22nd International Conference on Information Fusion
Ottawa, Canada - July 2-5, 2019

Large-scale Multi-dimensional Assignment: Problem Formulations and GPU Accelerated Solutions

Olivia Reynen, Samhita Vadrevu, Rakesh Nagi
Department of Industrial and Enterprise Systems Engineering
University of Illinois at Urbana-Champaign
Urbana, Illinois, USA
{reynen2, samhita3, nagi}@illinois.edu

Keith LeGrand
Sandia National Laboratories
Albuquerque, NM, USA
kalegra@sandia.gov

***Abstract*—In this paper, we present alternate integer programming formulations for the multi-dimensional assignment problem, which is typically employed for multi-sensor fusion, multi-target tracking (MTT) or data association in general. The first formulation is the Axial Multidimensional Assignment Problem with Decomposable Costs (MDADC). The decomposable costs comes from the fact that there are only pairwise costs between stages or scans of a target tracking problem or corpuses of a data association context. The difficulty with this formulation is the large number of transitivity or triangularity constraints that ensure if entity A is associated to entity B and entity B is associated with entity C, then it must also be that entity A is associated to entity C. The second formulation uses both pairs and triplets of observations, which offer more accurate representation for kinematic tracking of targets. This formulation avoids the large number of transitivity constraints but significantly increases the number of variables due to triples. Solution to large-scale problems has alluded researchers and practitioners alike. We present solution methods based on Lagrangian Relaxation and massively parallel algorithms that are implemented on Graphics Processing Units (GPUs). We test the problem formulations and solution algorithms on MTT problems. The triples formulation tends to be more accurate for tracking measures and the MDADC solver can solve much larger problems in reasonable computational time.**

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

I. Introduction

The multi-dimensional assignment problem (MAP) is prevalent in a number of target tracking and data fusion problems. When multiple sensors make observations of multiple targets, we need to correctly associate those observations to the targets as well to each other before fusion can be performed. In this case the "dimensions" or stages correspond to each sensor. In a graph representation, each observation is referred to as a vertex or node, and an edge between a node in one stage to another node in a different stage signifies that they, in fact, refer to the same target. Typically, there are no edges needed between nodes of the same dimension because the assumption is that each sensor has made one (and only one) observation of each target and therefore those observations are of distinct targets and don't need to be associated to each other. This results in a K-partite graph structure with K dimensions. Figure 1 shows a visual representation of the MAP.

More generally, data association in a variety of domains can be formulated using the MAP. Entity resolution and fusion after natural language processing for multiple documents of a corpus can be formulated as a MAP. See, for example [1]. In this context, each document corresponds to a dimension or stage and entities (people, places, organizations, etc.) refer to the nodes in that stage. The MAP looks to associate the same entities across different documents before fusion can be performed. Fusion in this case allows us to enrich the attribute set of the entities. For example, one source or document might have the person's name and address while the other might have name and age. Fusing them will provide all attributes of that individual.

Further references of MAP can be found in multi-scan multi-target tracking (MTT) [2], [3] as well as multi-sensor MTT applications [4], [5]. In one simple approach to solving these assignment problems, multi-scan and multi-sensor updates can be treated as sequential single-scan and single-sensor updates, respectively, both in which the measurement assignment problems are 2D. This approach is sometimes referred to as the iterated corrector approach in multi-sensor literature [6]. When using this approach in multiple-hypothesis based tracking, hypotheses must be pruned at each stage of association. Consequently, insignificant hypotheses pruned at early stages of assignment may become significant at later stages but cannot be recovered. In the context of multi-sensor MTT, this means that the order in which sensors are processed can affect the final solution.

In multiple hypothesis frameworks, such as MHT [7] and generalized labeled multi-Bernoulli (GLMB) filtering [8],

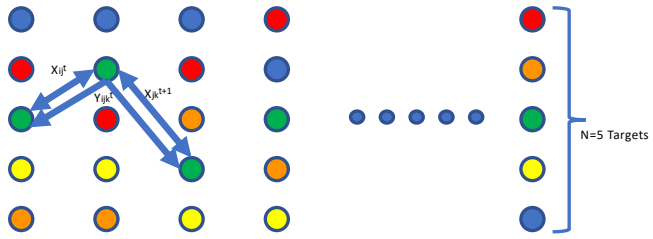

Fig. 1. Multi-Dimensional Assignment Problem Representation

978-0-9964527-8-6

there is great advantage in treating the collective multi-scan or multi-sensor assignment problem as an MAP. In treating the collection of assignment problems as a single MAP, hypothesis pruning is deferred, which results in a far less severe approximation of the optimal solution. When generating a collection of N-best solutions, there is a significant reduction in the number of infeasible solutions generated compared to the sequential 2D assignment approach.

What is common to these problems is that there are two or more stages or dimensions and there are multiple nodes in each stage that need to be associated to the other stage(s). For two dimensions, the problem can be formulated and solved efficiently using the Linear Assignment Problem (LAP). The LAP is also referred to as a weighted bipartite assignment problem. This problem can be solved in polynomial time. However, when the number of stages become 3 (or more) the problem is referred to as 3- (Multi-)Dimensional Matching (or Assignment/Association), which has been proven to be NP-hard. Solving large instances of these problems has been a challenge for researchers and practitioners. A recent attempt at solving large-scale multi-dimensional problems is [1].

