Multi-Target Tracking via Mixed Integer Optimization

Dimitris Bertsimas, Zachary Saunders, and Shimrit Shtern

***Abstract*—Given a set of target detections over several time periods, this paper addresses the multi-target tracking problem (MTT) of optimally assigning detections to targets and estimating the trajectory of the targets over time. MTT has been studied in the literature via predominantly probabilistic methods. In contrast to these approaches, we propose the use of mixed integer optimization (MIO) models and local search algorithms that are (a) scalable, as they provide near optimal solutions for six targets and ten time periods in milliseconds to seconds, (b) general, as they make no assumptions on the data, (c) robust, as they can accommodate missed and false detections of the targets, and (d) easily implementable, as they use at most two tuning parameters. We evaluate the performance of the new methods using a novel metric for complexity of an instance and find that they provide high quality solutions both reliably and quickly for a large range of scenarios, resulting in a promising approach to the area of MTT.**

***Index Terms*—data association; mixed integer optimization; multi-target tracking; optimization; trajectory estimation**

I. Introduction

Multi-target tracking is the problem of estimating the state of multiple dynamic objects, referred to as *targets*, over a fixed window of time. At various points of time within the window, the targets are observed in a *scan*, resulting in a set of *detections*. The multi-target tracking problem aims to extract information about target dynamics from these detections.

Solutions to this problem are sought across many civilian and military applications including, but not limited to, ballistic missile and aircraft defense, space applications, the movement of ships and ground troops, autonomous vehicles and robotics, and air traffic control. As each application has unique attributes and assumptions, various algorithms have been developed to solve this problem across the spectrum of contexts. As a result the field of multi-target tracking has expanded to numerous research venues, generating a wide range of literature on the topic. A complete overview of all MTT methods, including the classes of algorithms and their variants, as well as additional methods not discussed in this paper can be found in [1]. For a more exhaustive overview of estimation techniques, filtering, gating, and other methods see [2] and [3].

The field of multi-target tracking faces two primary challenges: (i) data association and (ii) trajectory estimation. Given a set of sensor detections, the data association problem consists of assigning the detections to a set of targets. Alternatively, this can be viewed as a labeling problem in which each detection must be labeled with a target identifier. The association problem is further complicated when sensors fail to report detections (missed detection) or incorrectly report detections (false alarm), resulting in ambiguity in the number of existing targets. The trajectory estimation problem consists of estimating the state space of a target (*i.e.*, position, velocity, acceleration, size, etc.) from the associated detections of the aforementioned assignment problem. Even when all of the associations are known, the estimation problem is challenging due to the presence of measurement noise. The two problems of data association and trajectory estimation are closely related and dependent on one another.

Some classical algorithms treat the data association and trajectory estimation problems separately using a combination of probabilistic approaches to determine data associations and filters to estimate trajectories. One such algorithm is the global nearest neighbor (GNN). The GNN algorithm is a naive 2-D assignment algorithm that evaluates one scan of detections at a time, globally assigning the nearest detection at each scan [4]. Once the data association has been determined, the detections are often passed through one of numerous filters, most commonly a Kalman filter [5], which updates the trajectory estimates before the algorithm progresses forward to the next scan. This process repeats sequentially through each scan of the data.

Modern algorithms in the field of multi-target tracking are most commonly statistically based, often relying on heavy probabilistic assumptions about the underlying target dynamics or detection process. The two most prevalent statistical algorithms in the field of multi-target tracking are the Multiple Hypothesis Tracker (MHT) and the Joint Probability Data Association Filter (JPDAF), along with their numerous variants and extensions. Both classes of algorithms attempt to solve the data association problem by generating a set of potential hypotheses, or possible detection-to-track assignments. Here a *track* is a set of labeled detections belonging to the same target. Probabilities are assigned to each hypothesis based on the likelihood of the trajectory's existence, and numerous approaches for accomplishing this task have been proposed.

The MHT, first proposed by Reid in [6], assigns likelihood values to hypotheses using a Bayesian maximum a posteriori estimator, which requires probabilistic assumptions on both object dynamics and detection process. This algorithm is generally considered to be the modern standard for solving the data association problem. Many variants have been proposed for implementation, which leverage techniques such as clustering, gating, hypothesis selection, hypothesis pruning, and merging of state estimates. Many of these methods are summarized in [7].

While the MHT has seen various forms of success, it faces several key challenges. Namely, the curses of dimensionality and complexity. The number of possible hypotheses grows

exponentially with the number of potential tracks and the number of scans. Consequently, it is considered intractable for large scenarios. Moreover, the MHT may potentially require extensive tuning and thus may be difficult to implement in practice, in addition to being computationally expensive. For these reasons, it is generally considered to be one of the most complex MTT algorithms.

A Probability Data Association (PDA) takes a Bayesian approach to solving the data association problem by finding detection-to-target assignment probabilities via a posterior PDF, which again requires heavy assumptions on object dynamics and the detection process. In similar fashion, a Joint PDA (JPDA) assigns probabilities that are computed *jointly* across all targets. The JPDAF is an algorithm that implements the JPDA along with filters and estimation methods as discussed previously in [2].

A limited number of optimization based algorithms have been applied to solve the MTT problem, most of which attempt to solve it by mapping the measurement set onto a trellis and seeking the optimal measurement association sequence. Some examples of such approaches include the Multi-Target Viterbi [8] and a following extension in [9], which formulates the method in [8] as a network flow, reducing the solve time from exponential to polynomial. Still other works, in particular [10], suggested adaptations of this approach that output a single best set of tracks, or a list of best sets of tracks, similar to the MHT.

Compared to the number of statistically based algorithms in the MTT literature, optimization based algorithms are relatively scarce. In fact, most of these algorithms in the MTT literature propose the use of optimization to leverage statistical algorithms such as, in particular, the MHT. For example, integer optimization has been used to improve the MHT hypothesis selection by solving an assignment problem that chooses the best hypothesis, but only after costs have been assigned (statistically based) and hypotheses have been pruned [11]. Somewhat similarly, linear optimization has also been used to assist in the hypothesis selection process for the MHT [12]. Other optimization approaches aim to improve the MHT hypothesis selection process via Lagrangian relaxation [13].

More recently, Andriyenko and Schindler have proposed formulating the MTT problem as a minimization of a continuous energy in [14], and then again as a minimization of discrete-continuous energy in [15]. These algorithms work towards more accurate representations of the nature of the problem, but sacrifice interpretability for complexity in the process. Rather than formulating the problem in a way that lends easily to traditional global optimization methods, the authors leverage the use of optimization techniques to find strong local minima of their proposed energy objective and achieve strong results in doing so. However, this approach calls for the use of several parameters that require tuning and the work provides limited recommendations for such a tuning process. Additionally, these methods require initialization heuristics to begin the solving process, which is in itself complicated to implement and is not directly connected to the final optimization problem solved.

In this paper, we propose the use of mixed integer optimization (MIO) to formulate and solve the multi-target tracking problem. Although MIOs are generally thought to be intractable (NP-Hard), in many practical cases near optimal solutions and even optimal solutions to these problems are obtainable efficiently [16]. This is largely attributed to the fact that MIO solvers have seen significant performance improvements in recent years due to advancements in both methodology and hardware. The development of new heuristic methods, discoveries in cutting plane theory, and improved linear optimization methods have all contributed to improvements in performance [17]. Modern solvers such as Gurobi and CPLEX have been shown to perform extremely well on benchmark tests. In the past six years alone, Gurobi has seen performance improvements by a factor of 48.7 [18]. CPLEX saw improvements by a factor of 29,000 from 1991 to 2007 [19]. From 1994 to 2014, the growth of supercomputing power as recorded by the TOP500 list has improved by a factor of 567,839 [20]. Thus, the total combined effective improvement of software and hardware advancements is on the scale of 800 billion in the past 25 years [21].

The existing literature is additionally lacking in performance metrics for the evaluation of MTT algorithms. There is no standard method of measuring scenario complexity or algorithm performance as a function of this complexity. In many cases, only the sensor's detection noise is taken into account and other factors such as target density are negated. Recent work in [22] proposes a mathematically rigorous performance metric for measuring the distance between ground truth and estimated track, but little attention is given to the complexity of generated scenarios. In this paper, we also introduce measures of complexity and performance related to those suggested in [22], but we show the value in relating a complexity measure to performance measures, namely in that it allows us to evaluate the data association and trajectory estimation problems separately. We evaluate the methods suggested in this paper using these complexity and performance measures on two simulated experiments.

The main contributions of this paper are as follows:

(i) We introduce a simple interpretable MIO model that solves the data association and trajectory estimation problems simultaneously for a sensor with no detection ambiguity. The model does not require assumptions on data generation or any parameter tuning. Furthermore, this MIO is practically tractable in that it obtains high quality solutions in a reasonable amount of time for the considered applications.

(ii) We propose a simple local search heuristic, motivated by the optimization problem, which provides feasible solutions that can be used as warm starts for the MIO in order to improve the quality of the solutions obtained as well as the running time. This heuristic is highly scalable and parallelizable, solving in milliseconds.

(iii) We extend this basic MIO model and the corresponding heuristic algorithm to incorporate detection ambiguity, i.e., the case where there are both missed detections and false alarms. This extension maintains interpretability while only introducing two additional parameters, for which we provide general tuning guidelines.

(iv) We present a novel measure of complexity for the data association problem and propose a simplified measure of performance for the trajectory estimation problem.

The paper structure is outlined as follows. We begin with a description of the MTT problem in Section II, as this is the setting we aim to model for the entirety of the work. In Section III we introduce a simple MIO formulation for a sensor with no detection ambiguity and extend it to a generalized formulation. We then present a randomized local search heuristic in Section IV, which we use as a warm start for the MIO. In Section V we discuss extensions to both the MIO model and the heuristic for the case of detection ambiguity. In order to quantify the performance of our suggested methods, we develop metrics for measuring scenario complexity and algorithm performance in Section VI. Experimental methods and computational results are presented in Section VII, including results for scenarios both with and without detection ambiguity. Finally, we summarize our contributions and describe future work in Section VIII.

General Notations: Unless specified otherwise, $\|\cdot\|$ is used to indicate a norm, and $|\cdot|$ refers to element-wise absolute value.

II. Problem Description

In this paper, we restrict our exploration of the MTT problem to the automatic tracking of multiple, independent point targets using a single sensor. A *target* is the object of interest. A point target's only identifiable attributes are features of its state space, which we restrict to position and velocity. The state space fully defines the field of *trajectories*, or paths, along which targets travel. A *detection* is collected from each target at sequential scans and is subject to noise. We consider two general scenarios: with and without detection ambiguity.

When there is no detection ambiguity the sensor produces exactly one detection for each target in each scan, without any other source of detections. Therefore, the number of detections in each scan is exactly equal to the number of existing targets. Under these conditions, the data association problem reduces to a one-to-one assignment problem. Our basic optimization model, presented in Section III, addresses this variant of the MTT problem.

The presence of detection ambiguity results in a more complex case where the sensor both generates false alarms and misses detections. A *false alarm* occurs when a detection is collected but in fact no target exists. This could be the result of measurement error or difficulties in the sensor's signal processing. A *missed detection* occurs when a data point is not collected in a given scan where a target actually exists. Due to such ambiguity the number of detections in each scan could be higher or lower than the actual number of existing targets. Thus, the number of targets can not be immediately deduced from the number of detections. Under these conditions each detection can be assigned to either a target, in the same manner as before, or classified as a false alarm. Furthermore, we wish to identify the location (scan and target ID) of a missed detection. In Section V we present extensions of our basic optimization model to a robust formulation that deals with this detection ambiguity, which we will refer to as the robust MIO model.

Throughout the paper we make the following assumptions:

Assumption 1.

(i) All targets have constant velocity. i.e., targets do not maneuver and no outside forces act on them.
(ii) Each target's dynamics are independent of any other target's dynamics.
(iii) The number of targets remains constant throughout the window of observation, i.e., there is no birth/death of targets.
(iv) The detection errors are independent of one another.

Notation: We observe P targets over a fixed time window in which T scans are collected. Without loss of generality, and for ease of notation, we assume the scans arrive at a fixed rate of 1Hz, such that the set of scans can be time stamped by $\{1, 2, ..., T\}$. The i^{th} detection of the t^{th} scan is indicated by x_{it}, such that a scan of data at time *t* is the unordered set of detections $\mathcal{X}_t = \{x_{1t}, x_{2,t}, ..., x_{P,t}\}$. The data for the problem is the ordered set of scans $\boldsymbol{\mathcal{X}} = (\mathcal{X}_1, \mathcal{X}_2, ..., \mathcal{X}_T)$. The state space of target trajectories is paramaterized by a true initial position α_j^{true} and a true constant velocity β_j^{true}.

