A Dual-Ascent Algorithm for the Multi-dimensional Assignment Problem: Application to Multi-Target Tracking

Samhita Vadrevu, Rakesh Nagi
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA
{samhita3, nagi}@illinois.edu

***Abstract*—Recently we proposed a new Mixed-Integer Linear Programming formulation for the Multi-Target Tracking (MTT) problem and used a standard optimization solver to demonstrate its viability [1]. Subsequently, we provided Graphics Processing Unit (GPU) accelerated algorithms for the underlying Multi-dimensional Assignment Problem (MAP) with decomposable costs or triplet costs using a Lagrangian Relaxation (LR) framework. Here, we present a Dual-Ascent algorithm that provides monotonically increasing lower bounds and converges in a fraction of iterations required for a subgradient scheme. This approach can handle a large number of targets for many time steps with massive parallelism and computational efficiency. The dual-ascent framework decomposes the MAP into a set of Linear Assignment Problems (LAPs) for adjacent time-steps, which can be solved in parallel using the GPU-accelerated method of [2], [3]. The overall dual-ascent algorithm is able to efficiently solve problems with 100 targets and 100 time-frames with high accuracy. We demonstrate the applicability of our new algorithm to MTT by including realistic issues of missed detections and false alarms. Computational results demonstrate the robustness of the algorithm with good MMEP and ITCP scores and solution times for the larger problems in less than 6 seconds.**

***Index Terms*—Multi-dimensional Assignment, Data Association, Multi-target tracking, Mixed-Integer Linear Programming, Lagrangian Relaxation, GPU computing.**

I. Introduction

The multi-target tracking problem has been largely studied and approached in various methods ranging from mathematical programming models, Multi-Hypothesis Tracking (MHT) to Joint Probability Data Association (JPDA). In this paper, we propose one such mathematical formulation, called the Multi-dimensional Assignment Problem (MAP) [4] that lends itself well to the MTT problem. A typical MAP setting has T dimensions, and each dimension has N nodes. MAP is a generalization of the Linear Assignment Problem (LAP), with a *T-partite* graph in consideration, as opposed to the *bipartite* graph in the LAP case.

MTT is formulated as a MAP in the following way. The sensors observe the targets for a particular time period. This time period is divided into multiple snapshots of time, also referred to as *time frames*, and a measurement of the target (information on the position, velocity, etc.) at each time frame is captured. We represent each target by a node, and the dimensions in the graph represent a time frame. The MAP is now solved for optimal assignments (or near-optimal) for each node across all the dimensions. The assignment of a node to another node in each time frame is interpreted as the track of the target.

In this paper, the formulation allows one to consider the triplet costs between the targets. As discussed in [5] and [6], accounting for the triplet costs has shown better results in terms of accuracy. This is indeed expected as higher order variables are able to capture more insights of the trajectory, that is induced in to the problem through the objective function coefficients.

A dual ascent-based technique is employed to obtain a near optimal solution to the MAP. The quadratic based formulation for MAP [7] is linearized using a creative linearization technique and is then relaxed using Lagrangian multipliers. As per our knowledge, the dual ascent-based technique has not yet been applied to a multi-dimensional assignment problem. Similar techniques have been applied to Quadratic Assignment Problems (QAP) [8] and resulted in strong bounds. The algorithm also guarantees a monotonic increase in the lower bound, thus avoiding any oscillatory behavior in search of better Lagrange multipliers for the lower bound. We note that MTT has been modeled using a variety of formulations [9], [10], [11]. Each formulation has different variables and constraints and, therefore have different solvability using a standard solver. Some formulations lend themselves to decomposition methods and others might not. The particular MTT formulation presented in this paper is unique in that it lends itself to decomposition and its relaxed dual formulation provides basis of monotonically increasing bounds. This makes it superior in solution time and possibly quality, which is the context of this contribution.

II. Related Literature

MTT has previously been studied using other approaches. Some of the prominent works are the formulation of MTT as combinatorial models by [12] and a Joint Probabilistic Data Association Model (JPDA) to solve the Multi-Hypothesis Tracking (MHT) in [13]. MTT has also been formulated as a

MAP by [14], and a sliding window technique was used. A series of genetic algorithms were proposed to solve MTT by [10], [15], [16] and [17].

MAP is NP-Hard, and exact algorithms become computationally prohibitive beyond a problem size. In this paper, we propose a dual ascent-based technique to obtain accurate solutions. MAP has also been employed to other applications in the fields of Weapon Target Assignment (WTA), resource allocation, Natural Language Processing (NLP), and other data association based applications. Considering the impact of these problems, a robust and scalable heuristic is desired.

III. Problem Formulation

The previously described MTT problem is modeled as a T-dimensional assignment problem in the following manner. Each dimension is a time frame $t = 1, \ldots, T$. The nodes in each dimension t correspond to the observations of a target in that time frame t. *Nota Bene*: We will deal with missing observations and spurious observations later. Each assignment, over the T dimensions, is the track associated with each target estimated based on the observations. The objective function is to minimize the score associated with each association. In this paper, the objective coefficients are both pairwise and triplet scores between observations, so as to better model maneuverable constant velocity targets, as explained in [1], [7], [18]. In Fig. 1, a tracking problem is illustrated as a T-dimensional assignment problem. The assignment (across all the dimensions) of node 1 originating in dimension $t = 0$ translates to the track of Target 1 in discrete time frames throughout the time T. MTT problems are modeled as MAP from [19] and is rewritten, by considering the set of trajectories as a union of edges on all paths. The typical LAP one-to-one assignment requirement between stages is assumed. That is each observation in time t is going to be assigned to one and only one observation in time $t + 1$. Our triplet formulation associates three adjacent stages simultaneously, akin to 3-dimensional matching, which is known to be NP-hard. In the next section, the mathematical formulation of MAP and its linearization are provided.

