering is employed to reduce the size of the association and is combined with an optimization approach.

In our previous work [19], we developed an approach for T2TA that employs Stochastic Optimization (SO) to find the best associations. The work was motivated by the SO-based approach [20] for multiple extended object tracking. Furthermore, we improved the greedy approach from [9] by adding a merging step. The SO-based approach [19] outperforms the greedy approach [9], [19] while remaining computationally efficient.

As autonomous driving is a safety-critical application, we are interested in reliable and repeatable deterministic approaches to T2TA. Herding [21] can be used to compute deterministic pseudo-samples that represent a given probability distribution. In many cases deterministic samples contain more information than random samples from the same distribution [22]. Herding has been applied to Gibbs sampling in [23], creating a deterministic Gibbs sampling approach. In our previous work we applied herded Gibbs sampling to the data association problem in multi-object tracking [24]. We could show that the deterministic Gibbs sampling converged faster than the random version. This suggests that deterministic sampling is a promising alternative to random sampling in data association. However, to the best of our knowledge, deterministic sampling has not been applied to track-to-track association yet.

A. Contribution

In this work, we develop a deterministic version of the stochastic optimization approach proposed in [19]. For this purpose, we apply herding [21] in order to deterministically choose the actions for creating feasible associations. Furthermore, we introduce a gating technique for herding, where we only consider the associations of clusters close to the current track for choosing the herding weight. This may improve the convergence especially in settings with many tracks.

We compare the random and herded stochastic optimization approaches with a Maximum Likelihood (ML) version and the greedy approach from [9], [19] on simulated data and compare the convergence behavior over the number of samples.

In Section II we define the problem setting. We revisit the stochastic optimization for T2TA in Section III and propose the herded stochastic optimization as well as the gated herded stochastic optimization in Section IV. Simulation results are presented in Section V, and the paper is concluded in Section VI.

II. Problem Description

In this work, we consider T2TA for multi-sensor fusion, where multiple objects are perceived and tracked by several sensors. The resulting tracks are fused in a central fusion center. Before a fusion is possible, corresponding tracks have to be associated, i.e., it has to be determined which tracks likely represent the same object. We do not incorporate prior knowledge. This has the advantage that our approach is also applicable if no prior knowledge is available, e.g, when new objects enter the area of interest. T2TA is a multidimensional assignment problem, which is NP-hard for more than two sensors [25].

As we receive tracks from each sensor and not measurements, we assume that there is at most one track per object from each sensor and no clutter tracks. Therefore, there can be at most one track per sensor in each cluster, i.e., the set of tracks that presumably belong to one object. We furthermore assume that the tracks are Gaussian distributed around the true objects' positions. For simplicity we assume that the detection probability p_D, i.e., the probability that a sensor has generated a track for a certain object, is constant and equal for all sensors. The number of objects is unknown. We denote the set of tracks as $\mathcal{T} = \{1, \ldots, N_\mathcal{T}\}$, where $N_\mathcal{T}$ is the number of tracks. The tracks are arbitrarily numbered from 1 to $N_\mathcal{T}$ and the corresponding states are given by $\mathbf{x}_1, \ldots, \mathbf{x}_{N_\mathcal{T}}$. The set of sensors is denoted as $\mathcal{S}$ and the number of sensors as $N_\mathcal{S} = |\mathcal{S}|$. The operator $s(t)$ returns the sensor that generated track $t \in \mathcal{T}$ and $S(\mathbf{C}_c) = \{s(t) | t \in \mathbf{C}_c\}$ returns the sensors that generated the tracks in cluster $\mathbf{C}_c$, where $\mathbf{C}_c \subseteq \mathcal{T}$.

We represent a clustering of tracks via a joint association variable $\boldsymbol{\theta}$, which maps each track to a cluster. All clusters are numbered, w.l.o.g. by positive integers:

$$\boldsymbol{\theta} : \mathcal{T} \mapsto \mathbb{N}^{N_\mathcal{T}}, \boldsymbol{\theta} = [\theta_1, \ldots, \theta_{N_\mathcal{T}}] \ , \tag{1}$$

where track t is mapped to cluster $\mathbf{C}_{\theta_t}$. The set of clusters $\mathcal{C} = \{c \mid \theta_t = c, t \in \mathcal{T}\}$ maintains the current cluster indices. A cluster is defined as $\mathbf{C}_c = \{t \mid \theta_t = c\}$, such that for all $t \in \mathcal{T}$ we have $t \in \mathbf{C}_c \iff \theta_t = c$. The association variable, however, is ambiguous, i.e., there are infinitely many association variables that represent the same clustering. In a valid cluster, all tracks have to originate from different sensors, i.e., $|S(\mathbf{C}_c)| = |\mathbf{C}_c|$, $c \in \mathcal{C}$.