III. MIO Model

In this section, we deal with the case of no detection ambiguity. Therefore, we add the following, more restrictive assumptions, to those presented in Assumption 1:

Assumption 2.

(i) The sensor generates exactly one detection for each target in each scan i.e., no missed detections.
(ii) The sensor does not generate any spurious detections i.e., no false alarms.

A corollary to Assumption 2 is that the number of detections at each scan will be constant and equal to the number of targets. This seemingly simple point is critical to developing models in the case of no detection ambiguity. We begin constructing our MIO model by introducing decision variables that define data associations as well as estimated trajectories. Using these decision variables, we then develop an objective function that mathematically quantifies the value of the model solutions. Finally, we restrict the set of feasible solutions using constraints that force the model to find solutions that are suitable for the MTT problem as we have defined it. A simple model is first developed step by step in the coming sections before a generalized formulation is presented.

A. Decision Variables

The data association and trajectory estimation problems each require unique decision variables. Because these two problems lie in different domains, the variables we use to represent these decisions also differ. First, we introduce continuous decision variables $\alpha_j \in \mathbb{R}^n$ and $\beta_j \in \mathbb{R}^n$ to represent the estimated initial position and velocity, respectively, of each trajectory j. In our interpretation of the MTT problem, we

allow the trajectory parameters to lie anywhere in the real-continuous domain. For the data association problem, we wish to assign detections to trajectories, which is a naturally discrete problem. Therefore, we introduce binary decision variables y_{itj} to indicate whether detection x_{it} is assigned to trajectory j or not:

$$y_{itj} = \begin{cases} 1, & \text{if detection } x_{it} \text{ is assigned to trajectory } j, \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

Next, we use these decision variables to develop an objective function that accurately scores the solutions found by the model.

B. Objective Function

We would like to develop a function that quantifies the quality of a feasible solution. Toward this goal, we aim to establish a single measure for both the data association and the trajectory estimation problems, though we will construct the objective function in steps by considering each of these problems individually. Beginning with the estimation problem, we define the quality of an estimated trajectory as the distance between the estimated position of the trajectory and its associated detections. Let $\hat{x}_{jt}$ denote the estimated position of target j in scan t. Then the distance between detection x_{it} and target j in scan t is:

$$\|x_{it} - \hat{x}_{jt}\|,$$

which represents of measure of the quality of the estimation for trajectory j in scan t. The total estimation quality for a trajectory is then given by:

$$\sum_{j=1}^{P}\sum_{t=1}^{T}\left\|\sum_{(i,j)\in\mathcal{A}_t} x_{it} - \hat{x}_{jt}\right\|, \quad (2)$$

where $\mathcal{A}_t$ is the set of pairs of detection-trajectory associations for scan t.

We can now separate the problem into two parts: 1) given a set of associations, find the estimated trajectories that minimizes (2), and 2) find the set of associations that result in the best estimated trajectories. Recall that each trajectory is defined by two parameters, α_j^{true} and β_j^{true}, such that the true position of target j in scan t is given by:

$$\bar{x}_{jt} = \alpha_j^{\text{true}} + \beta_j^{\text{true}} t. \quad (3)$$

Thus, an estimated trajectory can be analogously defined by α_j and β_j such that its estimated location at the time of scan t is given by:

$$\hat{x}_{jt} = \alpha_j + \beta_j t. \quad (4)$$

Therefore, given a complete set of associations $\mathcal{A} = (\mathcal{A}_1, \ldots, \mathcal{A}_T)$, the trajectory with the best estimation error is given by the solution to the following optimization problem:

$$\underset{\alpha_j,\beta_j}{\text{minimize:}} \sum_{j=1}^{P}\sum_{t=1}^{T}\left\|\sum_{(i,j)\in\mathcal{A}_t} x_{it} - (\alpha_j + \beta_j t)\right\|. \quad (5)$$

Notice that under the current assumptions, in which there is no detection ambiguity, (5) represents the cost of the association set $\mathcal{A}$.

Now we turn to the problem of choosing the associations, based on this measure. To this end we formulate the assignment cost (5) in terms of our decision variables for the association problem. Note that $(i,j) \in \mathcal{A}_t$ if and only if $y_{itj} = 1$. Thus,

$$\sum_{(i,j)\in\mathcal{A}_t} x_{it} = \sum_{i=1}^{P} y_{itj} x_{it}, \quad (6)$$

holds because all detections will be associated to a target and vice versa under Assumption 2. Making the appropriate substitutions, the cost of an assignment described by variables y_{itj} is given by:

$$\left\|\sum_{i=1}^{P} y_{itj} x_{it} - (\alpha_j + \beta_j t)\right\|. \quad (7)$$

Therefore, in order to find the assignment with the lowest cost, we are left to minimize cost (7) over all assignments, and we obtain the following final objective:

$$\underset{y_{itj},\alpha_j,\beta_j}{\text{minimize:}} \sum_{j=1}^{P}\sum_{t=1}^{T}\left\|\sum_{i=1}^{P} y_{itj} x_{it} - (\alpha_j + \beta_j t)\right\|. \quad (8)$$

At this point it is necessary to discuss the advantages and disadvantages of the two natural distance measures (norms) that will be considered: the ℓ_1 and the ℓ_2 norms. The ℓ_1 norm has the advantage that it can be reformulated using linear optimization (through the addition of continuous variables and constraints), and it is well known to be more robust to outliers. Furthermore, existing algorithms for MIO are better developed for linear rather than quadratic optimization. However, the ℓ_2 norm squared form, which is equivalent to the residual sum of squares (RSS), has the advantage that it can be quickly computed using simple linear algebra, making it more amenable to a heuristic. This concept will be discussed further in Section IV.

Because of the computational benefits of linear optimization over quadratic optimization, we choose to formulate the objective function to the MIO model using the ℓ_1 norm. This allows us to reformulate (8) using linear optimization by introducing continuous variables $\psi_{jt} \in \mathbb{R}^n$ and the following two constraints:

$$\sum_{i=1}^{P} y_{itj} x_{it} - \alpha_j - \beta_j t \le \psi_{jt}, \qquad \forall j, t, \quad (9)$$

$$-\left(\sum_{i=1}^{P} y_{itj} x_{it} - \alpha_j - \beta_j t\right) \ge \psi_{jt} \qquad \forall j, t. \quad (10)$$

The resulting objective function for the case of the ℓ_1 norm would then be:

$$\underset{\psi_{jt}}{\text{minimize:}} \sum_{j=1}^{P}\sum_{t=1}^{T}\psi_{jt}. \quad (11)$$

C. Constraints

In addition to the constraints used to linearize the objective function, we also require standard assignment constraints to ensure that only one detection is assigned to each target in each scan and vice versa. Specifically, for each target and each scan, each detection must be assigned to exactly one target *j*:

$$\sum_{j=1}^{P} y_{itj} = 1 \qquad \forall i, t. \tag{12}$$

Similarly, for each scan, each target must be assigned exactly one detection:

$$\sum_{i=1}^{P} y_{itj} = 1 \qquad \forall j, t. \tag{13}$$

D. Overall Formulation

Integrating all of these elements together, we arrive at the following MIO model:

$$\begin{aligned}
\underset{\psi_{jt}}{\text{minimize:}} \quad & \sum_{j=1}^{P}\sum_{t=1}^{T} \psi_{jt} && (14)\\
\text{subject to:} \quad & \sum_{j=1}^{P} y_{itj} = 1 \qquad \forall i, t \\
& \sum_{i=1}^{P} y_{itj} = 1 \qquad \forall j, t \\
& \sum_{i=1}^{P} y_{itj} x_{it} - \alpha_j - \beta_j t \leq \psi_{jt} \qquad \forall j, t \\
& -\left(\sum_{i=1}^{P} y_{itj} x_{it} - \alpha_j - \beta_j t\right) \geq \psi_{jt} \qquad \forall j, t \\
& y_{itj} \in \{0, 1\} \quad \forall i, t, j \\
& \alpha_j \in \mathbb{R}^n \quad \forall j, \quad \beta_j \in \mathbb{R}^n \quad \forall j.
\end{aligned}$$

This formulation is simple in the sense that it involves few variables and constraints, making it highly interpretable and easily implementable. However, it has the disadvantage of being ill suited for extensions to detection ambiguity because it heavily relies on the fact that exactly one of the detections in each scan is associated to a target, which implies:

$$\sum_{i=1}^{P} y_{itj} x_{it} = x_{it}, \tag{15}$$

will always hold true. However, in the case of detection ambiguity, (15) no longer holds true since there might be trajectories that are not associated with a detection in a given scan. Therefore, in the following section we present a generalized formulation, which is amenable to scenarios with detection ambiguity.

E. Generalized Formulation

Here we modify (14) so that it can be easily extended to handle false alarms and missed detections that occur in the case of detection ambiguity. We previously identified that (14) cannot extend to handle detection ambiguity because (15) will no longer hold true. Therefore, we seek an alternate method of representing the objective function. Toward this goal, we introduce a new continuous variable z_{jt}, and add the following constraint to the model:

$$M_t(1 - y_{itj}) \geq |z_{jt} - x_{it} y_{itj}| \qquad \forall i, t, j. \tag{16}$$

where $M_t = \max_i |x_{it}|$ for each scan. This constraint forces z_{jt} to take the value of x_{it} when $y_{itj} = 1$ and some arbitrary number when $y_{itj} = 0$.

$$z_{jt} = \begin{cases} x_{it}, & \text{if } y_{itj} = 1, \\ \textit{free}, & \text{otherwise.} \end{cases}$$

In the case of no detection ambiguity, $\sum_{i=1}^{P} y_{itj} = 1$ will always hold true as forced by (13). Thus, we recover (8) exactly because z_{jt} will always take on the value of exactly one of the x_{it}. Thus, the resulting alternate objective function is:

$$\underset{z_{jt}, \alpha_j, \beta_j}{\text{minimize:}} \quad \sum_{j=1}^{P}\sum_{t=1}^{T} \|z_{jt} - \alpha_j - \beta_j t\|. \tag{17}$$

This objective can be linearized in the same fashion as (8) by again introducing continuous variables ψ_{jt} and additional constraints as follows:

$$\underset{\psi_{jt}}{\text{minimize:}} \quad \sum_{j=1}^{P}\sum_{t=1}^{T} \psi_{jt} \tag{18}$$

$$z_{jt} - \alpha_j - \beta_j t \leq \psi_{jt} \qquad \forall i, j, t,$$

$$-(z_{jt} - \alpha_j - \beta_j t) \geq \psi_{jt} \qquad \forall i, j, t.$$

Also note that we can linearize (16) by substituting it for the following two linear constraints:

$$x_{it} y_{itj} + M_t(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j,$$

$$x_{it} y_{itj} - M_t(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j.$$

Again, we consolidate these elements together and arrive at the following generalized MIO model:

$$\begin{aligned}
\underset{\psi_{jt}}{\text{minimize:}} \quad & \sum_{j=1}^{P}\sum_{t=1}^{T} \psi_{jt} && (19)\\
\text{subject to:} \quad & \sum_{j=1}^{P} y_{itj} = 1 \qquad \forall i, t \\
& \sum_{i=1}^{P} y_{itj} = 1 \qquad \forall j, t \\
& x_{it} y_{itj} + M_t(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j \\
& x_{it} y_{itj} - M_t(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j \\
& z_{jt} - \alpha_j - \beta_j t \leq \psi_{jt} \qquad \forall i, j, t \\
& -(z_{jt} - \alpha_j - \beta_j t) \geq \psi_{jt} \qquad \forall i, j, t \\
& y_{itj} \in \{0, 1\} \quad \forall i, t, j \\
& \alpha_j \in \mathbb{R}^n \quad \forall j, \quad \beta_j \in \mathbb{R}^n \quad \forall j, \quad z_{jt} \in \mathbb{R}^n \quad \forall j, t.
\end{aligned}$$

Note that (14) and (19) are exactly identical formulations when detection ambiguity does not exist. Throughout the remainder of this paper, we refer to (14) as the *basic* MIO. In Section V we will extend (19) to account for false alarms and missed detections when detection ambiguity exits.

IV. Local Search Heuristic

In this section, we present a detailed description of a local search heuristic that finds high quality feasible solutions to (19), which we call the basic MIO model. These solutions can be used as a warm start to the basic MIO model, providing a performance boost in run time and improved solution quality. The heuristic utilizes randomized local search methods to find locally optimal solutions. Fundamentally, randomized local search methods begin with a random initial starting point and converge to a locally optimal solution through local improvements. Applying this scheme to a growing number of randomized starting points increases the probability of reaching high quality solutions or even the globally optimal solution.

