Hypothesis Exploration in Multiple Hypothesis Tracking with Multiple Clusters

1st Edmund Førland Brekke
Dept. of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
edmund.brekke@ntnu.no

2nd Lars-Christian Ness Tokle
Dept. of Engineering Cybernetics
Norwegian University of Science and Technology
Trondheim, Norway
lars-christian.n.tokle@ntnu.no

***Abstract*—Finding the most probable posterior hypotheses is a core task in hypothesis-oriented multiple hypothesis tracking (HO-MHT), and also in related tracking methods such as the Poisson multi-Bernoulli mixture (PMBM) filter. The traditional approach is to find the M best new hypotheses for each parent hypothesis by means of Murty's algorithm. In this paper we instead present an algorithm for finding the M best hypotheses ranging over all parent hypotheses. The algorithm is developed in the more general context of cluster management, where the goal is to merge several parent clusters, and to find the M best posterior hypotheses in any such supercluster.**

***Index Terms*—data association, multiple hypothesis tracking, cluster management, Poisson multi-Bernoulli mixture filter, M-best assignment**

I. Introduction

In several approaches to multi-target tracking, data association is addressed by discovering good association hypotheses and calculating their probabilities. In some approaches, often described as track-oriented, the goal is to find the one best hypothesis that describes the assignments between tracks and measurements over several time steps. In other approaches, often described as hypothesis-oriented, the goal is to update a collection of association hypotheses recursively every time a new set of measurements is received. The focus in this paper is on the hypothesis-oriented approach, which underlies the original formulation of both the multiple hypothesis tracker (MHT) [1] and the Poisson multi-Bernoulli mixture (PMBM) filter [2], which essentially is a generalization of the MHT [3].

A common technique for transforming a prior collection of hypotheses into a posterior collection of hypotheses is to use Murty's method [4] for discovering the M best posterior hypotheses that are children of a given prior hypothesis. By performing this process for each prior hypothesis, the collection of posterior hypotheses is obtained.

This process suffers from a couple of inefficiencies. First, what ultimately is most useful is the M best hypotheses among *all* the posterior hypotheses, rather than the M best children hypotheses of *each* parent hypothesis. To generate a fixed number of children hypotheses for each parent hypothesis is therefore inherently wasteful. Second, in a practical tracking system it is desirable to decompose the tracking problem into spatially independent clusters. Inevitably, situations will occur when clusters have to be merged. In this merging, it becomes necessary to construct the combinations of prior hypotheses from the different clusters. The exploration of posterior hypotheses must then be done conditional on each such combination. This can easily turn into a computational bottleneck.

As a solution to these challenges, this paper proposes a branch-and-bound method for hypothesis exploration that works by considering all prior hypotheses simultaneously, and switching from one prior hypothesis to another whenever there is reason to do so. The method is a generalization of Murty's method. It uses key steps of Murty's method in a cascaded manner, both for finding good prior hypotheses and for finding good track-to-measurement assignments.

The paper is organized as follows. In Section II we revisit related work in hypothesis exploration. In Section III we describe the context of cluster management in colloquial terms, before a formal problem statement follows in Section IV. In Section V we take a look at Murty's method to build the foundations for the proposed method. Section VI contains the main contributions of the paper, where the proposed branch-and-bound method is described. A simple worked example is presented in Section VII. More comprehensive simulation results are provided in Section VIII-B. The conclusion follows in Section IX.

II. Related works

By the term *hypothesis exploration* we understand the task of generating a set of posterior association hypotheses, which account for all sufficiently probable and mutually compatible combinations of tracks in the prior hypotheses with measurements in the current measurement scan. This has been a core research area in multi-target tracking since the MHT was invented. Many implementations of MHT and related methods are based on the approach proposed in [5]. Here, Murty's ranked assignment method is used to generate the M best children hypotheses for every parent hypothesis. This involves the solution of 2D assignment problems, i.e., between tracks and measurements, to find the best hypothesis, then the best hypotheses when each track-to-measurement assignment of the

This work was supported in part by the Research Council of Norway through Projects 223254, 331921 and 309230, and by NTNU through the Autoferry Project.

best hypothesis is forbidden, and so on. Popular techniques for the 2D assignment problem include the auction method [6], the Hungarian method [7] and the Jonker-Volgenant (JV) algorithm [8].

An accelerated variant of Murty-based exploration was proposed in [9]. Here, dual variables from the JV algorithm are used to determine lower bounds which are used in two ways. First, the bounds are used to sort the problems in in the priority queue, so that the most promising problems are solved first, hopefully leading to automatic exclusion of less promising problems. Second, the bounds are used as a heuristic for the order in which parent tracks are processed. This can be seen as a greedy search, with the aim of making the 2D assignment problems solved by the JV algorithm as small as possible.

The formulations in [4] and [9] assumed square reward matrices. In [10], the JV algorithm and Murty's method were adapted to rectangular reward matrices, which are more natural to work with and avoid unnecessary dummy elements.

In the PMBM implementation of [11] the number of hypotheses generated per parent hypothesis is given by the ceiling of M times the prior probability of the parent. In practice, this means that only a few parents get multiple offspring, while most parents are only allowed a single child. The same strategy was used for the Generalized Labeled Multi-Bernoulli (GLMB) filter in [12]. Later GLMB papers, such as [13], have used Gibbs sampling for hypothesis expansion. In this approach, one attempts to draw the associations from their true distribution.

Murty's method has been generalized to a method for finding the M best solutions to a general $0-1$ integer linear programming (ILP) problem in [14].