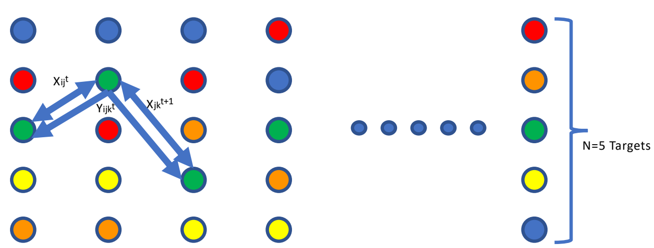

Fig. 1. Multi-Dimensional Assignment Problem Representation

A. Mathematical formulation of the MAP

An MAP can be described as having T dimensions and N nodes in each dimension. The objective is to find the assignment of each node across the dimensions such that the objective function is minimized. Consider the graph $G = (V_1, V_2, V_3, ..., V_T)$, with $|V_1| = |V_2| = |V_3|... = |V_T| = N$. The dimensions are denoted as p, and $p = 0, \ldots, T - 1$. The variables are of the form x_{ij}^p where node i is in frame p and node j in $p + 1$. The associated pairwise costs are denoted by C_{ij}^p, and the triplet costs for two variables from three consecutive stages are denoted by D_{ijk}^p, where nodes $i,\ j,\ k$ are from stages $p,\ p+1,\ p+2$. See [18] for additional details and the difference between MAP with decomposable costs and the one with triplet costs that span three dimensions.

MAP:

$$\min \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{p=0}^{T-2} C_{ij}^p x_{ij}^p + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\sum_{p=0}^{T-3} D_{ijk}^p x_{ij}^p x_{jk}^{p+1} \quad (1)$$

$$\text{s.t.} \sum_{i=0}^{N-1} x_{ij}^p = 1, \quad j = 0, \ldots, N-1,\ p = 0, \ldots, T-2; \quad (2)$$

$$\sum_{j=0}^{N-1} x_{ij}^p = 1, \quad i = 0, \ldots, N-1,\ p = 0, \ldots, T-2; \quad (3)$$

$$x_{ij}^p \in \{0, 1\}, \quad i, j = 0, \ldots, N-1,\ p = 0, ...T-2. \quad (4)$$

Linearization: We introduce a new variable y to denote the product of two x variables for linearizing the above formulation. It is expressed as $y_{ijk}^p = x_{ij}^p \times x_{jk}^{p+1}$, where the nodes i, j, k originate from dimensions p, $p+1$, $p+2$. The y variable connects the two x variables from three consecutive dimensions and is used to depict a chain of edges formed by the two x variables. We refer to this linearized formulation as Reformulated Multi-dimensional Assignment Problem (RMAP). In addition to the uniqueness constraints of x in the MAP formulation, RMAP consists of the constraints corresponding to the y variables. The non-zero property of y_{ijk}^p, is contingent upon the non-zero property of x_{ij}^p. That is for a fixed (i, j, p), the edge y_{ijk}^p, for any k in stage $p + 2$, exists (non-zero), if and only if the edge x_{ij}^p exists (non-zero). This property can be expressed mathematically as

$$\sum_{k=0}^{N-1} y_{ijk}^p = x_{ij}^p. \quad (5)$$

This is also true for the other end of the chain formed by y_{ijk}^p. That is y_{ijk}^p for any i in stage p, is non-zero if and only if the edge x_{jk}^{p+1} is non zero, and is expressed as

$$\sum_{i=0}^{N-1} y_{ijk}^p = x_{jk}^{p+1}. \quad (6)$$

RMAP is later relaxed using Lagrange multipliers. For the ease of multiplier update, which will be explained in Section IV-B, the equation (6) is replaced by another equivalent constraint:

$$\sum_{i=0}^{N-1} y_{ijk}^p = \sum_{i=0}^{N-1} y_{jki}^{p+1}. \quad (7)$$

With the above linearization and associated constraints, the final RMAP formulation is summarized as:

RMAP:

$$\min \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{p=0}^{T-2} C_{ij}^p x_{ij}^p + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\sum_{p=0}^{T-3} D_{ijk}^p y_{ijk}^p \quad (8)$$

$$\text{s.t.} \sum_{i=0}^{N-1} x_{ij}^p = 1, \quad j=0,\ldots,N-1,\ p=0,\ldots,T-2; \quad (9)$$

$$\sum_{j=0}^{N-1} x_{ij}^p = 1, \quad i=0,\ldots,N-1,\ p=0,\ldots,T-2; \quad (10)$$

$$\sum_{k=0}^{N-1} y_{ijk}^p = x_{ij}^p \quad i,j=0,\ldots,N-1,\ p=0,\ldots,T-3; \quad (11)$$

$$\sum_{k=0}^{N-1} y_{kij}^{T-3} = x_{ij}^{T-2} \quad i,j=0,\ldots,N-1; \quad (12)$$

$$\sum_{i=0}^{N-1} y_{ijk}^p = \sum_i y_{jki}^{p+1} \quad j,k=0,\ldots,N-1,\ p=0,\ldots,T-4; \quad (13)$$