We begin by describing the heuristic mechanism for a single starting point. A *starting point* is a randomized solution to the association problem that satisfies the assignment equations (12) and (13). To generate a random starting point, associations are randomly assigned to detections for each scan, in such a way that each ordered association, or permutation, has an equal probability of occurring. The assignment cost, temporarily denoted by f, of this starting point is then calculated by solving (7) for all scans of each trajectory. After initializing with a starting point, the heuristic begins a *sweep* through the scans, i.e., a single pass through the scans. At each scan of the sweep two detections are randomly selected from the current scan and their assignments are exchanged, an operation referred to as a *swap*. this process generates a new feasible solution. The assignment cost of this new solution is calculated, and the swap is kept if the cost of the new solution improves. Otherwise the swap is rejected. The heuristic continues to conduct sweeps until a full sweep is completed without accepting a single swap, at which point it terminates. Pseudocode for this local search heuristic, referred to as the *basic* heuristic, is provided in Figure 1.

Algorithm 1 Randomized local search with heuristic swaps

Input: $\boldsymbol{\mathcal{X}}$, P, T
Output: f, y
Initialization : Assign random initial assignment to y^0
Calculate $\alpha_j, \beta_j \quad \forall j$
Calculate f^0 - the cost of assignment y^0
swapped $\leftarrow true$
$k \leftarrow 1$
while swapped **do**
  swapped $\leftarrow false$
  for t in $\{t_1, t_2, ..., T\}$ **do**
    Randomly choose $j, m \in \{1, \ldots, P\}$
    Find i, l such that $y^{k-1}_{itm} = 1$ and $y^{k-1}_{ltj} = 1$
    Swap such that $y^k_{itj} \leftarrow 1$, $y^{k-1}_{itm} \leftarrow 0$, $y^k_{ltm} \leftarrow 1$ and $y^k_{ltj} \leftarrow 0$
    Calculate f^k the cost of assignment y^k as well as $\alpha_j, \beta_j, \alpha_m, \beta_m$
    if $(f^k \geq f^{k-1})$ **then**
      $y^k \leftarrow y^{k-1}$
    else
      swapped $\leftarrow true$
    end if
  end for
  $k \leftarrow k + 1$
end while
return f^k, y^k

Fig. 1: Pseudocode for heuristic for a single starting point.

The goal of any heuristic is to find good feasible solutions in an efficient manner. In particular, the goal of our heuristic is to find good feasible solutions that can serve as a warm start for the basic MIO model. In Section III we discussed our choice to use the ℓ_1 norm over the ℓ_2 norm for use in the objective function of our MIO models. We now turn to discuss why the ℓ_2 norm is the preferred choice for use in this heuristic. The two main areas of concern are 1) efficiency of the algorithm and 2) quality of the solution.

In the case of the MIO models, the preferred objective function utilizes the ℓ_1 norm because it lends itself easily to linear optimization solvers which have known performance advantages over quadratic optimization solvers. However, in the case of the heuristic, the objective function no longer needs to determine the associations because they are predetermined by the initialization and swapping processes. In addition, the heuristic objective score needs to be recomputed after each swap, even if it is eventually rejected, and hundreds to thousands of swaps may be carried out for a single starting point. This fact makes the computational cost of this calculation critical to the scalability of the heuristic. Computing the ℓ_1 norm objective would require solving an linear optimization problem. Even though this linear optimization problem can be computed quite quickly by state of the art optimization solvers, the ℓ_2 norm squared, or RSS, can be computed by simple linear algebra in a fraction of the time, as shown in [23]. Therefore, with respect to efficiency, the ℓ_2 norm is the clear choice for use in the local search heuristic.

In judging the quality of a heuristic solution, it is important to remember that it will serve as a warm start for the MIO, which uses the ℓ_1 norm in its objective. It is natural to assume that using the same norm for the heuristic objective would lead to higher quality solutions. While this holds true, we found that both norms find good quality solutions, likely due to the fact that they both represent measures of distance, making them highly correlated. Thus, the choice of the ℓ_2 over the ℓ_1 norm might not significantly degrade the solution quality. Although the use of ℓ_2 norm runs the risk of obtaining solutions which are not necessarily extremely good solutions for the ℓ_1 norm objective the potential loss in solution quality is far outweighed by the guaranteed efficiency improvements afforded by the ℓ_2

norm.

Furthermore, the local search heuristic can be parallelized by running unique starting points on separate processors, enabling the ability to run a larger number of starting points without increasing the run time. Increasing the number of starting points greatly increases the potential for the heuristic to improve solution quality and thus further reducing the risk of reduced solution quality. Therefore, we make the choice to use the ℓ_2 norm in the objective function of the heuristic.

V. Extensions to Detection Ambiguity

We transition to treat the case of detection ambiguity. Specifically, we now allow for false alarms, the instance in which a detection is triggered when no target exists, and missed detections, the instance in which a target exists but no detection is generated. Consequently, the number of detections at each scan varies, and the number of targets we wish to track becomes ambiguous. To define this explicitly, we introduce additional notation for the case of detection ambiguity. Let n_t be the number of detections at scan t. We denote:

$$N_0 = \min_t \, n_t,$$

as the largest number of detections across all scans. Similarly, we denote:

$$N_1 = \max_t \, n_t,$$

as the smallest number of detections across all scans. In the case of detection ambiguity we only assume that the true number of targets falls somewhere in the range of $[N_0, N_1]$. Specifically, we replace Assumption 2 with the following less restrictive assumption.

Assumption 3.

(i) The sensor generates at most one detection for each target in each scan i.e., there can be missed detections.

(ii) The sensor can generate detections that do not originate from any target i.e., there can be false alarms.

(iii) The number of true targets P satisfies $N_0 \leq P \leq N_1$.

Several new challenges emerge in the case of detection ambiguity. First, we need to estimate the number of targets. We denote the number of estimated targets as P_{est}. Second, we aim to identify detections as false alarms in addition to assigning them to targets, however, these cannot occur simultaneously for a single detection. Finally, we wish to identify the scans in which a particular target was not detected.

We identify two approaches for estimating the number of targets. In the first approach, the number of targets can be determined via the optimization model itself, through the use of additional decision variables and constraints. Alternatively, the second approach formulate a variant of the MIO model that assumes a fixed number of targets P and then uses the power of parallelization to run that model with all possible values of P which satisfy Assumption 3. At the completion of this process, we choose the solution for P that minimizes the total objective function value. These two approaches are equivalent, in the sense they have the same optimal solution set.

However, as it turns out, the first approach leads to a model that is not tractable for practical use, while the second approach yields smaller, more tractable models, which as we stated have the added benefit that they can be solved in parallel. Therefore, we turn our focus to discuss the second approach and defer the interested reader to Appendix A for a complete discussion of the first approach. In this section, we extend our basic MIO model to the case of detection ambiguity before discussing necessary adaptations to the basic heuristic, which will also assume a fixed number of targets and provide good quality warm start solutions to its corresponding MIO.

A. Robust MIO with Fixed Number of Targets

In this section, we extend (19) to account for missed detections and false alarms through the addition of new decision variables and constraints. The objective function is also updated to reflect the necessary additions.

1) Decision Variables: We need to establish new variables for identifying false alarms as well as missed detections. Toward this goal, we introduce new binary decision variables F_{it} to indicate whether or not a detection x_{it} is a false alarm.

$$F_{it} = \begin{cases} 1, & \text{if detection } i \text{ at time } t \text{ is a false alarm,} \\ 0, & \text{otherwise.} \end{cases}$$

Likewise, we introduce new binary decision variables M_{jt} to indicate whether or not trajectory j has a missed detection at time t.

$$M_{jt} = \begin{cases} 1, & \text{if detection for trajectory } j \\ & \text{at time } t \text{ is a missed detection,} \\ 0, & \text{otherwise.} \end{cases}$$

2) Objective Function: We can easily extend (17) to account for detection ambiguity by introducing penalties θ and ϕ for each missed detection and false alarm, respectively. This implies a linear penalty function, meaning that each missed detection (false alarm) contributes the same penalty to the objective function. Therefore, we simply need to penalize the total number of false alarms and missed detections in the objective function. If we denote the total number of false alarms TF and total number of missed detections TM, the resulting objective function takes the form:

$$\underset{\psi_{jt}}{\text{minimize:}} \; \sum_{j=1}^{P} \sum_{t=1}^{T} \psi_{jt} + \theta \cdot TF + \phi \cdot TM. \tag{20}$$

As a general rule, we would expect to increase θ or ϕ, as the number of expected false alarms or missed detections decreases, respectively. A more exhaustive discussion on the insight behind these penalties, in addition to recommendations for tuning them can be found in Appendix A.

3) Constraints: Finally, we restrict the set of feasible solutions to satisfy Assumption 3. This is accomplished by simply modifying (12) and (13) to account for false alarms and missed detections, respectively. Specifically, all detections must either be assigned to a trajectory j or to a false alarm:

$$\sum_{j=1}^{P} y_{itj} + F_{it} = 1 \qquad \forall i, t. \tag{21}$$

All trajectories *j* must either be assigned a detection or a missed detection:

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \qquad \forall j, t \tag{22}$$

Here we see why it was necessary to generalize the simple model to (19). When $M_{jt} = 1$, we have $\sum_{i=1}^{n_t} y_{itj} = 0$ by (22). Therefore (16) does not restrict z_{jt} at all and it is obvious, from the structure of the objective function, that in this case $z_{jt} = \alpha_j - \beta_j t$ is the optimal solution, since it results in no change to the objective score (no estimation penalty), which is precisely the desired effect. To state this explicitly, we have:

$$z_{jt} = \begin{cases} x_{it}, & \text{if } y_{itj} = 1, \\ \alpha_j - \beta_j t, & \text{otherwise,} \end{cases}$$

in the case of detection ambiguity.

Lastly, to properly penalize false alarms and missed detections in the objective function, we must force TF (TM, respectively) to equal the sum of all false alarms (missed detections, respectively).

$$\sum_{i=1}^{n_t} \sum_{t=1}^{T} F_{it} = TF$$

$$\sum_{j=1}^{P} \sum_{t=1}^{T} M_{jt} = TM \tag{23}$$

4) Full Formulation: Merging all of these elements together we arrive at our MIO model for a case of detection ambiguity and a fixed number of targets.

$$\begin{aligned} g(P) = \underset{\psi_{jt}}{\text{minimize:}} \; & \sum_{j=1}^{P} \sum_{t=1}^{T} \psi_{jt} + \theta \cdot TF + \phi \cdot TM \qquad (24) \\ \text{subject to:} \; & \sum_{j=1}^{P} y_{itj} + F_{it} = 1 \qquad \forall i, t \\ & \sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \qquad \forall j, t \\ & \sum_{i=1}^{n_t} \sum_{t=1}^{T} F_{it} = TF \\ & \sum_{j=1}^{P} \sum_{t=1}^{T} M_{jt} = TM \\ & x_{it} y_{itj} + M_t(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j \\ & x_{it} y_{itj} - M_t(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j \\ & z_{jt} - \alpha_j - \beta_j t \leq \psi_{jt} \qquad \forall j, t \\ & -(z_{jt} - \alpha_j - \beta_j t) \leq \psi_{jt} \qquad \forall j, t \\ & y_{itj} \in \{0, 1\} \quad \forall i, t, j \\ & \alpha_j \in \mathbb{R}^n \quad \forall j, \quad \beta_j \in \mathbb{R}^n \quad \forall j \\ & z_{jt} \in \mathbb{R}^n, \quad \forall j, t. \end{aligned}$$

We refer to this model as the *robust* MIO. Note that P_{est} would then be given by:

$$P_{\text{est}} = \min_{N_o \leq P \leq N_1} g(P),$$

and the trajectories and assignments will correspond to the optimal solution of the corresponding model.

B. Heuristic with Fixed Number of Targets

The local search heuristic for scenarios with detection ambiguity follows closely from the heuristic developed under the scenario without ambiguity, with a few key differences. In the first place, the process for establishing a random starting point during the initialization process requires refinement. Initial solutions should allow for false alarms and missed detections. With equal probability, each detection is randomly classified as a false alarm or selected to receive a target assignment. If a detection has been identified for assignment, assignments are randomly selected uniformly across all targets. The remaining scans for which trajectories do not receive an assignment are identified as missed detections.

Once a starting point has been initialized, the heuristic progresses in much the same manner as before. Again, it sweeps through all scans continuously making random swaps. However, the swapping process must be adjusted to include the addition of false alarms and missed detections. Specifically, the robust heuristic randomly chooses from the following options when making swaps:

1) Switch detection assignments between two targets.
2) Switch the detection assignment of a target with a false alarm.
3) Switch the detection assignment of a target with a missed detection for a different target.
4) Move the detection assignment of a target to a false alarm and replace it with a missed detection.
5) Move a false alarm into the position of a missed detection for a target

Similar to the basic heuristic, this robust extension will accept the switch/move if the objective score improves, and reject the switch/move otherwise, and the heuristic terminates once it completes a full sweep without accepting a single swap. The framework of this new heuristic, also referred to as the *robust* heuristic, is identical to the one presented in Figure 1, barring the appropriate modifications to the initialization and swapping steps outlined above.