III. Context

The underlying problem that motivates this work is the measurement update in the PMBM filter. In this setting, there exists a prior multi-target density [15], also known as set density, which can be factorized into a Poisson component and multi-Bernoulli mixture (MBM) component. The latter component can be seen as a weighed mixture over association hypotheses, where the mixture weights play the role of hypothesis probabilities. Conditional on each hypothesis, there is a multi-Bernoulli (MB) multi-target density, which again can be seen as the multi-target density of a union of several Bernoulli random finite sets. A Bernoulli random finite set models a target which may or may not exist with a given probability, and is parameterized by this probability and a kinematic pdf conditional on target existence. In this paper we will refer to the Bernoulli components as tracks.

It was shown in [2] that the PMBM form remains closed under the Bayes update. In other words, the multi-target density that results from performing a Bayes update using a set of measurements is also a PMBM density. This is often described as a conjugate prior property in the tracking literature. However, the number of hypotheses in the MBM density increases exponentially. Therefore, M best exploration techniques are necessary for a practical implementation of a PMBM filter.

A central premise of the present paper is that the MBM component can be further factorized into independent clusters. The MBM density restricted to a particular cluster is then a marginalization over the other clusters, and will also be an MBM density, which has a simpler structure with fewer subcomponents than the global MBM density.

In practice, independence is enforced through gating: Tracks can only be updated with measurements within a limited validation gate centered around the track. Then, independence between clusters breaks down whenever tracks in two different clusters gate the same measurement. In such a situation, the clusters involved must be merged into a new supercluster as part of the measurement update. The main objective of this paper is to propose an efficient solution for this task.

IV. Problem statement

A cluster is a collection of hypotheses, which again is a collection of pointers to tracks, a.k.a. Bernoulli components. Any track is uniquely given by a temporal sequence of measurements, possibly including dummy measurements. The collection of prior clusters is $\{\overline{c}_1, \ldots, \overline{c}_{\overline{n}_C}\}$. The prior cluster $\overline{c}_i$ contains $\overline{n}_{Hi}$ prior hypotheses $\{\overline{\theta}_{1(i)}, \ldots, \overline{\theta}_{\overline{n}_{Hi}(i)}\}$. The prior hypothesis $\overline{\theta}_l$ consists of $n_{\overline{\theta}_l}$ track pointers $\{\overline{\theta}_l^1, \ldots \overline{\theta}_l^{n_{\overline{\theta}_i}}\}$, which points to tracks at time $k-1$. We say "track pointers" and not "tracks" because the same track can be present in several hypotheses. A track can be present in at most one cluster.

Let the number of prior tracks be n_{k-1} and let the number of measurements be m_k. We define a dummy measurement for each of the prior tracks, representing missed detection. The number of elements in this extended measurement set is thus $\tilde{m}_k = m_k + n_{k-1}$. The potential matches between prior tracks and measurements are accounted for by a reward matrix $G \in \mathbb{R}^{n_{k-1} \times \tilde{m}}$. The element g^{tj} in G is a finite number if track t gates measurement j. Otherwise it is $-\infty$. In the measurement update, posterior tracks, hypotheses and clusters are to be constructed based on the $\tilde{m}_k = m_k + n_{k-1}$ real and dummy measurements, as well as the prior tracks, hypotheses and clusters.

The collection of superclusters is a partitioning of the prior cluster set under the following equivalence relation: Two prior clusters belong to the same supercluster if they contain tracks that share one or more measurements. We denote supercluster number i by σ_i. Formally it is a set of integers in $\{1, \ldots, \overline{n}_C\}$ pointing to the prior clusters belonging to σ_i.

For each supercluster, there is a corresponding posterior cluster. The collection of posterior clusters is $\{c_1, \ldots, c_{n_C}\}$. The posterior cluster c_i contains n_{Hi} posterior hypotheses $\{\theta_{1(i)}, \ldots, \theta_{n_{Hi}(i)}\}$. The posterior hypothesis θ_l consists of $n_{l\theta}$ track pointers $\{\theta_l^1, \ldots \theta_l^{n_{i\theta}}\}$, which point to tracks at time k. The score of a posterior hypothesis is the sum of scores of all the prior hypotheses in its supercluster, plus the sum of the reward values for all assignments between prior tracks and measurements involved in its posterior tracks.

Given the prior clusters, hypothesis and tracks, as well as the m_k measurements and the reward matrix G, our goal is to identify the M hypotheses with the highest scores in each posterior cluster. Each of the M solutions in posterior cluster c_i is a composition of two mappings between discrete sets. First, there is a mapping from the supercluster to the prior hypotheses that it contains:

$$\mathcal{S} : \sigma_i \to \mathbb{N} \text{ such that } \mathcal{S}(l) \leq \overline{n}_{Hl} \ \forall \ l \in \sigma_i.$$

Second, there is a mapping from the prior tracks in the output of $\mathcal{S}$ to the extended measurement set $\{1, \ldots, \tilde{m}_k\}$. This mapping is of the form

$$\mathcal{T} : \bigcup_{l \in \sigma_i} \mathcal{S}(l) \to \{1, \ldots, \tilde{m}_k\} \text{ such that } \mathcal{T}(i) = \mathcal{T}(j) \Rightarrow i = j.$$

V. Murty's method revisited

Murty's method can be formulated in terms of two loops, which together perform a branch-and-bound search. The outer loop runs until the M topmost posterior hypotheses have been generated, or no more hypotheses can be generated. The inner loop goes over all relevant tracks in a prior hypothesis and tries out new measurement assignments for these. When we discuss a generic Murty algorithm, we shall refer to the tracks as customers and the measurements as items.

