

General comment on the intro: Think what helps you make your arguments, and what is redundant, the intro is aimed at presenting known models in order to specify what is different or better about the model you are suggesting. You talk about recursive and batch models but except for the MHT you do not place the other models you discuss in these categories. You elaborate a lot about MHT but briefly mention 'operations research algorithms' without much explanation as well as the existing MIO which is our main competitor - why is it not 'good enough', why did we construct an alternative MIO?

Mixed Integer Optimization for the Multi-Target Tracking Problem

Dimitris Bertsimas, Shimrit Shtern, and Zachary Saunders

Abstract—The field of multi-target tracking consists of two primary challenges: (i) data association and (ii) trajectory estimation. MTT problems are well researched with many algorithms solving these two problems separately, however few algorithms attempt to solve these simultaneously and even fewer utilize optimization. In this paper we introduce a new mixed integer optimization (MIO) model which solves the data association and trajectory estimation problems simultaneously. Furthermore, we propose a greedy heuristic which provides good solutions very quickly and demonstrate its effectiveness as a warm start to the optimization solver.

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

I. INTRODUCTION

MULTI-target tracking is the problem of estimation 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 set of *detections*. From these detections, the multi-target tracking problem aims to extract information about target dynamics.

Solutions to this problem are sought across many civilian and military applications including: ballistic missile and aircraft defense, the movement of ships and ground troops, and air traffic control. Each application has unique attributes and assumptions, and various algorithms have been developed for each. As a result, the field of multi-target tracking is very well researched and there is a wide range of literature on the topic.

The field of multi-target tracking consists of 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 needs to be labeled with a target identifier. 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 under the simplest of conditions. This estimation problem is difficult, even when the associations are known, due to the presence of measurement noise. The problem is further complicated when sensors fail to report detections or report false detections, resulting in ambiguity in the number of existing targets.

Although these two challenges are closely related to one another, traditionally they are treated separately using a combination of probabilistic approaches to determine data associations and filters to estimate trajectories. The global nearest neighbor (GNN) algorithm, for example, is a naive 2-D assignment algorithm, which evaluates one scan of detections at a time,

globally assigning the nearest detection at each scan [1]. Once the data association has been determined, the detections are often passed through one of numerous filters, most commonly a Kalman filter [2], which updates the trajectory estimates before the algorithm progresses forward to the next scan. This process repeats sequentially through each scan of data.

Moreover, there are smoothing techniques which actually smooth the Kalman estimator backwards as well (citation). I would not put this here

There are two primary classes of MTT algorithms, recursive and batch. A *recursive* algorithm evaluates one scan at a time, making a hard decision on the data association, updating the trajectory estimates, and moves to the next scan. The GNN algorithm discussed previously falls in this class. A *batch* algorithm has the ability to consider multiple scans simultaneously, and progresses forward with a sliding window, only making hard decisions once a scan moves out of the window.

The two most prevalent algorithms in the field of multi-target tracking are the Multiple Hypothesis Tracker (MHT) and the Joint Probability Data Association Filter (JPDAF) and 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 merely a labeling of the detections, and probabilities are assigned to each hypothesis based on the likelihood of the trajectory's existence. do they deal with both problems or just the assign?

The MHT, first proposed by Reid in [3], assigns likelihood values to hypotheses using a Bayesian (MAP) estimator. This algorithm is heavily researched and is generally considered to be the modern standard for solving the data association problem at the time of writing. Many variants have been proposed for implementation which utilize various types of clustering, gating, hypothesis selection, hypothesis pruning, and merging of state estimates. Many of these methods are summarized by Blackman in [4]. must specify before using acronym

A PDA takes a Bayesian approach to solving the data association problem by finding detection to target assignment probabilities via a posterior PDF. Similarly, a Joint PDA (JPDA) assigns probabilities that are computed *jointly* across all targets. The JPDAF is an algorithm which implements the JPDA along with filters and estimation methods as discussed previously, [5].

(Both the JPDAF and MHT algorithms take advantage of a concept known as gating. At its simplest form gating eliminates infeasible detection assignments, limiting the set of possible hypotheses to consider. Generally, gating is based on limitations to target dynamics, assumed known a priori.) too much detail

A limited number of operations research algorithms have

what does this mean?
optimization) probabilistic
it's too vague

can you maybe add a reference to a review paper on applications?

but what methodology is used?

been applied to solve the MTT problem, most of which attempt to solve by mapping the measurement set onto a trellis and seek the optimal measurement association sequence. Some examples include the Multi-Target Viterbi [6] and an extension in [7] which formulates [6] as a network flow, reducing the solve time from exponential to polynomial. Others have suggested adaptations which allow this approach to be used similar to MHT methods by outputting a single best set of K tracks, or a list of L best sets of k tracks [8].