Due to the increase in potential combinations of solutions, we expect this variant of the heuristic to run slower. Though this effect can be mitigated by the fact that this variant is just as parallelizable as the basic heuristic. Next, we shift our discussion to focus on measuring the complexity of scenarios and the performance of our approaches.

VI. Scenario Complexity & Performance Metrics

In order to measure the quality of a solution obtained by any MTT algorithm one must have measures of both performance and scenario complexity. Unfortunately, as stated in [1], a unified approach for measuring scenario complexity does not exist, nor is there any clear measure of performance for either the trajectory estimation or the data association problem. In this paper, we argue that the data association problem has a natural performance metric but lacks a measure of complexity, while the trajectory estimation problem has a natural measure of complexity but lacks a clear performance metric. Thus, we construct the missing measures for each, respectively.

Intuitively, one can assume that the difficulty of a scenario is highly correlated with a sensor property that quantifies the deviation of the detections from the true targets. Therefore we first define σ to be a measure of this sensor property that quantitatively captures the noise in the sensor detections, which in most cases is the standard deviation of the detection error. We will show how to use σ to define complexity measures for different scenarios.

A. Data Association

In the case of the data association problem, the preferred performance metric often used in practice is % accuracy, *i.e.*, the number of correct detection assignments out of the number of possible correct assignments. For the case without sensor ambiguity, the number of possible assignments is simply the total number of detections, or equivalently, the number of targets multiplied by the number of scans:

$$Accuracy = \frac{\text{\# correct assignments}}{\text{Total \# of detections}} = \frac{\text{\# correct assignments}}{PT}.$$

In the case of sensor ambiguity, however, the number of possible correct assignments requires a deeper explanation. To develop a better understanding we consider our goal, which is to correctly assign detections to targets and identify both false alarms and missed detections. With this in mind, we define the number of possible correct assignments as the number of targets multiplied by the number of scans plus the number of false alarms:

$$Accuracy = \frac{\text{\# correct assignments}}{PT + \text{\# False Alarms}}.$$

Whereas accuracy serves as a good measure of performance for data association, there does not exist a corresponding measure of complexity that comparatively measures the difficulty of the data association problem. We argue that σ alone is not the best measure of difficulty for the data association problem. For example, in a scenario with very close target trajectories it may be difficult to ascertain data associations even for small σ values, and similarly with high enough σ values even widely spaced targets could be difficult to differentiate. Therefore, we introduce a metric ρ to quantify this complexity. For ease of notation in developing this metric we first define D_{ijt} as the distance between one true trajectory i and another true trajectory j in scan t:

$$D_{ijt} = \|\alpha_i^{\text{true}} + \beta_i^{\text{true}} t - \alpha_j^{\text{true}} + \beta_j^{\text{true}} t\|.$$

Additionally, we define a variable c_{ijt} that will take the value of 1 if the distance between trajectory *i* and trajectory *j* in scan *t* is greater than some monotonically increasing function of σ which we will denote by $h(\sigma)$:

$$c_{ijt} = \begin{cases} 1, & \text{if } D_{ijt} > h(\sigma), \\ 0, & \text{otherwise.} \end{cases}$$

Then the difficulty of a scenario in the sphere of the data association problem is quantified by the complexity measure

ρ, which is the proportion of detection pairs that fall within a closely defined proximity of each other:

$$\rho = \frac{\sum_{t=1}^{T} \sum_{i<j} c_{ijt}}{\binom{P}{2} T}.$$

This metric has several desirable attributes. First and foremost, it falls within the range of $[0, 1]$, which is identical to the range of accuracy and makes it easily comparable. Second, it is straightforward to understand and interpret. Higher values of ρ indicate easier scenarios because fewer targets are within close proximity for a shorter amount of time, and vice versa. Finally, as we have defined it, ρ has an inverse relationship with σ, which means that it serves as a connection between the scenario generation and performance measuring processes. While σ can be used more naturally for scenario generation, where it is useful as a parameter for signal noise, ρ can be calculated after the fact and used to quantify the difficulty of the scenario as it pertains to the data association problem.

B. Trajectory Estimation

In the case of the trajectory estimation problem, the preferred complexity metric often used in practice is σ itself. Increasing the noise may often lead to stronger bias in the trajectory estimation, especially in scenarios with fewer scans, and results in a deteriorated quality of the estimation. Therefore, we believe that σ is the correct metric for use in measuring the difficulty of the trajectory estimation problem.

However, establishing a performance metric for the trajectory estimation problem is necessary. We choose to apply a metric that captures the core goal of this problem: to estimate a trajectory as close as possible to the true ground track. Therefore, we use the following metric as a measure of this error:

$$\delta = \frac{\sum_{t=1}^{T} \sum_{j=1}^{P} \|\bar{x}_{jt} - \hat{x}_{jt}\|}{PT}. \tag{25}$$

Lower values of δ correspond to higher performance because the distance between the estimated and true ground trajectories is smaller. We match the true trajectories to the estimated trajectories using an one-to-one assignment problem that can be formulated using linear optimization. Note that in the case of detection ambiguity, P is unknown, and so we may generate either more or less trajectories than the true number. Thus when assigning the trajectories to each other, we must take P in equation (25) to be the minimum between the true number of targets and the estimated number of targets. See Appendix C for more details and a complete formulation of the assignment problem.

In the next section, we will see how these measures of complexity and performance are useful in quantifying the strengths and weaknesses of our methods.

VII. Experimental Simulations & Computational Results

We evaluate our approaches on a wide variety of simulated scenarios and compare the results against two benchmark solutions. As a first benchmark we randomly generate detection assignments, including randomly assigning false alarms and missed detections in the case of detection ambiguity. We will refer to this solution as the *random* solution. In the second benchmark the detection assignments are perfectly known, meaning that all assignments are exactly correct including the classification of false alarms and identification of missed detections. This solution is referred to as the *ideal* solution; while it is only ideal with respect to the data association problem, this solution provides a means for bounding the expected error in the trajectory estimation problem. Note that we do not compare our methods to any known MTT algorithms, such as the MHT or JPDAF, due to the complexity in parameter tuning and the overhead of implementation of these algorithms.

The existing literature does not currently propose a clearly defined comprehensive set of standard test scenarios, as pointed out in [1] where it is noted that two particularly important types of scenarios include crossing trajectories and parallel trajectories. In order to test our methods across scenarios with a wide range of complexity, for both the data association and trajectory estimation problems, it is necessary to create scenarios that capture these key types. With this in mind, we choose to generate scenarios of both trajectory types using a simple methodology that will be outlined in our discussion on experimental methods.

We run two separate experiments, one with detection ambiguity and one without. Both experiments, including the scenario generation process, heuristic, and MIO, were implemented in the development software *julia* 0.4.3 [24] using the optimization package *JuMP* [25]. The optimization software Gurobi 6.5.0 [26] was used to solve the MIOs and the optimization process was restricted to the use of a single core. Each simulation was run on a single compute node of the unclassified TX-Green cluster located at Lincoln Laboratories. The cluster utilizes DL165 G7 compute nodes consisting of 2.2 GHz compute cores with 8 GB of RAM each, for a total peak performance of 77.1 TFLOPS [27].

We begin by outlining our experimental methods for scenarios without detection ambiguity and discuss the results of our approaches on these scenarios.

A. Scenarios without Detection Ambiguity

In order to evaluate the scalability of our algorithms we test our methods across a range of scenarios with varying numbers of targets and scans. In particular, we consider $P \in \{4, 6, 8, 10\}$ targets and $T \in \{4, 6, 8, 10\}$ scans, where the scans are collected at a rate of 1 Hz. The cartesian product of P and T creates sixteen unique scenario sizes. We generate ten unique crossing scenarios and ten unique parallel scenarios of each size.

To generate trajectories, we first establish a state space as the segment $[-\tau, \tau]$. For our experiments, we elected for $\tau = 20$. Two points within this state space are selected to define a trajectory, where the first is referred to as the trajectory's *initial position* and the second as the trajectory's *final position*. To generate crossing trajectories, the initial and final positions are

randomly selected from within the full range of the state space. To generate parallel trajectories, the state space is divided into P equal non overlapping segments, such that within the ith segment we randomly select the initial and final positions for target i. This ensures the generation of trajectories that do not cross or overlap, but they remain within close proximity of one other.

For each scenario, we randomly generate ten realizations of data by first perturbing each true position measurement by an error $\epsilon \sim \mathcal{N}(0, \sigma)$, where σ represents the signal to noise ratio. We also refer to σ as the noise parameter, and we consider $\sigma \in \{0.1, 0.5, 1.0, 2.0, 3.5, 5.0\}$. Adding this detection error to the true position results in a detection:

$$x_{it} = \alpha_i^{\text{true}} + \beta_i^{\text{true}} t + \epsilon.$$

Scans $\mathcal{X}_t$ are simulated by randomizing the order of x_{it} for each t. Each unique $\boldsymbol{\mathcal{X}}$ generated is referred to as a *simulation*. For every simulation, we run the heuristic with each of the fixed number of starting points from the set $N \in \{100 \quad 1,000 \quad 10,000\}$, and then we use each of these solutions as a warm start for the MIO. The optimization process is set to terminate after 3T seconds, with solutions recorded at intervals of $\{1, T, 2T, 3T\}$ seconds.

At the conclusion of the experiment we calculate the difficulty of each scenario, the accuracy of each solution, and the trajectory estimation error δ. When measuring the difficulty of scenarios in terms of ρ, we propose the use of $h(\sigma) = 2\sigma$ because it is difficult to distinguish between detections originating from target trajectories that are closer together.

1) Scenario Generation: We begin with a discussion of the relationship between ρ and σ and demonstrate how this relationship benefits both scenario generation and complexity measurement by allowing each to occur in its own natural domain. Figure 2 shows the relationship between σ and ρ for the twenty scenarios simulated in our experiments. The plot is broken down by scenario type between crossing and parallel trajectories.

Note that higher values of ρ indicate a lower proportion of detections within very close proximity to one another. This plot shows that the parallel method of scenario generation clearly generates easier scenarios, as measured by ρ. This suggests that it may be more difficult to discern correct associations for crossing scenarios than for parallel scenarios. In addition, we can conclude from Figure 2 that σ and ρ are highly correlated; increasingly higher values of σ correspond to increasingly lower values of ρ, which is intuitive based on the definition of ρ.

We also note that the range of ρ values corresponding to a single value of σ decreases as the number of targets increases. In other words, as the number of targets increases, it becomes more difficult to generate a wider variety of ρ values from a given σ. For example, in the crossing scenarios with $T = 10$ and $\sigma = 5$ in the case of four targets, the range of ρ is approximately $[0, 0.5]$, while in the case of ten targets it is limited to $[0.125, 0.375]$. This most likely results from using a fixed state space; as the number of targets increases, the density of trajectories also increases,

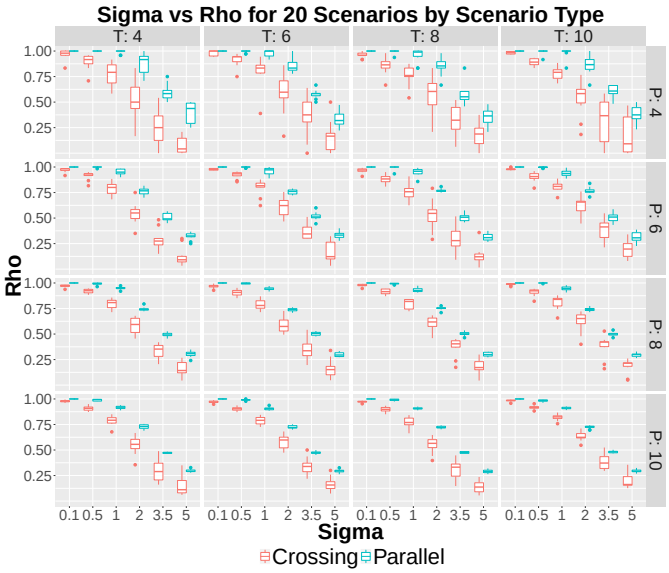

Fig. 2: Relationship between σ and ρ summarized by scenario type for all 20 generated scenarios in this experiment.

ultimately limiting the range of scenario complexity. Although the variety of difficulty measures decreases as the number of targets increases, the measure ρ still provides a meaningful quantification of difficulty for the data association problem.

2) Basic Heuristic: We now transition to evaluating the scalability of the heuristic run times. Table I summarizes the minimum, mean, and maximum run times of the heuristic for a single starting point, arranged by the number of targets (P) and number of scans (T). Times are shown in milliseconds.

