

A Mixed-Integer Conic Program for the Multi-Agent Moving-Target Traveling Salesman Problem

Allen George Philip¹, Zhongqiang Ren², Sivakumar Rathinam³ and Howie Choset²

Abstract—The Moving-Target Traveling Salesman Problem (MT-TSP) seeks a shortest path for an agent that starts at a stationary depot, visits a set of moving targets exactly once, each within one of their respective time windows, and returns to the depot. In this paper, we introduce a new Mixed-Integer Conic Program (MICP) formulation for the Multi-Agent Moving-Target Traveling Salesman Problem (MA-MT-TSP), a generalization of the MT-TSP involving multiple agents. Our approach begins by restating the current state-of-the-art MICP formulation for MA-MT-TSP as a Nonconvex Mixed-Integer Nonlinear Program (MINLP), followed by a novel reformulation into a new MICP. We present computational results demonstrating that our formulation outperforms the state-of-the-art, achieving up to two orders of magnitude reduction in runtime, and over 90% improvement in optimality gap.

I. INTRODUCTION

Given a set of fixed target locations (targets) and the traversal costs between all target pairs, the Traveling Salesman Problem (TSP) aims to find a tour of minimum cost for an agent, that visits all the targets exactly once. The TSP is a classical problem in combinatorial optimization, with several applications including unmanned vehicle planning [1]–[4], transportation and delivery [5], monitoring and surveillance [6], [7], disaster management [8], precision agriculture [9], and search and rescue [10], [11]. The Moving-Target Traveling Salesman Problem (MT-TSP) is a generalization of the TSP where the targets traverse some predefined paths. The targets may also have associated time windows only during which an agent can visit them. Different variants of the MT-TSP were introduced in the literature, motivated by practical applications such as defending an area from oncoming hostile rockets or Unmanned Aerial Vehicles [12]–[14], monitoring and surveillance [15]–[18], resupply missions with moving targets [12], dynamic target tracking [19], and industrial robot planning [20].

It is generally assumed that the maximum speed of the agent is greater than the speed of the fastest target [12]. When the speeds of all the targets reduces to 0, the MT-TSP simplifies to the TSP. Hence, MT-TSP is NP-Hard. In the literature, we find several heuristic based approaches for the MT-TSP [15]–[17], [19], [21]–[25] that finds feasible solutions, but gives no information on how far they are from

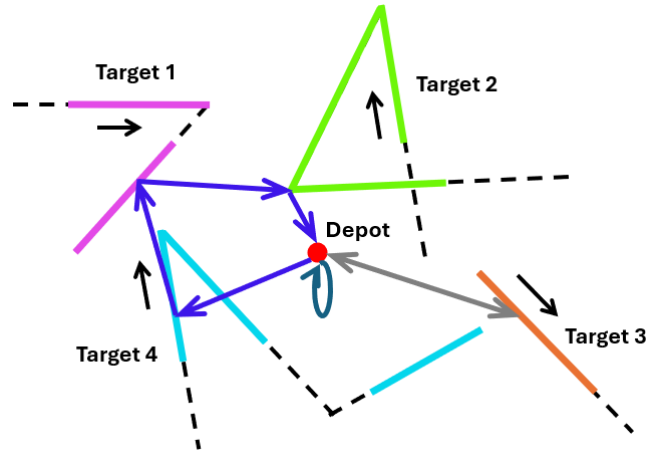


Fig. 1. A feasible solution for an example instance of the MA-MT-TSP. The solid, colored portions of the target trajectories correspond to their time windows. The agents begin and end their tour at the depot. Note how one of the three agents is not assigned any targets in this solution, and simply waits at the depot.

the optimum. The literature also presents exact and approximation algorithms for some MT-TSP variants. However, they only apply to very restricted cases where the targets are assumed to move in the same direction with the same speed [20], [26], move along the same line [12], [27], or move along lines passing through the depot [12]. In a recent paper [28], we presented an algorithm capable of handling targets moving along arbitrary curves, offering tight lower bounds for the MT-TSP. Also, in [29], we introduced a new mixed-integer conic program (MICP) that finds the optimum for the MT-TSP, for the case where each target moves along its own line, and have one associated time window. Finally, a complete algorithm for MT-TSP with stationary obstacles was also introduced recently in [30].

The objective of this paper is to find exact solutions for the Multi-Agent Moving-Target Traveling Salesman Problem (MA-MT-TSP), a generalization of the MT-TSP with multiple agents. Currently, the only exact solver for this problem is the MICP introduced in [14]. Hence, we take this as our baseline. Here, each target is assumed to move along its own line, and have one associated time window. However, in this paper, we consider a less restricted case where each target traverse a path made of piecewise-linear segments, and have more than one associated time window (Fig. 1). Therefore, we first extend the baseline MICP to accommodate these changes. Then, we pose this extended formulation as a nonconvex mixed-integer nonlinear program

¹ Mechanical Engineering, Texas A&M University, College Station, TX 77843-3123. Email: y262u297@tamu.edu

² Robotics Institute, Carnegie Mellon University, 5000 Forbes Ave., Pittsburgh, PA 15213, USA.

³ Mechanical Engineering, and Computer Science and Engineering, Texas A&M University, College Station, TX 77843-3123.

(nonconvex MINLP), which we then finally reformulate as a new MICP, similar to the works in [29], and [31].

We prove that our approach finds the optimum for the MA-MT-TSP, and present computational results to corroborate the performance of our formulation. We observe that our MICP formulation significantly outperforms the baseline, and scales much better with increasing number of targets and agents, and larger time window durations, achieving up to a two-order-of-magnitude reduction in runtime, and up to over 90% tighter optimality gap.