$$x_{ij}^p \in \{0,1\} \quad i,j=0,\ldots,N-1,\ p=0,\ldots,T-2; \quad (14)$$

$$y_{ijk}^p \geq 0 \quad i,j,k=0,\ldots,N-1,\ p=0,\ldots,K-3. \quad (15)$$

IV. Lagrangian Relaxation for RMAP

RMAP formulation has a large number of constraints and variables, making it quite difficult to solve. To overcome this challenge, RMAP is relaxed using Lagrangian multipliers (LRMAP) and a linear program with the binary condition on x relaxed (LPLRMAP) is derived. We then take the dual (DLRMAP) of this (primal) LP. This is because the dual problem can be decomposed into smaller subproblems which makes it easier to solve. In the RMAP formulation as shown above, the constraint (13) is relaxed and added to the objective function with the corresponding multiplier $\hat{\mathbf{v}}$.

The Lagrangian Relaxation of the RMAP formulation is then:

$$\min \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{p=0}^{T-2} C_{ij}^p x_{ij}^p + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\sum_{p=0}^{T-3} D_{ijk}^p y_{ijk}^p - \sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\sum_{p=0}^{T-4} \hat{v}_{jk}^p \Big(\sum_{i=0}^{N-1} y_{ijk}^p - \sum_{i=0}^{N-1} y_{jki}^{p+1}\Big) \quad (16)$$

$$\text{s.t. } (9)-(10) \quad (17)$$

$$(11)-(12) \quad (18)$$

$$x_{ij}^p \in \{0,1\},\ y_{ijk}^p \geq 0 \qquad \forall i,j,k,\ p=0,\ldots,T-2. \quad (19)$$

To save memory, similar to [8], the multipliers of the associated y variables from (13), are stored in the same variable as the single y variable. Whenever a multiplier is updated, the other associated multipliers are adjusted accordingly, to keep the sum of the multipliers the same.

The Lagrangian relaxation formulation of the MAP after the above modifications is:

LRMAP:

$$\min \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{p=0}^{K-2} C_{ij}^p x_{ij}^p + \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}\sum_{k=0}^{N-1}\sum_{p=0}^{T-3} (D_{ijk}^p - \hat{v}_{jk}^p) y_{ijk}^p \quad (20)$$

$$\text{s.t. } (9)-(10) \quad (21)$$

$$(11)-(12) \quad (22)$$

$$x_{ij}^p \in \{0,1\},\ y_{ijk}^p \geq 0 \qquad \forall i,j,k,\ p=0,\ldots,T-2. \quad (23)$$

In order to dualize RMAP, the Lagrangian Relaxation is relaxed in terms of the binary condition on x.

A. Dualization and Decomposition

The LP relaxation of the above formulation is dualized with $\alpha, \beta, \gamma, \delta$ as the dual variables corresponding to the constraints (9), (10), (11), (12):

DLRMAP:

$$\max \sum_{j=0}^{N-1}\sum_{p=0}^{T-2} \alpha_{jp} + \sum_{i=0}^{N-1}\sum_{p=0}^{T-2} \beta_{ip} \quad (24)$$

$$\text{s.t. } \alpha_{jp} + \beta_{ip} - \gamma_{ij}^p \leq C_{ij}^p, \quad \forall i,j,\ p=0,\ldots,T-3; \quad (25)$$

$$\alpha_{j(T-2)} + \beta_{i(T-2)} - \delta_{ij} \leq C_{ij}^{T-2}, \forall i,j; \quad (26)$$

$$\gamma_{ij}^p \leq D_{ijk}^p - \hat{v}_{jk}^p, \quad \forall i,j,k,\ p=0,\ldots,T-4; \quad (27)$$

$$\gamma_{ij}^{(T-3)} + \delta_{jk} \leq D_{ijk}^{T-3} - \hat{v}_{jk}^{T-3}, \quad \forall i,j,k; \quad (28)$$

$$\alpha_{jp}, \beta_{ip}, \gamma_{ij}^p \in \mathbb{R}, \quad \forall i,j,k,\ p=0,\ldots,T-2. \quad (29)$$

It is easy to observe that the dual problem is decomposed into x and y subproblems.

From (25), for $\sum_j \sum_p \alpha_{jp} + \sum_i \sum_p \beta_{ip}$ to be maximized, γ_{ij}^p has to be large. However by (27), the γ_{ij}^p variable has to be constrained according to the following y subproblem:

$$\Delta_{ij}^p(\hat{v}) = \max \left\{ \gamma_{ij}^p \middle| \gamma_{ij}^p \leq \hat{D}_{ijk}^p, \forall\ i,j,k,p \right\},$$

where $\hat{D}_{ijk}^p = D_{ijk}^p - \hat{v}_{jk}^p$.

There are N^2T y subproblems and for every i,j,p, the cost $\Delta_{ij}^p(\hat{v})$, is transferred to the x costs C_{ij}^p in (25) and the x subproblem then becomes:

$$\nu(\textbf{DLRMAP}(\hat{v})) = \max \sum_{j=0}^{N-1}\sum_{p=0}^{T-2} \alpha_{jp} + \sum_{i=0}^{N-1}\sum_{p=0}^{T-2} \beta_{ip}$$
$$\text{s.t. } \alpha_{jp} + \beta_{ip} \leq C_{ij}^p + \Delta_{ij}^p(\hat{v}), \forall i,j,p. \quad (30)$$

The x subproblem, as shown above, can be decomposed into a set of LAPs (along T) in terms of dual variables corresponding to the x variable in the LPLRMAP formulation. These can be solved using the GPU accelerated Hungarian algorithm for assignment problems by [2] and its dynamic counterpart in [3].

