Mixed Integer Optimization for the Multi-Target Tracking Problem

Dimitris Bertsimas, Shimrit Shtern, and Zach 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

suggestions: Add the type of MTT models - sequential vs. batch , one sweep vs. multi sweep batch when describing types of MTT's since you use these terms later. I am still missing a more thorough discussion of robustness - what are the issues with missed detections and false alarms, how is this dealt with in current algorithms, and in the optimization algorithm that we cite, maybe one or two words about gating. This is very important since this is one of the goals of this work. I suggest maybe calling the case with false alarms and missed detections - detection ambiguity which justified the term robust. Add a brief discussion about evaluation of scenario difficulty as well as performance measures (we have three of them) and add as one of the papers contributions

you can not start the paper with the challenges of multi target tracking without first addressing what is target tracking, shortly at least of the audience is familiar with the matter and in more detail if they are not. So I would suggest something along the lines of:

Multi-target tracking the problem of estimation the state of multiple dynamic objects, in various points in time, which are referred to as targets, by using a set of unassociated detections from one or more sensors. The solution of this problem is of utmost importance in various military and civilian security applications (maybe some references).

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 these problems are solved separately using a combination of probabilistic approaches to determine data associations and filters to estimate trajectories. The global nearest neighbor algorithm, for example, is a naive 2-D assignment algorithm, which evaluates one scan you introduce a new term ”scan” without explaining what is means, later you do the same for the term ”pass” - maybe use a specific point in time instead of scan or define scan at a time, globally assigning the nearest detection at each scan [1]. Once the data association has been determined, the detections are passed through a Kalman filter why specifically the Kalman filter? although it might be the most common one various filters are used, so I would be more careful of my wording and use Kalman as an example rather then saying Kalman filter. Moreover, there are smoothing techniques which actually smooth the Kalman estimator backwards as well to update the trajectory estimates and the algorithm progresses forward sequentially [2].

Adding another depth of complexity, the Multiple Hypothesis Tracker (MHT) is that the only one? what about JPDAF (joint probabilistic association filter) and I'm sure there are some more, you can look in Bar-Shalom's book (the one I sent you) and there are probably more recent review papers as well. Actually the PDA uses a Bayesian approach to try to combine the two problems but in an ”online” fashion rather then a batch., first proposed by Reid in [3] has the ability to evaluate N scans back before making hard associations. However, the combination of associations grows exponentially with each scan, leading to tractability issues. This algorithm is heavily researched and is generally considered to be the modern standard for solving the data association problem, with numerous implementation methods as summarized by Blackman in [4].

All of the methods described so far are recursive, meaning that the algorithm advances sequentially through scans of data to solve the data association this is not necessarily true since you can apply the Kalman on a batch of detections or rather smooths the tracks according to the decisions and the MHT you describe also is a batch process, maybe you should start by defining the two ”modes” of data tracking batch vs. sequential giving examples of algorithms and uses,advantages and disadvantages of both, continually updating the state estimates along the way. Alternatively, batch method algorithms aim to solve data associations and state estimates of all scans known up to the point of time of interest. However, these are generally solved via multiple passes you didn't define pass here and should define it before use, but I'm sure that as I said Kalman can be applied in batch rather than multiple pass even though it might not be common to use it in that way rather than a single pass, refining the solution with each iteration [citation].

Compared to the number of probabilistic approaches available in the MTT literature, optimization focused methods are relatively lacking. The first presence of optimization applied to the MTT problem is the use of integer programming to leverage the MHT data association decision making process [5]. Somewhat similarly, linear programming has also been used to leverage the power of optimization with the MHT [6]. However, these methods still treat the data association and trajectory estimation problems separately. Finally, some aim to solve the MHT problem using Lagrangian relaxation [7].

Recently, there has been an effort to utilization optimization to solve both problems [8]. However, this approach requires the use of two methods solved iteratively to update one another, and is therefore considered a multi-pass batch algorithm. what about the number of tuning parameters or assumptions in the model? . I would rewrite the next sentence to say something along the lines of : "To the best of our knowledge, this paper is the first to suggest an optimization based approach that uses a single pass batch process to solve both the association and estimation problem simultaneously." The authors have not found a single pass batch algorithm which utilizes optimization to solve both problems simultaneously within the literature we have reviewed.

There is no shortage of literature on MTT methodologies. A more exhaustive overview of all MTT methods, including additional methods not discussed in this paper, can be found in [9].