II. PROBLEM DEFINITION

All the targets and agents move on a 2D Euclidean plane (\mathbb{R}^2). All the agents share the same maximum speed v_{max} , and the same depot s . Without loss of generality, we define a copy of the depot, denoted by s' , and require the agents to return to s' after completing their respective tours. Let $V_{tar} := \{1, 2, \dots, n\}$ denote the set of n moving targets, and $V_{agt} := \{1, 2, \dots, m\}$ denote the set of m agents. For a target $u \in V_{tar}$, the portion of u 's trajectory within any of its time windows consists of linear trajectory segments. The set of all such segments¹ across all of u 's time windows is denoted by C_u . The set $V_{seg} := \bigcup_{u \in V_{tar}} C_u$ consists of all the linear trajectory segments corresponding to all targets. For a segment $i \in V_{seg}$, $\tau_i \in V_{tar}$ denotes its associated target. Also, \underline{t}_i and \bar{t}_i represent the times, and \underline{p}_i and \bar{p}_i represent the positions of the segment's start and end points. Finally, v_i denotes the velocity of τ_i along the segment. Let $V := V_{seg} \cup \{s, s'\}$. For s and s' , we define velocities v_s , $v_{s'}$, and set them to $(0, 0)$ since they are stationary. Also, we define time windows $[\underline{t}_s, \bar{t}_s]$ and $[\underline{t}_{s'}, \bar{t}_{s'}]$ for s and s' respectively. We set $\underline{t}_s = \bar{t}_s = 0$ since the agents depart from s at time 0. We also set $\underline{t}_{s'} = 0$ and $\bar{t}_{s'} = T$ where T is the time horizon over which the trajectories of the targets are defined. For some $i \in \{s, s'\}$, \underline{p}_i and \bar{p}_i simply refers to the fixed depot location. An agent visits a moving target (say u) if the position of the agent coincides with the position of u at some time t within time interval $[\underline{t}_i, \bar{t}_i]$ corresponding to some segment $i \in C_u$. The objective of the MA-MT-TSP is to find tours for the agents such that each target is visited once by one of the agents, and the sum of the distances traversed by the agents is minimized.

III. BASELINE MICP FOR MA-MT-TSP

This section presents the current state-of-the-art MICP for the MA-MT-TSP introduced in [14]. We extend this formulation to accommodate piecewise-linear trajectories, and more than one time window for each target. Before proceeding further, we define a directed graph (V, E) , with all the nodes in C_u for each $u \in V_{tar}$ forming a cluster. The edges in E are added as follows: From s to all nodes in V_{seg} , from each node belonging to a cluster to all the other nodes belonging to different clusters, and finally, from all the nodes in V_{seg} , to s' . For any node $i \in V$, E_i^{in} and E_i^{out} denotes the set of all edges entering and exiting i .

We now define the decision variables for the MICP. For each node $i \in V_{seg}$, the variable $t_{i,k} \in \mathbb{R}_{\geq 0}$ denotes the time at which an agent $k \in V_{agt}$ visits target τ_i at node i . The position of τ_i at time $t_{i,k}$ is denoted by $p_{i,k} \in \mathbb{R}^2$. Variables $t_{i,k}$ are also defined for $i \in \{s, s'\}$, denoting the time at which agent k departs or arrives at the depot. Here, $p_{i,k}$ denotes the fixed depot position. For each edge $e = (i, j) \in E$, the binary variable $y_{e,k} \in \{0, 1\}$ denotes the choice of whether agent k traverses the edge. The auxiliary variable $l_{e,k} \in \mathbb{R}_{\geq 0}$ denotes the cost incurred by agent k if it were to traverse the edge, which can be modeled by $y_{e,k} \|p_{j,k} - p_{i,k}\|_2$. To define each $l_{e,k}$ variable, the MICP relies on two other auxiliary variables $l_{e,k}^{xy} \in \mathbb{R}^2$, and $\bar{l}_{e,k} \in \mathbb{R}_{\geq 0}$. The variables $l_{e,k}^{xy}$ represent $p_{j,k} - p_{i,k}$, and variables $\bar{l}_{e,k}$ relate to $l_{e,k}$ through a set of big- M constraints in the MICP. The big- M here is the parameter R , which denotes the diagonal length of the square area that contains the depot and all the target trajectories. We now present the baseline formulation below.

$$\min \sum_{k \in V_{agt}} \sum_{e \in E} l_{e,k} \quad (1)$$

subject to constraints

$$\sum_{e \in E_s^{out}} y_{e,k} \leq 1, \quad \forall k \in V_{agt}, \quad (2)$$

$$\sum_{e \in E_{s'}^{in}} y_{e,k} \leq 1, \quad \forall k \in V_{agt}, \quad (3)$$

$$\sum_{k \in V_{agt}} \sum_{i \in C_u} \sum_{e \in E_i^{in}} y_{e,k} = 1, \quad \forall u \in V_{tar}, \quad (4)$$

$$\sum_{e \in E_i^{in}} y_{e,k} = \sum_{e \in E_i^{out}} y_{e,k}, \quad \forall i \in V_{seg}, k \in V_{agt}, \quad (5)$$

$$\underline{t}_i \leq t_{i,k} \leq \bar{t}_i, \quad \forall i \in V, k \in V_{agt}, \quad (6)$$

$$l_{e,k} \leq v_{max}(t_{j,k} - t_{i,k} + T(1 - y_{e,k})), \quad (7)$$

$$l_{e,k}^{xy} = ((\underline{p}_j + t_{j,k}v_j - \underline{t}_jv_j) - (\underline{p}_i + t_{i,k}v_i - \underline{t}_iv_i)), \quad (8)$$

$$\forall e = (i, j) \in E, k \in V_{agt}, \quad (9)$$

$$\bar{l}_{e,k} = l_{e,k} + R(1 - y_{e,k}), \quad \forall e = (i, j) \in E, k \in V_{agt}, \quad (10)$$

The objective (1) is to minimize the sum of tour lengths of all the agents. The condition that an agent can depart from and arrive at the depot at most once is described by (2) and (3) respectively. The constraints (4) ensure that each target is visited exactly once by one of the agents, at one of the target's segments. The flow conservation constraints for all the segment nodes, for all agents, are described by (5). Constraints (2) to (5) are flow constraints that ensure that each agent can have at most one tour, which starts and ends at the depot, and that each target must be visited once by one of the agents. Constraints (6) require that the time at which an agent visits a depot node $i \in \{s, s'\}$ or visits a target τ_i

¹We represent a segment using a unique label $i \in \mathbb{Z}^+$.

at a segment node $i \in V_{seg}$, corresponds to the time interval $[\underline{t}_i, \bar{t}_i]$ for that node. Constraints (8) captures the definition of the auxiliary variables $l_{e,k}^{xy}$.