Where the MAP differs is in the node to node assignment costs (or weights). In the Multi-dimensional Assignment Problem with Decomposable Costs (MDADC), node to node costs are provided for pairs of stages. That is, the edge between a node i in stage p to node j in stage q is provided a cost of C_{ij}^{pq}. This makes sense in the multi-sensor problem or the entity resolution problem because the scoring between a pair of observations from two dimensions does not depend on how they are associated to other dimensions. This may not be true in the MTT problem, where costs could be provided to trajectory segments for three or more adjacent dimensions. MAP formulations may also differ in the constraints that define a valid/feasible solution. In a typical MDADC formulation, if entity A is associated to entity B and entity B is associated with entity C, then it must also be that entity A is associated to entity C. This results in a large number of transitivity or triangularity constraints. If there are N nodes per dimension, and K dimensions in all, there are $O(N^3K^3)$ such constraints. Another way to ensure that the associations are consistent is to define an association variable for nodes between three – usually adjacent – stages simultaneously. This intrinsically avoids the need for a large number of transitivity constraints because of the "chaining" effect of these triple variables. By chaining we mean if node i in stage p, node j in stage $p+1$ and node k in stage $p+2$ are simultaneously associated to each other, and if node j in stage $p+1$, node k in stage $p+2$ and node l in stage $p+3$ are simultaneously associated to each other, then node i in stage p is implicitly associated to node l in stage $p+3$. The downside of this formulation is that the number of variables goes up from $O(N^2K^2)$ to $O(N^3K^2)$ [9].

As mentioned earlier, MAP is NP-hard, and thus, approximate forms must be used for all but the most trivial problems. Common approaches to this problem generally involve decomposing assignment costs such that the problem may be solved suboptimally as a series of pairwise assignment problems. In addition to pairwise assignments, the proposed method considers higher-order assignments through the use of triplet costs. These triplet costs can exploit unmodeled statistical relationships to improve assignment solution accuracy and, ultimately, tracking performance. For instance, whereas pairwise assignments may only capture first-order target dynamics, higher-order or nonparametric models could be leveraged through triplet costs to better track maneuvering objects. When evaluating the assignment of two sequential detections to an object, the agreement of those detections' metadata (such as color, SNR, shape, etc.), may be captured by a triplet cost without adding variables to the state space and explicitly modeling those metadata relationships in the observation likelihood function. As a result, fewer infeasible assignment solutions are produced, and assignments containing triplets with metadata agreement are promoted.

In this paper, our goal is to present alternative integer programming formulations of the MAP and compare them in terms of solution speed and accuracy. We also present the promise of massively-parallel Graphics Processing Unit (GPU) accelerated solution algorithms for these computationally demanding problems. We leverage the heuristic upper bounding procedures that follow along the iterations of the Lagrangian Relaxation procedure to provide alternative hypotheses for tracking problems.

The rest of the paper is organized as follows. In Section II, we review additional literature related to our problem. In Section III, we formally describe our mixed integer programming formulations for the MAP. In Section IV, we discuss our Lagrangian Relaxation approaches to solving MAP. In Section V we discuss the Lagrangian relaxation algorithms used in the GPU solvers. In Section VI we discuss the computational experiments. In Section VII, we conclude with a summary and directions of future research.

II. Related Literature

The MAP is very useful in modeling several scenarios including MTT or data association problems. There are different ways to formulate the MAP as well as different solving methods that can be used. The problem can be formulated as an MDADC as described in [10], [11], a minimum cost flow problem as shown in [12], or several others. An overview of these methods can be found in [2].

Lagrangian relaxation is commonly used as a way to solve NP-hard problems such as the MAP presented here. The mathematical optimization models of the MAP grow quickly in the size of the problem–both the number of targets and the number of frames–and given that it is NP-hard, off-the-shelf solvers are unable to handle problems of practical dimensions. Tauer and Nagi [1] developed a Lagrangian Relaxation approach to the MDADC and handled the large number of Lagrange multipliers on a Hadoop framework (Cloud computer). While successful in handling larger problems, the solution time for each multiplier update iteration on the Hadoop architecture is excessive, making it unusable in real-time situations. More

recently [13] has attempted to develop a version of this algorithm for a cluster of computers with Graphics Processing Units (GPUs). This is explained in Section V. They leverage the extremely fast GPU implementation of the LAP provided in [14].

III. Problem Formulation

A. MAP with Pairwise Decomposable Costs - MDADC

The axial MAP is a generalization of the LAP to $K \geq 3$ dimensions, and therefore, it is a minimum-cost K-partite matching problem. Specifically, let us assume that we have N vertices representing observations/reports from K (radar) scans, also referred to as partitions (or dimensions), and each pair of reports from two different partitions are connected by an edge forming a K-partite graph. Then the goal of the axial MAP is to identify N vertex disjoint subgraphs with K vertices each (one vertex per dimension), such that the sum of the costs of the edges chosen in the subgraphs is minimal. Different variants of the MAP can be conceived for different strategies of defining the subgraph costs in the objective function. Various mathematical formulations of MAP are thoroughly discussed by [15]–[19]. The MDADC formulation [20], [21] is one such variant in which a vertex from a partition is assumed to be connected to all the vertices from other partitions by edges that have a certain weight. The goal of MDADC is to construct N vertex disjoint "cliques" so as to minimize the sum of the weights of the edges included in those cliques. Cliques are nothing but subgraphs in which every vertex is connected to every other vertex by an edge. The "decomposability" aspect comes from the fact that a clique cost can be decomposed into the costs of individual edges included in that clique. Let us consider the following Mixed Integer Linear Programming (MILP) formulation for the MDADC. Note that the indices $i, j, k \in \{1, \cdots, N\}$, and indices $p, q, r \in \{1, \cdots, K\}$.