Without proof, we note that the integrality property [20] holds for LRMAP, and therefore, even with the optimal multipliers, it can best approach the LP relaxation of RMAP. In the next section, the dual ascent procedure and determination of the step size is explained in detail.

B. Dual Ascent Procedure and Multiplier Update

We note that the constraints (26) and (28) correspond to the last stages. With these constraints, dual ascent for multiplier update requires solving a linear program for δ_{ij} and γ_{ij}^{T-3}. To avoid this complication, a dummy stage is added to the end so that the last stage is accounted for in the computations and does not require additional multiplier update and the associated computational effort. Therefore, in all the computations p ranges from 0 to T in x_{ij}^p and for y_{ijk}^p, p can take values from 0 to $T-2$. Thus, the total number of x subproblems are T and y subproblems are $T-1$. The multiplier update scheme is divided into two steps for easier understanding.

a) STEP 1: In the dual formulation DLRMAP, the x subproblem is an LAP in the dual form, and the y subproblem is a simple linear program. The y subproblem is solved first, and then the value obtained from the y subproblem, Δ_{ij}^p is transferred to the x subproblem which is then solved as a LAP. For any fixed $i,\ j,\ k,\ p$, the corresponding y subproblem is:

$$\Delta_{ij}^p(\hat{v}) = \max\left\{\gamma_{ij}^p \middle| \gamma_{ij}^p \leq \hat{D}_{ijk}^p, \forall\ i,j,k,p\right\}.$$

To solve this LP, the minimum of $\hat{D}_{ijk}^p$ is assigned to γ_{ij}^p, so as to maintain dual feasibility. After solving, consider $\pi(y_{ijk}^p)$ to be the slack of the variable y_{ijk}^p. Since not all constraints are active, a multiplier $\hat{v}_{jk}^p$ can be updated using a fraction (κ_1) of the slack variable. As explained in [21], the minimum over i of all the N slacks is considered and is used to update the multiplier $\hat{v}_{jk}^p$.

$$\hat{v}_{jk}^p \leftarrow \hat{v}_{jk}^p + \kappa_1 \min_i \pi(y_{ijk}^p).$$

Similarly, the multiplier $\hat{v}_{ki^*}^p$ is updated as,

$$\hat{v}_{ki^*}^p \leftarrow \hat{v}_{ki^*}^p - \kappa_2 \min_i \pi(y_{jki}^{p+1}).$$

Here, $0 \leq \kappa_1, \kappa_2 \leq 1$ and $\kappa_1 + \kappa_2 \leq 1$ as the sum of the multipliers remains constant. (Note: For the computational experiments, we have set $\kappa_1 = 0.56$ and $\kappa_2 = 0.44$.)

As shown in [8], to reduce memory usage, it is sufficient to store, adjust, and update the costs of $\hat{D}$ and $\hat{C}$, instead of explicitly storing and updating the multipliers separately. In a similar spirit, we perform the cost update in terms of the $\hat{D}$ and $\hat{C}$ costs as follows:

$$\hat{D}_{i_1jk}^p = \kappa_1 \min_i \hat{D}_{ijk}^p + \kappa_2 \min_i \hat{D}_{jki}^{p+1}.$$

$$\hat{D}_{jki_2}^{p+1} = \kappa_1 \min_i \hat{D}_{ijk}^p + \kappa_2 \min_i \hat{D}_{jki}^{p+1}.$$

where $i_1 = \arg\min_i D_{ijk}^p$ and $i_2 = \arg\min_i \hat{D}_{jki}^{p+1}$, and the cost distribution from $\hat{C}$ costs is

$$\hat{D}_{ijk}^p \leftarrow \hat{D}_{ijk}^p + \hat{C}_{ij}^p,\ \forall i,j,k,p.$$

There will be discrepancy in the multiplier values if all the $D's$ are updated corresponding to a fixed j, k and p, since we are only updating the multiplier $\hat{v}_{jk}^p$. Thus, **argmin** is considered when adjusting the y costs.

V. Dual Ascent Algorithm

An accompanying benefit of the Lagrangian Relaxation approach is that one not only has a provable (and hopefully tight) lower bound, but the infeasible solution can be repaired in the neighborhood to result in excellent (possibly optimal) solution(s). Since these repaired solutions are feasible to the original problem, the best of these can serve as an upper bound (UB) to RMAP.

A. Finding a feasible solution (Upper Bound)

A simple (possibly lazy) approach to construct a feasible solution from the DLRMAP solution is to connect the pair-wise assignments of adjacent time steps (x) through the T dimensions. The y variables are simply computed from the x values as $y_{ijk}^p = x_{ij}^p \times x_{jk}^{p+1}$. Such a solution will not have any conflicts in the y and the x variables. The objective value (8) is computed using this feasible solution. Now that we have an UB from the feasible solution and a LB from DLRMAP, we can assess an optimality gap. This gap is defined as

$$\text{gap} = \frac{\text{UB} - \text{LB}}{\text{UB}}.$$

The algorithm iterates until: (1) the gap is greater than 0.1% (or some user-defined threshold); (2) the gap improvement has slowed down; or (3) the maximum number of iterations permitted is reached.

B. Algorithm Pseudo code

Algorithm 1 presents the dual ascent algorithm without GPU acceleration. For sake of brevity we don't discuss the GPU accelerated versions in this paper. An interested reader can refer to it in [21].