Now, consider the big- M constraints (7) and (9) for each edge $e = (i, j) \in E$, and each agent $k \in V_{agt}$. The big- M values used here are T , and R , respectively. Constraints (9) along with (8) and (10) describe the condition that if $y_{e,k} = 1$, then $l_{e,k} \geq \|p_{j,k} - p_{i,k}\|_2$ and if $y_{e,k} = 0$, then $l_{e,k}$ is free to take any value. Note that to minimize the objective (1), the condition: $l_{e,k} = \|p_{j,k} - p_{i,k}\|_2$ if $y_{e,k} = 1$, and $l_{e,k} = 0$ if $y_{e,k} = 0$, must be satisfied. The time-feasibility constraints (7) describe the condition that if $y_{e,k} = 1$, then $l_{e,k} \leq v_{max}(t_{j,k} - t_{i,k})$ and if $y_{e,k} = 0$, then no restrictions are placed on $t_{i,k}$ and $t_{j,k}$ (since $l_{e,k} = 0$ in this case).

Although this MICP describes the MA-MT-TSP well, It is challenging to solve in practice. This is possibly due to the big- M constraints (7) and (9), which may lead to poor convex relaxations, as well as numerical instabilities, as noted by the authors in [14]. In the next section, we present an alternative MICP based on the ideas in [29] and [31], that finds the optimum for the MA-MT-TSP, much faster than the baseline. Prior to presenting this formulation, we restate the baseline as a nonconvex MINLP. This will aid us in proving that an optimal solution for our new MICP indeed provides an optimal solution for the MA-MT-TSP.

IV. A NEW MICP FOR THE MA-MT-TSP

A. Nonconvex MINLP Formulation for MA-MT-TSP

This section details how the baseline MICP formulation for the MA-MT-TSP can be restated as a nonconvex MINLP. We refer to this program as the nonconvex formulation for simplicity. First, we define the decision variables. For each node $i \in V$ and agent $k \in V_{agt}$, we reuse the variable $t_{i,k}$ from the baseline. In addition, we also define $p_{i,k}$ as a new auxiliary variable. For each edge $e = (i, j) \in E$, we reuse the variables $y_{e,k}$, $l_{e,k}$, and $l_{e,k}^{xy}$ from the baseline. The $l_{e,k}$ variable is defined with the help of new real auxiliary variables $z_{e,k,p}$ and $z_{e,k,t}$ representing the products $y_{e,k}p_{i,k}$ and $y_{e,k}t_{i,k}$ respectively, and variables $z'_{e,k,p}$ and $z'_{e,k,t}$ representing $y_{e,k}p_{j,k}$ and $y_{e,k}t_{j,k}$ respectively. Observe how in the baseline, $l_{e,k}$ was defined with the help of $\bar{l}_{e,k}$ and big- M constraints instead. We now present the nonconvex formulation below.

$$\min \sum_{k \in V_{agt}} \sum_{e \in E} l_{e,k} \quad (11)$$

subject to constraints (2), (3), (4), (5), (6),

$$p_{i,k} = \underline{p}_i + t_{i,k}v_i - \underline{t}_i v_i, \quad \forall i \in V, k \in V_{agt}, \quad (12)$$

$$l_{e,k} \leq v_{max}(z'_{e,k,t} - z_{e,k,t}), \quad \forall e = (i, j) \in E, k \in V_{agt}, \quad (13)$$

$$l_{e,k}^{xy} = (z'_{e,k,p} - z_{e,k,p}), \quad \forall e = (i, j) \in E, k \in V_{agt}, \quad (14)$$

$$\|l_{e,k}^{xy}\|_2^2 \leq l_{e,k}^2, \quad \forall e = (i, j) \in E, k \in V_{agt}, \quad (15)$$

$$\begin{aligned} z_{e,k,p} &= y_{e,k}p_{i,k}, \quad z_{e,k,t} = y_{e,k}t_{i,k}, \quad z'_{e,k,p} = y_{e,k}p_{j,k}, \\ z'_{e,k,t} &= y_{e,k}t_{j,k}, \quad \forall e = (i, j) \in E, k \in V_{agt}. \end{aligned} \quad (16)$$

This formulation shares the flow constraints (2) to (5) as well as the time of visit constraints (6) from the baseline. Constraints (12) describe the variables $p_{i,k}$. Now, we explain the remaining constraints (13) to (16). Note how apart from the binary requirement for the flow variables $y_{e,k}$, the nonconvexities in this program arise from the product of flow variables and node variables in (16). These constraints help achieve the role satisfied by the big- M constraints in the baseline. When $y_{e,k} = 1$, (16) gives $z_{e,k,p} = p_{i,k}$ and $z'_{e,k,p} = p_{j,k}$. Consequently, constraints (14) and (15) give $l_{e,k} \geq \|p_{j,k} - p_{i,k}\|_2$. However, when $y_{e,k} = 0$, (16) gives $z_{e,k,p} = z'_{e,k,p} = (0, 0)$ and consequently, $l_{e,k} = 0$ from (14) and (15). Like the baseline, in order to minimize the objective, the condition $l_{e,k} = \|p_{j,k} - p_{i,k}\|_2$ if $y_{e,k} = 1$ and $l_{e,k} = 0$ if $y_{e,k} = 0$, must be satisfied. Now, consider the time-feasibility constraints (13). When $y_{e,k} = 1$, $z_{e,k,t} = t_{i,k}$ and $z'_{e,k,t} = t_{j,k}$ from (16). Consequently, (13) gives $l_{e,k} \leq v_{max}(t_{j,k} - t_{i,k})$. However, when $y_{e,k} = 0$, $t_{i,k}$ and $t_{j,k}$ are free to take any values since $l_{e,k} = z_{e,k,t} = z'_{e,k,t} = 0$ in this case. Hence, the nonconvex formulation and the baseline MICP have the same optimal objective value. Moreover, an optimal solution for the MA-MT-TSP can be obtained from an optimal solution for the nonconvex formulation by retrieving the values taken by $y_{e,k}$, $t_{i,k}$, and $p_{i,k}$ variables.