Murty's method is typically formulated in terms of its operation on problem-solution pairs $q = \langle \mathsf{P}, \mathsf{S} \rangle$. The problem P can be constructed as a collection of 4-tuples $T = \langle y, z, l, \mathrm{e} \rangle$ where y is a customer, z is an item, l is a reward value and e is an upper bound on the possible score values of solutions S that contains T. A solution S to the problem P is a subset of the 4-tuples in P so that all customers y in P are included, and no item z is repeated. The score $q.\mathrm{s}$ of a solution $q.\mathsf{S}$ is the sum of all the l-values in the solution. The score bound $T.\mathrm{e}$ can be found by means of the dual variables in the JV algorithm using the technique described in [9]. The score bound can also be omitted, or, equivalently, made trivial. Pseudo-code for the outer Murty loop is given in Algorithm 1.

Algorithm 1 The outer loop of Murty's method

```
procedure MURTY($\mathsf{P}_1$, $M$)
    $\mathsf{Q}[1] \leftarrow$ SOLVE( $\mathsf{P}_1$)
    $\mathsf{R} \leftarrow [\,]$
    while $|\mathsf{R}| < M$ and $\mathsf{Q}$ is non-empty do
        $\mathsf{Q}[*] \leftarrow$ Entry of $\mathsf{Q}$ with the highest score
        Remove $\mathsf{Q}[*]$ from $\mathsf{Q}$
        $\mathsf{R} \leftarrow [\mathsf{R}, \mathsf{Q}[*]]$
        $\mathsf{Q}' \leftarrow$ INNERMURTY($\mathsf{Q}[*], \mathsf{Q}[*].\mathsf{P}, \mathsf{Q}[*].\mathsf{S}$)
        $\mathsf{Q} \leftarrow [\mathsf{Q}, \mathsf{Q}']$
    end while
    return $\mathsf{R}$
end procedure
```

Pseudo-code for the inner loop follows in Algorithm 2. The actual assignment solving is done by the procedure SOLVE, which we will elaborate later (in Algorithm 5). This could for example be a wrapper of a JV implementation. Notice that SOLVE as used in INNERMURTY takes both the problem P' and the priority queue entry $\mathsf{Q}[*]$ as separate argument. This is done to make its usage in the proposed branch-and-bound algorithm as streamlined as possible. It outputs a new priority queue entry q which is of the the same structure as $\mathsf{Q}[*]$. A key step in the algorithm is the enforcement of the tuple $T = \langle y, z, \ldots \rangle$ on line 12, which is a shorthand notation for removing all tuples involving the customer y or the item z except for T itself. This leads to a partitioning of the original problem, so that problems generated subsequently in the inner Murty loop become smaller.

Algorithm 2 The inner loop of Murty's method

```
1:  procedure INNERMURTY($\mathsf{Q}[*], \mathsf{P}, \mathsf{S}$)
2:      $\mathsf{Q}' \leftarrow [\,]$
3:      $o \leftarrow$ order of customers by a heuristic
4:      for $T = \langle y, z, \ldots \rangle \in \mathsf{S}$ ordered according to $o$ do
5:          $\mathrm{e} \leftarrow \max T'.e$ s.t. $T'.y = T.y$ and $T'.z \neq T.z$
6:          if $\mathrm{e} \geq$ LSB then
7:              $\mathsf{P}' \leftarrow \mathsf{P}$
8:              Remove $T$ from $\mathsf{P}'$
9:              $q \leftarrow$ SOLVE( $\mathsf{P}'$ ; $\mathsf{Q}[*]$ )
10:             if $q.\mathsf{S}$ is valid and $q.\mathrm{s} \geq$ LSB then
11:                 $\mathsf{Q}' \leftarrow [\mathsf{Q}', q]$
12:             end if
13:             Enforce $T$ in $\mathsf{P}$
14:         end if
15:     end for
16:     return $\mathsf{Q}'$
17: end procedure
```

The formulation of INNERMURTY also makes use of an ordering heuristic o which decides the order of processing the customers. It was recommended in [9] to sort the customers according to the bounds in an ascending order. The rationale behind this is that by treating the problems most likely to contain good solutions last, the sizes of these problems are reduced, leading to better efficiency. Furthermore, the formulation of INNERMURTY makes use of a lower score bound (LSB), which also has been included due to its importance in the proposed branch-and-bound algorithm. The purpose of the LSB is to immediately discard solutions which cannot possibly be among the M best solutions.

Murty's method generates a search tree whose root node is $\mathsf{Q}[1].\mathsf{S}$, and whose parental relationships are defined by lines 7 and 10 in Algorithm 2. The solutions contained in Q can be seen as a set of active leaf nodes, which potentially may get children in subsequent iterations. All paths from the root node, i.e. $\mathsf{Q}[1].\mathsf{S}$, towards the leafs must be non-increasing in score. All nodes that are not part of Q are nodes that have been admitted into R. Consequently, the best of the leaf nodes, which is to be admitted into R at the current iteration, cannot be worse than subsequent additions to R. It follows from this that R is guaranteed to contain the M best solutions at iteration number M of the outer Murty loop.

VI. Joint clustering and hypothesis exploration

The M-best problem posed in Section IV has a cascaded structure which is more complicated than the structure assumed by Murty's method. Nevertheless, Murty's method can be generalized in various ways to solve this problem as well. One possible methodology could be what we may call a double Murty method: Once for discovering good prior hypotheses, and once for discovering good track-to-measurement associations. However, it is not straightforward to determine how many solutions should be returned in the two stages in order to find the M best hypotheses in each posterior cluster.