VI. Computational Experiments

The scoring scheme is a crucial part of algorithms based on mathematical programming methods. The scores, which become the objective function coefficients need to represent the system under study accurately. In this paper, the data simulates the behavior of maneuverable targets moving under constant velocity and the scores are computed as described in [1]. The idea behind this method is to obtain pairwise costs, C_{ij}^p, and compute the lengths of spline curves (l) between two successive pairs of coordinate positions $(x_i(p), y_i(p))$ and $(x_j(p{+}1), y_j(p{+}1))$, measured at time frames p and $p{+}1$. For the triplet costs, the absolute value of the difference between the curve lengths (l_{ij}^p, l_{jk}^{p+1}) is computed to provide D_{ijk}^p. For each problem of sizes N, T = {(5, 5), (5, 10), (10, 5), (10, 10)}, five instances have been generated. For the larger problems of sizes {(20, 20), (25, 25), (30, 30), (40, 40), (50, 50), (100, 100)}, one problem instance for each size has been generated.

Gurobi, a standard MILP solver [22], can be used to compare the solutions for problems of size less than 40. But

Algorithm 1: RMAP DUAL ASCENT (RMAP-DA)

1) Cost Initialization:
   a) Initialize $m \leftarrow 0$, $\mathbf{v}^m \leftarrow \mathbf{0}$, $\bar{\nu}(\text{DLRMAP}) \leftarrow -\infty$, and GAP $\leftarrow \infty$.
   b) Initialize cost matrices C and D.
2) Termination: Stop if $m >$ ITN_LIM or GAP $<$ MIN_GAP or LB improvement over last 5 iterations is $< 0.1\%$
3) Cost Distribution and solving the y subproblem:
   a) Update the dual multipliers $(\hat{v}^p_{jk})^m \leftarrow (\hat{v}^p_{jk})^{m-1} + \lambda_{ijkp}$
   b) Update $\hat{D}^p_{ijk} \leftarrow \hat{D}^p_{ijk} - (\hat{v}^p_{jk})^m, \forall (i,j,k,p)$
4) Solving the y subproblem:
   a) Solve y subproblem and cost coefficients $\hat{D}$.
   b) Let $\Delta^p_{ij}(\mathbf{u}^m) \leftarrow \nu(y\text{ subproblem}(i,j,p)), \forall\ (i,j,p)$.
   c) Update $\hat{C}^p_{ij} \leftarrow \hat{C}^p_{ij} + \Delta^p_{ij}(\mathbf{v}^m),\ \forall\ (i,j,p)$.
5) Solve x-LAPs:
   a) Solve T x LAPs of size $N \times N$ and cost coefficients $\hat{C}$.
   b) Update $\nu(\text{DLRMAP}(\mathbf{v}^m)) \leftarrow \nu(x - \text{LAPs})$.
   c) Compute Upper Bound (UB) and GAP.
6) Update $m \leftarrow m + 1$. Return to Step 2.

like most off-the-shelf solvers, Gurobi too fails to scale and cannot be used for larger problem instances. On the other hand, the dual ascent-based MAP solver is a robust algorithm that is scalable and highly accurate. We report and compare the solution times for the smaller instances solved using Gurobi and RMAP-DA. For the larger sized problems ($N, T \geq 40$), RMAP-DA is used to find the solutions and the scores and times have been reported.

A. Evaluation Criteria

To evaluate the performance of the algorithm for data simulated in the target tracking domain, the ITCP and MMEP scores [23] are computed. The original trajectories are compared against the solution and compute the errors.

1) Incorrect Trajectory Count Percentage (ITCP): An incorrect trajectory is defined as a trajectory with at least one assignment turning out to be incorrect. ITCP computes the number of incorrect trajectories as a percentage of the total number of trajectories.

2) Mismatch Error Percentage (MMEP): The MMEP computes the total number of mismatch errors (MME) in every trajectory as a percentage of all the assignments made. The number of total assignments in our system is $N \times (T-1)$ and MMEP can be computed as MMEP $= \frac{\sum \text{MME}}{N \times (T-1)} \times 100$.

Both these above measures are equally necessary for deciding the quality of the solution. While MMEP reports a score that is more focused on the assignment level, ITCP gives the essence of how good the overall trajectories are. A solution could have a high MMEP, but a low ITCP score and still be a good solution. There might be a scenario where two trajectories are close together, leading to multiple assignments being mismatched. This situation would result in a high MMEP score, as all the mismatched assignments are accounted for, but a low ITCP score as only two trajectories are incorrect overall. Another situation that might arise is where all the trajectories are close together, and at least one error in every trajectory reported, but not many errors overall, which implies a high ITCP and a low MMEP score.

B. Experimental Results

The ITCP and MMEP scores are reported for problems of different sizes. For every problem instance, the top five solutions (decided based on the duality gap) are generated. The ITCP and MMEP scores are computed for each of these solutions, and the best solution, the one with the lowest ITCP and the lowest MMEP scores, is chosen. For smaller problem sizes ($N \leq 10$), five problem instances are generated for each size. The average of these best scores is computed over the five instances, which is reported in Table I. For some of the tests, the 'best' score could not be decided. This was because we could not find a unique solution that had the lowest values for both ITCP and MMEP scores. That is while one solution had the lowest ITCP score, there was another solution with the lowest MMEP score. To handle this situation, we report a range (lowest and the highest values) of ITCP and MMEP scores. For example, if the ITCP range says [3, 5], the lowest ITCP score was 3 among all the solutions, and the highest is 5. The same applies for the range of MMEP scores as well. The solution times of both RMAP and Gurobi, have been compared and shown in Fig 2. Gurobi fails to scale for problem sizes greater 40 and dual ascent continues to perform well. Another advantage that the dual ascent-based RMAP-DA brings to the table is the monotonic increase in the lower bound. This avoids redundant oscillations in search of the lower bound when compared to other subgradient schemes, that might take longer time to converge.