B. A New MICP Formulation for MA-MT-TSP

In this section, we reformulate the nonconvex formulation, as a new MICP for the MA-MT-TSP. First, we define the decision variables. We discard the node variables $t_{i,k}$ and $p_{i,k}$ used in the nonconvex formulation. Additionally, we do not define a separate set of edge variables for each agent $k \in V_{agt}$. We then end up with a single set of decision variables y_e , l_e , l_e^{xy} , $z_{e,p}$, $z_{e,t}$, $z'_{e,p}$, $z'_{e,t}$ for each edge $e \in E$. Now we present the MICP.

$$\min \sum_{e \in E} l_e \quad (17)$$

subject to constraints

$$\sum_{e \in E_s^{out}} y_e = \alpha, \quad (18)$$

$$\sum_{e \in E_s^{in}} y_e = \alpha, \quad (19)$$

$$1 \leq \alpha \leq m, \quad (20)$$

$$\sum_{i \in C_u} \sum_{e \in E_i^{in}} y_e = 1, \quad \forall u \in V_{tar}, \quad (21)$$

$$\sum_{e \in E_i^{in}} (z'_{e,t}, y_e) = \sum_{e \in E_i^{out}} (z_{e,t}, y_e), \quad \forall i \in V_{seg}, \quad (22)$$

$$y_e t_i \leq z_{e,t} \leq y_e \bar{t}_i, \quad \forall e = (i, j) \in E, \quad (23)$$

$$y_e \bar{t}_j \leq z'_{e,t} \leq y_e \bar{t}_j, \quad \forall e = (i, j) \in E, \quad (24)$$

$$z_{e,p} = v_i z_{e,t} + y_e (\underline{p}_i - \underline{t}_i v_i), \quad \forall e = (i, j) \in E, \quad (25)$$

$$z'_{e,p} = v_j z'_{e,t} + y_e (\underline{p}_j - \underline{t}_j v_j), \quad \forall e = (i, j) \in E, \quad (26)$$

$$l_e \leq v_{\max}(z'_{e,t} - z_{e,t}), \quad \forall e = (i, j) \in E, \quad (27)$$

$$l_e^{xy} = (z'_{e,p} - z_{e,p}), \quad \forall e = (i, j) \in E, \quad (28)$$

$$\|l_e^{xy}\|_2^2 \leq l_e^2, \quad \forall e = (i, j) \in E. \quad (29)$$

This formulation defines multiple agent tours without defining a separate set of edge variables for each agent. This is done by allowing an integer flow α where $1 \leq \alpha \leq m$, in and out of the depot, and restricting the flow in and out of the other nodes to be equal, and be at most 1. Recall that m denotes the total number of agents. These conditions are described by the new flow constraints (18) to (22), which effectively replaces (2) to (5) in the nonconvex formulation. Note that in (22), an additional condition, $\sum_{e \in E_i^{\text{in}}} z'_{e,t} = \sum_{e \in E_i^{\text{out}}} z_{e,t}$ is added for each node $i \in V_{\text{seg}}$. This is to ensure that the incoming and outgoing edge for some segment node i in a tour, coincide at the same time corresponding to that node. This additional condition is used since node variables for time are not introduced in this formulation.

Now, consider the remaining constraints (23) to (29). Constraints (27) to (29) are similar to constraints (13) to (15) in the nonconvex formulation, with the only changes coming from not having a separate set of edge variables and corresponding constraints for each agent. Finally, constraints (23) to (26) in the MICP replaces constraints (6), (12), and (16) in the nonconvex formulation. The key idea here is as follows: For each $e = (i, j) \in E$, we define $(z_{e,p}, z_{e,t}) \in y_e X_i$ and $(z'_{e,p}, z'_{e,t}) \in y_e X_j$, where $(p_i, t_i) \in X_i$ represent the below two inequalities, similar to constraints (6) and (12), for some node $i \in V$.

$$\underline{t}_i \leq t_i \leq \bar{t}_i, \quad (30)$$

$$p_i = \underline{p}_i + t_i v_i - \underline{t}_i v_i. \quad (31)$$

Consequently, we let $(z_{e,p}, z_{e,t}) \in X_i$ and $(z'_{e,p}, z'_{e,t}) \in X_j$ when $y_e = 1$, and become $(0, 0, 0)$ when $y_e = 0$, without explicitly defining node variables t_i and p_i , or multiplying them with the flow variables like in (16).

From an optimal solution to the MICP, optimal agent tours for the nonconvex formulation can be obtained by considering the subset of edges $E_{\text{tour}} := \{e \in E : y_e = 1\}$ and finding all the $1 \leq \alpha \leq m$ vertex-disjoint paths starting at s and ending at s' within E_{tour} . This can be done using a graph search algorithm such as Depth-First Search (DFS). Since all the agents are homogeneous and share the same depot, any of the paths can be assigned to any agent. Let $E_{\text{tour}}^k \subseteq E_{\text{tour}}$ denote the path assigned to an agent $k \in V_{\text{agt}}$. Note that $E_{\text{tour}} = \bigcup_{k \in V_{\text{agt}}} E_{\text{tour}}^k$. Also note that it is possible $E_{\text{tour}}^k = \emptyset$ for some k in the case where some of the agents are not assigned tours. Finally, note that for some $k \in V_{\text{agt}}$, $E_{\text{tour}}^k = \{e \in E : y_{e,k} = 1\}$ for the nonconvex

formulation. Hence, for each $e = (i, j) \in E_{\text{tour}}^k$, we set $y_{e,k} = 1$, and set $(z_{e,k,p}, z_{e,k,t}) = (p_{i,k}, t_{i,k}) = (z_{e,p}, z_{e,t})$ and $(z'_{e,k,p}, z'_{e,k,t}) = (p_{j,k}, t_{j,k}) = (z'_{e,p}, z'_{e,t})$.