		Basic Heuristic Run Times (in milliseconds)		
P	T	Min	Mean	Max
4	4	0.07	0.10	0.18
4	6	0.18	0.24	0.38
4	8	0.34	0.45	0.62
4	10	0.58	0.76	1.02
6	4	0.11	0.15	0.25
6	6	0.31	0.39	0.58
6	8	0.64	0.81	1.05
6	10	1.24	1.56	2.02
8	4	0.14	0.19	0.30
8	6	0.46	0.57	0.86
8	8	0.95	1.24	1.58
8	10	2.07	2.53	3.37
10	4	0.19	0.25	0.41
10	6	0.63	0.80	1.03
10	8	1.44	1.84	2.44
10	10	2.96	3.73	4.56

TABLE I: Heuristic run times (in milliseconds) for a single starting point.

Close examination shows that the heuristic scales more efficiently with increases in the number of targets than increases in the number of scans. For example, increasing from four to six targets for four scans increases the computational cost by 50%, while increasing from four to six scans for four targets increases the computational cost by 140%. The same trend holds true across all targets and scans. Although the heuristic

scales more efficiently in P than T, it does not exceed 5 milliseconds across all cases for any tested starting point.

The true scalability of the heuristic is fully realized when we consider the power of parallelization. By running the heuristic on several processors, we can reduce the number of starting points run on each processor and thus in turn reduce the total running time. As a result, given enough processors, the heuristic can run several thousand starting points and still find solutions in a fraction of a second. To illustrate this concept, consider the task of running 50,000 heuristic starting points for a scenario with six targets and six scans. The average run time for a single starting point of this size is about 0.4 milliseconds. Running all of these starting points in sequence would require approximately 20 seconds of total run time; however, those same starting points parallelized onto 100 processors would only require a run time of 0.2 seconds. Thus the run time of the heuristic can be reduced to meet the efficiency needs of the system, subject only to the limitation of available processors.

In order to determine the appropriate number of starting points for the heuristic we examine the MIO objective function value of the heuristic solutions for various numbers of starting points. Recall that regardless of the number of starting points, we will always choose the solution with the best objective function value. For ease of notation, let f_H be the MIO objective score of the heuristic solution and f_I be the MIO objective score of the ideal solution, which refers to the solution in which the data association problem is exactly correct. We then compute the ratio of f_H/f_I, providing us with a normalized measure for comparing the objective score across several scenarios. Figure 3 plots f_H/f_I (log scale) against σ.

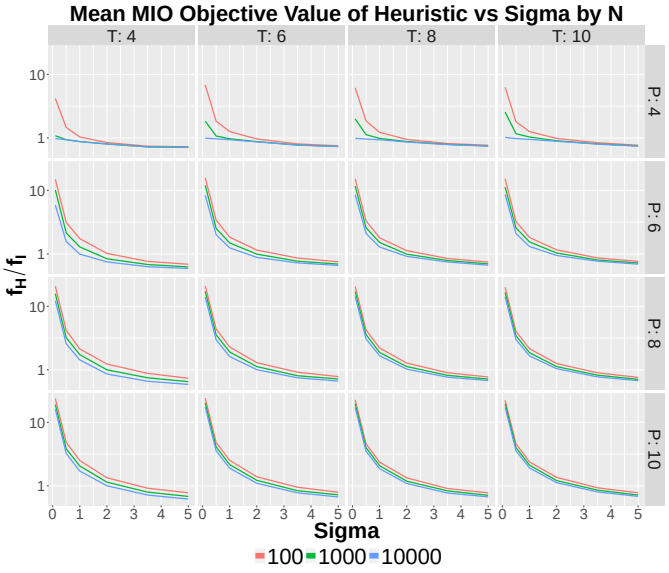

Fig. 3: Quality of heuristic solution as compared to the ideal solution's MIO objective value summarized by number of starting points.

In all cases, increasing the number of starting points improves the heuristic solution relative to the ideal solution. This improvement is greatest for scenarios of four targets, in which case the objective function value gain is on the order of about five times when N increases from 100 to 10,000 starting points. However, for scenarios with more targets, there is only slight improvement in the objective score with increases in the number of starting points, suggesting the need for higher value of N. This marginal gain is likely a result of the added computational complexity that grows exponentially as the number of targets increases. In any sense, we found that there is not a significant difference in heuristic performance for the range of N values that we explored. Therefore, for simplification as we move forward in our analysis, we will restrict our discussions of the heuristic to $N = 1,000$.

It is also interesting to note that in some instances the ratio f_H/f_I falls below the value 1, indicating that the heuristic outperforms the ideal solution as measured by the objective function. This occurs for higher values of σ, for example when σ is larger than 2, regardless of the values of P and T. To explain this phenomenon we recall that the ideal solution is only ideal in the sphere of data association, while the MIO objective function serves to provide a high quality solution that balances the tradeoff between data association and trajectory estimation. This suggests that it may be necessary to sacrifice some correct data associations in order to improve the trajectory estimation in cases where there is more noise in the detections.

With these points in mind we continue our analysis to evaluate solution quality for both the data association problem and trajectory estimation for both the heuristic and MIO approaches.

3) Data Association: We shift our focus to analyze the performance of both the basic heuristic and the basic MIO model in the context of the data association problem. Figure 4 plots the mean accuracy of both solutions against ρ, our measure of difficulty for data association. The heuristic shown on the plot was initialized with 1,000 starting points and its corresponding solution was provided to the MIO model as a warm start. Note that the results for the MIO model do not include the solution at $3T$ seconds because it exhibited little to no improvement over the MIO solution at $2T$ seconds. The ideal solution, which trivially always achieves an accuracy of 1.0, has also been excluded for the sake of clarity.

Figure 4 shows that the quality of the associations found by both the heuristic and the MIO are indeed highly correlated with the scenario complexity parameter ρ. The number of correct associations in both methods increase as ρ reaches 1, as we see in the cases of four and six targets, where the association is extremely close or exactly equal to the ideal association when ρ is equal to 1. By contrast, the random association approach is unaffected by the value of ρ and its accuracy is always lower than the heuristic's accuracy.

In general it seems that running the MIO for T or less seconds is optimal, since longer running times do not significantly improve the accuracy attained. Specifically, for scenarios of four targets the best accuracy is reached after 1 second, while for scenarios of eight and ten targets, the best accuracy is reached after T seconds.

The heuristic improves over the random solution in scenarios of all sizes, suggesting that the heuristic finds good quality

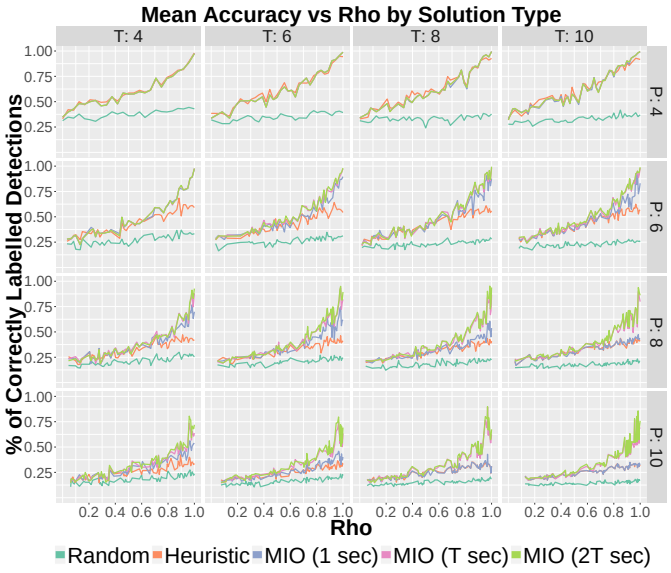

Fig. 4: Accuracy of MIO compared against the heuristic and a randomized solution.

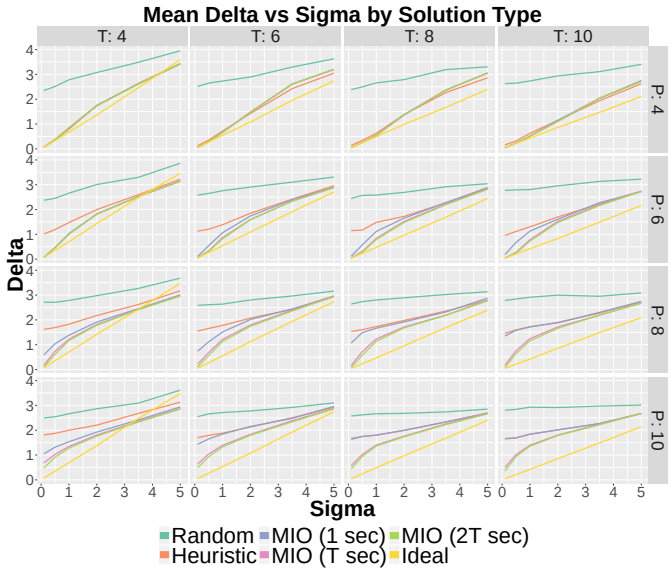

Fig. 5: Trajectory estimation performance

solutions. In fact, for scenarios of four targets, the heuristic performs as well as the MIO model across all numbers of scans. Although the heuristic does not perform as well as the MIO in scenarios with more targets, it still provides good solutions from which the MIO benefits. In particular, for larger scenarios with eight and ten targets, the MIO provides about a 30% improvement over the heuristic solution after T seconds. Moreover, it seems that as the number of targets increases, the accuracy deteriorates due to the added combinatorial difficulty of the association problem.

Both algorithms appear to scale well with increases in the number of scans. In fact, the accuracy of MIO solutions appears to improve as the number of scans grows. In the case of ten targets, the accuracy of the MIO after T (or $2T$) seconds is higher for eight and ten scans than for four and six scans, especially for high values of ρ. This suggests that the MIO benefits from additional scans, which is likely a result of the additional information gained from increasing the number of detections. For a fixed number of targets, there appears to be no change in heuristic solution quality as the number of scans changes.

4) Trajectory Estimation: We next evaluate the performance of the basic heuristic and MIO model through the lens of trajectory estimation. As previously discussed, we are interested in comparing δ, our performance measure proxy for ground track error, against σ, our measure of difficulty for trajectory estimation, in order to analyze the performance of our methods in the sphere of estimation. Figure 5 plots σ against δ for each of the previous solution types, in addition to the ideal solution.

Recall that lower values of delta correspond to trajectory estimations that are close to that of the true ground track. We note that as σ approaches zero, both the heuristic and MIO trajectory estimates approach the trajectory estimates of the ideal solution, suggesting that both algorithms find very high quality trajectory estimates in these cases. Similar to the data association problem, we find that the best estimates are reached after 1 second for scenarios with four targets and after T seconds for all other target scenarios.

The heuristic and MIO outperform the ideal solution in scenarios with only four scans and high values of σ. However, the algorithms do not outperform the ideal solution for larger numbers of scans. Moreover, δ of the ideal solution changes linearly with sigma, but the slope of this linear dependence decreases as the number of scans increases. This suggests that as the number of scans approaches infinity, the estimated trajectories given by the ideal data association will converge to the true ground tracks, even for large sigma values. To a lesser extent this is also true for the MIO and heuristic solutions, which is demonstrated by the fact they move further away from the ideal solution when the number of scans increases, showing the tradeoff between added information and computational complexity.

We find that the estimation errors of the MIO and heuristic solutions have a bounded distance from the error of the ideal solution, and that our methods always outperform the random solution even for large values of σ. This implies that although we do not necessarily obtain the optimal solution, the guiding objective prevents us from finding solutions that are very poor. In particular, the gap between the ideal estimation error and the heuristic/MIO estimation error remains relatively constant across all values of P for a fixed number of scans.

5) Summary of Results: We have shown that in the case of no detection ambiguity,

- The heuristic is highly scalable and we can find good quality solutions in fractions of a second using parallelization.
- Using the heuristic solutions as a warm start, the MIO achieves high quality solutions to both the data association and trajectory estimation problems after T or fewer seconds.

- The MIO is scalable with respect to increases in both the number of targets and scans.
- There exists a tradeoff between making correct data associations and improving trajectory estimation, particularly in cases of high signal to noise ratios.
- Increasing the number of scans while adding computational complexity to the model helps to obtain better solutions.

With these results in mind, we proceed to discussing the case of scenarios with detection ambiguity.

B. Scenarios with Detection Ambiguity

We now extend our discussion to analyze the performance of our methods in scenarios with detection ambiguity. We first summarize our experimental methods before discussing performance of both the robust heuristic and the robust MIO in the spheres of both the data association and trajectory estimation problems.