Compared to the number of probabilistic approaches available in the MTT literature, methods which utilize the power of optimization are relatively lacking. Most of the optimization methods found in the MTT literature aim to leverage current algorithms such as the MHT. integer programming has been used to leverage MHT hypothesis selection through the use of an assignment which chooses the best hypothesis once costs have been assigned. Somewhat similarly, linear programming has also been used to leverage the power of optimization with the MHT [10]. Other attempts have aimed to solve the MHT hypothesis selection problem using Lagrangian relaxation [11].

Recently, [12] was the first to formulate the MTT problem using discrete-continuous optimization with label costs, jointly solving the data association and trajectory estimation problems via minimization of a unified discrete-continuous energy. However, this approach utilizes several parameters that must be tuned or assumptions in the model.

There is no shortage of literature on MTT methodologies. A more 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 [13]. For a more exhaustive overview of estimation techniques, filtering, gating, and more please see [5] or [14].

In this paper we suggest a MIO approach to solving the multi-target tracking problem. Although mixed integer linear programs (MILPs) are generally thought as intractable due to their theoretical categorization as NP-Hard problem, in many practical cases good solutions, and even optimal solutions to these problems can be obtained in reasonable time [15]. This is due to the fact that MIO solvers have seen significant performance improvements in recent years due to advancements in both the literature and hardware. The development of new heuristic methods, discoveries in cutting plane theory, and improved linear programming methods have all contributed to improvements in performance [16]. 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 of a factor of 48.7% [17].

The literature is also 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 is negated. Recent work [18] proposes a mathematically rigorous performance metric for measuring the distance between ground truth and estimated track, but there is not much attention given to the complexity of generated sce-

narios. In this paper we also suggest measures of complexity and performance which are related to the ones suggested in [18] but we show the value in relating a complexity measure to performance measures, namely that it allows you 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.

In this work we aim to simultaneously solve the data association and trajectory estimation problems using a single interpretable MIO model which can be solved in practical time for the applications considered. We propose a heuristic that gives us feasible solutions to this problem and show how it can be used as warm start to the MIO in order to improve the quality of the solutions obtained as well as the running time. The main contributions of this paper are as follows:

(i) we introduce a simple interpretable MIO model which solves the data association and trajectory estimation problems simultaneously for a sensor with no detection ambiguity. (ii) We extend this basic model for the case of detection ambiguity, i.e., the case where there are both missed detections and false alarms, keeping interoperability while only adding two tunable parameters, as well as, provide general guidelines as to how tune this parameters. (iii) we present several measures of scenarios complexity algorithms performance, and discuss the connection between these measures, in order to create a performance threshold for our algorithm. We believe that this measures are good indicator of problem complexity and performance and can be utilized for other algorithms as well (vi)...

The paper structure is as follows...we begin by explaining the MTT problem as we wish to model it. Then we develop our MIO model, and follow it up with a discuss on our proposed heuristic. Then we show how these methods are adapted to scenarios where the number of targets is unknown. Finally, we test our methods in the simulated experiments section, discuss performance metrics, and summarize our experimental results.

General Notations: We use ' to indicate the transpose of a matrix. Unless specified otherwise, $\| \cdot \|$ is used to indicate the L1 norm, and $|\cdot|$ is 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 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. Detections are subject to noise. We treat two general scenarios: with and without detection ambiguity.

When there is no detection ambiguity, the sensor produced exactly one detection for each target at each time, and there is no other source of detections. Therefore, the number of detections at each point in time is the same as the number of targets, and the data association problem at each point in time is equivalent to a simple assignment problem. Our basic

optimization model, presented in section [add reference] will this with this problem.

Detection ambiguity refers to the more complex case where the sensor generates both false alarms and missed detections. A *False Alarm* occurs when a detection is collected when no target exists. This could be the result of measurement error or difficulties in signal processing. A *Missed Detection* occurs when a data point is not collected at a given time when a target actually exists. Therefore, the number of detections at each point in time may be either higher or lower than the actual number of targets, and each detection can be classified in either of these categories in addition to assigning targets to trajectories as before. In section [add reference] we will extend the formulation of basic model to a robust formulation dealing with this ambiguity, and refer to it 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 one another*
(iii) *The number of targets remains constant throughout the window of observation i.e. there is no birth/death of targets*
(iv) *Each target produces at most one detection per scan*
(v) *The detection errors are independent of one another*