C. Proof of Validity

In this section, we will show the correctness of the MICP formulation by proving the following theorem.

Theorem 1 *The optimal value of the MICP formulation is equal to that of the nonconvex formulation for the MA-MT-TSP. Additionally, optimal agent tours for the nonconvex formulation can be recovered from an optimal solution to the MICP by following the procedure explained previously.*

Proof: Let E_{tour} be the set of all edges selected at optimality by either of the two formulations. For the nonconvex formulation, $E_{\text{tour}} := \{e \in E : \exists k \in V_{\text{agt}} \text{ such that } y_{e,k} = 1\}$, and for the MICP formulation, $E_{\text{tour}} := \{e \in E : y_e = 1\}$. The flow constraints in both formulations require that the edges in E_{tour} form $1 \leq \alpha \leq m$ vertex-disjoint paths, that starts at s and ends at s' , such that for each target $u \in V_{\text{tar}}$, exactly one node in C_u is visited by one of the α paths. First, consider an edge $e \notin E_{\text{tour}}$. In the nonconvex formulation, for all $k \in V_{\text{agt}}$, $y_{e,k} = 0$. Consequently, constraints (16) require $(z_{e,k,p}, z_{e,k,t}) = (z'_{e,k,p}, z'_{e,k,t}) = (0, 0, 0)$ for all $k \in V_{\text{agt}}$. Similarly, in the MICP formulation, $y_e = 0$ leading to constraints (23) to (26) requiring $(z_{e,p}, z_{e,t}) = (z'_{e,p}, z'_{e,t}) = (0, 0, 0)$. Hence, the cost addend is $\sum_{k \in V_{\text{agt}}} l_{e,k} = l_e = 0$ for either formulation. Now, consider an edge $e = (i, j) \in E_{\text{tour}}$. In the nonconvex formulation, since edge e is assigned to some unique agent $k^* \in V_{\text{agt}}$, $y_{e,k^*} = 1$ when $k = k^*$ and $y_{e,k} = 0$ for all other $k \in V_{\text{agt}} \setminus \{k^*\}$. Consequently, (16) requires $(z_{e,k^*,p}, z_{e,k^*,t}) = (p_{i,k^*}, t_{i,k^*})$ and $(z'_{e,k^*,p}, z'_{e,k^*,t}) = (p_{j,k^*}, t_{j,k^*})$. Additionally, (6) and (12) require $(p_{i,k^*}, t_{i,k^*}) \in X_i$ and $(p_{j,k^*}, t_{j,k^*}) \in X_j$. These two requirements, with the help of flow conservation, (5) reads $(z_{e,k^*,p}, z_{e,k^*,t}) \in X_i$, $(z'_{e,k^*,p}, z'_{e,k^*,t}) \in X_j$, and for adjacent edges $e = (i, j)$ and $f = (j, k)$ in E_{tour} , $(z'_{e,k^*,p}, z'_{e,k^*,t}) = (z_{f,k^*,p}, z_{f,k^*,t})$. Similarly, in the MICP formulation, $y_e = 1$ and therefore, constraints (23) to (26) require $(z_{e,p}, z_{e,t}) \in X_i$, $(z'_{e,p}, z'_{e,t}) \in X_j$. Additionally, the new requirement in (22) ensures that for adjacent edges $e = (i, j)$ and $f = (j, k)$ in E_{tour} , $(z'_{e,p}, z'_{e,t}) = (z_{f,p}, z_{f,t})$. Finally, the time-feasibility constraints in both the formulations (13) and (27) have the same form. Therefore, corresponding to an edge $e \in E_{\text{tour}}$, the cost addend in the nonconvex formulation is $\sum_{k \in V_{\text{agt}}} l_{e,k} = l_{e,k^*} = \|z'_{e,k^*,p} - z_{e,k^*,p}\|_2$ which is the same as the cost addend $l_e = \|z'_{e,p} - z_{e,p}\|_2$ by the MICP formulation. Since the E_{tour} corresponding to an optimal solution for one formulation, when used in the other formulation, results in a feasible solution with the same cost, both formulations share the same optimal value. Moreover, from an optimal solution to the MICP, optimal agent tours for the nonconvex formulation can be obtained by following the previously described procedure. ■

V. NUMERICAL RESULTS

A. Test Settings and Instance Generation

All the tests were run on a laptop with an Intel Core i7-7700HQ 2.80GHz CPU, and 16GB RAM. The implementation was in Python 3.11.6, and both the baseline MICP formulation (MICP-Baseline) and the new MICP formulation (MICP) were solved using Gurobi 10.0.3 optimizer [32]. All the Gurobi parameters were set to their default values, except for *TimeLimit*², which was set to 1800.

A total of 80 instances were generated, with 20 instances each for 5, 10, 15, and 20 targets. Each instance was defined by the number of targets n , a square area of fixed size $S = 100$ units (with diagonal length $R = \sqrt{2}S$), a fixed time horizon $T = 150$ secs, the depot location fixed at the center (0,0) of the square, the maximum speed of the agents $v_{max} = 4$ units/sec, and finally, a set of randomly generated piecewise-linear trajectories corresponding to the n targets such that each target has a constant speed within $[0.5, 1]$ unit/sec, and is confined within the square area.