Instead, we propose a branch-and-bound technique, which employs only the inner Murty loop as a tool for discovering both good prior hypotheses and good track-to-measurement associations. The algorithm consists of three phases. In phase one we find the best local posterior hypothesis for each parent hypothesis. In phase two we initialize the priority queue of posterior candidate hypotheses for the supercluster. In phase three we search for good posterior hypotheses in the supercluster, by adaptively exploring changes of parent hypotheses (*switches*) and track-to-measurement assignments (*expansions*). Pseudocode is given by Algorithms 2 - 7, which together provide all the building blocks. The problems to be solved, on local expansion level, switch level and clustered expansion level, respectively, are constructed by means of the procedures in Algorithm 3.

Algorithm 3 Problem construction algorithms

```
                ▷ Track to measurement for given prior hypothesis
procedure ProblemConstruct1($\overline{\theta}_l$)
    P ← $\{< y, z, g^{yz} >$ such that $y \in \overline{\theta}_l$ , $g^{yz} > -\infty\}$
    return P
end procedure
                                    ▷ Prior cluster to prior hypothesis
procedure ProblemConstruct2(B)
    P ← $\{\langle y, z, l\rangle$ such that $y$ is a prior cluster,
                $z$ is a prior hypothesis in cluster $y$,
                $l$ is $\mathsf{B}(y,z).s$ $\}$
    return P
end procedure
                            ▷ Track to measurement within supercluster
procedure ProblemConstruct3($\tilde{\mathsf{S}}$)
    P ← $\{\langle y, z, g^{yz}, \mathrm{e}\rangle$ s.t. $y \in T.\overline{\theta}$ for some $T \in \tilde{\mathsf{S}}$,
                $g^{yz} > -\infty$ and
                $\mathrm{e} \geq$ score of any S containing $\langle y, z, \ldots\rangle$ $\}$
    return P
end procedure
```

A. Phase one (local assignments)

For each prior hypothesis in the prior clusters of the supercluster we solve the track-to-measurement assignment problem, and store the result together with the posterior score. We term these solutions *best-local posterior hypotheses*. Pseudocode for this is given in Algorithm 4.

Algorithm 4 Initialize MHT priority queue

```
function InitMHTqueue($\overline{c}$)
    B ← []
    for each hypothesis $\overline{\theta}_l \in \overline{c}$ do
        P ← ProblemConstruct1($\overline{\theta}_l$)
        [S, s] ← Assign2D(P)
        B ← [B, < S, s >]
    end for
    return SortDescending(B)
end function
```

B. Phase two (Priority queue initialization)

The goal of this phase is to find one posterior hypothesis which is a promising top contender. To do this, we concatenate the prior hypotheses that gave the top solutions for each prior cluster during phase one, and solve the resulting track-to-measurement assignment problem. The solving is done by means of the procedure described in Algorithm 5. If there is measurement contention, the result may be different from what we would get by concatenating the local posterior hypotheses from phase one, and the resulting score may be lower than the sum of scores of the best-local hypotheses involved.

Consequently, the initialization returns not only a global candidate solution, its score and its building blocks, but also a bound. This bound is equal to the sum of the best-local scores involved (line 9 in Algorithm 7), ensuring that any global hypothesis discovered later in the search tree cannot have a higher score. This bound is declared the *upper undiscovered bound* (UUB), meaning that no undiscovered hypotheses can have a higher score. The LSB, which was introduced in Section V, is simultaneously initialized at $-\infty$.

Algorithm 5 The generalized solve algorithm

```
procedure Solve($\pi$, Q[∗])
    if $\pi = \tilde{\mathsf{P}}$ then
        [$\tilde{\mathsf{S}}$, b] ← Assign2D($\tilde{\mathsf{P}}$)
        P ← ProblemConstruct3($\tilde{\mathsf{S}}$)
        [S, s, e] ← Assign2D(P)
        $q$ ← $<\mathsf{S}, \tilde{\mathsf{S}}, \mathsf{P}, \tilde{\mathsf{P}}, \mathrm{b}, \mathrm{s}, \mathrm{e}>$
    else if $\pi = \mathsf{P}$ then
        [S, s, e] ← Assign2D(P)
        $q$ ← $<\mathsf{S}, \mathsf{Q}[*].\tilde{\mathsf{S}}, \mathsf{P}, \mathsf{Q}[*].\tilde{\mathsf{P}}, \mathrm{s}, \mathrm{s}, \mathrm{e}>$
    end if
    return $q$
end procedure
```

C. Phase three (Switch-expand search)

The search for the M best posterior hypotheses is implemented as a loop which runs until the UUB coincides with the LSB, or no more feasible candidates can be found. The LSB remains at $-\infty$ until M global hypotheses have been generated, after which it always takes the score value of hypothesis number M sorted from the top. In every iteration we process the hypothesis in the priority queue that holds the

UUB. The hypotheses that are generated become descendants of the UUB holder in the search tree. During every iteration, the algorithm considers three possibilities in turn, which we call *aggressive switching*, *lazy switches* and *expansions*.

Aggressive switching is triggered if the UUB holder has a lower score than the UUB. The goal is to find a new UUB holder whose score is equal to the UUB before we proceed to lazy switches and expansions. To initialize aggressive switching the UUB holder is marked as *unresolved*. The aggressive switching is itself a while-loop that runs until no more unresolved hypotheses exist. During each iteration, the unresolved hypothesis with the highest bound is picked for processing. An inner Murty loop is performed, with the clusters as customers and the available switches as items. It should be noted that this switch-level assignment problem is without contention, because each prior cluster has a separate set of prior hypotheses. However, the calls to SOLVE must solve both the switch-level assignment problem and the resulting track-to-measurement assignment problem, which yields the hypothesis score. For each parent cluster, a best case loss value is calculated. This is the difference between the the best local score for the next best parent, and the best local score for the current parent. Customers are skipped if no feasible switches exist, or if the resulting best case loss is lower than the LSB.