$$\text{MDADC: } \min \sum_{i} \sum_{j} \sum_{p} \sum_{q>p} C_{ij}^{pq} x_{ij}^{pq}; \tag{1}$$

$$\text{s.t. } \sum_{j} x_{ij}^{pq} = 1, \quad \forall (i,p,q) : p < q; \tag{2}$$

$$\sum_{i} x_{ij}^{pq} = 1, \quad \forall (j,p,q) : p < q; \tag{3}$$

$$\left.\begin{aligned} x_{ij}^{pq} + x_{jk}^{qr} - x_{ik}^{pr} - 1 \leq 0, \\ x_{jk}^{qr} + x_{ik}^{pr} - x_{ij}^{pq} - 1 \leq 0, \\ x_{ik}^{pr} + x_{ij}^{pq} - x_{jk}^{qr} - 1 \leq 0, \end{aligned}\right\}$$

$$\forall (i,j,k,p,q,r) : p < q < r; \tag{4}$$

$$x_{ij}^{pq} \in \{0,1\}, \quad \forall (i,j,p,q) : p < q. \tag{5}$$

The decision variable $x_{ij}^{pq} = 1$, if vertex i from partition p is assigned to vertex j from partition q, and 0 otherwise. Constraints (2) and (3) enforce that a vertex from one of the partitions should be assigned to exactly one vertex from any other partition. C_{ij}^{pq} is the cost of assigning vertex i from partition p to vertex j from partition q. In data association problems, this can also be seen as a measure of similarity between a pair of vertices that belong to two different partitions.

Recall that the overarching goal of any MAP formulation is to establish K vertex disjoint, connected substructures (cliques, stars, Hamiltonian paths, Hamiltonian cycles, etc.), from the given graph, containing N vertices each. In MDADC, cliques are established through the transitivity constraints (4), which enforce that if vertex i in partition p is connected (or associated) to vertex j from partition q ($x_{ij}^{pq} = 1$), and vertex j is assigned to vertex k from partition r ($x_{jk}^{qr} = 1$), then either: (1) Vertex i should be assigned to vertex k by setting $x_{ik}^{pr} = 1$ (if it is beneficial to the objective function), or (2) One or both of the other two assignments should be made zero.

MDADC is routinely used to model the Data Association problem, especially when the relationships of vertices within a particular graph are not important. Data association [22]–[25] is a fundamental problem in data science applications involving multiple data sources. Data gathered by these sources may be in different formats. The goal of data association is to merge these data into a cumulative evidence that can be used for sense-making tasks or to obtain information about the current state of the real world.

B. MAP with Triple Costs - Quadratic Linearized-MAP

We first introduce the following quadratic programming formulation for MAP/MTT (see [26] for further discussion): The indices $i, j, k \in \{1, \cdots, N\}$.

$$\min \sum_{t=1}^{T-2} \sum_{i} \sum_{j} \sum_{k} D_{ijk}^{t} x_{ij}^{t} x_{jk}^{t+1} \tag{6}$$

$$\text{s.t. } \sum_{j} x_{ij}^{t} = 1, \ \forall i, \ \forall t \in \{1, \cdots, T-1\}; \tag{7}$$

$$\sum_{i} x_{ij}^{t} = 1, \ \forall j, \ \forall t \in \{1, \cdots, T-1\}; \tag{8}$$

$$x_{ij}^{t} \in \{0,1\}, \ \forall i, j, \ \forall t \in \{1, \cdots, T-1\}. \tag{9}$$

The decision variable $x_{ij}^{t} = 1$ if vertex i from frame t is assigned to vertex j from frame $t+1$, and 0 otherwise. Constraints (7) and (8) enforce that, for any pair of consecutive frames, a vertex from one of the frames should be assigned to exactly one vertex from the other frame. The objective coefficient D_{ijk}^{t} represents a triplet score that is in effect only when $x_{ij}^{t} = 1$ and $x_{jk}^{t+1} = 1$, simultaneously.

The formulation described above is difficult to solve in its native form. One of the ways to solve this problem is to convert it into a MILP by introducing higher order variables and linearization. This process is called the *reformulation-linearization technique* (RLT) or *Quadratic-Linearization*, which has been well established for the quadratic assignment problem (QAP) [27]–[29]. Quadratic-Linearization can be applied to the MTT formulation as follows. We can replace T with K for the number of dimensions for sake of consistency with the earlier formulation.

1) First, we multiply constraint set (7) by variables $x_{hi}^{t-1}, \forall h, i, \forall t = \{2, \cdots, K\}$ and multiply constraint set (8) by variables $x_{jk}^{t+1}, \forall j, k, \forall t = \{1, \cdots, K-1\}$ to

create two new sets of (redundant) quadratic constraints. These quadratic constraints are added to the formulation.

2) Then, we define new continuous variables y^t_{ijk} for every product term $x^t_{ij} \cdot x^{t+1}_{jk}, \forall i,j,k,\forall t = \{1,\cdots,K-2\}$, to obtain the desired QL-MTT formulation as below.

$$\min \sum_{t=1}^{K-2} \sum_i \sum_j \sum_k D^t_{ijk} y^t_{ijk} \tag{10}$$

$$\text{s.t. } (7)-(9); \tag{11}$$

$$\sum_k y^t_{ijk} = x^t_{ij},\ \forall i,j,\forall t \in \{1,\cdots,K-2\}; \tag{12}$$

$$\sum_i y^t_{ijk} = x^{t+1}_{jk},\ \forall j,k,\forall t \in \{1,\cdots,K-2\}; \tag{13}$$

$$y^t_{ijk} \geq 0,\ \forall i,j,k\ \forall t \in \{1,\cdots,K-2\}. \tag{14}$$

The QL-MTT formulation has $O(TN^2)$ binary variables, $O(TN^3)$ continuous variables, and $O(TN^2)$ constraints, which makes it reasonably compact. Alternatively, MTT problem can also be formulated as a network flow problem [30], [31], which are subsumed by the QL-MTT formulation.