For each instance, we conducted experiments where we varied two test parameters: (a) the total time window duration for each target, which is the sum of durations of all the time windows associated with the target, and (b) the number of agents m . The total time window duration was varied to be 20, 40, and 60 secs, and the number of agents were varied to be from 1 to 5. Each target was assigned two time windows of equal duration in all the experiments. The time windows were selected such that for all generated instances, a feasible solution exists even when restricting the number of agents to 1. This was done by randomly choosing a sequence in which all the targets are visited by a single agent, and then finding the quickest tour for that sequence by fixing the speed of the agent to be v_{max} . If the time to complete the tour exceeded T , the process was repeated with another sequence. Otherwise, for all choices of total time window durations, two non-intersecting time windows of half the total duration were defined for each target such that one of the time windows contains the time at which the target was visited in the tour.

To evaluate the MICP-Baseline and MICP formulations, we use % Gap, and runtime, which we will now explain. For a given total time window duration, number of agents, and formulation of choice, the solver is first run on all the 20 instances corresponding to a given number of targets. The optimality gap value from the solver for an instance is defined as $\frac{|c_f - c_{lb}|}{|c_f|} \times 100$, where c_f is the feasible (primal) objective, and c_{lb} is the lower bound (dual) objective. % Gap denotes the average of the smallest gap values output by the solver within the time limit for all these instances, and runtime denotes the average of the run-times output by the solver for all these instances.

²Limits the total time expended (in seconds).

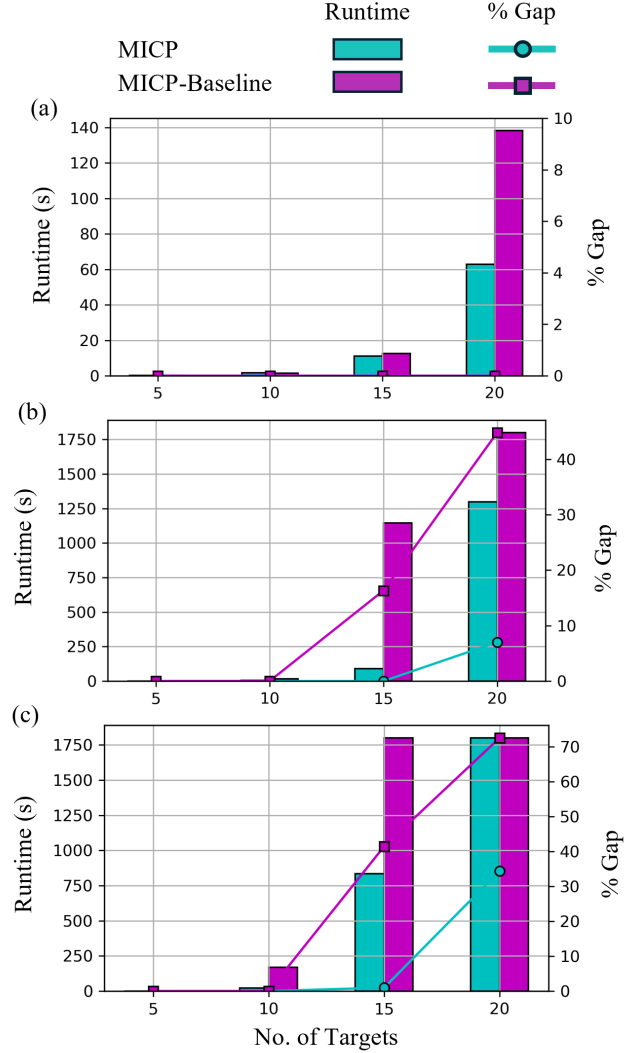


Fig. 2. Numerical results comparing the % Gap and runtime for MICP-Baseline and MICP, when the number of agents is fixed at 1, and the total time window duration is varied to be 20 (a), 40 (b), and 60 (c) secs. The MICP scales significantly better than MICP-Baseline with larger time window durations and more targets. For 10 targets in (c) and 15 targets in (b), the MICP shows a runtime improvement of two orders of magnitude. It also shows a % Gap improvement of more than 15 for 15 targets in (b). Similarly, for 15 targets in (c) and 20 targets in (b) and (c), the MICP runs up to 1000 seconds faster, while also providing a % Gap improvement within a 35-40 range.

B. Varying the Total Time Window Duration

In this section, we present the results of experiments where the total time window duration was varied. The number of agents were fixed at 1, and all the instances were solved using both formulations for the varying durations (20, 40, and 60 secs). The results of this experiment are illustrated in Fig. 2, with panels (a), (b), and (c) corresponding to durations 20, 40, and 60, respectively. We observe that for both the formulations, the problem becomes more challenging to solve as the number of targets increases. Moreover, this difficulty becomes more prominent as the total time window duration increases. However, we observe that the MICP

scales significantly better than MICP-Baseline against larger number of targets and bigger time windows. This can be seen especially in the case of 15 targets where the % Gap always converge to 0 for the MICP, and its runtime increases noticeably only with the largest time window duration of 60, as opposed to the MICP-Baseline whose % Gap and runtime grows dramatically with increasing time window durations. Note how at 20 targets, the problem is challenging for both formulations. However, MICP still significantly outperforms MICP-Baseline in terms of runtime or % Gap for all the time window durations considered.

C. Varying the Number of Agents

In this section, we present the results of experiments where the number of agents was varied. The total time window duration was fixed at 40 *secs*, and all the instances were solved using both formulations for 2, 3, 4, and 5 agents. The results of this experiment are illustrated in Fig. 3, with panels (a), (b), (c), and (d) corresponding to 2, 3, 4, and 5 agents, respectively. Recall that the 1 agent case was already presented in Fig. 2 (b). Hence, it is not repeated in Fig. 3. Like before, the % Gap and runtime worsen with more targets for both formulations. With additional agents, this effect is further pronounced for the MICP-Baseline. However, for the MICP, we observe an improvement in results in this case. From Fig. 3, we observe that the MICP-Baseline struggles significantly with 2 or more agents for 15-target instances, and with 4 or more agents for 10-target instances. On average, we find the solver reaches the time limit in these cases, with a remaining % Gap of more than 60. However, for the MICP, the % Gap always converges to 0. Moreover, we observe a significant improvement in runtime, with up to two orders of magnitude for 10 and 15 targets, and one order of magnitude for 20 targets. This can be attributed to the MICP not using a separate set of decision variables for each agent and to the improved convexification process it uses, which does not rely on big- M constraints.