Lazy switching is also conducted in terms of an inner Murty loop identical to the one used for aggressive switching. In this case, the loop is always used once and only once in each iteration.

Expansions are also generated by an inner Murty loop. Here, the customers are the active tracks in the parents of the UUB holder and the items are the available measurements. Here, the score bounds used in line 6 of Algorithm 2 are obtained from the dual JV variables. Customers are skipped if their bounds will push the posterior hypothesis score below the LSB. Switches are not allowed in the descendants of an expansion in the search tree.

REMARK 1 (The need for aggressive switching)**.** If there is a gap between the bound and the score of the UUB holder, then there must be measurement contention between the prior tracks of the UUB holder. In such a situation, it cannot be ruled out that a switch may lead to a score that is higher than the current score. However, we are not allowing switches among the descendants of expansions in the search tree. Therefore, we are only ready to perform expansions after we have accounted for all possible switches that potentially can reduce the gap.

REMARK 2 (No switches downstream from expansions)**.** Switches are not allowed to follow expansions in the descents of the search tree. This requirement is enforced in order to prevent the same hypothesis from being reached through multiple descents. This also makes it possible to use the score of any expansion as its bound.

The pseudocode for the complete procedure is given in Algorithm 7. In addition to the procedures presented in Algorithms 2 - 5, it also makes use of a procedure for maintaing and sorting the priority queue, given in Algorithm 6.

Algorithm 6 Queue maintenance procedure

```
procedure QUEUEMAINTAIN($\mathsf{Q}$)
    Sort $\mathsf{Q}$ in decreasing order according to $\mathsf{Q}.\mathrm{b}$
    $\alpha \leftarrow \arg\max_i \mathsf{Q}[i].b$, $\mathsf{Q}[i]$ not investigated before
    UUB $\leftarrow \mathsf{Q}(\alpha).\mathrm{b}$
    if $|\mathsf{Q}| \geq M$ then
        $L \leftarrow$ sort $\mathsf{Q}.\mathrm{s}$ in descending order
        LSB $\leftarrow L[M]$
    end if
    return $\mathsf{Q}$ , UUB, LSB, $\alpha$
end procedure
```

Algorithm 7 The switch-expand algorithm

```
procedure SWITCHEXPAND
                                        ▷ Phase 1: Best-local searches
    Initialize empty matrix structure $\mathsf{B}$
    for each cluster $\bar{c}$ do
        $\mathsf{B}[\bar{c},:] \leftarrow$ INITMHTQUEUE($\bar{c}$)
    end for
                              ▷ Phase 2: Initialization of priority queue
    $\tilde{\mathsf{S}} \leftarrow$ Concatenate topmost solutions in $\mathsf{B}$.
    $b \leftarrow$ sum of topmost scores in $\mathsf{B}$.
    $\tilde{\mathsf{P}} \leftarrow$ PROBLEMCONSTRUCT2($\mathsf{B}$)
    $\mathsf{P} \leftarrow$ PROBLEMCONSTRUCT3($\tilde{\mathsf{S}}$)
    $[\mathsf{S}, \mathrm{s}, \mathrm{e}] \leftarrow$ ASSIGN2D($\mathsf{P}$)
    $\mathsf{Q}[1] \leftarrow <\mathsf{S}, \tilde{\mathsf{S}}, \mathsf{P}, \tilde{\mathsf{P}}, \mathrm{b}, \mathrm{s}, \mathrm{e}>$
    $\alpha \leftarrow 1$
    LSB $\leftarrow -\infty$
    UUB $\leftarrow \mathsf{Q}[\alpha].\mathrm{b}$
                                  ▷ Phase 3: Branch-and-bound search
    while UUB $>$ LSB do
        while $\mathsf{Q}[\alpha].\mathrm{b} > \mathsf{Q}[\alpha].\mathrm{s}$ do
            $\tilde{\mathsf{Q}} \leftarrow$ INNERMURTY($\mathsf{Q}[\alpha], \mathsf{Q}[\alpha].\tilde{\mathsf{P}}, \mathsf{Q}[\alpha].\tilde{\mathsf{S}}$)
            $\mathsf{Q} \leftarrow [\mathsf{Q}, \tilde{\mathsf{Q}}]$
            [$\mathsf{Q}$ , UUB, LSB, $\alpha$ ] $\leftarrow$ QUEUEMAINTAIN($\mathsf{Q}$)
        end while
        $\tilde{\mathsf{Q}}_s \leftarrow$ INNERMURTY($\mathsf{Q}[\alpha], \mathsf{Q}[\alpha].\tilde{\mathsf{P}}, \mathsf{Q}[\alpha].\tilde{\mathsf{S}}$)
        $\tilde{\mathsf{Q}}_e \leftarrow$ INNERMURTY($\mathsf{Q}[\alpha], \mathsf{Q}[\alpha].\mathsf{P}, \mathsf{Q}[\alpha].\mathsf{S}$)
        $\mathsf{Q} \leftarrow [\mathsf{Q}, \tilde{\mathsf{Q}}_s, \tilde{\mathsf{Q}}_e]$
        [$\mathsf{Q}$ , UUB, LSB, $\alpha$ ] $\leftarrow$ QUEUEMAINTAIN($\mathsf{Q}$)
    end while
    $I \leftarrow$ indices of the $M$ highest entries of $\mathsf{Q}.\mathrm{s}$
    return $\mathsf{Q}[I].\mathsf{S}$
end procedure
```