In more general data association problems where there are not only spatio-temporal trajectories as MTT, there is a need to associate all pairs and all triples (not just ones corresponding to adjacent stages). Such a formulation generalizes QL-MTT and is represented as the following QL-MAP.

$$\min \sum_i \sum_j \sum_p \sum_{q>p} C^{pq}_{ij} x^{pq}_{ij} + \sum_i \sum_j \sum_k \sum_p \sum_{q>p} \sum_{r\neq p\neq q} D^{rpq}_{kij} y^{rpq}_{kij} \tag{15}$$

$$\text{s.t. } \sum_i x^{pq}_{ij} = 1, \qquad \forall j,p,q>p; \tag{16}$$

$$\sum_j x^{pq}_{ij} = 1, \qquad \forall i,p,q>p; \tag{17}$$

$$\sum_k y^{rpq}_{kij} = x^{pq}_{ij}, \qquad \forall i,j,p,q>p, r\neq p\neq q; \tag{18}$$

$$y^{pqr}_{ijk} = y^{qpr}_{jik} = y^{rpq}_{kij} \qquad \forall i,j,k,p,q>p, r>q; \tag{19}$$

$$x^{pq}_{ij} \in \{0,1\}, \qquad \forall i,j,p,q>p; \tag{20}$$

$$y^{rpq}_{kij} \geq 0, \qquad \forall i,j,k,p,q>p, r\neq p\neq q. \tag{21}$$

Most of the constraints have been introduced earlier, and we note (19) as symmetry variables.

IV. Lagrangian Relaxation Solution Approach

In this section we present the Lagrangian Relaxations of the two MAP formulations presented in Section III.

A. LR for MDADC

Let us consider the MDADC formulation (1)-(5). The transitivity constraints (4) can be relaxed and added to the objective function using Lagrange multipliers $\theta^{(pq)r}_{(ij)k}$, corresponding to constraint $\phi^{(pq)r}_{(ij)k} = x^{pq}_{ij} + x^{qr}_{jk} - x^{pr}_{ik} - 1 \leq 0$. $\phi^{(pq)r}_{(ij)k} > 0$ signifies the amount by which the constraint is violated. Let $\boldsymbol{\theta} = \left\langle \theta^{(pq)r}_{(ij)k} \right\rangle$ and $\boldsymbol{\phi} = \left\langle \phi^{(pq)r}_{(ij)k} \right\rangle$, represent the corresponding vectors. Then, we obtain the following Lagrangian relaxation problem LR($\boldsymbol{\theta}$).

$$\min \sum_i \sum_j \sum_p \sum_{q>p} \left(\left(C^{pq}_{ij} + \Theta^{pq}_{ij} \right) x^{pq}_{ij} - \Omega^{pq}_{ij} \right);$$

$$\text{s.t. } (2),(3),(5). \tag{22}$$

Here,

$$\Theta^{pq}_{ij} = \sum_k \left(\sum_r \left(\theta^{(pq)r}_{(ij)k} \right) + \sum_{r<p,q} \left(\theta^{(rp)q}_{(ki)j} - \theta^{(rq)p}_{(kj)i} \right) + \sum_{r>p,r<q} \left(-\theta^{(pr)q}_{(ik)j} + \theta^{(rq)p}_{(kj)i} \right) + \sum_{r>p,q} \left(-\theta^{(qr)p}_{(jk)i} + \theta^{(pr)q}_{(ik)j} \right) \right). \tag{23}$$

$$\Omega^{pq}_{ij} = \sum_k \left(\sum_r \left(\theta^{(pq)r}_{(ij)k} \right) + \sum_{r<p,q} \left(\theta^{(rp)q}_{(ki)j} + \theta^{(rq)p}_{(kj)i} \right) + \sum_{r>p,r<q} \left(\theta^{(pr)q}_{(ik)j} + \theta^{(rq)p}_{(kj)i} \right) + \sum_{r>p,q} \left(\theta^{(qr)p}_{(jk)i} + \theta^{(pr)q}_{(ik)j} \right) \right). \tag{24}$$

Let $\nu(\cdot)$ represent the objective function of a problem. Then for any $\boldsymbol{\theta} \geq \mathbf{0}$, $\nu(\text{LR}(\boldsymbol{\theta}))$ provides a lower bound on $\nu(\text{MDADC})$, i.e., $\nu(\text{LR}(\boldsymbol{\theta})) \leq \nu(\text{MDADC})$. To find the best possible lower bound, we need to solve the Lagrangian dual problem LD($\boldsymbol{\theta}$): $\max_{\boldsymbol{\theta}\geq 0} \nu(\text{LR}(\boldsymbol{\theta}))$. Hence, the primary goal is to systematically search for the Lagrange multipliers which maximize the objective function value of the Lagrangian dual problem. The Lagrangian dual problem LD($\boldsymbol{\theta}$) is solved using a traditional subgradient scheme. Note that for a given $\boldsymbol{\theta}$, LR($\boldsymbol{\theta}$) is simply a collection of LAPs, which are readily solved using the GPU accelerated approach of [14]. To summarize in brief, the multiplier adjustment scheme and the LAPs are solved iteratively until convergence of some kind is achieved: primal feasibility, desired gap with the best upper bound, or number of iterations.

B. LR for QL-MAP

The Lagrangian relaxation of the QL-MAP formulation is (LRQL($\hat{\boldsymbol{u}}$)).

$$\min \sum_{i,j} \sum_{p,q>p} C^{pq}_{ij} x^{pq}_{ij} + \sum_{i,j,k} \sum_{p,q>p,r\neq p\neq q} (D^{pqr}_{ijk} - \hat{u}^{pqr}_{ijk}) y^{pqr}_{ijk} \tag{25}$$

$$\text{s.t. } \sum_i x^{pq}_{ij} = 1, \qquad \forall j,p,q>p; \tag{26}$$

$$\sum_j x^{pq}_{ij} = 1, \qquad \forall i,p,q>p; \tag{27}$$

$$\sum_k y^{rpq}_{kij} = x^{pq}_{ij}, \quad \forall i,j,p,q>p, r\neq p\neq q; \tag{28}$$

$$x^{pq}_{ij}, y^{pqr}_{ijk} \geq 0 \qquad \forall i,j,k,p,q>p, r\neq p\neq q \tag{29}$$