You suddenly switch to talking about MIOs. Give a preliminary sentence to explain it: In this paper we suggest a mixed integer optimization (MIO) approach to solving the multi-target tracking problem. Although MIO are generally thought as intractable since 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. 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 improvement in performance [10]. 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 48.7x [11].

When evaluating MTT algorithms the literature is also lacking, since no is no standard way 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 [add citation] tries to deal with these issues by suggesting a unified measure of complexity and performance. In this paper we also suggest measures of complexity and performance which are related to the ones suggested in [put citation again] and address the connection between then on various scenarios, as well as apply them to measure the performance of the suggested model

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*). Our experiments demonstrate significant improvement in ....

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.

II. Problem Description

this description is much better and clearer so you can leave it as is

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 time steps. 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.

rearranged. Last assumption was redundant. changed the wording on the one before last, also added a assumption about the detection error.

Throughout the paper we make the following assumptions.

Assumption 1. *(i) All targets have constant velocity. i.e. Targets cannot change 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 at a given time.

(v) The detections generated by the sensor for each target at each point in time have independent errors.

Notation: We use P to indicate the true number of targets under observation, which for now we have assumed is known a priori. Targets are observed over a time window in which detections are collected at T time steps. T is always known because we know the number of detections which have been collected a priori. The i^{th} detection at the t^{th} time step is indicated by x_{it}. Trajectories *j* are parameterized by an initial position α_j and an constant velocity β_j, which we aim to estimate. Notice that x_{it}, α_j and β_j, may all be vectors however for simplicity we treat them in the following as scalars.

when describing the objective function it might be worth mentioning that for vectors we can extend the result for either l1 or l2 norm for the sum absolute values and RSS, respectively.

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).

bad constructed setence you use decision to explain decision. most of the description is to general you need to relate it to the problem at hand.

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 to 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 initial position and velocity of each trajectory *j*. In our interpretation of the MTT problem we have 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:

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

B. Objective Function

Next, we would like to develop a function which accurately captures 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 decision variables, the estimated position of the linear trajectory *j* at time *t* is given by:

$$\hat{x}_{jt} = \alpha_j - \beta_j * t \tag{2}$$

For the trajectory estimation problem we wish to minimize the distance between x_{it} and $\hat{x}_{jt}$. In other words we wish to minimize $\|x_{it} - \hat{x}_{jt}\|$ for some norm, for detections x_{it} assigned to trajectory j. [Enter discussion of RSS vs. l1 here and use the corresponding norm in your formulation and also show how the l1 norm can be reformulated as a linear program, which is not available for the RSS] Substituting (2) for $\hat{x}_{jt}$ we arrive at our objective function:

$$\underset{\alpha_j,\beta_j}{\text{minimize:}} \sum_{i->j} \sum_{t=1}^{T} \|x_{it} - \alpha_j - \beta_j * t\| \tag{3}$$

For the data association problem, we wish to only penalize the objective function when x_{it} has been assigned to trajectory *j*, which occurs when $y_{itj} = 1$. An easy method to enforce this using our current variables would be to construct an interaction term such as: $|y_{itj}x_{it} - \alpha_j - \beta_j * t|$. However, this creates an undesirable non-linearity this is still a linear problem since x_{it} are given, and the norm is an l1 norm..... actually come to think about it why didn't we just use

$$\| \sum_i y_{itj}x_{it} - \alpha_j - \beta_j * t\|_1$$

I mean we actually need z_{jt} only when there is a missed detection...which actually means that the use of big M here is quite redundant. We may want to consider this reformulation but present the original formulation in order to use it later when describing ambiguity in detections. Can you actually check if this formulation is faster for the 10 by 10? this can not be done for the RSS by the way which is another instantiate to show both forms which can be linearized through the introduction of 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 $y_{itj} = 0$. Using this method we arrive at the final objective function.

$$\underset{\alpha_j,\beta_j,z_{jt}}{\text{minimize:}} \sum_{j=1}^{P}\sum_{t=1}^{T} |z_{jt} - \alpha_j - \beta_j * t| \tag{4}$$