III. Stochastic optimization for T2TA

The main principle of our stochastic optimization approach is to iterate through the tracks and for each track we sample and perform an action, yielding a new joint association. In the end, the joint association with the highest likelihood can be used or, alternatively, several associations can be used in a multi-hypothesis approach.

In [19], we employed a likelihood for a clustering depending on the assumed detection probability. The likelihood of a cluster consists of a likelihood of its size depending on the detection probability as well as a Gaussian spatial likelihood, where we assume that the tracks are Gaussian distributed around the true object position with a constant covariance Σ. A discussion of other likelihood functions for T2TA can be found in [26]. Our likelihood function is defined as follows

$$w_{\mathbf{C}_c} = p_D^{|\mathbf{C}_c|}(1-p_D)^{N_\mathcal{S}-|\mathbf{C}_c|} \cdot \prod_{t \in \mathbf{C}_c} \mathcal{N}\left(\mathbf{x}_t; \frac{1}{|\mathbf{C}_c|}\sum_{t \in \mathbf{C}_c} \mathbf{x}_t, \left(1 + \frac{1}{|\mathbf{C}_c|}\right)\Sigma\right) \ . \tag{2}$$

As we do not know the true object position and use no prior information, we use the sample mean. It has a covariance of $\frac{1}{|\mathbf{C}_c|}\Sigma$ and hence the covariance of the spatial likelihood is $\left(1 + \frac{1}{|\mathbf{C}_c|}\right)\Sigma$. Σ would have to be estimated in a real world setting.

The likelihood of a joint association is then given by

$$w_{\boldsymbol{\theta}} = \prod_{c \in \{c' | \theta_t = c', t \in \mathcal{T}\}} w_{\mathbf{C}_c} \ . \tag{3}$$

In order to circumvent numerical problems, the logarithm of the likelihoods can be considered.

We consider four possible actions: (i) remain in the current cluster, i.e., the current association does not change (r); (ii) split the current track into a new singleton cluster (s); (iii) move the current track to another cluster (m) and (iv) merge the current cluster with another cluster (M). This yields the following possible actions

$$\mathbf{A} = [r, s, m_1, \ldots, m_{|\mathcal{C}|}, M_1, \ldots, M_{|\mathcal{C}|}] \ , \tag{4}$$

with the corresponding probabilities

$$\mathbf{p}_\mathbf{A} \propto [p^r, p^s, p_1^m, \ldots, p_{|\mathcal{C}|}^m, p_1^M, \ldots, p_{|\mathcal{C}|}^M] \ . \tag{5}$$

The likelihood p^a of action a is computed as the ratio of likelihoods $\frac{w_{\boldsymbol{\theta}^{(a)}}}{w_{\boldsymbol{\theta}}}$, where $\boldsymbol{\theta}$ is the current joint association and $\boldsymbol{\theta}^{(a)}$ the resulting joint association after performing action a. This yields the following likelihood for remaining in the current cluster

$$p^r = 1 \ , \tag{6}$$

as $\boldsymbol{\theta}$ does not change performing this action.

Algorithm 1 Stochastic optimization algorithm for T2TA

function SO($\mathcal{T}$, $\mathcal{S}$, N, p_D)
  $\boldsymbol{\theta} \leftarrow [1, \ldots, N_{\mathcal{T}}]$
  $k \leftarrow N_{\mathcal{T}} + 1$
  for $n = 1, \ldots, N$ **do**
    for $t = 1, \ldots, N_{\mathcal{T}}$ **do**
      calculate $\mathbf{p}_{\mathbf{A}}$ and $\mathbf{A}$ according to (5) and (4)
      normalize $\mathbf{p}_{\mathbf{A}}$
      $a \leftarrow \text{randChoice}(\mathbf{A}, \mathbf{p}_{\mathbf{A}})$
      performAction$(a, \mathcal{C})$
      $\boldsymbol{\theta}^{(n,t)} \leftarrow \boldsymbol{\theta}$
    end for
  end for
  return $\boldsymbol{\theta}^{(1,1)} \ldots, \boldsymbol{\theta}^{(N,N_{\mathcal{T}})}$
end function