The Lagrangian relaxation gives a valid lower bound. Thus, it can be established that,

$$\nu(\text{LRQL}) \leq \nu(\text{QL-MAP}) \tag{30}$$

V. Lagrangian Relaxation Algorithms

A. LR Algorithm for MDADC

It is not difficult to recognize that $\text{LR}(\boldsymbol{\theta})$ can be decomposed into pairwise sub-problems $\text{SP}_{pq}(\boldsymbol{\theta})$ $(p < q)$ that are LAPs and solvable in polynomial time. The objective function of these sub-problems is:

$$\text{SP}_{pq}(\boldsymbol{\theta}) : \min \sum_i \sum_j \left(C_{ij}^{pq} + \Theta_{ij}^{pq}\right) x_{ij}^{pq}. \tag{31}$$

Subsequently, we employ a standard surrogate sub-gradient approach to find the best values of the multipliers $\boldsymbol{\Theta}$ that result in the tightest lower bound possible. A summary of the LR algorithm is as follows:

Algorithm 1: Lagrangian subgradient optimization.

1) Initialize $k = 0, \boldsymbol{\theta}^k = \mathbf{0}, \text{LB} = -\infty$, and $\text{GAP} = \infty$.
2) Initialize step-size λ^k and multiplier $0 < \alpha \leq 1$.
3) LAP solution:
   a) Solve $\binom{K}{2}$ LAPs of size $N \times N$ and obtain $\nu(\text{SP}_{pq}(\boldsymbol{\theta}^k)) \forall p, q : p < q$.
   b) $\nu(\text{LR}(\boldsymbol{\theta}^k)) = \left(\sum_p \sum_{q>p} \nu(\text{SP}_{pq}(\boldsymbol{\theta}^k))\right)$.
   c) If $\text{LB} < \nu(\text{LR}(\boldsymbol{\theta}^k))$, update $\text{LB} = \nu(\text{LR}(\boldsymbol{\theta}^k))$.
   d) Update $\text{GAP} = \frac{\text{UB}-\text{LB}}{\text{LB}}$. Stop if $\text{GAP} < \text{MIN\_GAP}$.
4) Perform feasibility check.
   a) Stop if primal feasible. We have the optimum!
5) Lagrangian dual update:
   a) Update dual multipliers $\boldsymbol{\theta}^{k+1} \leftarrow \max\{0, \boldsymbol{\theta}^k + \lambda \cdot \boldsymbol{\phi}^k\}$.
   b) Update step-size $\lambda^{k+1} \leftarrow \alpha \cdot \lambda^k$.
   c) Update cost coefficients C_{ij}^{pq} using $\boldsymbol{\theta}^{k+1}$.
6) Update $k \leftarrow k + 1$. Stop if $k >$ ITN_LIM. Else, return to Step 3.

B. LR Algorithm for QL-MAP

The LR algorithm for the QL-MAP is similar to Algorithm 1, except we solve it using a Dual Ascent framework. The advantage of the Dual Ascent scheme for multiplier update is that it results in monotonically increasing lower bound values (opposed to oscillations in the subgradient adjustment procedure). The details of this are procedure are omitted for sake of brevity.

VI. Computational Experiments

The purpose of the following computational experiments is to compare the aforementioned formulations for the MAP to solve an MTT problem. In this experiment, a graph-based approach to MTT is considered, which prioritizes data association over state estimation. Problems with 20 targets and 20 frames or less can be solved using Gurobi [32] for the QL-MTT formulation. The same costs and tracking data generated for these problems are used to test the Lagrangian relaxation method for the MDADC formulation as well as the Lagrangian relaxation for the QL-MAP formulation. The tracking data and corresponding cost values are generated using both cubic spline interpolation and linear spline interpolation given the positions of the trajectories. This cost generation method is described in detail in Section IV of [9]. As shown in this paper, generating costs using clamped spline interpolation is more costly but will typically lead to a more accurate solution. Figure 2 shows an example of some trajectories that are generated using this method.

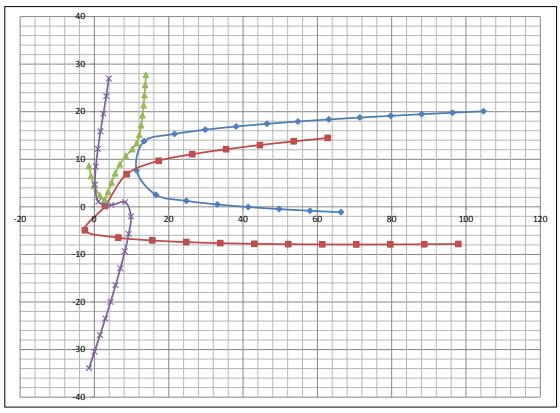

Fig. 2. Example trajectories [9]

For each $(N, K) \in \{(5, 10), (10, 5), (10, 10), (10, 20), (20, 10)\}$, 10 problem instances are generated. These same 10 problem instances are solved using Gurobi and both Lagrangian relaxation methods. Problems larger than 20 targets and 20 frames cannot be solved in reasonable time or results in memory failure when using a commercial solver such as Gurobi with any formulation. This is why alternative solution methods such as Lagrangian relaxation are necessary. Larger problems of size $(N, T) \in \{(30, 30), (50, 50), (100, 100)\}$ are solved using the MDADC algorithm to show that it is capable of solving larger problems that Gurobi is unable to solve.