VI. CONCLUSION AND FUTURE WORK

In this paper, we presented a new Mixed-Integer Conic Programming formulation that finds optimal solutions to the Multi-Agent Moving-Target Traveling Salesman Problem. We considered scenarios where targets move along piecewise-linear segments and may have multiple associated time windows. We proved the validity of our formulation, and provided numerical results to corroborate its performance. Our experiments show that the proposed MICP outperforms the state-of-the-art MICP, both in terms of runtime and optimality gap across various experiments. The formulation is currently limited to cases where agents are homogeneous and share a common depot. As future work, we plan to extend our approach to accommodate heterogeneous agents and multiple depots.

ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant 2120219 and Grant

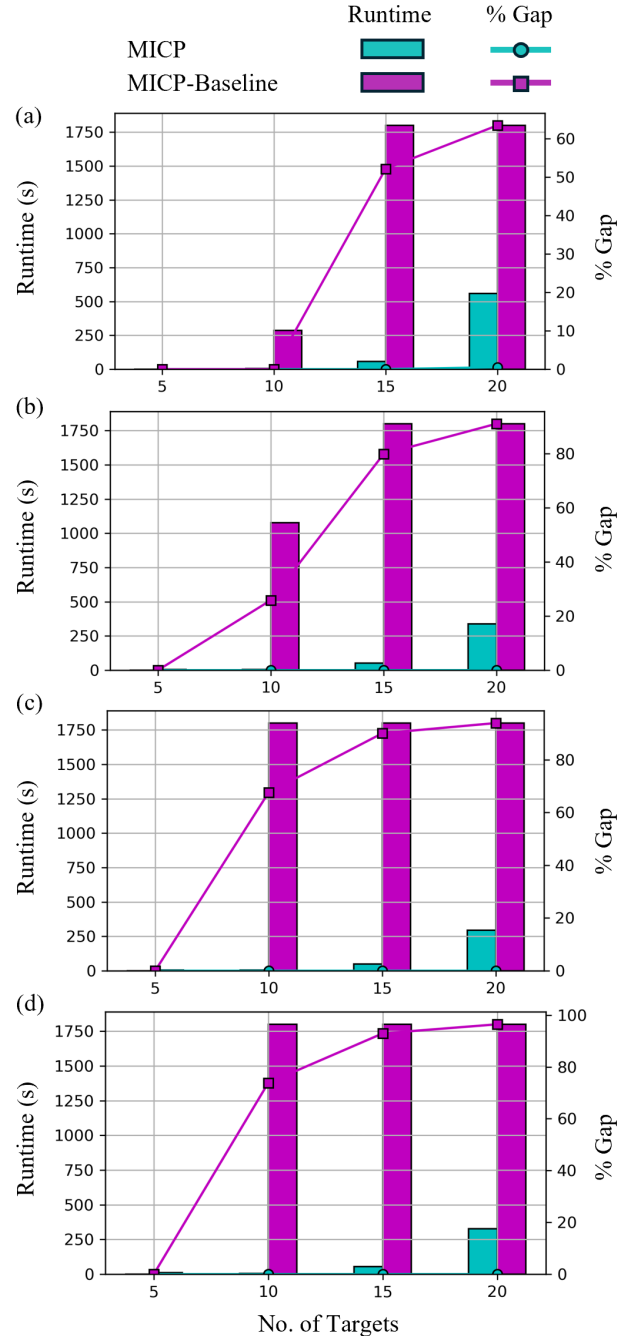


Fig. 3. Numerical results comparing the % Gap and runtime for MICP-Baseline and MICP, when the total time window duration is fixed at 40, and the number of agents is varied to be 2 (a), 3 (b), 4 (c), and 5 (d). The MICP significantly outperforms MICP-Baseline for all number of targets and agents. Note how for 15 targets, MICP runs two orders of magnitude faster, while giving a % Gap improvement of more than 50 for 2 agents to more than 90 for 5 agents. Similarly, for 20 targets, the MICP runs one-order of magnitude faster, while giving a % Gap improvement of 60 for 2 agents to more than 90 for 5 agents.