Algorithm 2 perform action

function PERFORMACTION(a, $\mathcal{C}$)
  if $a = s$ **then**
    $\theta_t \leftarrow k$
    $k \leftarrow k + 1$
  else if $a = m_c, c \in \mathcal{C}$ **then**
    $\theta_t \leftarrow c$
  else if $a = M_c, c \in \mathcal{C}$ **then**
    $\theta_{t'} \leftarrow c, t' \in \mathbf{C}_{\theta_t}$
  end if
end function

The likelihood of creating a singleton is given by

$$p^s = \begin{cases} 0 & \text{if } |\mathbf{C}_{\theta_t}| = 1, \\ \dfrac{w_{\{t\}} w_{\mathbf{C}_{\theta_t} \setminus \{t\}}}{w_{\mathbf{C}_{\theta_t}}} & \text{else.} \end{cases} \tag{7}$$

The likelihood of creating a singleton cluster is 0, if it already is a singleton. In that case the action is equivalent to remaining in the current cluster.

The likelihood of moving the current track to cluster $\mathbf{C}_c$ is given by

$$p_c^m = \begin{cases} \dfrac{w_{\mathbf{C}_c \cup \{t\}} w_{\mathbf{C}_{\theta_t} \setminus \{t\}}}{w_{\mathbf{C}_c} w_{\mathbf{C}_{\theta_t}}} & \text{if } S(\mathbf{C}_c) \cap \{s(t)\} = \emptyset \wedge c \neq \theta_t \ , \\ 0 & \text{else} \ . \end{cases} \tag{8}$$

We do not consider moving the current track to its current cluster or merging with its current cluster, as this would be equivalent to the remain action. Furthermore all tracks in the resulting cluster have to stem from different sensors, which is ensured by $S(\mathbf{C}_c) \cap \{s(t)\} = \emptyset$.

Finally, the likelihood of merging the current cluster with cluster $\mathbf{C}_c$ is given by

$$p_c^M = \begin{cases} \dfrac{w_{\mathbf{C}_c \cup \mathbf{C}_{\theta_t}}}{w_{\mathbf{C}_c} w_{\mathbf{C}_{\theta_t}}} & \text{if } S(\mathbf{C}_c) \cap S(\mathbf{C}_{\theta_t}) = \emptyset \wedge c \neq \theta_t \ , \\ 0 & \text{else} \ . \end{cases} \tag{9}$$

To speed up computation, it is possible to add a gating step. Then, only those clusters are considered for the moving and merging actions, that are within a certain gating distance g of the current track

$$\|\overline{\mathbf{x}}^c - \mathbf{x}_t\| \leq g \ , \tag{10}$$

where $\overline{\mathbf{x}}^c$ is the center of the cluster $\mathbf{C}_c$.

After sampling an action, the joint association variable $\boldsymbol{\theta}$ is updated (Alg. 2). The complete SO algorithm can be found in Alg. 1. It creates a new joint association after each action, in total $N \cdot N_{\mathcal{T}}$ associations, where N is the number of sweeps through all tracks.

IV. Herded Stochastic Optimization

In order to create a deterministic T2TA algorithm that retains the same advantages of the random stochastic optimization approach, we apply herding to it. Herding [21] is a deterministic procedure that can be used to create deterministic pseudosamples of a given discrete probability distribution. It has been applied to Gibbs sampling [23] as well as data association in multi-object tracking [24].

The key aspect of the herding procedure is a weight vector, which contains a weight for every possible sample and accumulates information on previous samples. To compute a new sample from the probability distribution, the weight vector is maximized, yielding the new sample. The weight vector is then updated using the indicator vector of the generated sample.

The i-th sample $x^{(i)}$ and weight vector $\mathbf{w}^{(i)}$ are computed as follows:

$$x^{(i)} = \arg\max_{x \in \mathcal{X}} \langle \mathbf{w}^{(i-1)}, \boldsymbol{\phi}(x) \rangle \ , \tag{11}$$

$$\mathbf{w}^{(i)} = \mathbf{w}^{(i-1)} + \boldsymbol{p} - \boldsymbol{\phi}(x^{(i)}) \ , \tag{12}$$

where $\mathcal{X}$ is the sample space, $\mathbf{p}$ is the probability density and $\boldsymbol{\phi}(x)$ the indicator vector, which is 1 at the position indicated by x and 0 otherwise. The weight vector has the same dimension as $\mathbf{p}$ and can be initialized with it, i.e., $\mathbf{w}^{(0)} = \mathbf{p}$. In that case the first sample is always the maximum likelihood result.