The MDADC algorithm is terminated after a feasible solution is found or after 500 iterations, whichever comes first. Due to the way the algorithm is terminated by the iteration limit before finding a feasible solution during some instances, it is important to note how the final solution that is analyzed for the test results is obtained. In this case, the pairwise assignments from the current solution between consecutive time steps, t and $t + 1$, is used. The possible disagreements between these pairwise assignments and ones that go between time step t and $t+2$ are ignored. More sophisticated methods of projecting the final solution into the feasible region could be studied in the future. Similarly, the QL-MAP algorithm is also limited to 100 iterations. For the larger problems, the MDADC algorithm was limited to 200 iterations for the sake of time. It was shown that the algorithm is able to reach a good solution in this amount of iterations, so 500 iterations was not necessary.

A. Evaluation Criteria

To measure the accuracy of the solutions obtained by all three formulations, the total mismatch error percentage (MMEP) and incorrect trajectory count percentage (ITCP) metrics are employed [33]. Both measures compare the solution to the original trajectories and count the errors. The MMEP counts the total number of mismatch errors (MME) in all trajectories as a percentage of all assignments made. In this problem there are $N \times (T-1)$ total assignments, so $\text{MMEP} = \frac{\sum \text{MME}}{N \times (T-1)} \times 100$. The ITCP counts the number of incorrect trajectories as a percentage of the total number of trajectories. An incorrect trajectory is a trajectory with at least one incorrect assignment. In this problem there are N total trajectories. Both measures are needed to show the overall accuracy of the problem because it is possible to have a high MMEP but a low ITCP and, therefore, not a bad solution. If there are two trajectories that are very close together, multiple assignments might be mismatched between them along the way. This would lead to high MMEP since this measure counts every mismatched assignment, but low ITCP since this measure will only count the two incorrect trajectories. It is also possible to have a high ITCP, but low MMEP if the trajectories are all close together. If this happens there could be at least one error in every trajectory leading to a high ITCP, but not many errors overall leading to a low MMEP.

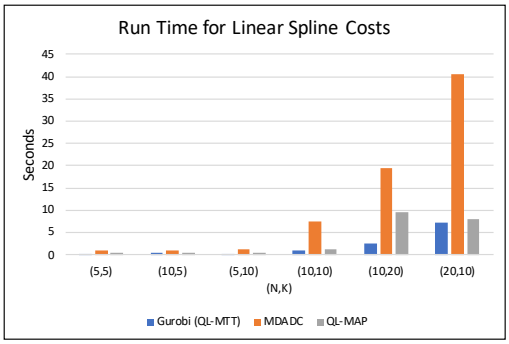

Fig. 3. Run Time for Linear Spline Costs

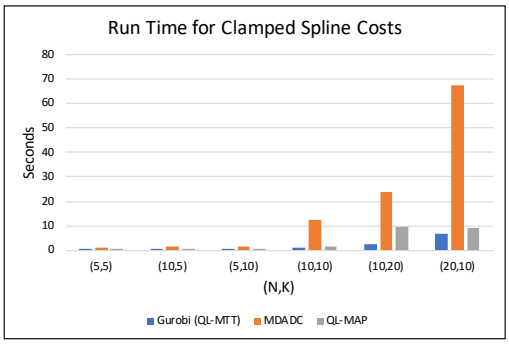

Fig. 4. Run Time for Clamped Spline Costs

B. Experimental Results

The first set of tests considers smaller problems that are solvable using all three solution methods. The average total run time for each problem size is recorded and shown in Figures 3 and 4. The run times for the MDADC and QL-MAP programs are longer than Gurobi for certain tests, which can be explained by the iteration limit. These longer run times indicate problem instances where the algorithm is unable to find a feasible solution and reaches the iteration limit. The iteration limit could possibly be lowered to decrease run time without sacrificing solution accuracy. A more sophisticated approach could involve utilizing the optimality gap as a second termination criteria for the algorithm. In other words, the algorithm would terminate if it reached the iteration limit or reached a small enough optimality gap. However, the potential savings in total iterations must be traded with the additional cost of computing the upper bound used in the optimality gap calculation, the consideration of which is beyond the scope of this work. Furthermore, the Lagrangian-based GPU algorithms are really not intended for small problems that a standard solver like Gurobi can solve. Their true benefit will surface for large-dimension problems.

The average MMEP and ITCP measures for each problem size can be found in the Figures 5, 6, 7, and 8. As mentioned previously, a low ITCP and MMEP value is ideal. All three formulations achieve relatively low scores for all of these problem instances, which means that they are all able to find a solution relatively close to the true trajectories. Note that, because of noisy nature of measurement data, the optimal assignment solution will not necessarily guarantee an optimal tracking solution. This fact is reinforced by the presence of any nonzero ITCP or MMEP values associated with the Gurobi solutions, which are guaranteed to be optimal with respect to assignment cost. One can see that this discrepancy occurs less when working with costs generated by the clamped spline method than with the linear spline method, which is to be expected based on the findings of [9]. The clamped spline costs provide a better representation of the true trajectories. Higher scores in the two Lagrangian Relaxation methods will come from both not solving to optimality as well as from cost coefficient inaccuracy issues while the high Gurobi scores are only from misrepresentation by the cost method used. The QL-MAP formulation achieves significantly better scores than the MDADC formulation for all problem sizes, which suggests that it is a better formulation for the MTT problem. Another interesting result that can be seen in Figure 5 is that the QL-MAP formulation actually did better than Gurobi for the $(10, 10)$ problem size. Although Gurobi is always able to solve to optimality, the QL-MTT formulation that it models only uses triplet costs. The QL-MAP formulation uses both triplet and pairwise costs so it has a better representation of the true trajectories, which is why its solution could be closer to the true tracking data.