This experiment serves as an extension of the basic one in order to test the performance of our algorithms under detection ambiguity. We use the same scenarios generated from the basic experiment, however due to the additional difficulty inherent in the case of detection ambiguity, we limit the range of signal noise to $\sigma \in \{0.1, 0.5, 1.0, 2.0\}$, choosing to exclude the more extreme noise values. In addition, we simulate both missed detections and false alarms. A detection is removed with probability γ, where we consider $\gamma \in \{0.2, 0.15, 0.1, 0.05\}$. We do not allow empty scans. For each scan, we generate false alarms according to a Poisson distribution with parameter λ, where we consider $\lambda \in \{0.1, 0.5, 0.1, 2.0\}$. False alarm positions are selected uniformly at random from within the state space. The false alarms are then added to $\mathcal{X}_t$ and the detection order of $\mathcal{X}_t$ is randomly shuffled in the same manner as described in the first experiment.

After generating the data we follow the same sequence of steps as outlined for the basic experiment, running the heuristic first and then feeding the corresponding solution into the MIO as a warm start. Note that the heuristic is only initialized with 1,000 starting points, as determined from the results of the basic experiment. We terminate the optimization process after $2T$ seconds since the previous experiment showed no added benefit in running it for longer time periods. Solutions are recorded at intervals of $\{1, T, 2T\}$ seconds.

Prior to running this experiment we also performed a mini experiment and used its results to tune the penalties θ and ϕ; a summary of the exact penalties used, along with an explanation of the insight behind them, can be found in Appendix A. With respect to the penalties, it is important to note that while the estimation error and the number of missed detections are highly correlated with the number of targets, the number of false alarms are not; thus, the penalty θ is set to be linearly dependent on the number of estimated targets. In order to decide on a final estimate for the number of targets, we compared normalized performance scores, i.e., the MIO objective function divided by the estimated number of targets, and ultimately chose this estimate to be the number of targets with the best resulting ratio.

1) Robust Heuristic: Table II summarizes the minimum, mean, and maximum run times of the robust heuristic for a single starting point, arranged by the number of estimated targets (P_{est}) and number of scans (T). Times are shown in milliseconds.

		Robust Heuristic Run Times (in milliseconds)		
$P_{\text{estimated}}$	T	Min	Mean	Max
2	4	0.15	0.23	0.41
2	6	0.42	0.56	0.93
2	8	0.77	1.04	2.24
2	10	1.27	1.73	3.07
4	4	0.15	0.34	1.04
4	6	0.50	0.94	2.69
4	8	1.09	1.88	3.87
4	10	2.12	3.25	7.20
6	4	0.14	0.42	0.96
6	6	0.57	1.29	4.45
6	8	1.33	2.66	5.82
6	10	2.53	4.61	9.4
8	4	0.16	0.50	1.10
8	6	0.60	1.59	3.46
8	8	1.38	3.37	6.87
8	10	2.63	5.84	12.40
10	4	0.18	0.55	1.10
10	6	0.72	1.82	3.98
10	8	1.53	3.96	8.18
10	10	3.42	6.93	13.93
12	4	0.16	0.56	0.99
12	6	0.99	1.95	3.96
12	8	1.74	4.33	8.69
12	10	3.40	7.71	15.10

TABLE II: Robust heuristic run times (in milliseconds) for a single starting point.

As expected, the robust heuristic requires longer run times than the basic heuristic due to the increase in combinatorial solutions to the assignment problem. Comparing Table I and Table II we see that robust run times for four estimated targets are roughly four times longer than the corresponding run times in the basic heuristic for a fixed number of scans. However, the magnitude of this effect appears to decay as the number of targets increases. For example, in the case of eight targets the robust heuristic times are only about twice that of the basic heuristic. This suggests that the robust heuristic still scales well with increases in the number of targets. It is also important to note that the run time variance in the robust case is much wider than that of the basic case. However, parallelization may somewhat relieve this issue, enabling the actual runtime to fall closer to the average.

As demonstrated by the symmetric result for the basic heuristic, the scalability with respect to P_{est} is better than with respect to T. For example, increasing from eight to ten targets for eight scans inflates the computational cost by 18%, while increasing from eight to ten scans for eight targets inflates the computational cost by 75%. We note, as a necessary clarification, that for each scenario we must check several P_{est} values, which requires additional processing time and parallelization. Through our simulations we found that the range of possible P_{est} values was generally limited to a set of size six.

2) Number of Targets: In order to correctly estimate the trajectories in the case of detection ambiguity we must first

correctly identify the number of targets, which consequently makes the task of estimating the correct number of targets extremely important. Therefore, we begin our analysis of detection ambiguity by evaluating the difference between the true and estimated number of targets. Explicitly, we define:

$$P_{\text{diff}} = P_{\text{true}} - P_{\text{est}}, \tag{26}$$

where P_{est} is the number of estimated targets and P_{true} is the number of true targets. Note that $P_{\text{diff}} = 0$ indicates that we have correctly estimated the number of targets. When $P_{\text{diff}} < 0$ we have overestimated the number of targets, whereas when $P_{\text{diff}} > 0$ we have underestimated the number of targets. Figure 6 plots the distribution of P_{diff} for scenarios with four targets and eight scans and, for comparison, Figure 7 plots the same result for scenarios with eight targets and eight time scans.

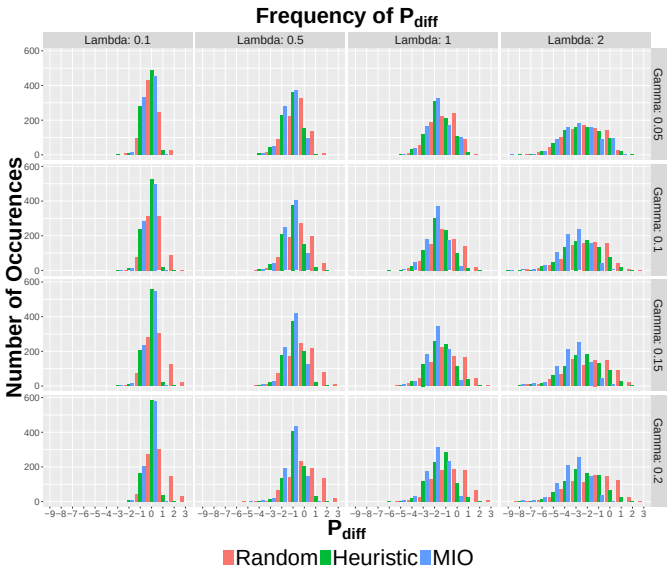

Fig. 6: Distribution of the difference in true and estimated number of targets for scenarios with 4 targets and 8 scans, arranged by γ and λ.

We see that when $\lambda = 0.1$, both the robust heuristic and the robust MIO estimate the number of targets correctly a very high proportion of the time in both scenario sizes. As λ increases, we find that there is a tendency to overestimate the number of targets. This trend appears more prominently in Figure 6, indicating that for large values of λ, the scenario with four targets has a higher tendency to overestimate than the scenario with eight targets. Further examination shows that the solutions for both methods have slightly fewer false alarms than the ideal solution for scenarios with eight targets; this effect is again slightly more exaggerated for the scenarios with four targets. These results suggest that θ was set too high. In addition to the above conclusions, analysis also reveals that both the heuristic and the MIO identified too many missed detections; this suggests that the missed detection penalty ϕ was too low.

Ultimately, the combination of these effects results in a tendency to overestimate the number of targets. Finer tuning

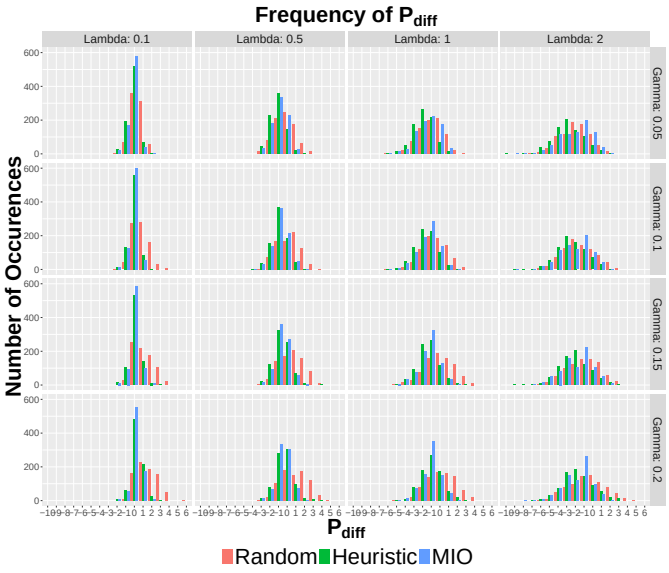

Fig. 7: Distribution of the difference in true and estimated number of targets for scenarios with 8 targets and 8 scans, arranged by γ and λ.

of θ and ϕ can resolve this propensity for overestimation. However, we found it challenging to tune these parameters for higher levels of the false alarm rate λ, which explains why the overestimation effect grows worse as λ increases. Moreover, it is difficult to tune the parameters in such a way that we attain the same tendencies for different numbers of targets. Unfortunately, we do not know the true number of targets, which leads us to believe that more sophisticated penalty functions are necessary in order to neutralize this difference.

3) Data Association: Bearing in mind the fact that we tend to overestimate the number of targets with the given penalties, we move on in our analysis to measuring the accuracy of our robust approaches. Figures 8 and 9 plot accuracy against the difficulty metric, ρ, for scenarios of four and eight targets, respectively. Both scenarios have eight scans and both figures have been arranged by γ and λ.

In Figure 8 we see the accuracy results for four targets. In this case, the heuristic solution is a significant improvement over the random solution and is close to the best solution achieved by the MIO. The MIO achieves this best solution after 1 second and no significant improvement is gained from running it for a longer period of time. However, in larger scenarios with eight targets, as shown in Figure 9, the accuracy of the heuristic offers only a small improvement over the random solution and the MIO improves upon the heuristic solution only after T seconds (when it reaches its best accuracy).

As we stated earlier, even with small values of λ and γ, the case of detection ambiguity has the added difficulty of estimating the correct number of targets. Let us consider the simplest scenarios with detection ambiguity where $\lambda = 0.1$ and $\gamma = 0.05$. We compare the accuracy for four or eight targets in Figure 4, where the number of targets was known, to the accuracy for the corresponding scenarios in Figures 8

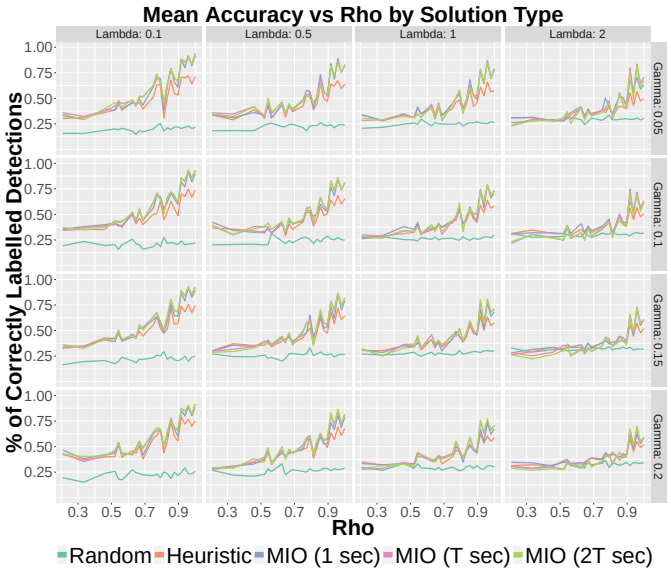

Fig. 8: Accuracy of robust heuristic and MIO as compared to random solutions for scenarios of 4 targets and 8 scans, arranged by γ and λ.

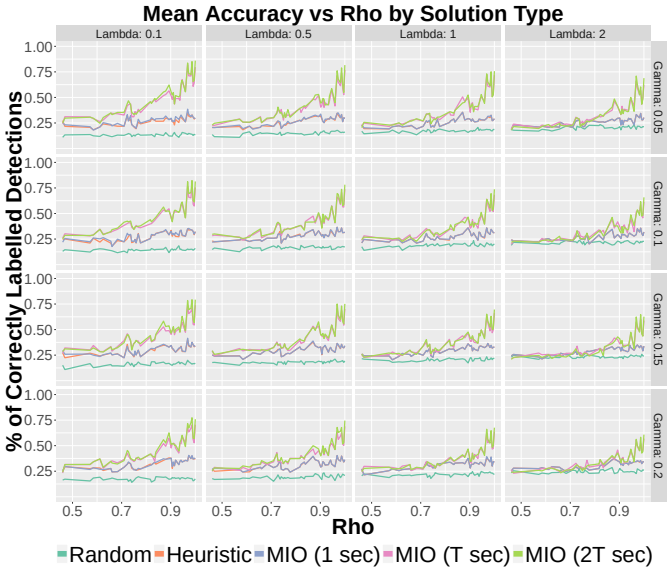

Fig. 9: Accuracy of robust heuristic and MIO as compared to random solutions for scenarios of 8 targets and 8 scans, arranged by γ and λ.

and 9, where the number of targets is unknown. We see a drop in accuracy of only about $5\% - 10\%$ in the presence of detection ambiguity across all values of ρ, suggesting that our approaches deal well with this challenge.