When replacing the random sampling in Alg. 1 by herded sampling, we need multiple herding processes, as we sample from different densities ($\mathbf{p}_{\mathbf{A}}$) depending on the current association ($\boldsymbol{\theta}$) of the tracks. Each $\boldsymbol{\theta}$ in combination with the current track t defines the corresponding $\mathbf{p}_{\mathbf{A}}$. However, one clustering can be represented by different joint association variables, as the numbering is arbitrary. Therefore, we introduce the operator $\mathcal{U}$ that numbers the clusters in ascending order with respect to the order they first appear in $\boldsymbol{\theta}$. This yields an unambiguous cluster numbering depending on the track order.

In the following example, $\boldsymbol{\theta}^{(1)}$ and $\boldsymbol{\theta}^{(2)}$ are distinct but represent the same underlying clustering, as we have $\mathcal{U}(\boldsymbol{\theta}^{(1)}) = \mathcal{U}(\boldsymbol{\theta}^{(2)})$.

$$\boldsymbol{\theta}^{(1)} = [3, 7, 3, 1, 1, 3] \tag{13}$$

$$\boldsymbol{\theta}^{(2)} = [5, 2, 5, 4, 4, 5] \tag{14}$$

$$\mathcal{U}(\boldsymbol{\theta}^{(1)}) = \mathcal{U}(\boldsymbol{\theta}^{(2)}) = [1, 2, 1, 3, 3, 1] \tag{15}$$

The weight vectors $\mathbf{w}$ and probability densities $\mathbf{p}_{\mathbf{A}}$ of each herding process are stored in a hash table H after first use.

Algorithm 3 Herded Stochastic Optimization-based Algorithm for T2TA

1: **function** HERDEDSO($\mathcal{T}$, $\mathcal{S}$, N, p_D)
2: $\quad \boldsymbol{\theta} \leftarrow [1, \ldots, N_\mathcal{T}]$
3: $\quad k \leftarrow N_\mathcal{T} + 1$
4: $\quad$ initialize hash table H
5: $\quad$ **for** $n = 1, \ldots, N$ **do**
6: $\quad\quad$ **for** $t = 1, \ldots, N_\mathcal{T}$ **do**
7: $\quad\quad\quad \boldsymbol{\theta} \leftarrow \mathcal{U}(\boldsymbol{\theta})$
8: $\quad\quad\quad i_{t,\boldsymbol{\theta}} \leftarrow [t, \theta_1, \ldots, \theta_{N_\mathcal{T}}]$
9: $\quad\quad\quad$ **if** $i_{t,\boldsymbol{\theta}}$ in H **then**
10: $\quad\quad\quad\quad \mathbf{w}, \mathbf{p_A} \leftarrow H[i_{t,\boldsymbol{\theta}}]$
11: $\quad\quad\quad$ **else**
12: $\quad\quad\quad\quad$ calculate $\mathbf{p_A}$ and $\mathbf{A}$ according to (5),(4)
13: $\quad\quad\quad\quad$ normalize $\mathbf{p_A}$
14: $\quad\quad\quad\quad \mathbf{w} \leftarrow \mathbf{p_A}$
15: $\quad\quad\quad$ **end if**
16: $\quad\quad\quad a \leftarrow \arg\max_{a' \in \mathbf{A}} \langle \mathbf{w}, \boldsymbol{\phi}(a') \rangle$
17: $\quad\quad\quad \mathbf{w} \leftarrow \mathbf{w} + \mathbf{p_A} - \boldsymbol{\phi}(a)$
18: $\quad\quad\quad$ update $\mathbf{w}$ in $H[i_{t,\boldsymbol{\theta}}]$
19: $\quad\quad\quad$ perform_action($a, \mathcal{C}$)
20: $\quad\quad\quad \boldsymbol{\theta}^{(n,t)} \leftarrow \boldsymbol{\theta}$
21: $\quad\quad$ **end for**
22: $\quad$ **end for**
23: $\quad$ **return** $\boldsymbol{\theta}^{(1,1)} \ldots, \boldsymbol{\theta}^{(N,N_\mathcal{T})}$
24: **end function**

This allows for saving storage space, as there are intractable many possible clusterings and most of them are unlikely or unfeasible. After an action is sampled by maximizing the weight vector, $\mathbf{w}$ is updated and stored in the hash table H again. The full algorithm can be found in Alg. 3.