As mentioned before, the advantage to the MDADC algorithm is that it can solve much larger problems than Gurobi. In these cases, an approximate solution found with Lagrangian relaxation will be used. Five problems each of sizes $(N, K) \in \{(30, 30), (50, 50), (100, 100)\}$ were tested with the MDADC solver. The QL-MAP solver is a current work in progress and is only able to solve the smaller problems so far. The average

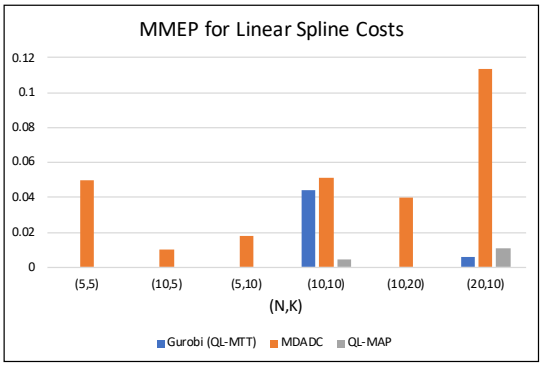

Fig. 5. MMEP for Linear Spline Costs

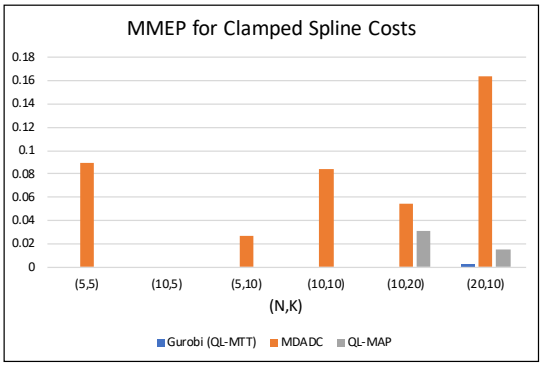

Fig. 6. MMEP for Clamped Spline Costs

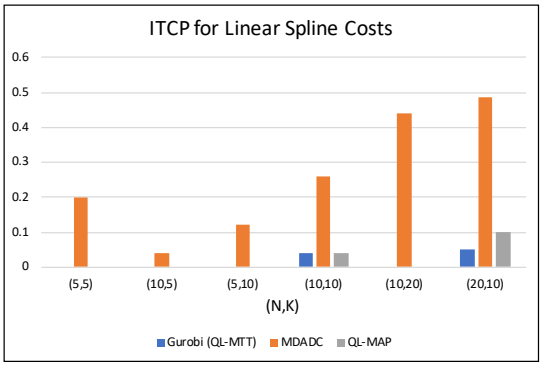

Fig. 7. ITCP for Linear Spline Costs

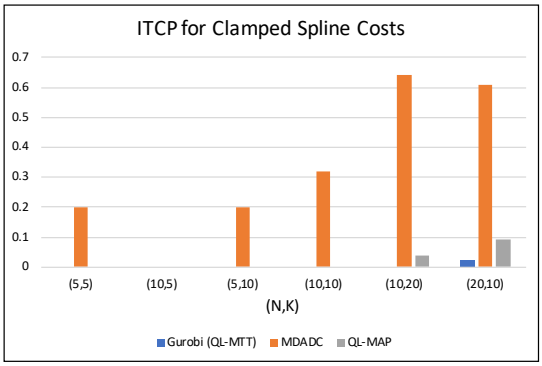

Fig. 8. ITCP for Clamped Spline Costs

run time, ITCP, MMEP and optimality gap values can be found in Table I. The ITCP and MMEP measures capture how close the solutions are to the true tracking data, but this can be influenced by the accuracy of the cost methods. Therefore, the optimality gap is provided here to show how close the solver is to finding the optimal solution given the costs. A small enough optimality gap indicates that this solver is an acceptable alternatives to a solver like Gurobi, whereas the ITCP and MMEP scores are more representative of how well the costs and formulations represent the true tracking problem.

TABLE I
MDADC TEST RESULTS

(N,K)	Run Time	MMEP	ITCP	Optimality Gap
(30,30)	209.4	10.92%	90.00%	1.20%
(50,50)	1012	8.96%	90.00%	0.636%
(100,100)	18924	7.12%	100.0%	0.222%

Table I shows that the MDADC tends to have a high ITCP score for problems of larger size, but relatively low MMEP. This indicates that although few errors were made in the assignment solutions, there was at least one error in almost every one of the trajectories. When there is a large number of targets in the space you are observing it is expected that many of the trajectories would be very close together, which would increase the likelihood of one target getting mistaken for another. The high ITCP scores observed during these tests show that almost every target is confused with another at least once in the solution. However, the low MMEP score shows that there are not many mistakes made overall. The MDADC solver is able to achieve an optimality gap of 1.2% or less for all tests, which suggests that it is a very good replacement for Gurobi for these larger problems.

VII. CONCLUSIONS

In this paper, we presented three different formulations, QL-MTT, QL-MAP and MDADC, to solve the multi-dimensional assignment problem. We tested three different solvers. One used Gurobi for the QL-MTT formulation, which is able to solve the problem to optimality, but only works with smaller problems. The other two solvers use Lagrangian relaxation algorithms implemented on GPUs to solve larger problems.

We performed several tests on all three solvers to compare their solutions and performance metrics. It was shown that both Lagrangian relaxation methods are acceptable alternatives to an exact solver as they were able to get very close to the optimal solution for most problems. It was also shown that the MDADC algorithm is able to solve much larger problems than an exact solver would be able to do while still maintaining low error percentages when compared to the true trajectories. The QL-MAP performed better than the MDADC formulation for the problem sizes that it is currently able to handle.

This paper was focused mostly on the abilities of the proposed solvers when applied to the tracking problem, however, there is future work to be done to make the problem more realistic. Missed detections and false detections happen very frequently in real tracking problems. Future work will incorporate these into the model while maintaining good solution

quality. Introducing these other factors greatly increases the difficulty of the problem, so it is important to start with an accurate and efficient solver before introducing them. These enhancements would allow for a more realistic comparison between the proposed algorithms and traditional state-of-the-art tracking methods such as MHT.