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*:

$$\sum_{j=1}^{P} y_{itj} = 1 \qquad \forall i, t \tag{5}$$

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

$$\sum_{i=1}^{P} y_{itj} = 1 \qquad \forall j, t \tag{6}$$

Additionally, constraints can be used to reduce the number of variables in the problem, increasing the simplicity and interpretability of a model. This can also greatly improve the performance of the model. 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* ($y_{itj} = 1$). We enforce this effect using the following constraint:

$$M_t(1 - y_{itj}) \geq |z_{jt} - x_{it}y_{itj}| \qquad \forall i, t, j \tag{7}$$

where $\underset{y_{itj}}{\text{maximize}}\{x_{it}\}$ for each time step. This can be done This can be written equivalently as the following set of two linear constraints:

$$x_{it}y_{itj} + M(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j \tag{8}$$
$$x_{it}y_{itj} - M(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j \tag{9}$$

D. Full Formulation

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

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

IV. Heuristic

In swap k for time step t choose $i, l \in \{1, \ldots, P\}$ detections and for $j, m \in \{1, \ldots, P\}$ targets such that $y^k_{itj} = 1$ and $y^k_{ltm} = 1$ switch the detection association so that $y^{k+1}_{ltj} = 1$, and $y^{k+1}_{itm} = 1$, recalculate $\alpha_j, \beta_j, \alpha_m$ and β_m . Compute the objective using the new associations and return the swap if the objective does not improve. 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 a subset of the N starting points on a separate core.

V. Robust MIO Model

In this section we treat the case of detection ambiguity. 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 time steps.*

(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 $N_0 \leq P \leq N_1$.

Here add why we first treat a fixed number of targets and how we can use this formulation via parallelization to solve the problem. Then state that 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 earlier formulation 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.

$$\sum_{j=1}^{P} y_{itj} + F_{it} = 1 \qquad \forall i, t \tag{10}$$

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

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \qquad \forall j, t \tag{11}$$

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

$$\sum_{i=1}^{n_t}\sum_{t=1}^{T} F_{it} = TF \tag{12}$$

$$\sum_{j=1}^{N_1}\sum_{t=1}^{T} M_{jt} = TM \tag{13}$$

3) Full Formulation:

$$\begin{aligned}
\underset{y_{itj},\alpha_j,\beta_j,z_{jt},F_{it},M_{jt}}{\text{minimize:}} \quad & \sum_{j=1}^{P}\sum_{t=1}^{T} |z_{jt} - \alpha_j - \beta_j * t| + \theta_o TF + \phi_0 TM \\
\text{subject to:} \quad & \sum_{j=1}^{P} y_{itj} + F_{it} = 1 \qquad \forall i, t \\
& \sum_{i=1}^{n_t} y_{itj} + M_{jt} = 1 \qquad \forall j, t \\
& \sum_{i=1}^{n_t}\sum_{t=1}^{T} F_{it} = TF \\
& \sum_{j=1}^{P}\sum_{t=1}^{T} M_{jt} = TM \\
& x_{it} y_{itj} + M_1(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j \\
& x_{it} y_{itj} - M_1(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j \\
& y_{itj} \in \{0, 1\} \quad \forall i, t, j \\
& M_{jt} \in \{0, 1\} \quad \forall j, t \quad F_{it} \in \{0, 1\} \quad \forall i, t \\
& \alpha_j \in \mathbb{R}^n, \quad \beta_j \in \mathbb{R}^n \quad \forall j \\
& z_{jt} \in \mathbb{R}^n, \quad \forall j, t \\
& TF \in \mathbb{Z}, \quad TM \in \mathbb{Z}
\end{aligned}$$

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.

$$\sum_{j=1}^{N_1} y_{itj} + F_{it} = 1 \qquad \forall i, t \tag{14}$$

Similarly, the sum of all missed detections is TM.

$$\sum_{j=1}^{N_1}\sum_{t=1}^{T} M_{jt} = TM \tag{15}$$

as we discussed you can state here that 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. Since they are binary the constraint which binds them between N_0 and N_1 is redundant and the additional constraint $w_{N_0+1} \geq ... \geq w_{N_1}$ will guarantee a unique w solution

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

$$\sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \qquad \forall j, t \tag{16}$$

Working of the assumption that the true number of targets *P* falls between N_0 and N_1, we must introduce this as a constraint to our model. The sum of all existing targets is bound by N_0 and N_1.

$$N_0 \leq \sum_{j=1}^{N_1} w_j \leq N_1 \tag{17}$$

Finally, 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.

$$|\alpha_j| + |\beta_j| \leq M_0 w_j \qquad \forall j \tag{18}$$

3) Full Formulation 2: Incorporating these additional variables and constraints, we arrive at the following complete alternative formulation.