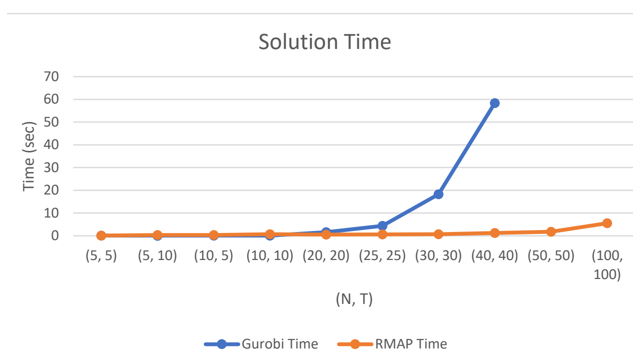

Fig. 2. Solution Time Comparison between RMAP and Gurobi

TABLE I
ITCP AND MMEP SCORES OF RMAP

N	T	ITCP	MMEP(%)	Solution Time (s)
5	5	0	0	0.103
5	10	1.75	8.89	0.1518
10	5	0.8	2	0.134
10	10	6.2	12.67	0.2818
20	20	17	9.74	0.525
25	25	16	6.17	0.59
30	30	29	10.68	0.726
40	40	30	5.38	1.255
50	50	[48, 50]	[6.45, 7.43]	1.821
100	100	[96, 98]	[4.68, 5.32]	5.546

C. Handling Missed Detections and False Alarms

A prominent issue in the MTT problems is handling spurious measurements from the RADARs and missing measurements of a target. We address these problems as 'False Alarms' and 'Missed Detections', respectively. In this paper, we simulate the scores in a way as to account for these situations and demonstrate the robustness of the algorithm in the presence of these abnormalities. The primary challenge with these issues is that the number of nodes (targets) in the graph per dimension is not the same anymore. So dummy variables need to be added to the graph, as required, to maintain the same number of nodes across all the time-scans or dimensions. The costs associated with these dummy variables are assigned to be a large number M.

To simulate the missed detection scenarios, a random detection from any of the time-scans is removed, with a probability threshold $p_m \in \{0.0, 0.1, 0.2, 0.3\}$. For the false alarm scenarios, a spurious detection in each frame is added with a probability threshold $p_f \in \{0.0, 0.1, 0.2, 0.3\}$. The spurious detections are created by adding a copy of an already existing detection, which are then randomly rotated and their coordinates are translated, as explained in [1]. For experimental purposes, we consider the missed detection and the false alarm scenarios in separate situations. The MMEP and the ITCP scores are reported for each of these probability thresholds in Tables IV and III. We observe that the solution time does not change in the presence of missed detections or false alarms.

D. Handling birth and death targets

Unexpected births and deaths of targets is another significant issue in the target tracking domain. When dealing with tracking multiple targets, it is possible that targets leave the field of view or new targets enter it. This might lead to two important scenarios: *Birth* and *Death* of targets. A *birth* of a target implies an appearance of measurements for a new target that did not exist previously. This node will not have any measurements until an arbitrary time frame p and then has sensor measurements from $p+1$ through T. The *death* of a target implies that there were measurements for a node until an arbitrary time frame p and after $p+1$, a sudden disappearance of these measurements through T is observed. To simulate the behavior of birth and death of targets in our system, we have considered a problem instance of size $(N, T) = (30, 30)$ and three targets for birth/death have been chosen at random. There could be three possible situations.

- **All Births**: All the three nodes are expected to appear at different time frames.
- **All Deaths**: All the three nodes are expected to disappear at different time frames.
- **Mixed Birth and Deaths**: Two of the nodes undergo a birth situation and the third node undergoes a death situation, at different time frames.

All the edges that involve these three nodes are given a cost of 0. These tests demonstrate the robustness of the algorithm in presence of births and deaths of targets. From the results shown in Table II, it can be seen that the scores are not drastically different from the base case of a problem size $(N, T) = (30, 30)$, from Table I, which shows that the algorithm, in the presence of births/deaths, is able to perform almost as good as the base case (Table I), and is thus robust. The MMEP seems to be only slightly higher.

TABLE II
SCORES WITH MISSED DETECTION PROBABILITY p_m

N	T	p_m	ITCP	MMEP (%)
5	5	0.1	0.8	4.00
		0.2	1.2	6.00
		0.3	1	6.00
5	10	0.1	4	16.00
		0.2	3.8	17.33
		0.3	3.4	15.11
10	5	0.1	0.8	2.00
		0.2	1	3.50
		0.3	2	5.50
10	10	0.1	6.6	17.11
		0.2	6.8	16.45
		0.3	7.6	19.12
20	20	0.1	17	9.74
		0.2	15	7.63
		0.3	16	10.79
25	25	0.1	19	9.00
		0.2	19	9.00
		0.3	20	9.33
30	30	0.1	29	10.68
		0.2	28	9.77
		0.3	29	10.35
40	40	0.1	[32, 34]	[6.67, 7.5]
		0.2	35	6.85
		0.3	37	7.88
50	50	0.1	[48,50]	[6.45,7.43]
		0.2	49	7.34
		0.3	48	8.16
100	100	0.1	97	5.10
		0.2	[97,100]	[5.97,6.61]
		0.3	97	5.80