These robust approaches also appear to be more sensitive to higher false detection rates than higher missed detection probabilities. In order to illustrate this result, let us consider the following example with eight targets where we analyze data association accuracy; in this example we (i) fix γ and alter λ, and then (ii) fix λ and alter γ. If we choose $\gamma = .05$ and increase λ from 0.1 to 2.0, we find that the MIO solution after T seconds decreases in accuracy from about 90% to 65%. If we now instead fix $\lambda = .1$ and increase γ from $.05$ to $.2$, the resulting MIO solution after T seconds only reduces in accuracy to 75%, showing the aforementioned sensitivity property. This effect can be attributed to the challenge in tuning the penalties for higher levels of λ, which in turn results in an overestimation of the number of targets and therefore a reduced accuracy.

4) Trajectory Estimation: We conclude our analysis of the robust approaches with a discussion of their performance in the sphere of the trajectory estimation problem. Figures 10 and 11 plot the δ performance metric against the detection error σ for scenarios of four and eight targets, respectively.

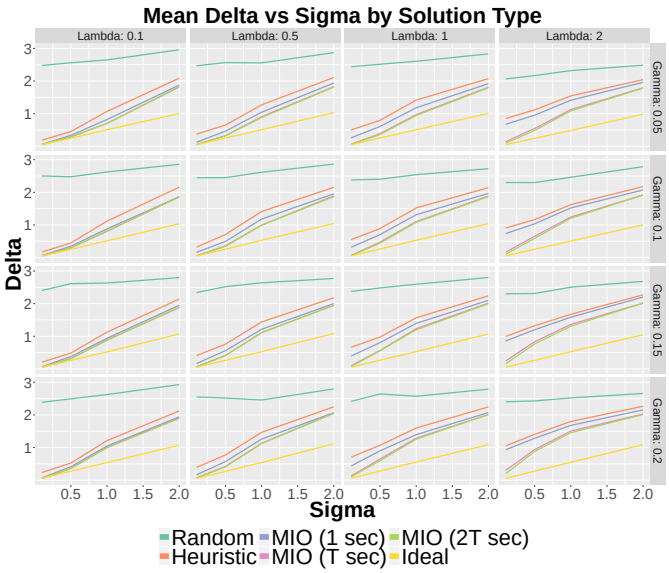

Fig. 10: δ of robust heuristic and MIO as compared to random solutions for scenarios of 4 targets and 8 scans.

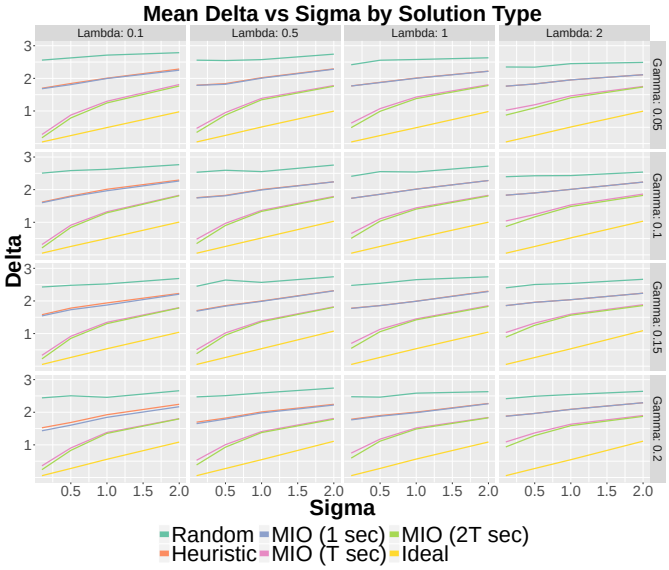

Fig. 11: δ of robust heuristic and MIO as compared to random solutions for scenarios of 8 targets and 8 scans.

Note that δ is calculated using the $\min\{P_{\text{true}}, P_{\text{est}}\}$. The fact

that we tend to overestimate means that δ will be calculated using P_{true} much more often. This implies that in most cases we match all of the true trajectories to estimated ones, and thus the number of trajectories for the δ calculation is the same as in the basic scenarios without detection ambiguity. Therefore, we can compare and interpret their values directly.

If we compare Figure 11 with the graph in Figure 5 corresponding to $P = 8$ and $T = 8$, we find that, for $\lambda = 0.1$ and $\gamma = 0.05$, there is no observable loss of performance in adding the detection ambiguity in the MIO solution after T seconds. Furthermore, this observation also holds true for the heuristic solution. By contrast, if we instead compare Figure 10 with the graph in Figure 5 corresponding to $P = 4$ and $T = 8$, at the same values of λ and γ, the δ value for the MIO increases for σ values larger than 0.5, and this increase becomes worse as σ grows. Moreover, in this same case, the heuristic performance deteriorates at an even higher rate than that of the MIO.

Similar to what we observed in the data association problem, our approaches in the trajectory estimation problem are also more robust to increases in γ than increases in λ. Figures 10 and 11 both show that there is little to no loss in the quality of the trajectory estimation when increasing γ for both the heuristic and MIO, which holds true across all values of σ and λ. However as λ increases, particularly for small values of σ, we find that the estimation of the MIO at T seconds deteriorates. For example, in the case of eight targets and $\sigma = 0.1$, when $\lambda = 0.1$ we obtain δ close to zero, but when λ increases to $\lambda = 2$ the value of delta grows to 1. This deterioration effect is not as prominent in the case of only four targets.

5) Summary of Detection Ambiguity Results: Incorporating detection ambiguity presents two additional challenges: (i) the assignment problem is more complex for a fixed number of targets and, (ii) there is the additional problem of estimating the correct number of targets. The results in this setting indicate that,

- Tuning the parameters leads to a correct estimation of the number of targets, even for a relatively large scenario.
- The accuracy of the data association solutions deteriorates by 5%-10% with the inclusion of detection ambiguity.
- Although we tend to overestimate the number of targets, the quality of estimated trajectories that correspond to the true trajectories is similar to the basic setting with no detection ambiguity.
- The solution of the MIO is more robust to changes in missed detection probability than to changes in the false alarm rate.

VIII. Summary and Future Work

In this paper, we present a new approach to the multi-target tracking problem that jointly solves the problems of data association and trajectory estimation via global optimization methods using a single objective function. In order to solve these problems efficiently we propose the use of a randomized local search heuristic as a warm start for an MIO model, while addressing scenarios with and without detection ambiguity. The described approach is both general, because it makes no assumptions on the data generation process, and easily implementable, as it is based on a simple model with little to no tunable parameters. We demonstrate that the local search heuristic finds high quality feasible solutions very quickly, and that the MIO model generates improvement over the heuristic in seconds. Furthermore, we show that while the introduction of detection ambiguity deteriorates the performance of the data association, it does not significantly effect the quality of the trajectory estimation. We are able to conclude that the quality of the obtained solution is robust to the missed detection rate, but less so to the false alarm rate.

In the process of analyzing this widely-studied problem through a new optimization lens, we identified challenges in both model formulation and successful implementation that can be addressed in future work. Due to the fact that the runtimes of the MIO and the heuristic are proportional to the number of scans, these models have limited scalability in that sense. However, they show strong potential for larger applications in a sliding window scheme, which would use past decisions to fix detection assignments and thus contribute additional information to the current tracking window. This would allow for the tracking of targets in real time systems for longer periods of observation. Additionally, we observed that one of the key difficulties in the case of detection ambiguity involves the correct estimation of the true number of targets. Our results suggest that tuning the penalties for a scenario with a certain number of targets may lead to over or under-estimation for scenarios with differing numbers of targets. Therefore, we suggest exploration into more complex penalties that further analyze this dependency. Other directions for consideration also lie in relaxing some of the scenario based assumptions, in particular, extensions to non-linear trajectories or the birth/death of targets.

Appendix A
Detection Ambiguity Penalty Values

Here we provide recommendations for the tuning of the false alarm penalty θ and the missed detection penalty ϕ. We begin with an explanation grounded in logic. It can be shown that the expected number of false alarms increases as the false alarm rate λ increases. Therefore, it stands to reason that the false alarm penalty should decrease as the false alarm rate increases, as a general rule of thumb. Similarly, the expected number of missed detections increases as the missed detection probability γ increases, and so too the missed detection penalty ϕ should decrease. Furthermore, there is reason to believe that the value of both of these penalties should be tied to the value of σ. Specifically, the estimation error (first term in (20)) will naturally increase as the noise σ increases. Therefore, the values of θ and ϕ should be adjusted to account for this tradeoff. In particular, we should consider increasing both penalties as σ increases.

We also note that the number of false alarms is independent of the number of targets. In contrast, both the estimation error and number of missed detections are directly proportional to the number of targets. This presents a challenge because the

true number of targets is unknown and we cannot tune the penalties to suit this number. Instead, we suggest normalizing θ for the number of targets the MIO is currently estimating. This effectively balances the terms in the objective function by making each term proportional to the number of targets. For example, we tuned θ for scenarios with eight targets and used the following linear dependence to update $\theta(P_{\text{est}})$ for the estimated number of targets P_{est}:

$$\theta(P_{\text{est}}) = \frac{P_{\text{est}}}{8}\theta. \tag{27}$$

Additionally, we empirically observed that the missed detection penalty ϕ benefits from tuning for γ, λ and σ, while the false alarm penalty θ benefits from tuning for λ and σ, but gains no added benefit from tuning for γ.

Through examination and experimentation we found these conclusions to generally hold true across a variety of scenario sizes and difficulties. Using the insight gained from these results, we tuned both penalties for the full scale experiment with detection ambiguity outlined in Section VII. The false alarm penalties for the eight target scenarios in the robust experiment are shown in Table III and the missed detection penalties are shown in Table IV.

	σ			
λ	0.1	0.5	1.0	2.0
0.1	1.7	2.6	3.1	3.5
0.5	1.1	1.9	2.3	2.5
1.0	0.9	1.2	1.6	1.8
2.0	0.5	0.9	0.9	1.0

TABLE III: False alarm penalties (θ) as a function of λ and σ.

		σ			
λ	γ	0.1	0.5	1	2
0.10	0.05	0.20	0.50	0.80	0.70
0.10	0.10	0.10	0.30	0.50	0.50
0.10	0.15	0.10	0.20	0.40	0.40
0.10	0.20	0.10	0.10	0.30	0.40
0.50	0.05	0.20	0.50	0.80	0.80
0.50	0.10	0.20	0.30	0.50	0.60
0.50	0.15	0.20	0.25	0.40	0.40
0.50	0.20	0.10	0.20	0.30	0.40
1.00	0.05	0.30	0.70	0.80	0.80
1.00	0.10	0.20	0.40	0.50	0.60
1.00	0.15	0.20	0.25	0.40	0.40
1.00	0.20	0.10	0.20	0.30	0.40
2.00	0.05	0.30	0.70	0.90	1.00
2.00	0.10	0.20	0.50	0.60	0.60
2.00	0.15	0.20	0.25	0.40	0.50
2.00	0.20	0.10	0.20	0.30	0.40

TABLE IV: Missed detection penalties (ϕ) as a function of λ, γ, and σ.

APPENDIX B
ROBUST MIO WITH NUMBER OF TARGETS AS A DECISION VARIABLE

For completeness we take the time to present an alternative approach to solving the MTT problem with detection ambiguity. We extend (24) to develop a new MIO model that directly determines the number of targets via optimization. We accomplish this by incorporating additional decision variables and constraints into the framework of the MIO formulation.

A. Decision Variables

Rather than assume a fixed number of targets for the model, we now allow this decision to be made by the model itself. Toward this goal, we introduce a new binary decision variable w_j to indicate whether or not trajectory *j* corresponds to an existing target:

$$w_j = \begin{cases} 1, & \text{if trajectory } j \text{ exists,} \\ 0, & \text{otherwise.} \end{cases}$$

We make two important notes in regard to this modification:

- Detections assigned to a non-existing target would be classified as a false alarm.
- w_j ranges from $1, ..., N_1$. As a result, all instances of j in the new MIO must be adjusted accordingly.