Notation: We observe P targets over a fixed time window over which T scans are collected. Scans occur at a fixed rate, usually of about 1Hz. 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 $\mathcal{X}_t = \{x_{1t}, x_{2t}, \dots, x_{Pt}\}$. The data for the problem is the ordered set of scans $\mathcal{X} = \{\mathcal{X}_1, \mathcal{X}_2, \dots, \mathcal{X}_T\}$. The state space of target trajectories is paramaterized by a true initial position α_j^T and a true constant velocity β_j^T . Therefore, the true position \bar{x}_{jt} of trajectory j at scan t is given by:

$$\bar{x}_{jt} = \alpha_j^T + \beta_j^T * t \quad (1)$$

III. BASIC 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 at each time (no missed detections).*
(ii) *The sensor does not generate any additional detections (no false alarms).*

We begin constructing our MIO model by defining decision variables that represent the desired detection to target associations and target estimated trajectories. Next, using these decision variables, we develop an objective function which mathematically quantifies the value of the model decisions, in this case as a measure of distance of the estimated trajectories from the associated detections. Finally, we restrict these variables using constraints that force the model to find solutions that are feasible for the MTT problem. The model is developed

step by step in the coming sections before the full model is presented.

A. Decision Variables

The data association and trajectory estimation problems 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 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 estimation problem, we wish to assign detections to trajectories, 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:

$$y_{itj} \begin{cases} 1 & \text{if detection } x_{it} \text{ is assigned to trajectory } j \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

B. Objective Function

Next, we would like to develop a function which accurately scores the quality of a feasible solution. An ideal objective function would jointly provide a single quantifiable measure of goodness for both the data association and trajectory estimation problems. Therefore we want the objective to take into account the assignments of detections in addition to the estimated trajectory determined by those assignments.

In terms of our established decision variables, the estimated position of the linear trajectory j at time t is given by:

$$\hat{x}_{jt} = \alpha_j - \beta_j * t \quad (3)$$

For the trajectory estimation problem we wish to minimize the distance between x_{it} and \hat{x}_{jt} . In other words, for detections x_{it} assigned to trajectory j , we wish to minimize $\|x_{it} - \hat{x}_{jt}\|$ for some norm. The two natural norms to consider here are the L1 and L2 norms. The L1 norm has the advantage that it can be reformulated as a linear program (through the addition of continuous variables and constraints), and it is well known to be more robust to outliers. Additionally, existing algorithms for MIO are more well developed for linear rather than quadratic optimization. However, the L2 norm square form (RSS) has the advantage that it can be quickly computed using a matrix formulation, making it more predisposed to a heuristic. This concept will be discussed further in section IV.

Substituting (3) for \hat{x}_{jt} we arrive at our objective function:

$$\underset{\alpha_j, \beta_j}{\text{minimize:}} \quad \sum_{(i,j) \in \mathcal{A}} \sum_{t=1}^T \|x_{it} - \alpha_j - \beta_j * t\| \quad (4)$$

where \mathcal{A} is the set of pairs that indicate the assignment of detection i to trajectory j .

For the data association problem, we wish to only penalize the objective function when detection x_{it} has been assigned to trajectory j , which occurs when $y_{itj} = 1$. An easy method

to enforce this using our established variables would be to construct an interaction term like in (5) below.

$$\underset{y_{itj}, \alpha_j, \beta_j}{\text{minimize:}} \sum_{i=1}^P \sum_{t=1}^T \|y_{itj} x_{it} - \alpha_j - \beta_j * t\| \quad (5)$$

We now show that this objective can be converted to a linear program in the case of the L1 norm by introducing continuous variables θ_{jt} and the following additional constraints.

$$y_{itj} x_{it} - \alpha_j - \beta_j * t \leq \theta_{jt} \quad \forall i, j, t \quad (6)$$

$$-(y_{itj} x_{it} - \alpha_j - \beta_j * t) \geq \theta_{jt} \quad \forall i, j, t \quad (7)$$

The resulting objective function for the case of the L1 norm would then be:

$$\underset{\theta_{jt}}{\text{minimize:}} \sum_{j=1}^P \sum_{t=1}^T e' \theta_{jt} \quad (8)$$

where e is the vector of ones.

For the case of the L2 norm, the objective function would be:

$$\underset{\theta_{jt}}{\text{minimize:}} \sum_{j=1}^P \sum_{t=1}^T \|\theta_{jt}\|_2^2 \quad (9)$$

C. Constraints

Next, we add constraints to our model. Constraints ensure that the model satisfies our assumptions, restricting the set of feasible solutions to those that reflect reality. In particular, we need to set assignment constraints to define the set of legal data associations.