VII. WORKED EXAMPLE

Consider a scenario involving 7 tracks and 2 non-dummy measurements, with reward values organized in the matrix

$$\begin{array}{c|ccc} t\backslash j & 1 & 2 & 3-8 \\ \hline 10 & 4.30 & 4.77 & \\ 11 & -0.19 & 3.38 & \\ 12 & 3.85 & 1.13 & \\ 13 & 1.65 & -1.25 & -2.30 \cdot \mathbf{I}_6^*. \\ 14 & 4.66 & 4.62 & \\ 15 & 1.77 & 4.13 & \end{array} \tag{1}$$

$\mathbf{I}_6^*$ denotes an identity matrix where all zeros have been replaced with $-\infty$. Tracks 10, 11, 14 and 15 are in prior cluster 1, while tracks 12 and 13 are in prior cluster 2. The prior MBM in cluster 1 contains two hypotheses $[11, 14]$ and $[10, 15]$ with prior scores 0 and -6.1407. The prior MBM in cluster 2 contains two hypotheses $[12]$ and $[13]$ with prior scores 0 and -5.7303. Our goal is to find the $M = 6$ best posterior hypotheses.

A. Phase 1

In Phase 1 we find the best local posterior hypotheses for each of the prior hypotheses. The outcome of Phase 1 is displayed in Table I.

B. Phase 2

In Phase 2 we are to initialize by combining the prior hypotheses $\overline{\theta}_{1(1)}$ and $\overline{\theta}_{1(2)}$, and solving the resulting assignment problem, so that we find an initial posterior hypothesis θ_1. That is, we are to find the optimal assignment of measurements to the track collection $[11, 14, 12]$. Notice that there is measurement contention: Both track 11 and track 12 got measurement 1 assigned in the best-local hypotheses of Table I. The assignment problem has the solution $[4, 2, 1]$ with total score

$$6.17 = 0 + 0 - 2.30 - 4.62 + 3.85.$$

On the other hand, the bound of θ_1, i.e., the upper bound for the score values of its descendants in the search tree, is

$$11.88 = 8.03 + 3.85.$$

C. Phase 3

The algorithm spends a total of 5 iterations of the outermost loop before the LSB and the UUB meet. A total of 7 hypotheses are included in the priority queue. The search tree is displayed in Figure 1, while the details of the hypotheses are listed in Table II. The evolution of the LSB and UUB during the iterations is displayed in Table III.

First iteration: In the first iteration of Phase 3 we first consider aggressive switching, because the bound of the initialization was higher than its score. However, switching to $\overline{\theta}_{2(1)}$ will reduce the bound to $6.13 = 11.88 - (2.29 - 8.04)$, while switching to $\overline{\theta}_{2(2)}$ will reduce the bound to $3.95 = 11.88 - (-4.08 - 3.85)$. Since both these bounds are below the score, aggressive switching is dismissed.

We proceed to perform lazy switches. We consider changing the default prior combination $[\overline{\theta}_{1(1)}, \overline{\theta}_{1(2)}]$ to $[\overline{\theta}_{2(1)}, \overline{\theta}_{1(2)}]$ and $[\overline{\theta}_{1(1)}, \overline{\theta}_{2(2)}]$, which yield hypothesis 2 and 3 in the search tree, respectively. At a first glance, the bound of hypothesis 2 appears to be $6.14 = 2.29 + 3.85$. However, the bound is not tight, because we have measurement contention. Therefore, we check whether subsequent switches could yield a higher score. Since $\overline{\theta}_{1(1)}$ already is invalidated, the only remaining switch is changing $\overline{\theta}_{1(2)}$ to $\overline{\theta}_{2(2)}$. Compared to preliminary bound of 6.14 this will lead reduce the bound to $-1.79 = 6.14 - 3.85 - 4.08$. On the other hand, solving the assignment problem of hypothesis 2 yields the score 0.1689. Thus, we can only know

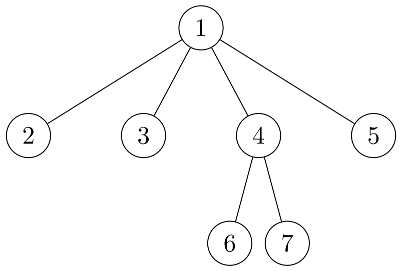

Fig. 1. Search tree for the worked example which includes all hypotheses obtained through solving a track-to-measurement assignment problem.

Prior hypothesis	$\overline{\theta}_{1(1)}$	$\overline{\theta}_{2(1)}$	$\overline{\theta}_{1(2)}$	$\overline{\theta}_{2(2)}$
Tracks	$[11, 14]$	$[10, 15]$	$[12]$	$[13, 40]$
Measurements	$[2, 1]$	$[1, 2]$	$[1]$	$[6]$
Score	8.04	2.29	3.85	-4.08

TABLE I
THE BEST LOCAL HYPOTHESES IN BOTH CLUSTERS.

that any hypotheses further down in the search tree below hypothesis 2 cannot have a higher score than 0.1689, which thus becomes the bound of hypothesis 2. Similar reasoning is used for hypothesis 3.

We proceed to make the expansions of hypothesis 1. From the dual variables obtained in the initialization we have score bounds that are used to decide the order of expansions. In the case that the assignment of track 11 with measurement 4 is invalidated, the best remaining assignment is measurement 2 with score bound 6.1713. Similarly, the best remaining assignments to track 14 and 12, after invalidating measurements 2 and 1, are measurements 1 and 5, with score bounds 6.1713 and 5.7369, respectively. Going from least to most promising, as suggested in [9], the order becomes 12, 14 and 11.