The herded SO approach has the same theoretical complexity as the original SO approach, namely $\mathcal{O}(N{N_\mathcal{T}}^2)$ for generating $N N_\mathcal{T}$ possible joint associations [19]. The main difference lies in the computation of the sample, which is $\mathcal{O}(N_\mathcal{T})$ for random sampling and the herding procedure [24]. Storing the weight vectors in a hash table can be done in $\mathcal{O}(1)$.

A. Gated Herding

In scenarios with many objects or many sensors, there are many tracks and therefore many different $\mathbf{p_A}$ to sample from. If a different clustering is present every time an action is sampled, the weight vectors are never reused. In this case, the advantages of herding do not come to bear as the first sampled action is always the one with the highest likelihood. However, the association of far away tracks is most likely not relevant to the current track, especially when using the gating technique described in Sec. III.

The idea of gated herding is to ignore all clusters in $\boldsymbol{\theta}$ that are beyond the gating threshold for determining which weight vector is used in order to have a higher reuse of probability densities. The filtered joint association variable $\widetilde{\boldsymbol{\theta}}^t = [\widetilde{\theta}^t_1, \ldots, \widetilde{\theta}^t_{N_\mathcal{T}}]$ only includes the local clustering around the current track t and masks out the rest. $\widetilde{\theta}^t_1$ is defined as follows for all $t' \in \mathcal{T}$

$$\widetilde{\theta}^t_{t'} = \begin{cases} \theta_{t'} & \text{, if } \left\| \overline{\mathbf{x}}^{\theta_{t'}} - \mathbf{x}_t \right\| \leq g \ , \\ -1 & \text{, else } . \end{cases} \tag{16}$$

An example of gating can be seen in Fig. 1. The grayed out tracks are not relevant for the association of the current track and therefore do not influence the herding process. In contrast, in the original herding, there would be a separate herding process for each association of the grayed out tracks, even though the likelihoods of the relevant clusters are the same.

Then, in line 7 and 8 of Alg. 3 $\widetilde{\boldsymbol{\theta}}^t$ is used to identify $\mathbf{p_A}$, although the updates to the clusters are performed on $\boldsymbol{\theta}$ in Alg. 2. As the clusters are renamed using the $\mathcal{U}$ mapping, it is necessary to apply the actions to the original clusters, determined by the inverse $\mathcal{U}$ mapping. A full implementation can be found in our GitHub repository[1].

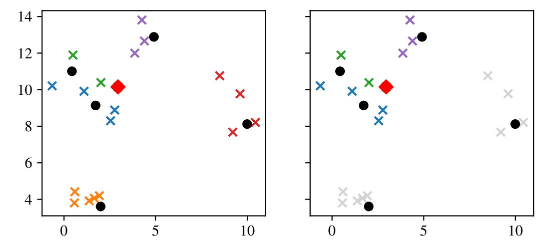


Fig. 1. Gating example. On the left is the full joint association and on the right the clusters that are beyond the gating threshold are grayed out. The current track t is marked as a red diamond.

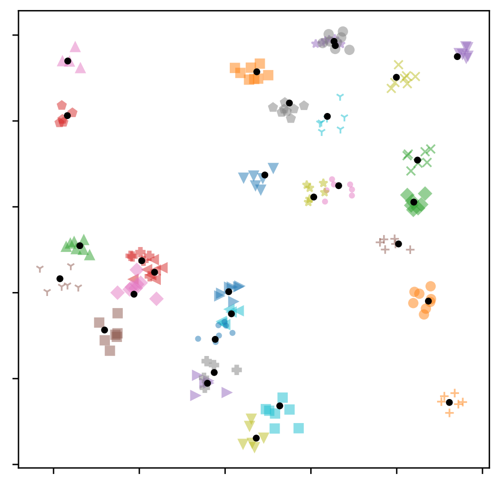
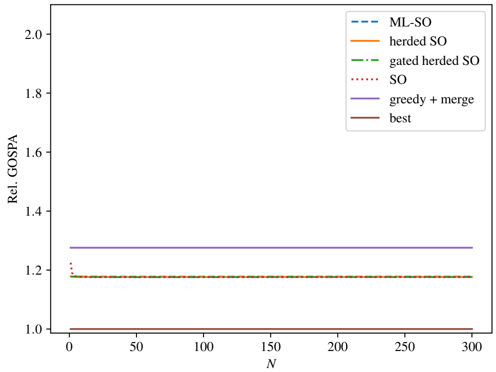


Fig. 2. Example scenario with 30 objects, 8 sensors and $p_D = 0.8$. The marker shape and color indicate the ground truth clustering and black dots the true object positions.