At each time step, each detection x_{it} must be assigned to exactly one target j :

$$\sum_{j=1}^P y_{itj} = 1 \quad \forall i, t \quad (10)$$

Similarly, at each time step, each target must be assigned exactly one detection:

$$\sum_{i=1}^P y_{itj} = 1 \quad \forall j, t \quad (11)$$

D. Simple Formulation

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

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

E. Generalized Formulation

The previous formulation, although simple and easily interpretable, has the disadvantages of being (i) dense and (ii) ill suited for extension to detection ambiguity. Therefore, next, we present a generalized formulation which can be linearized through the introduction of additional binary decision variables. Alternatively, we can create a new variable z_{jt} which takes on the value x_{it} when $y_{ijt} = 1$ and some arbitrary number when $y_{ijt} = 0$. Using this method we must adjust the objective function below.

$$\underset{z_{jt}, \alpha_j, \beta_j}{\text{minimize:}} \sum_{j=1}^P \sum_{t=1}^T \|z_{jt} - \alpha_j - \beta_j * t\| \quad (12)$$

This objective can then be linearized by again introducing θ_{jt} and similar constraints as follows.

$$\underset{\theta_{jt}}{\text{minimize:}} \sum_{j=1}^P \sum_{t=1}^T \theta_{jt} e \quad (13)$$

$$z_{jt} - \alpha_j - \beta_j * t \leq \theta_{jt} \quad \forall i, j, t \quad (14)$$

$$-(z_{jt} - \alpha_j - \beta_j * t) \geq \theta_{jt} \quad \forall i, j, t \quad (15)$$

Furthermore, we must ensure that the decision variable z_{jt} will only take on the value of x_{it} in the objective function if x_{it} is assigned to target j ($y_{ijt} = 1$). We enforce this effect using the following constraint:

$$M_t(1 - y_{ijt}) \geq |z_{jt} - x_{it} y_{ijt}| \quad \forall i, t, j \quad (16)$$

$$(17)$$

where $M_t = \max_j |x_{it}|$ for each time step. Furthermore, we can write this equivalently as a linear program by using the following set of two linear constraints:

$$x_{it} y_{ijt} + M(1 - y_{ijt}) \geq z_{jt} \quad \forall i, t, j \quad (18)$$

$$x_{it} y_{ijt} - M(1 - y_{ijt}) \leq z_{jt} \quad \forall i, t, j \quad (19)$$

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

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

IV. HEURISTIC

Next we present a detailed description of an algorithm which finds good feasible solutions. These solutions can be used as a warm start to the MIO, providing a performance boost to the MIO. The algorithm hinges on a single heuristic which makes decisions that lead to locally optimal solutions. If the algorithm is repeated with numerous starting points, the best locally optimal solution is kept.

An important distinction to discuss here is the difference in objective functions. As discussed before the two natural choices are the L1 and L2 norms. For the heuristic, we desire an objective which can be calculated efficiently. Therefore, in this case the L2 norm square (RSS) is the preferred choice because it can be calculated quickly using matrix algebra. [19] shows how a design matrix M can be using to quickly compute the RSS.

The algorithm is presented in Algorithm 1. The algorithm initializes by randomizing a solution which satisfies equations 6 and 7. The initial parameters are calculated as well as the objective score. In swap k for time step t choose $i, l \in \{1, \dots, P\}$ detections and for $j, m \in \{1, \dots, P\}$ targets such that $y_{itj}^k = 1$ and $y_{ltm}^k = 1$. Switch the detection association so that $y_{litj}^{k+1} = 1$ and $y_{itm}^{k+1} = 1$. Compute $\alpha_j, \beta_j, \alpha_m$, and β_m , along with the new objective. If the objective improves, the swap is kept, otherwise it is rejected. The heuristic then advances to the next time step where the same process is completed. The algorithm terminates once it makes a single pass through every time step without accepting a single switch. As we will see in the computational results section, this algorithm runs very efficiently, providing good solutions very quickly. Furthermore, this algorithm can be parallelized by running subsets of the N starting points on separate cores.

A proposed pseudocode for the heuristic is provided below.