The expansion in track 12 results from invalidating measurement 1, leading to hypothesis number 4 with score 5.7308. Measurement 1 is then enforced to track 12 for the remainder of iteration 1. The expansion in track 14 results from invalidating measurement 2, leading to hypothesis number 5 with score 4.9256. Measurement 2 is then enforced to track 14 for the remainder of iteration 1. The expansion in track 11 is then activated by invalidating measurement 4. However, since measurements 1 and 2 already are enforced to other tracks, the problem is not solvable. We have now generated all hypothesis from iteration 1. The UUB is passed onto the second best hypothesis, which is hypothesis 4. The LSB is still $-\infty$ because we have not yet generated 6 hypotheses.

Second iteration: Since hypothesis 4 is an expansion, no switches are considered, and we move straight onto possible expansions. This track-to-measurement assignment problem is similar to that of hypothesis 1, except that measurement 1 has been invalidated for track 12. We use the dual variables from the assignment solving of hypothesis 4, which yield the score bounds 5.5954, 2.2695 and 5.7369 for tracks 11, 14 and 12 respectively. This yields the track order 14, 11 and 12. Again, we see that the tracks that will get the contended measurements

Hypothesis	Tracks	Measurements	Bound	Score
1	$[11, 14, 12]$	$[4, 2, 1]$	11.8482	6.1652
2	$[10, 15, 12]$	$[2, 8, 1]$	0.1689	0.1689
3	$[11, 14, 13, 40]$	$[2, 1, 6]$	0.0054	0.0054
4	$[11, 14, 12]$	$[2, 1, 5]$	5.7308	5.7308
5	$[11, 14, 12]$	$[2, 7, 1]$	4.9256	4.9256
6	$[11, 14, 12]$	$[1, 2, 5]$	2.1218	2.1218
7	$[11, 14, 12]$	$[4, 1, 2]$	3.4793	3.4793

TABLE II
THE 8 POSTERIOR HYPOTHESES FOR THE WORKED EXAMPLE.

Iteration	LSB	UUB	Found hypotheses
1	$-\infty$	11.88	[①]
2	$-\infty$	5.7308	[1, ④, 5, 2, 3]
3	0.169	4.9256	[1, 4, ⑤, 8, 7, 2, 3]
4	0.169	3.4793	[1, 4, 5, ⑦, 6, 2, 3]
5	0.169	2.1218	[1, 4, 5, 7, ⑥, 2, 3]

TABLE III
THE 5 ITERATIONS FOR THE WORKED EXAMPLE.

1 and 2 enforced in subsequent expansions, are prioritized. Thus, we again find an impossible problem in the third expansion. Before that, hypothesis 6 and 7 are generated. We have now generated all hypothesis from iteration 2. The UUB is passed to the second best hypothesis, which is hypothesis 5. We have now a total of 7 hypotheses, and the LSB takes the value of the second worst hypothesis, which is 0.169.

Third to fifth iteration: During these iterations, several more hypotheses are considered, but in every case it is found that the score must be lower than the LSB. Eventually, at the end of iteration 5,hypothesis number 2, which is the sixth best hypothesis, becomes the UUB holder while also holding the LSB, and we are done.

VIII. SIMULATIONS

The proposed branch-and-bound method has been developed as part of a complete PMBM-style tracking system with cluster management. A detailed exposition of this system is beyond the scope of present paper. Therefore, we do not analyze the performance of the overall tracking system. Instead, we study the performance of the method as a standalone component, by comparing it with benchmark alternatives on 80000 single- and multi-cluster assignment problems generated by the complete tracking system.

A detailed exposition of the simulation setup used is also beyond the space constraints of this paper. The assignment problems have been generated from two scenarios: A track initialization scenario with potentially up to about 30 targets, mostly well separated, and a formation tracking scenario with 8 targets which often, but not always, are so close that several of the validation gates intersect. Table IV gives an impression of how the assignment problems vary in size. We see that single-cluster single-target problems dominate by far with 75% of the cases, but thanks to the large number of simulations

TABLE IV
FREQUENCY OF CARDINALITIES AND CLUSTER NUMBERS

Ave. Card.	$\overline{n}_C = 1$	$\overline{n}_C = 2$	$\overline{n}_C = 3$	$\overline{n}_C = 4$	$\overline{n}_C = 5$
0	2114	0	0	0	0
1	60632	207	7	0	0
2	11590	3043	98	4	0
3	2963	2004	626	24	6
4	885	940	620	161	10
5	306	478	424	205	87
6	97	230	232	182	100
7	49	116	160	118	91
8	16	59	105	74	39
9	4	41	55	56	29
10	4	41	55	56	29
11	1	9	29	42	36
12	0	14	18	29	49
13	0	0	10	19	33
14	0	1	3	10	11
15+	0	0	2	6	8

there are also a significant number of fairly complex assignment problems. The observant reader may question why there are 2114 cases of average cardinality zero, and why we have 7 cases with three clusters and average cardinality 1, etc. This is because clusters have been allowed to contain empty hypotheses, which may pull the average cardinality down.

A. *Benchmark methods*

We consider 4 methods for hypothesis exploration. The first method, "Many", uses the proposed algorithm with a very large number (1000) of hypotheses per supercluster. This is only implemented for benchmarking purposes. The second method, "Branch-and-bound", uses the proposed algorithm with a reasonable number (150) of hypotheses per supercluster. In the third method, "Double", Murty's method is first used to discover the 30 best combinations of prior hypotheses from the clusters involved, based on their prior scores. Then, Murty's method is used to discover the 30 best posterior hypotheses for each such combination of prior hypotheses. The fourth method, "Proportional", uses proportional representation of prior hypotheses. Here the best 150 combinations of prior hypotheses are generated, in the same way as in the third approach. The prior probabilities of the combinations are then calculated, by normalizing the exponential score sums. Denote the probability of combination number l by P_l. Combination number l is then allowed to generate at most $\lceil 150 \cdot P_l \rceil$ posterior hypotheses through standard Murty.