ACKNOWLEDGMENTS

This work was partially supported by the Laboratory Directed Research and Development program at Sandia National Laboratories, a multi-mission laboratory managed and operated by National Technology and Engineering Solutions of Sandia, LLC, a wholly owned subsidiary of Honeywell International, Inc., for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-NA0003525. 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] G. Tauer and R. Nagi, "A map-reduce lagrangian heuristic for multi-dimensional assignment problems with decomposable costs," *Parallel Computing*, vol. 39, no. 11, pp. 653–668, 2013.

[2] C. Chong, S. Mori, and D. B. Reid, "Forty years of multiple hypothesis tracking - a review of key developments," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 452–459.

[3] B.-T. Vo and B.-N. Vo, "Multi-scan generalized labeled multi-bernoulli filter," in *21st International Conference on Information Fusion (FUSION)*, 2018, pp. 195–202.

[4] Y. Bar-Shalom, P. K. Willett, and X. Tian, *Tracking and Data Fusion*. YBS publishing, 2011.

[5] R. P. Mahler, *Advances in Statistical Multisource-Multitarget Information Fusion*. Artech House, 2014.

[6] F. Papi, "Multi-sensor δ-GLMB filter for multi-target tracking using Doppler only measurements," in *2015 European Intelligence and Security Informatics Conference*, 2015, pp. 83–89. [Online]. Available: http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=7379727

[7] S. Blackman and R. Popoli, *Design and Analysis of Modern Tracking Systems*. Artech House, 1999.

[8] B.-T. Vo and B.-N. Vo, "Labeled random finite sets and multi-object conjugate priors," *IEEE Transactions on Signal Processing*, vol. 61, no. 13, pp. 3460–3475, 2013.

[9] 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.

[10] 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.

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

[12] G. Battistelli, L. Chisci, F. Papi, A. Benavoli, A. Farina, and A. Graziano, "Optimal flow models for multiscan data association," *IEEE Trans. on Aerospace and Electronic Systems*, vol. 47, no. 4, pp. 2405–2422, 2011.

[13] S. Natu, K. Date, and R. Nagi, "GPU-accelerated lagrangian heuristic for multidimensional assignment problems with decomposable costs," *Parallel Computing*, vol. submitted, 2019.

[14] K. Date and R. Nagi, "GPU-accelerated hungarian algorithms for the linear assignment problem," *Parallel Computing*, vol. 57, pp. 52–72, 2016.

[15] 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.

[16] A. Poore, "Multidimensional assignment formulation of data association problems arising from multitarget and multisensor tracking," *Computational Optimization and Applications*, vol. 3, no. 1, pp. 27–57, 1994. [Online]. Available: http://dx.doi.org/10.1007/BF01299390

[17] A. Poore and A. J. Robertson, "A new lagrangian relaxation based algorithm for a class of multidimensional assignment problems," *Computational Optimization and Applications*, vol. 8, no. 2, pp. 129–150, 1997.

[18] H. J. Bandelt, A. Maas, and F. C. R. Spieksma, "Local search heuristics for multi-index assignment problems with decomposable costs," *The Journal of the Operational Research Society*, vol. 55, no. 7, pp. 694–704, 2004. [Online]. Available: http://www.jstor.org/stable/4102016

[19] J. L. Walteros, C. Vogiatzis, E. L. Pasiliao, and P. M. Pardalos, "Integer programming models for the multidimensional assignment problem with star costs," *European Journal of Operational Research*, vol. 235, no. 3, pp. 553–568, 2014.

[20] H.-J. Bandelt, Y. Crama, and F. C. Spieksma, "Approximation algorithms for multi-dimensional assignment problems with decomposable costs," *Discrete Applied Mathematics*, vol. 49, no. 1-3, pp. 25–50, 1994.

[21] Y. Kuroki and T. Matsui, "An approximation algorithm for multidimensional assignment problems minimizing the sum of squared errors," *Discrete Applied Mathematics*, vol. 157, no. 9, pp. 2124–2135, 2009.

[22] G. Tauer, R. Nagi, and M. Sudit, "The graph association problem: mathematical models and a lagrangian heuristic," *Naval Research Logistics (NRL)*, vol. 60, no. 3, pp. 251–268, 2013.

[23] K. Date, G. A. Gross, S. Khopkar, R. Nagi, and K. Sambhoos, "Data association and graph analytical processing of hard and soft intelligence data," in *16th International Conference on Information Fusion*, 2013, pp. 404–411.

[24] K. Date, G. A. Gross, and R. Nagi, "Test and evaluation of data association algorithms in hard+soft data fusion," in *17th International Conference on Information Fusion*, 2014, pp. 1–8.

[25] G. A. Gross, K. Date, D. R. Schlegel, J. J. Corso, J. Llinas, R. Nagi, and S. C. Shapiro, "Systemic test and evaluation of a hard+ soft information fusion framework: Challenges and current approaches," in *17th International Conference on Information Fusion*, 2014, pp. 1–8.

[26] 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.

[27] W. P. Adams and H. D. Sherali, "Linearization strategies for a class of zero-one mixed integer programming problems," *Operations Research*, vol. 38, no. 2, pp. 217–226, 1990.

[28] W. P. Adams and T. A. Johnson, "Improved linear programming-based lower bounds for the Quadratic Assignment Problem," *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 16, pp. 43–77, 1994.

[29] 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.

[30] A. Andriyenko and K. Schindler, "Globally optimal multi-target tracking on a hexagonal lattice," in *Computer Vision – ECCV 2010*, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 466–479.

[31] A. A. Butt and R. T. Collins, "Multi-target tracking by lagrangian relaxation to min-cost network flow," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1846–1853.

[32] I. Gurobi Optimization, "Gurobi optimizer reference manual," http://www.gurobi.com, 2016.

[33] 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.