Algorithm 1 Randomized local search with heuristic swaps

Input: \mathcal{X}, P, T, M

Initialization : Assign random initial assignments for y_{itj}^0

- 1: Calculate $\alpha_j, \beta_j \quad \forall j$
- 2: Calculate RSS^0
- 3: swapped = true
- 4: $k = 1$
- 5: **while** swapped **do**
- 6: swapped = false
- 7: **for** t in scans **do**
- 8: Randomly choose $j, m \in \{1, \dots, P\}$
- 9: Find i, l such that $y_{itm}^{k-1} = 1$ and $y_{litj}^{k-1} = 1$
- 10: Swap such that $y_{itj}^k = 1$ and $y_{ltm}^k = 1$
- 11: Calculate $RSS^k, \alpha_j, \beta_j, \alpha_m, \beta_m$
- 12: **if** ($RSS^k \geq RSS^{k-1}$) **then**
- 13: $y^k \leftarrow y^{k-1}$
- 14: **else**
- 15: swapped \leftarrow true
- 16: **end if**
- 17: **end for**
- 18: $k \leftarrow k + 1$
- 19: **end while**
- 20: **return** RSS^k, y_{itj}^k

V. ROBUST MIO MODEL

In this section we treat the case of detection ambiguity. The key difference is that now the number of targets is unknown, and this becomes a third problem which we wish to solve in addition to the data association and trajectory estimation problems which remain once the number of targets has been determined. Since in this case both missed detections and false alarms are present the number of targets is unknown and we may no longer have the same number of detections at each time step. Therefore, we must introduce additional notation for this scenario. We let n_t represent the number of detections at time t . We can then identify the fewest and largest number of detections in a time step with $N_0 = \min_t n_t$ and $N_1 = \max_t n_t$, respectively.

Specifically, in this case we replace assumption 2 by the following less restrictive assumptions.

- Assumption 3.** (i) *The sensor does not generate a detection for any target for any time with probability P_d which is constant and independent between targets and scans.*
- (ii) *At each point in time the sensor generates false alarms according to a Poisson distribution with rate λ_{FA} , which are located uniformly in the space.*
- (iii) *The number of true targets P satisfies $N_0 \leq P \leq N_1$.*

We first show that this problem can be solved by dividing it into a subset of simpler problems. A MIO formulation is presented that assumes a fixed number of targets. This formulation allows us to leverage the power of parallelization to solve the problem by solving each subproblem separately. The results can then be gathered and compared to find the globally optimal solution. For completeness we also present

a formulation which solves the original problem without the need for multiple parallelized MIOs.

A. Fixed Number of Targets (P)

If we first assume that the number of targets is fixed, we can more easily adapt the generalized formulation presented in Section III to handle the addition of false alarms and missed detections. This simple adaptation requires the introduction of two additional variable types and minimal constraint changes. We can then run these formulations for each possible value of fixed number of targets P across the range of N_0 to N_1 and choose the solution with the best objective overall. Furthermore, this is an advantageous strategy because each independent experiment can be run in parallel.

1) *Decision Variables:* We first introduce new binary decision variables F_{it} to indicate whether or not a detection x_{it} is a false alarm.

$$F_{it} = \begin{cases} 1 & \text{if detection } i \text{ at time } t \text{ is a False Alarm} \\ 0 & \text{otherwise} \end{cases}$$

Similarly, we introduce binary decision variables M_{jt} to indicate whether or not an *existing* 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) *Constraints:* All detections must either be assigned to a trajectory j or a false alarm.

$$\sum_{j=1}^P y_{itj} + F_{it} = 1 \quad \forall i, t \quad (20)$$

All trajectories j must either be assigned a detection or a missed detection.

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \quad \forall j, t \quad (21)$$

The sum of all false alarms is TF , and similarly the sum of all missed detections is TM .

$$\sum_{i=1}^{n_t} \sum_{t=1}^T F_{it} = TF \quad (22)$$

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

3) Formulation 2:

$$\begin{aligned} & \text{minimize:} \\ & y_{itj}, \alpha_j, \beta_j, z_{jt}, F_{it}, M_{jt} \quad \sum_{j=1}^P \sum_{t=1}^T |z_{jt} - \alpha_j - \beta_j * t| + \theta_o TF + \phi_o TM \\ & \text{subject to:} \\ & \sum_{j=1}^P y_{itj} + F_{it} = 1 \quad \forall i, t \\ & \sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \quad \forall j, t \\ & \sum_{i=1}^{n_t} \sum_{t=1}^T F_{it} = TF \\ & \sum_{j=1}^P \sum_{t=1}^T M_{jt} = TM \\ & x_{it} y_{itj} + M(1 - y_{itj}) \geq z_{jt} \quad \forall i, t, j \\ & x_{it} y_{itj} - M(1 - y_{itj}) \leq z_{jt} \quad \forall i, t, j \\ & y_{itj} \in \{0, 1\} \quad \forall i, t, j \\ & \alpha_j \in \mathbb{R}^n, \quad \beta_j \in \mathbb{R}^n \quad \forall j \\ & z_{jt} \in \mathbb{R}^n, \quad \forall j, t \end{aligned}$$

B. Number of Targets as a Decision Variable

Now let's say we wish to allow the number of targets to be chosen via optimization. We can further adapt the formulation to account for the number of targets. We must be careful to identify targets which are *existing*. We say that with think a trajectory j *exists* if the detections assigned to that trajectory correspond to an actual target. As a result the detections assigned to a trajectory j that does not exist will naturally be false alarms.