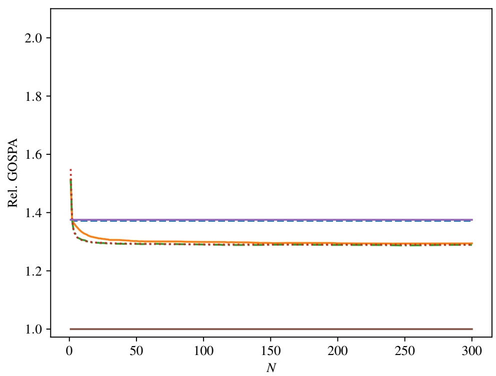
V. EVALUATION

We evaluated the algorithms on simulated data. The scenarios were randomly created with 30 objects in an area of

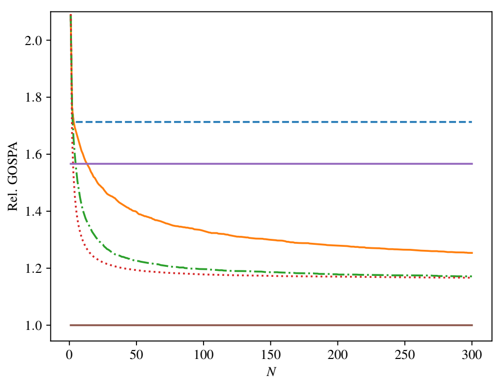
[1]https://github.com/Fusion-Goettingen



(a) $p_D = 0.2$



(b) $p_D = 0.5$



(c) $p_D = 0.8$

Fig. 3. Relative GOSPA error over the number of sweeps for different detection probabilities. In (a) all SO variants overlap, while in (b) the greedy appraoch an ML-SO overlap, as well as SO and gated herded SO.

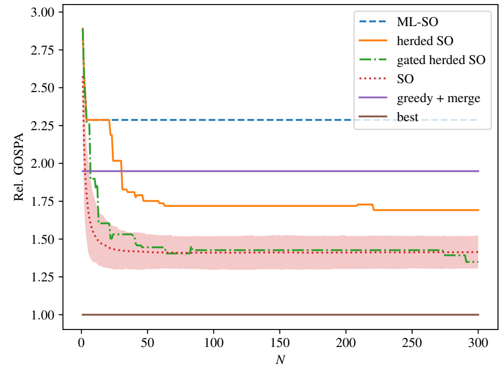


Fig. 4. Relative GOSPA error for one specific scenario of objects and tracks and $p_D = 0.8$. The 90 % confidence interval of 1000 Monte Carlo runs are shaded.

$100\,\mathrm{m} \times 100\,\mathrm{m}$. We used 8 sensors that generated Gaussian distributed tracks around the true objects' positions with a covariance of $\Sigma = 2^2 \cdot \mathbf{I}_2$. We applied the 3σ bound of Σ as gating threshold g. An example scenario can be found in Fig. 2.

Besides the random, herded and gated herded SO algorithms, we also evaluated a deterministic variant of SO, where we deterministically select the action with the highest likelihood each time, which we call Maximum Likelihood SO (ML-SO). Furthermore, we used the greedy approach with merging that we presented in [19] as a deterministic approach not based on sampling.

To evaluate the algorithms, we computed the cluster centers of each cluster as fused estimates. We then computed the GOSPA [27] error on the fused estimates and the true positions. In order to compare different scenarios, we evaluated the relative GOSPA error, i.e., the GOSPA error divided by the GOSPA error of the best association. It has to be noted though that the SO does not optimize the GOSPA directly and the ground truth association does not necessarily have the highest likelihood or GOSPA error.

Fig. 3 shows the relative GOSPA error for all methods over the number of sweeps, each sweep considering the association with the highest likelihood so far. The results are based on 1000 Monte Carlo runs and for each set of objects and tracks 25 Monte Carlo runs for the SO.