E. Demonstrating the prominence of the triplet scores

We have claimed in Section III that the triplet scores capture the properties of non-linear trajectories better than the pairwise

TABLE III
SCORES WITH FALSE ALARM PROBABILITY p_f

N	T	p_f	ITCP	MMEP(%)
6	5	0.1	0	0
		0.2	0.8	4.17
		0.3	0.8	5.00
6	10	0.1	3.6	10.81
		0.2	3	9.25
		0.3	3.6	11.92
11	5	0.1	1	2.35
		0.2	1.4	4.63
		0.3	2.8	6.79
11	10	0.1	5.8	14.10
		0.2	8	17.37
		0.3	7.2	16.28
21	20	0.1	[12,13]	[7.01,7.26]
		0.2	[11,13]	[7.01,7.5]
		0.3	[13,14]	[9.02,11.53]
26	25	0.1	[19,21]	[9.46, 10.42]
		0.2	[18,20]	[7.69, 8.97]
		0.3	19	7.69
31	30	0.1	31	10.00
		0.2	31	10.00
		0.3	29	11.68
41	40	0.1	32	5.37
		0.2	31	5.56
		0.3	[33,35]	[6.66, 7.75]
51	50	0.1	[44, 45]	[5.52, 7.04]
		0.2	45	5.96
		0.3	47	6.52
101	100	0.1	[98,99]	[4.88,5.53]
		0.2	100	6.12
		0.3	101	10.1

TABLE IV
SCORES WITH BIRTH AND DEATH SCENARIOS FOR $(N, T) = (30, 30)$

Type of Birth/ Death	ITCP	MMEP (%)
All Births	28	12.99
All Deaths	28	12.99
Mixed Birth and Deaths	30	16.89

scores, and thus, provide better tracking outcomes. In this section we compare the RMAP-DA algorithm with triplet scores and only pairwise scores to verify this claim. For testing the algorithm with pairwise costs, we solve the problem as a group of adjacent time-wise Linear Assignment Problems (LAP) with pairwise scores as costs. As expected, the results, shown in Table V, validate our claim. The ITCP scores, when using only the pairwise costs are higher than the ITCP scores of the regular RMAP-DA approach (with triplet costs). The MMEP scores too are higher (almost double in some cases) when using the pairwise costs only. These results endorse the assertion behind using the triplet costs which help in capturing the properties of the trajectories from a "look-ahead" perspective; they provide more accurate tracking results than the "Markovian" pairwise costs only.

TABLE V
COMPARISON OF ITCP AND MMEP SCORES FOR TESTS WITH AND WITHOUT TRIPLET COSTS

N	T	Triplet Scores		Only Pairwise Scores	
		ITCP	MMEP (%)	ITCP	MMEP (%)
5	5	0	0	1.8	7.8
5	10	1.75	8.89	3.2	16.89
10	5	0.8	2	1.8	5
10	10	6.2	12.67	8.2	22.88
20	20	17	9.74	15	11.58
25	25	16	6.17	22	17.17
30	30	29	10.68	29	15.51
40	40	30	5.38	37	11.9
50	50	[48, 50]	[6.45, 7.43]	49	11.39
100	100	[96, 98]	[4.68, 5.32]	100	7.89

VII. CONCLUSIONS

In this paper, the multi-target tracking problem has been formulated as a multi-dimensional assignment problem and a robust and a scalable dual ascent-based algorithm (RMAP-DA) is developed to obtain near-optimal solutions. The MAP formulation allows the usage of the triplet costs, enabling us to induce some more global properties of the trajectory into the algorithm, via the cost coefficients. The quadratic-based formulation is linearized with the introduction of a new variable y. This linearized formulation is relaxed using Lagrangian multipliers and then the Linear Program is dualized. The problem is then decomposed into x and y sub problems and solved. The algorithm also guarantees a monotonic increase in the lower bound and does not exhibit an oscillatory behavior in search of the bound, thus saving time. For the computational experiments, tests cases of various sizes have been generated. These tests simulate the behavior of maneuverable targets in a target tracking environment. Evaluation metrics such as **ITCP** and **MMEP** scores are computed to verify the quality of the solution. For all the tests, the solution time, the MMEP and the ITCP scores have been reported. The algorithm takes very little computational time when compared to Gurobi. However, the robustness of the algorithm can only be verified in the presence of abnormalities or disruptions in the system. Some of the most prominent issues faced in the tracking domain are the presence of false alarms, missed detections and sudden appearance and disappearance of target measurements (birth and death situations). Data sets have been generated to simulate these conditions and the MMEP and ITCP scores have been reported for these tests. The algorithm did seem to perform well in such scenarios too. Moreover, leveraging the technological advancements of the modern day, we have developed an algorithm that is decomposable and we implement it on GPUs, using CUDA for thread level parallelism and thus, able to handle large-sized problems. Therefore, the contribution of this paper is mainly focused on employing a robust technique to MTT, and providing fast and highly accurate solutions, even in the presence of abnormalities. Some of the mainstream tracking algorithms such as PMHT, JPDAF [24], [25] consider probabilistic scores. In this paper, all the pairwise and triplet scores are deterministic and the the emphasis was placed on demonstrating the robustness of the algorithm. For future work, we would like to collaborate with researchers working on such tracking problems, so that probabilistic methods can be integrated with discrete combinatorial optimization methods.