1) *Decision Variables:* We introduce a new binary decision variable w_j to indicate whether or not trajectory j corresponds to an existing target.

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

2) *Constraints:* Most constraints remain similar to their original counterparts, except now we must account for the possibility that some trajectories may not exist. Therefore, where before we summed over P , we will now be summing over N_1 . This affects two constraints.

All detections must either be assigned to a trajectory j or a false alarm. This can be implemented similarly to (20), except now we sum over N_1 because the number of targets is unknown but limited by N_1 .

$$\sum_{j=1}^{N_1} y_{itj} + F_{it} = 1 \quad \forall i, t \quad (24)$$

Similarly, (23) must be adjusted to sum over the maximal number of targets allowed N_1 .

$$\sum_{j=1}^{N_1} \sum_{t=1}^T M_{jt} = TM \quad (25)$$

All *existing* trajectories must either be assigned a detection or a missed detection.

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \quad \forall j, t \quad (26)$$

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.

$$|\alpha_j| + |\beta_j| \leq M_0 w_j \quad \forall j \quad (27)$$

Because of **Assumption 3 (iii)** we can set $w_j = 1$ for all $j = 1, \dots, N_0$ which leaves us with only $N_1 - N_0$ additional binary variables. We simply need the following additional constraint which will guarantee a unique w solution.

$$w_{N_0+1} \geq \dots \geq w_{N_1} \quad (28)$$

Furthermore, we can replace (26) with the following two constraints.

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \quad \forall j = 1, \dots, N_0, t \quad (29)$$

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \quad \forall j = N_0, \dots, N_1, t \quad (30)$$

3) *Formulation 3*: Incorporating these additional variables and constraints, we arrive at the following complete alternative formulation.

$$\begin{aligned} & \underset{y_{itj}, \alpha_j, \beta_j, F_{it}, M_{jt}}{\text{minimize:}} \quad \sum_{j=1}^{N_1} \sum_{t=1}^T |z_{jt} - \alpha_j - \beta_j * t| + \theta_o TF + \phi_o TM \\ & \text{subject to:} \quad \sum_{j=1}^{N_1} y_{itj} + F_{it} = 1 \quad \forall i, t \\ & \quad \sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \quad \forall j = 1, \dots, N_0, t \\ & \quad \sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \quad \forall j = N_0, \dots, N_1, t \\ & \quad \sum_{i=1}^{n_t} \sum_{t=1}^T F_{it} = TF \\ & \quad \sum_{j=1}^{N_1} \sum_{t=1}^T M_{jt} = TM \\ & \quad w_{N_0+1} \geq \dots \geq w_{N_1} \\ & \quad |\alpha_j| + |\beta_j| \leq M_0 w_j \quad \forall j \\ & \quad x_{it} y_{itj} + M_1(1 - y_{itj}) \geq z_{jt} \quad \forall i, t, j \\ & \quad x_{it} y_{itj} - M_1(1 - y_{itj}) \leq z_{jt} \quad \forall i, t, j \\ & \quad y_{itj} \in \{0, 1\} \quad \forall i, t, j \\ & \quad \alpha_j \in \mathbb{R}^n, \quad \beta_j \in \mathbb{R}^n, \quad w_j \in \mathbb{R}^n \quad \forall j \\ & \quad z_{jt} \in \mathbb{R}^n, \quad \forall j, t \end{aligned}$$

C. Adaptation of Heuristic

The heuristic for the scenario with ambiguity follows similarly from the heuristic developed under the scenario without ambiguity. The main difference is that now the options for making switches must include false alarms and missed detections. Therefore, framework of the algorithm is the same as for Algorithm 1, but now the heuristic randomly chooses from the following options.