It can be seen in Fig. 3 that all SO approaches outperform the greedy approach, except for the ML-SO, which only yields comparable results for $p_D = 0.5$ and worse results for high detection probabilities ($p_D = 0.8$).

For low detection probabilities (Fig. 3a), the SO algorithms do not differ. For $p_D = 0.5$ (Fig. 3b), the ML-SO yields results comparable to the greedy approach, herded SO does not quite converge to the results of SO and the gated herded SO yields the same results as SO. For high detection probabilities (Fig. 3c), the ML-SO is even worse than the greedy approach. The gated herded SO converges a bit slower than SO, but still converges to the same results as opposed to the herded SO. This shows that high detection probabilities with a high number of tracks pose a challenge to the deterministic approaches, though the gated herded SO still shows a comparable convergence behavior to SO.

The results of ML-SO do not change after a couple of sweeps, as the ML approach gets stuck in a local maximum, and this is not sufficient for higher detection probabilities. This illustrates that the herded variants reuse the weight vectors as otherwise the results would be the same as with the ML approach. Furthermore, especially with high detection probabilities, the gated herding converges faster, pointing towards a higher reuse of weight vectors when only local clusters contribute to the herding process.

One advantage of deterministic algorithms is that they only have one possible outcome, whereas the result of random algorithms can vary. We computed the relative GOSPA errors on one scenario, but computed 1000 Monte Carlo runs for the SO. The results can be found in Fig. 4, where the shaded red area shows the span between the $5\,\%$ and the $95\,\%$ quantile. Even though the SO yields a stable mean, the results vary. The comparison with the reliable results from the deterministic algorithms underlines the advantages of the latter for safety-critical applications.

VI. Conclusion and Future Work

In this paper, we proposed a novel deterministic data association approach for T2TA by applying herding to the previously presented stochastic optimization approach [19]. The herded stochastic optimization approach did not converge as fast as the random one for higher detection probabilities. The gated herding version, which only considers the local surroundings for the herding process, however, drastically improves the convergence and yielded comparable results to the random version. Both herding variants yielded better results than the other two deterministic options, the maximum likelihood stochastic optimization and the greedy approach.

We could thus illustrate that herded SO, especially the gated variant, can be a competitive alternative in terms of sampling quality while providing the advantage of reliability in safety-critical applications.

Improving the convergence behavior of the herded variants is a promising field for future research. We will further test the association approaches on more realistic scenarios from advanced simulators and investigate how to estimate the detection probability and spatial likelihood from tracking data. We further plan to evaluate different cluster likelihoods such as the ones in [26], [28].

References

[1] Chee-Yee Chong, S. Mori, W. Barker, and Kuo-Chu Chang, "Architectures and algorithms for track association and fusion," *IEEE Aerospace and Electronic Systems Magazine*, vol. 15, no. 1, pp. 5–13, Jan. 2000.

[2] M. Aeberhard, S. Schlichtharle, N. Kaempchen, and T. Bertram, "Track-to-track fusion with asynchronous sensors using information matrix fusion for surround environment perception," *IEEE Transactions on Intelligent Transportation Systems*, vol. 13, no. 4, pp. 1717–1726, Dec. 2012.

[3] C. Allig and G. Wanielik, "Dynamic dissemination method for collective perception," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*. Auckland, New Zealand: IEEE, Oct. 2019, pp. 3756–3762.

[4] A. Chtourou, P. Merdrignac, and O. Shagdar, "Collective perception service for connected vehicles and roadside infrastructure," in *2021 IEEE 93rd Vehicular Technology Conference (VTC2021-Spring)*, Apr. 2021, pp. 1–5.

[5] Y. Bar-Shalom, "On the track-to-track correlation problem," *IEEE Transactions on Automatic Control*, vol. 26, no. 2, pp. 571–572, Apr. 1981.

[6] Y. Bar-Shalom and H. Chen, "Multisensor track-to-track association for tracks with dependent errors," in *2004 43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 3, Dec. 2004, pp. 2674–2679.

[7] D. Fränken and A. Hüpper, "Improved fast covariance intersection for distributed data fusion," in *2005 7th International Conference on Information Fusion*. Philadelphia, PA, USA: IEEE, 2005, p. 7 pp.

[8] K. Chang, T. Zhi, and R. Saha, "Performance evaluation of track fusion with information matrix filter," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 38, no. 2, pp. 455–466, Apr. 2002.