Acknowledgments

The authors receive support from IBM-ILLINOIS Center for Cognitive Computing Systems Research (C3SR) – a research collaboration as part of the IBM AI Horizons Network. Rakesh Nagi was also partially supported with ONR award N000014-16-1-2245 under the program management of Dr. Warren Adams. These sources of support are gratefully appreciated.

References

[1] K. Date and R. Nagi, "Tracking multiple maneuvering targets using integer programming and spline interpolation," in *21st International Conference on Information Fusion*, 9 2018, pp. 1293–1300.

[2] ——, "GPU-accelerated hungarian algorithms for the linear assignment problem," *Parallel Computing*, vol. 57, pp. 52–72, 2016.

[3] ——, "Level 2 reformulation linearization technique–based parallel algorithms for solving large quadratic assignment problems on graphics processing unit clusters," *Informs Journal on Computing*, vol. 31, no. 4, pp. 633–845, 2019.

[4] R. Burkard, M. Dell'Amico, and S. Martello, *Assignment Problems: Revised Reprint*. Society for Industrial and Applied Mathematics, Jan. 2012. [Online]. Available: http://epubs.siam.org/doi/book/10.1137/1.9781611972238

[5] A. A. Butt and R. T. Collins, "Multiple target tracking using frame triplets," in *Computer Vision – ACCV 2012*, K. M. Lee, Y. Matsushita, J. M. Rehg, and Z. Hu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 163–176.

[6] R. T. Collins, "Multitarget data association with higher-order motion models," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, June 2012, pp. 1744–1751.

[7] C. Arora and A. Globerson, "Higher order matching for consistent multiple target tracking," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013, pp. 177–184.

[8] W. P. Adams, M. Guignard, P. M. Hahn, and W. L. Hightower, "A level-2 reformulation–linearization technique bound for the Quadratic Assignment Problem," *European Journal of Operational Research*, vol. 180, no. 3, pp. 983–996, 2007.

[9] C. J. Veenman, M. J. Reinders, and E. Backer, "Resolving motion correspondence for densely moving points," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 23, no. 1, pp. 54–72, 2001.

[10] Zne-Jung Lee, Shun-Feng Su, and Chou-Yuan Lee, "Efficiently solving general weapon-target assignment problem by genetic algorithms with greedy eugenics," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 33, no. 1, pp. 113–121, Feb 2003.

[11] S. Silven, "A neural approach to the assignment algorithm for multiple-target tracking," *IEEE Journal of Oceanic Engineering*, vol. 17, no. 4, pp. 326–332, Oct 1992.

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

[13] Y. Bar-Shalom and X.-R. Li, *Multitarget-multisensor tracking: Principles and techniques.* YBs London, UK:, 1995, vol. 19.

[14] T. Kirubarajan, H. Wang, Y. Bar-Shalom, and K. R. Pattipati, "Efficient multisensor fusion using multidimensional data association," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 2, pp. 386–400, April 2001.

[15] F.-X. Liu and Q.-H. Xing, "Optimized target assign method based on mixed genetic algorithms," vol. 22, 07 2002.

[16] W. Wang, S.-C. Cheng, and Y.-Z. Zhang, "Research on approach for a type of weapon target assignment problem solving by genetic algorithm," vol. 30, pp. 1708–1711, 09 2008.

[17] J.-X. Yu, S.-H. Wang, and W.-X. Cheng, "Target allocation decision making based on improved genetic algorithms with local search," vol. 30, pp. 1114–1117+1162, 06 2008.

[18] O. Reynen, S. Vadrevu, R. Nagi, and K. LeGrand, "Large-scale multi-dimensional assignment: Problem formulations and GPU accelerated solutions," in *22nd International Conference on Information Fusion*, July 2019, pp. 1–8.

[19] A. B. Poore, "Multidimensional assignment problems arising in multitarget and multisensor tracking," in *Nonlinear Assignment Problems*. Springer, 2000, pp. 13–38.

[20] A. M. Geoffrion, *Lagrangean relaxation for integer programming*. Berlin, Heidelberg: Springer Berlin Heidelberg, 1974, pp. 82–114. [Online]. Available: https://doi.org/10.1007/BFb0120690

[21] S. Vadrevu and R. Nagi, "A GPU accelerated algorithm for solving the multi-dimensional assignment problem for a multi-target tracking application," *submitted INFORMS Journal of Computing*, Dec 2019.

[22] L. Gurobi Optimization, "Gurobi optimizer reference manual," 2020. [Online]. Available: http://www.gurobi.com

[23] K. Bernardin and R. Stiefelhagen, "Evaluating multiple object tracking performance: the clear mot metrics," *Journal on Image and Video Processing*, vol. 2008, p. 1, 2008.

[24] A. Logothetis, V. Krishnamurthy, and J. Holst, "On maneuvering target tracking via the PMHT," in *Proceedings of the 36th IEEE Conference on Decision and Control*, vol. 5. San Diego, CA, USA: IEEE, 1997, pp. 5024–5029. [Online]. Available: http://ieeexplore.ieee.org/document/649857/

[25] C. Rago, P. Willett, and R. Streit, "A comparison of the JPDAF and PMHT tracking algorithms," in *1995 International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. Detroit, MI, USA: IEEE, 1995, pp. 3571–3574. [Online]. Available: http://ieeexplore.ieee.org/document/479758/