- 1) Switch detection assignments between two existing targets
- 2) Switch the detection assignment of an existing target with a false alarm
- 3) Switch the detection assignment of an existing target with a missed detection identifier for a different existing target
- 4) Move the detection assignment of an existing target to a false alarm and replace with a missed detection identifier
- 5) Move a false alarm into the location of a missed detection identifier for an existing target

Like in the heuristic introduced early, this algorithm will accept the switch/move if the objective score improves, and reject the switch/move otherwise. The algorithm terminates under the same condition as the deterministic heuristic. We expect this algorithm to run slightly slower due to the vast increase in possible solutions.

VI. SCENARIO COMPLEXITY & PERFORMANCE METRICS

There does not exist a unified approach for measuring scenario complexity as stated by [13] nor does there exist an explanation of difficulty as it pertains to each of the MTT problems of data association and trajectory estimation. Here we suggest several metrics which can be used to quantitatively compare the complexity of numerous generated scenarios. Although the two challenges of data association and trajectory estimation are interrelated and we have shown that the two problems can be jointly solved, we argue that it is important to identify measures which score the difficulty of scenarios and performance of results separately. We begin by developing coefficients ρ which is an indicator of the level of complexity and difficulty of a given scenario. The first two metrics aim to capture the difficulty of the estimation problem, however they measure different difficulties. For ease of notation, let

$$D_{ijt} = \|\alpha_i^T + \beta_i^T * t - \alpha_j^T + \beta_j^T * t\| \quad (31)$$

The first proposed measure of ρ aims to capture the average distance between all pairwise trajectories.

$$\rho_1 = \frac{\sum_{t=1}^T \sum_{i < j} D_{ijt}}{\binom{P}{2} T \sigma} \quad (32)$$

The second measure of ρ aims to capture the average minimal distance between each trajectory and its closest trajectory.

$$\rho_2 = \frac{\sum_{i=1}^P \min_j \sum_{t=1}^T D_{ijt}}{PT\sigma} \quad (33)$$

These metrics have several desirable attributes. For example, the error is equivalent to the noise when $\rho = 1$. Lower values of ρ correspond to increasingly difficult scenarios, while higher values of ρ correspond to easier scenarios. For a set of fixed trajectories, raising σ increases the difficulty, while lowering σ decreases the difficulty.

The last measure of ρ aims to capture the difficulty of the data association problem. Let

$$c_{ijt} = \begin{cases} 1 & \text{if } D_{ijt} > 2\sigma \\ 0 & \text{otherwise} \end{cases}$$

The metric offers insight into the density of the scenario by measuring the percentage of close proximity detections.

$$\rho_3 = \frac{\sum_{t=1}^T \sum_{i < j} c_{ijt}}{\binom{P}{2} T} \quad (34)$$

Similarly, higher values of ρ_3 indicate easier scenarios because fewer detections are within close proximity, presumably leading to a decreased difficulty in the data association problem.

Similarly, we propose the use of different metrics to measure the performance of the data association and trajectory estimation problems separately. For the data association problem, which is naturally discrete, we propose the use of % accuracy *i.e* the ratio of the number of correct detection assignments to the total number of detections.

$$\text{Accuracy} = \frac{\# \text{ correct assignments}}{\text{Total \# of detections}} \quad (35)$$

Naturally the trajectory estimation problem calls for a performance metric which compares the distance of the ground truth to the estimation trajectory. Therefore, we propose the following metric.

$$\delta = \frac{\sum_{t=1}^T \sum_{j=1}^P \|\bar{x}_{jt} - \hat{x}_{jt}\|}{PT} \quad (36)$$

We match the true trajectories to the estimated trajectories using a one to one assignment problem which can be formulated as a linear program with the absolute distance between true and estimated as label costs.

The classification of simulated scenarios by difficulty and complexity is no easy task, and to the best of the authors' knowledge there is no accepted standard among the MTT community. We have proposed various metrics which allow for the intuitive and quantitative representation of these scenarios. Furthermore, the suggested performance metrics allow us to independently evaluate the performance of MTT algorithms, as it pertains to the data association and trajectory estimation problems.

VII. EXPERIMENTAL SIMULATIONS

There does not exist among the literature a clearly defined comprehensive set of simulation scenarios as pointed out by [13]. However, this work also noted that two types of scenarios of particular importance include crossing trajectories

and parallel trajectories. In agreement, we choose to develop scenarios of both types. We evaluated our methods on two separate experiments, one with detection ambiguity and one without.