2120529. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

- [1] P. Oberlin, S. Rathinam, and S. Darbha, "Today's traveling salesman problem," *IEEE robotics & automation magazine*, vol. 17, no. 4, pp. 70–77, 2010.
- [2] Y. Liu and R. Bucknall, "Efficient multi-task allocation and path planning for unmanned surface vehicle in support of ocean operations," *Neurocomputing*, vol. 275, pp. 1550–1566, 2018.
- [3] J. L. Ryan, T. G. Bailey, J. T. Moore, and W. B. Carlton, "Reactive tabu search in unmanned aerial reconnaissance simulations," in *1998 Winter Simulation Conference. Proceedings (Cat. No. 98CH36274)*, vol. 1. IEEE, 1998, pp. 873–879.
- [4] Z. Yu, L. Jinhai, G. Guochang, Z. Rubo, and Y. Haiyan, "An implementation of evolutionary computation for path planning of cooperative mobile robots," in *Proceedings of the 4th World Congress on Intelligent Control and Automation (Cat. No. 02EX527)*, vol. 3. IEEE, 2002, pp. 1798–1802.
- [5] A. M. Ham, "Integrated scheduling of m-truck, m-drone, and m-depot constrained by time-window, drop-pickup, and m-visit using constraint programming," *Transportation Research Part C: Emerging Technologies*, vol. 91, pp. 1–14, 2018.
- [6] S. Venkatachalam, K. Sundar, and S. Rathinam, "A two-stage approach for routing multiple unmanned aerial vehicles with stochastic fuel consumption," *Sensors*, vol. 18, no. 11, p. 3756, 2018.
- [7] H. A. Saleh and R. Chelouah, "The design of the global navigation satellite system surveying networks using genetic algorithms," *Engineering Applications of Artificial Intelligence*, vol. 17, no. 1, pp. 111–122, 2004.
- [8] O. Cheikhrouhou, A. Koubâa, and A. Zarrad, "A cloud based disaster management system," *Journal of Sensor and Actuator Networks*, vol. 9, no. 1, p. 6, 2020.
- [9] J. Conesa-Muñoz, G. Pajares, and A. Ribeiro, "Mix-opt: A new route operator for optimal coverage path planning for a fleet in an agricultural environment," *Expert Systems with Applications*, vol. 54, pp. 364–378, 2016.
- [10] W. Zhao, Q. Meng, and P. W. Chung, "A heuristic distributed task allocation method for multivehicle multitask problems and its application to search and rescue scenario," *IEEE transactions on cybernetics*, vol. 46, no. 4, pp. 902–915, 2015.
- [11] B. L. Brumitt and A. Stentz, "Dynamic mission planning for multiple mobile robots," in *Proceedings of IEEE International Conference on Robotics and Automation*, vol. 3. IEEE, 1996, pp. 2396–2401.
- [12] C. S. Helvig, G. Robins, and A. Zelikovsky, "The moving-target traveling salesman problem," *Journal of Algorithms*, vol. 49, no. 1, pp. 153–174, 2003.
- [13] C. D. Smith, "Assessment of genetic algorithm based assignment strategies for unmanned systems using the multiple traveling salesman problem with moving targets," 2021.
- [14] A. Stieber and A. Fügenschuh, "Dealing with time in the multiple traveling salespersons problem with moving targets," *Central European Journal of Operations Research*, vol. 30, no. 3, pp. 991–1017, 2022.
- [15] R. S. de Moraes and E. P. de Freitas, "Experimental analysis of heuristic solutions for the moving target traveling salesman problem applied to a moving targets monitoring system," *Expert Systems with Applications*, vol. 136, pp. 392–409, 2019.
- [16] Y. Wang and N. Wang, "Moving-target travelling salesman problem for a helicopter patrolling suspicious boats in antipiracy escort operations," *Expert Systems with Applications*, vol. 213, p. 118986, 2023.
- [17] D. Marlow, P. Kilby, and G. Mercer, "The travelling salesman problem in maritime surveillance—techniques, algorithms and analysis," in *Proceedings of the international congress on modelling and simulation*, 2007, pp. 684–690.
- [18] A. Maskooki and M. Kallio, "A bi-criteria moving-target travelling salesman problem under uncertainty," *European Journal of Operational Research*, 2023.
- [19] B. Englot, T. Sahai, and I. Cohen, "Efficient tracking and pursuit of moving targets by heuristic solution of the traveling salesman problem," in *52nd IEEE conference on decision and control*. IEEE, 2013, pp. 3433–3438.
- [20] P. Chalasani and R. Motwani, "Approximating capacitated routing and delivery problems," *SIAM Journal on Computing*, vol. 28, no. 6, pp. 2133–2149, 1999.
- [21] J.-M. Bourjolly, O. Gurtuna, and A. Lyngvi, "On-orbit servicing: a time-dependent, moving-target traveling salesman problem," *International Transactions in Operational Research*, vol. 13, no. 5, pp. 461–481, 2006.
- [22] N. S. Choubey, "Moving target travelling salesman problem using genetic algorithm," *International Journal of Computer Applications*, vol. 70, no. 2, 2013.
- [23] C. Groba, A. Sartal, and X. H. Vázquez, "Solving the dynamic traveling salesman problem using a genetic algorithm with trajectory prediction: An application to fish aggregating devices," *Computers & Operations Research*, vol. 56, pp. 22–32, 2015.
- [24] Q. Jiang, R. Sarker, and H. Abbass, "Tracking moving targets and the non-stationary traveling salesman problem," *Complexity International*, vol. 11, no. 2005, pp. 171–179, 2005.
- [25] U. Ucar and S. K. İşleyen, "A meta-heuristic solution approach for the destruction of moving targets through air operations," *International Journal of Industrial Engineering*, vol. 26, no. 6, 2019.
- [26] M. Hammar and B. J. Nilsson, "Approximation results for kinetic variants of tsp," in *Automata, Languages and Programming: 26th International Colloquium, ICALP'99 Prague, Czech Republic, July 11–15, 1999 Proceedings 26*. Springer, 1999, pp. 392–401.
- [27] M. Hassoun, S. Shoval, E. Simchon, and L. Yedidsion, "The single line moving target traveling salesman problem with release times," *Annals of Operations Research*, vol. 289, pp. 449–458, 2020.
- [28] A. G. Philip, Z. Ren, S. Rathinam, and H. Choset, "C*: A new bounding approach for the moving-target traveling salesman problem," *arXiv preprint arXiv:2312.05499*, 2023.
- [29] —, "A mixed-integer conic program for the moving-target traveling salesman problem based on a graph of convex sets," *arXiv preprint arXiv:2403.04917*, 2024.
- [30] A. Bhat, G. Gutow, B. Vundurthy, Z. Ren, S. Rathinam, and H. Choset, "A complete algorithm for a moving target traveling salesman problem with obstacles," *arXiv preprint arXiv:2409.09852*, 2024.
- [31] T. Marcucci, J. Umenberger, P. Parrilo, and R. Tedrake, "Shortest paths in graphs of convex sets," *SIAM Journal on Optimization*, vol. 34, no. 1, pp. 507–532, 2024.
- [32] Gurobi Optimization, LLC, "Gurobi Optimizer Reference Manual," 2023. [Online]. Available: <https://www.gurobi.com>