B. Objective Function

We can utilize the same objective function from (20), except for slight adjustments needed to account for the possibility that some trajectories may not exist. Explicitly, the number of possible trajectories is now N_1 so the objective should be adjusted to sum over *j* the full range of j from 1 to N_1:

$$\underset{\psi_{jt}}{\text{minimize:}} \sum_{j=1}^{N_1}\sum_{t=1}^{T}\psi_{jt} + \theta \cdot TF + \phi \cdot TM$$

C. Constraints

In the same fashion, most constraints remain similar to their original counterparts in (24), with the exception that the summations must be adjusted appropriately. For example, we adjust (21) and (23) as follows:

$$\sum_{j=1}^{N_1} y_{itj} + F_{it} = 1 \qquad \forall i, t,$$

$$\sum_{j=1}^{N_1}\sum_{t=1}^{T} M_{jt} = TM.$$

By the same accord the RHS of (22) no longer equals one because some trajectories may not exist. Therefore we say that all *existing* trajectories must either be assigned a detection or a missed detection, which implies the following constraint:

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \qquad \forall j, t. \tag{28}$$

Also, we restrict α_j and β_j to be zero if trajectory *j* does not exist. This ensures only existing trajectories are penalized in the objective function.

$$|\alpha_j| + |\beta_j| \leq M_0 w_j \qquad \forall j.$$

This model is symmetric with respect to w_j variables. Such symmetry in models is an undesirable quality because multiple optimal solutions may exist, resulting in a less efficient formulation. However, we can actually reduce the symmetry through

careful design. Since $N_0 \leq P \leq N_1$, we can set $w_j = 1$ for all $j = 1, \ldots, N_0$, which leaves us with only $N_1 - N_0$ unknown w_j variables. In addition, we can add the following constraint:

$$w_{N_0+1} \geq ... \geq w_{N_1},$$

to further reduce the number of equivalent solutions and increase the efficient resolvability of the model.

D. Full Formulation

Incorporating these additional variables and constraints, we arrive at the full formulation of an MIO model that uses decision variables to solve for the number of estimated targets when detection ambiguity exists.

$$\begin{aligned}
\underset{\psi_{jt}}{\text{minimize:}} \quad & \sum_{j=1}^{N_1}\sum_{t=1}^{T} \psi_{jt} + \theta \cdot TF + \phi \cdot TM && (29)\\
\text{subject to:} \quad & \sum_{j=1}^{N_1} y_{itj} + F_{it} = 1 \qquad \forall i, t\\
& \sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \qquad \forall j = 1, ..., N_0, t\\
& \sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \qquad \forall j = N_0, ..., N_1, t\\
& \sum_{i=1}^{n_t}\sum_{t=1}^{T} F_{it} = TF\\
& \sum_{j=1}^{N_1}\sum_{t=1}^{T} M_{jt} = TM\\
& w_{N_0+1} \geq ... \geq w_{N_1}\\
& |\alpha_j| + |\beta_j| \leq M_0 w_j \qquad \forall j\\
& x_{it} y_{itj} + M_t(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j\\
& x_{it} y_{itj} - M_t(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j\\
& z_{jt} - \alpha_j - \beta_j t \leq \psi_{jt} \qquad \forall j, t\\
& -(z_{jt} - \alpha_j - \beta_j t) \leq \psi_{jt} \qquad \forall j, t\\
& y_{itj} \in \{0, 1\} \quad \forall i, t, j\\
& \alpha_j \in \mathbb{R}^n, \quad \beta_j \in \mathbb{R}^n, \quad w_j \in \mathbb{R}^n \quad \forall j\\
& z_{jt} \in \mathbb{R}^n, \quad \forall j, t.
\end{aligned}$$

E. Extension of Robust Heuristic

The robust local search heuristic presented in section V-B can also be used to find warm start solutions to (29). Although the robust heuristic requires a given fixed number of targets for the heuristic, it does not require running the heuristic in parallel for all values of P. Alternatively, the number of targets can be selected at random prior to initializing the heuristic. No matter the case, the heuristic solutions can be used as a warm start to (29) by simply setting $w_j = 1$ for all $j = 1, ..., P$ and $w_j = 0$ for all remaining w_j up to N_1.

APPENDIX C
TRAJECTORY ASSIGNMENT PAIRING

In order to analyze the performance of a multi-target tracking algorithm, we must first match the true trajectories of the scenario to the estimated trajectories of the algorithm solution. In other words, we wish to the best pairings of true and estimated trajectories. With this in mind, we present an integer optimization model that solves for the globally optimal pairings of true and estimated trajectories. In addition, we extend this assignment problem to scenarios with detection ambiguity, where the MTT algorithms may overestimate or underestimate the number of targets.

A. Decision Variables

The goal of this assignment problem is to optimally assign pairs of true trajectories *i* to estimated trajectories *j* if there exists such a pairing to be made. Only a single set of decision variables are needed to determine this pairing:

$$y_{ij} = \begin{cases} 1, & \text{if true trajectory } i \text{ is assigned} \\ & \text{to estimated trajectory } j, \\ 0, & \text{otherwise.} \end{cases}$$

B. Objective Function

Remember that we denote the true position of trajectory *i* in scan *t* with $\bar{x}_{it}$ and the estimated position of trajectory *j* in scan *t* with $\hat{x}_{jt}$. Then the cost c_{ij} of assigning true trajectory *i* to estimated trajectory *j* is the norm distance between these two trajectories in all scans.

$$c_{ij} = \sum_{t=1}^{T} \|\bar{x}_{it} - \hat{x}_{jt}\|.$$

If we denote the true number of targets as P_{true} and the estimated number of targets as P_{est} then the objective of the integer optimization model is:

$$\underset{y_{ij}}{\text{minimize:}} \quad \sum_{i=1}^{P_{\text{true}}}\sum_{j=1}^{P_{\text{est}}} c_{ij} y_{ij}.$$

C. Constraints

When the number of true targets is equal to the number of estimated targets ($P_{\text{true}} = P_{\text{est}} = P$), we simply require two equality constraints to ensure that each true trajectory *i* is assigned to exactly one estimated trajectory *j* and vice versa.

$$\sum_{i=1}^{P} y_{ij} = 1 \qquad \forall j = 1, ..., P \qquad (30)$$

$$\sum_{j=1}^{P} y_{ij} = 1 \qquad \forall i = 1, ..., P. \qquad (31)$$

However, when the number of estimated trajectories differs from the number of true trajectories, these constraints must be adjusted appropriately. For the case in which the number of true targets exceeds the estimated number of targets ($P_{\text{true}} \geq$

P_{est}), we restrict each true trajectory i to the assignment of *at most* one estimated trajectory j. In effect, Equation 30 is modified to:

$$\sum_{i=1}^{P_{\text{true}}} y_{ij} \leq 1 \qquad \forall j = 1, ..., P_{\text{est}}.$$

By contrast, when the number of estimated targets exceeds the true number of targets ($P_{\text{true}} \leq P_{\text{est}}$), then we restrict each estimated trajectory j to the assignment of *at most* one true trajectory i, and Equation 31 is modified to:

$$\sum_{j=1}^{P_{\text{est}}} y_{ij} \leq 1 \qquad \forall i = 1, ..., P_{\text{true}}.$$

D. Generalized Assignment Pairing Model

In summary, we arrive at the following generalized integer optimization assignment model:

$$\begin{aligned}
\underset{y_{ij}}{\text{minimize:}} \quad & \sum_{i=1}^{P_{\text{true}}} \sum_{j=1}^{P_{\text{est}}} c_{ij} y_{ij} \\
\text{subject to:} \quad & \sum_{i=1}^{P_{\text{true}}} y_{ij} = 1 \qquad \forall j = 1, ..., P_{\text{est}} \\
& \sum_{j=1}^{P_{\text{est}}} y_{ij} = 1 \qquad \forall i = 1, ..., P_{\text{true}} \\
& y_{ij} \in \{0, 1\} \quad \forall i = 1, ..., P_{\text{true}}, j = 1, ..., P_{\text{est}}.
\end{aligned}$$

This model is vital to scoring the performance of an MTT algorithm's solution because it ensures the globally optimal pairing of true and estimated trajectories.

ACKNOWLEDGMENTS

The authors would like to thank Sung-Hyun Son, Ph.D. and Steven Relyea at Lincoln Laboratories for introducing us to the MTT problem and for their ongoing guidance throughout this work. Additionally, we would like to thank Lincoln Laboratories and the LLGrid team for their continual support in running our experiments.

This material is based upon work supported by the Air Force under Air Force Contract No. FA8721-05-C-0002 and/or FA8702-15-D-0001. The views expressed in this document are those of the authors and do not reflect the official policy or position of the United States Air Force, the United States Department of Defense or the United States Government.

REFERENCES

[1] G. Pulford, "Taxonomy of multiple target tracking methods," *Radar, Sonar and Navigation, IEE Proceedings -*, vol. 152, no. 5, pp. 291–304, October 2005.

[2] Y. Bar-Shalom and X. Li, *Multitarget-multisensor Tracking: Principles and Techniques*. Yaakov Bar-Shalom, 1995. [Online]. Available: https://books.google.com/books?id=GfOoMQEACAAJ

[3] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[4] S. Blackman, *Multiple-target Tracking with Radar Applications*, ser. Radar Library. Artech House, 1986. [Online]. Available: https://books.google.com/books?id=Ag9TAAAAMAAJ

[5] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

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

[7] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, Jan 2004.

[8] J. K. Wolf, A. M. Viterbi, and G. S. Dixon, "Finding the best set of k paths through a trellis with application to multitarget tracking," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 25, no. 2, pp. 287–296, Mar 1989.

[9] D. Castanon, "Efficient algorithms for finding the k best paths through a trellis," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 26, no. 2, pp. 405–410, Mar 1990.

[10] R. Perry, A. Vaddiraju, and K. Buckley, "Multitarget list viterbi tracking algorithm," in *Signals, Systems amp; Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, vol. 1, Nov 1998, pp. 436–440 vol.1.

[11] C. Morefield, "Application of 0-1 integer programming to multitarget tracking problems," *Automatic Control, IEEE Transactions on*, vol. 22, no. 3, pp. 302–312, Jun 1977.

[12] C. Carthel and S. Coraluppi, "Multi-hypothesis sonar tracking," in *Fusion 2004: Seventh International Conference on Information Fusion*, 2004.

[13] A. P. Poore and N. Rijavec, "A lagrangian relaxation algorithm for multidimensional assignment problems arising from multitarget tracking," *SIAM Journal on Optimization*, vol. 3, no. 3, pp. 544–563, 1993. [Online]. Available: http://dx.doi.org/10.1137/0803027

[14] A. Andriyenko and K. Schindler, "Multi-target tracking by continuous energy minimization," in *Computer Vision and Pattern Recognition (CVPR)*, 2011.

[15] A. Andriyenko, K. Schindler, and S. Roth, "Discrete-continuous optimization for multi-target tracking," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 1926–1933.

[16] A. Lodi, *50 Years of Integer Programming 1958-2008.From the Early Years to the State-of-the-Art*, M. Jnger, T. M. Liebling, D. Naddef, G. L. Nemhauser, W. R. Pulleyblank, G. Reinelt, G. Rinaldi, and L. A. Wolsey, Eds. Springer Berlin Heidelberg, 2010.

[17] R. E. Bixby, "Mixed-integer programming: It works better than you may think," http://www.ferc.gov/CalendarFiles/20100609110044-Bixby,%20Gurobi%20Optimization.pdf, 2010, accessed 4 April 2016s.

[18] I. Gurobi Optimization, "Gurobi 6.5 performance benchmarks," http://www.gurobi.com/pdfs/benchmarks.pdf, 2015, accessed 4 April 2016s.

[19] G. Nemhauser, "Integer programming: Global impact," https://smartech.gatech.edu/bitstream/handle/1853/49829/presentation.pdf, 2013, accessed 4 April 2016.

[20] Top500 Supercomputer Sites, "Performance development," http://www.top500.org/statistics/perfdevel/, 2015, accessed 4 April 2016.

[21] B. Bertsimas and J. Dunn, "Optimal trees," submitted for publication, 2015.

[22] B. Ristic, B. N. Vo, D. Clark, and B. T. Vo, "A metric for performance evaluation of multi-target tracking algorithms," *IEEE Transactions on Signal Processing*, vol. 59, no. 7, pp. 3452–3457, July 2011.

[23] C. Heij, P. de Boer, P. Franses, T. Kloek, H. van Dijk, and A. Rotterdam, *Econometric Methods with Applications in Business and Economics*. OUP Oxford, 2004. [Online]. Available: https://books.google.com/books?id=PJf6GUAavhAC

[24] N. J. Stor, I. Slapnicar, and J. L. Barlow, "Accurate eigenvalue decomposition of real symmetric arrowhead matrices and applications," *Linear Algebra and Its Applications*, vol. 464, pp. 62–89, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0024379513006265

[25] M. Lubin and I. Dunning, "Computing in operations research using julia," *INFORMS Journal on Computing*, vol. 27, no. 2, pp. 238–248, 2015. [Online]. Available: http://dx.doi.org/10.1287/ijoc.2014.0623

[26] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com

[27] N. Bliss, R. Bond, J. Kepner, H. Kim, and A. Reuther, "Interactive grid computing at lincoln laboratory," *MIT Lincoln Laboratory Journal*, vol. 16, no. 1, 2006.