A. Experiment 1

In order to evaluate scalability we test our methods across a range of scenarios with varying numbers of targets and scans. In particular we consider: $P \in \{4, 6, 8, 10\}$ and $T \in \{4, 6, 8, 10\}$. Scans are collected at a rate of 1 Hz. The cartesian product of P and T creates 16 unique scenario sizes. We generate 50 crossing scenarios and 50 parallel scenarios of each size. Additionally, we randomly generate 30 noise perturbations $\gamma \sim \mathcal{N}(0, 1)$. We consider noise coefficients $\sigma \in [0.1, 2.0]$ with step size 0.1. The error η of each detection is then $\sigma\gamma$. Detections are assembled by adding the simulated error to the true position.

$$x_{jt} = \alpha_j^T + \beta_j^T * t + \gamma\sigma \quad (37)$$

Scans \mathcal{X}_t are simulated by randomizing the order of x_{jt} for all t . Each unique \mathcal{X} generated is referred to as a *simulation*. Next, we begin testing our methods feeding the data into our algorithms, starting with the heuristic. For each simulation, we ran the heuristic with a range of starting points $N \in \{100, 1,000, 10,000\}$. Finally, we feed each heuristic solution as a warm start into the MIO and run the optimization solver. The optimization process was exited after $\text{minimin}\{\text{Run time: 2 min, Number of IP solutions: 15}\}$

B. Experiment 2

The second experiment serves as an extension of the first in order to test the performance of our algorithms under detection ambiguity. We consider the same ranges for P, T, γ , and σ

C. Implementation

Both experiments were implemented in the development software *julia* 0.4.3 [20] using the optimization package *JuMP* [21]. The implemented MIO utilized the optimization solver Gurobi 6.5.0[22]. Gurobi was limited to the use of a single core for the optimization processes. Each simulation was run on a single node of the unclassified TX-Green supercomputer located at Lincoln Laboratories [23]. [Insert computer specs (RAM, CPU, etc.) if available.]

VIII. COMPUTATIONAL RESULTS

IX. CONCLUSION AND FUTURE WORK

We presented a multi-target tracking approach which jointly solves the problems of data association and trajectory estimation. We accomplish this without the need of a trajectory bank nor the a prior computation of trajectory hypothesis. We demonstrated that the proposed method outperforms for linear trajectories....

ACKNOWLEDGMENT

The authors would like to thank Sung Son, Ph.D and Steve Relyea at Lincoln Laboratories for their efforts in problem clarification as well as for providing feedback and other expert knowledge on the problem details. Additionally, we would like to thank Lincoln Laboratories for and the LLGrid team for its assistance in running our experiments.

REFERENCES

- [1] S. Blackman, *Multiple-target Tracking with Radar Applications*, ser. Radar Library. Artech House, 1986. [Online]. Available: <https://books.google.com/books?id=Ag9TAAAMAAJ>
- [2] 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.
- [3] D. Reid, "An algorithm for tracking multiple targets," *Automatic Control, IEEE Transactions on*, vol. 24, no. 6, pp. 843–854, Dec 1979.
- [4] S. Blackman, "Multiple hypothesis tracking for multiple target tracking," *Aerospace and Electronic Systems Magazine, IEEE*, vol. 19, no. 1, pp. 5–18, Jan 2004.
- [5] 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>
- [6] 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.
- [7] 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.
- [8] R. Perry, A. Vaddiraju, and K. Buckley, "Multitarget list viterbi tracking algorithm," in *Signals, Systems and Computers, 1998. Conference Record of the Thirty-Second Asilomar Conference on*, vol. 1, Nov 1998, pp. 436–440 vol.1.
- [9] 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.
- [10] C. Carthel and S. Coraluppi, "Multi-hypothesis sonar tracking," in *Fusion 2004: Seventh International Conference on Information Fusion*, 2004.
- [11] 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>
- [12] 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.
- [13] G. Pulford, "Taxonomy of multiple target tracking methods," *Radar, Sonar and Navigation, IEE Proceedings -*, vol. 152, no. 5, pp. 291–304, October 2005.
- [14] Y. Bar-Shalom and X.-R. Li, *Estimation with Applications to Tracking and Navigation*. New York, NY, USA: John Wiley & Sons, Inc., 2001.
- [15] 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.
- [16] R. E. Bixby, "Mixed-integer programming: It works better than you may think," 2010. [Online]. Available: <http://www.ferc.gov/CalendarFiles/20100609110044-Bixby,%20Gurobi%20Optimization.pdf>
- [17] I. Gurobi Optimization, "Gurobi 6.5 performance benchmarks," 2015. [Online]. Available: <http://www.gurobi.com/pdfs/benchmarks.pdf>
- [18] 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.
- [19] 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>
- [20] 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>
- [21] 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>
- [22] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: <http://www.gurobi.com>
- [23] 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.