$$\begin{aligned}
\underset{y_{itj},\alpha_j,\beta_j,z_{jt},F_{it},M_{jt}}{\text{minimize:}} \quad & \sum_{j=1}^{N_1}\sum_{t=1}^{T} |z_{jt} - \alpha_j - \beta_j * t| + \theta_o TF + \phi_0 TM \\
\text{subject to:} \quad & \sum_{j=1}^{N_1} y_{itj} + F_{it} = 1 \qquad \forall i, t \\
& \sum_{i=1}^{n_t} y_{itj} + M_{jt} = w_j \qquad \forall j, t \\
& \sum_{i=1}^{n_t}\sum_{t=1}^{T} F_{it} = TF \\
& \sum_{j=1}^{N_1}\sum_{t=1}^{T} M_{jt} = TM \\
& N_0 \leq \sum_{j=1}^{N_1} w_j \leq N_1 \\
& |\alpha_j| + |\beta_j| \leq M_0 w_j \qquad \forall j \\
& x_{it} y_{itj} + M_1(1 - y_{itj}) \geq z_{jt} \qquad \forall i, t, j \\
& x_{it} y_{itj} - M_1(1 - y_{itj}) \leq z_{jt} \qquad \forall i, t, j \\
& y_{itj} \in \{0, 1\} \quad \forall i, t, j \\
& \text{Erase no need} - M_{jt} \in \{0, 1\} \quad \forall j, t \quad F_{it} \in \{0, 1\} \quad \forall i, t \\
& \alpha_j \in \mathbb{R}^n, \quad \beta_j \in \mathbb{R}^n \quad \forall j \\
& z_{jt} \in \mathbb{R}^n, \quad \forall j, t \\
& \text{Erase no need} TF \in \mathbb{Z}, \quad TM \in \mathbb{Z}
\end{aligned}$$

C. Adaptation of Heuristic

The heuristic for the robust scenario follows similarly from the heuristic from the deterministic scenario. 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 deterministic heuristic, the 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. Performance Metrics

add definition of scenario complexity here and relate to the appropriate performance metrics, add the additional performace metric of RSS (or abslute value) as well as association accuracy

Although two challenges of data association and trajectory estimation are interrelated, we argue that it is important to identify performance metrics which are unique to each problem. For the data association problem, which is naturally discrete, we propose the use of % accuracy *i.e* the number of correction detection labels. 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.

$$\delta = \frac{\sum_{t=1}^{T}\sum_{i=1}^{P} |Distance_{true,estimated}|}{PT}$$

VII. Experimental Simulations

There does not exist among the literature a clearly defined comprehensive set of simulation scenarios as pointed out by [9]. 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, with the key difference in the two experiments being that the number of targets is known in the first but unknown in the second.

A. Experiment 1

We considered a fixed state space of [-5,5], only allowing target trajectories to exist within this window. We began by randomly generate 200 unique scenarios of true trajectories of various scenario sizes. We tested a range of possible number targets: [4, 6, 8, 10] and similarly a range of time steps: [4, 6, 8, 10] Additionally, we randomly generate 30 unique normally distributed perturbations with mean 0 and standard deviation 1. We considered σ values in the range of [0.1, 2.0] with step size 0.1. We assemble the noise for each detection by multiplying the perturbation by σ. Detections were assembled by adding noise to the true trajectories and randomizing the detection order. and randomizing the detection order.

Next, we began testing our methods by running the heuristic. For each simulation, we ran the heuristic with a range of starting points (N), [100 1,000 10,000].

Finally, we feed each heuristic solution as a warm start into the MIO and run. The optimization process was exited after minimin{Run time: 2 min, Number of IP solutions: 15}

B. Experiment 2

VIII. Scenario Complexity Coefficient (ρ)

We develop a coefficient ρ for use in describing the difficulty of a given scenario. As mentioned previously, ambiguity exists in describing scenario complexity. Yet it is important to have metrics which capture the differences among scenarios. Furthermore, the data association and trajectory estimation problems are very different by nature and therefore have. The first three metrics

$$\rho = \frac{\sum_{t=1}^{T} \sum_{i<j} |Distance_{ijt}|}{\binom{P}{2} T} \tag{19}$$

This metric has 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.

IX. Computational Results

X. 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.

References

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

[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] 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.

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

[7] 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

[8] 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.

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

[10] 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

[11] I. Gurobi Optimization, "Gurobi 6.5 performance benchmarks," 2015. [Online]. Available: http://www.gurobi.com/pdfs/benchmarks.pdf