B. *Performance analysis*

We lump all the cases of $\overline{n}_C$ together in our study of performance. The same patterns can be observed for all values $\overline{n}_C$. In Table V we analyze how much of the probability mass is missing for Branch-and-bound, Double and Proportional, compared to Many. The numbers are to be read as follows: For instance, when Branch-and-bound attains the value 0.014 for average cardinality 4, it means that the probability mass not contained in the output of Branch-and-bound is less than 0014 in 90% of the cases.

TABLE V
90% QUANTILE OF MISSING PROBABILITY MASS

Ave. Card.	Branch-and-bound	Double	Proportional
0	0	0	0
1	0	0	0
2	0	0.009	0.003
3	0.004	0.037	0.021
4	0.031	0.116	0.059
5	0.0706	0.196	0.108
6	0.133	0.300	0.1786
7	0.173	0.366	0.223
8+	0.1978	0.376	0.289

TABLE VI
PROPORTION OF CASES WHERE A SIGNIFICANT HYPOTHESIS IS LOST

Ave. Card.	Double	Proportional
0	0.000	0.000
1	0.000	0.000
2	0.003	0.001
3	0.018	0.005
4	0.055	0.018
5	0.101	0.420
6	0.168	0.071
7	0.210	0.103
8+	0.266	0.150

The improvement of Branch-and-bound over Proportional is most dramatic in the cases with average cardinality 2 and 3. The limited performance gains of Branch-and-bound over Proportional for higher cardinalities may possibly reflect that such problems are much more difficult, so that very large numbers of hypotheses are needed to encapsulate all the probability mass.

Another perspective can be seen by comparing the probability of the best hypothesis that Double or Proportional failed to find, with the probability of the topmost hypothesis, as calculated by Branch-and-bound. In Table VI we look at frequencies for how often the ratio between these probabilities is higher than 0.1. Again Proportional does significantly better than Double. For both methods these frequencies appear to increase as a function of cardinality.

We may ask how many of the top 150 hypotheses are missing in Double or Proportional, regardless of their probabilities. In Table VII we look at the median for this proportion. Notice that Proportional loses as much as 0.222 of the hypotheses already at cardinality 2. This is because Proportional frequently distributes very many of its slots for new hypotheses to prior hypotheses which in reality have few children despite the high prior probability, and therefore fails to include the many children from prior hypotheses with lower prior probabilities.

IX. CONCLUSION

Various approaches can be devised for hypothesis exploration in multiple hypothesis tracking with multiple clusters. We have proposed a systematic approach that finds the M best posterior hypotheses in each supercluster. Simulations indicate that alternative, and more heuristic approaches to hypothesis exploration, may suffer from a higher loss of probability mass.

TABLE VII
MEDIAN FOR THE PROPORTION OF HYPOTHESES LOST

Ave. Card.	Double	Proportional
0	0.000	0.000
1	0.000	0.000
2	0.000	0.250
3	0.170	0.375
4	0.343	0.347
5	0.347	0.327
6	0.393	0.287
7	0.400	0.268
8+	0.420	0.267

We intend to report the detailed description of a complete PMBM filter utilizing the proposed exploration method in a forthcoming publication. Potential topics of future research include tighter bounds, more efficient ordering heuristics, and possibly also techniques from probabilistic graphical model theory to further boost performance.

REFERENCES

[1] D. Reid, "An algorithm for tracking multiple targets," *IEEE Transactions on Automatic Control*, vol. 24, no. 6, pp. 843–854, Dec. 1979.

[2] J. Williams, "Marginal multi-Bernoulli filters: RFS derivation of MHT, JIPDA, and association-based MeMBer," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 51, no. 3, pp. 1664–1687, July 2015.

[3] E. F. Brekke and M. A. Chitre, "The multiple hypothesis tracker derived from finite set statistics," in *Proceedings of Fusion*, Xi'An, China, July 2017.

[4] K. G. Murty, "An algorithm for ranking all the assignments in order of increasing cost," *Operations Research*, vol. 16, no. 3, pp. 682–687, 1968.

[5] R. Danchick and G. E. Newnam, "Reformulating Reid's MHT method with generalised Murty K-best ranked linear assignment algorithm," *IEE Proceedings - Radar, Sonar and Navigation*, vol. 153, pp. 13–22, Feb. 2006.

[6] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Norwood, MA, USA: Artech House, 1999.

[7] C. H. Papadimitriou and K. Steiglitz, *Combinatorial Optimization: Algorithms and Complexity*. Mineola, NY, USA: Dover, 1998.

[8] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, no. 4, pp. 325 – 40, 1987.

[9] M. L. Miller, H. S. Stone, and I. J. Cox, "Optimizing Murty's ranked assignment method," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 33, no. 3, pp. 851–862, Jul. 1997.

[10] D. F. Crouse, "On implementing 2D rectangular assignment algorithms," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 52, no. 4, pp. 1679–1696, 2016.

[11] A. F. Garcia-Fernandez, J. L. Williams, K. Granström, and L. Svensson, "Poisson multi-Bernoulli mixture filter: direct derivation and implementation," *IEEE Transactions on Aerospace and Electronic Systems*, pp. 1–1, 2018.

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

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

[14] E. L. Lawler, "A procedure for computing the K best solutions to discrete optimization problems and its application to the shortest path problem," *Management Science*, vol. 18, no. 7, pp. 401–405, 1972.

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