[9] A. Houenou, P. Bonnifait, V. Cherfaoui, and J. Boissou, "A track-to-track association method for automotive perception systems," in *2012 IEEE Intelligent Vehicles Symposium*, Jun. 2012, pp. 704–710.

[10] A. M. Aziz, "Fuzzy track-to-track association and track fusion approach in distributed multisensor–multitarget multiple-attribute environment," *Signal Processing*, vol. 87, no. 6, pp. 1474–1492, Jun. 2007.

[11] A. J. Robertson, "A set of greedy randomized adaptive local search procedure (GRASP) implementations for the multidimensional assignment problem," *Computational Optimization and Applications*, vol. 19, no. 2, pp. 145–164, Jul. 2001.

[12] S. Deb, M. Yeddanapudi, K. Pattipati, and Y. Bar-Shalom, "A generalized S-D assignment algorithm for multisensor-multitarget state estimation," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 2, pp. 523–538, Apr. 1997.

[13] R. Popp, K. Pattipati, and Y. Bar-Shalom, "M-best S-D assignment algorithm with application to multitarget tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 1, pp. 22–39, Jan. 2001.

[14] K.-H. Lee, Y. Kanzawa, M. Derry, and M. R. James, "Multi-target track-to-track fusion based on permutation matrix track association," in *2018 IEEE Intelligent Vehicles Symposium (IV)*, Jun. 2018, pp. 465–470.

[15] Y. Zhang, H. Leung, M. Blanchette, T. Lo, and J. Litva, "An efficient decentralized multiradar multitarget tracker for air surveillance," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 4, pp. 1357–1363, Oct. 1997.

[16] T. Li, F. De la Prieta Pintado, J. M. Corchado, and J. Bajo, "Multi-source homogeneous data clustering for multi-target detection from cluttered background with misdetection," *Applied Soft Computing*, vol. 60, pp. 436–446, Nov. 2017.

[17] A. L. Bereson and R. N. Lobbia, "Efficient track-to-task assignment using cluster analysis," in *2006 9th International Conference on Information Fusion*, Jul. 2006, pp. 1–8.

[18] M. Chummun, T. Kirubarajan, K. Pattipati, and Y. Bar-Shalom, "Fast data association using multidimensional assignment with clustering," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 898–913, Jul. 2001.

[19] L. M. Wolf, S. Steuernagel, K. Thormann, and M. Baum, "Track-to-track association based on stochastic optimization," in *2023 26th International Conference on Information Fusion (FUSION)*, Jun. 2023, pp. 1–7.

[20] K. Granström, L. Svensson, S. Reuter, Y. Xia, and M. Fatemi, "Likelihood-based data association for extended object tracking using sampling methods," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 1, pp. 30–45, Mar. 2018.

[21] M. Welling, "Herding dynamical weights to learn," in *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. Montreal, Quebec, Canada: ACM Press, 2009, pp. 1–8.

[22] Y. Chen, M. Welling, and A. Smola, "Super-samples from kernel herding," in *Proceedings of the 26th Conference on Uncertainty in Artificial Intelligence (UAI 2010)*. AUAI Press, 2010, pp. 109–116.

[23] Y. Chen, L. Bornn, N. de Freitas, M. Eskelin, J. Fang, and M. Welling, "Herded Gibbs sampling," *Journal of Machine Learning Research*, vol. 17, no. 10, pp. 1–29, 2016.

[24] L. M. Wolf and M. Baum, "Deterministic gibbs sampling for data association in multi-object tracking," in *2020 IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, Sep. 2020, pp. 291–296.

[25] M. R. Garey and D. S. Johnson, *Computers and Intractability; A Guide to the Theory of NP-Completeness*. USA: W. H. Freeman & Co., 1990.

[26] L. M. Kaplan, Y. Bar-Shalom, and W. D. Blair, "Assignment costs for multiple sensor track-to-track association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 44, no. 2, pp. 655–677, Apr. 2008.

[27] A. S. Rahmathullah, Á. F. García-Fernández, and L. Svensson, "Generalized optimal sub-pattern assignment metric," in *2017 20th International Conference on Information Fusion (Fusion)*, Jul. 2017, pp. 1–8.

[28] T. L. Ogle and P. Willett, "An alternative derivation of generalized likelihood tests for track-to-track correlation," in *2019 22th International Conference on Information Fusion (FUSION)*, Jul. 2019, pp. 1–7.