Applied Mathematics

Helmut Schmidt Universität
Universität der Bundeswehr Hamburg

Angewandte Mathematik und Optimierung Schriftenreihe
Applied Mathematics and Optimization Series
AMOS # 61(2017)

Anke Stieber, Armin Fügenschuh

# The Multiple Traveling Salesmen Problem with Moving Targets and Nonlinear Trajectories

Herausgegeben von der
Professur für Angewandte Mathematik
Professor Dr. rer. nat. Armin Fügenschuh

Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg
Fachbereich Maschinenbau
Holstenhofweg 85
D-22043 Hamburg

Telefon: +49 (0)40 6541 3540
Fax: +49 (0)40 6541 3672

e-mail: appliedmath@hsu-hh.de
URL: http://www.hsu-hh.de/am

Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Print 2199-1928
Angewandte Mathematik und Optimierung Schriftenreihe (AMOS), ISSN-Internet 2199-1936

# The Multiple Traveling Salesmen Problem with Moving Targets and Nonlinear Trajectories

Anke Stieber and Armin Fügenschuh

Helmut Schmidt University / University of the Federal Armed Forces Hamburg
Holstenhofweg 85, 22043 Hamburg, Germany
{anke.stieber,fuegenschuh}@hsu-hh.de

**Abstract.** We consider the multiple traveling salesmen problem with moving targets (MTSPMT), which is a generalization of the classical traveling salesman problem (TSP). Here the nodes (objects, targets) are not static, they are moving over time on trajectories. Moreover, each target has a visibility time window and it can be reached by a salesman only within that time window. The objective function is to minimize the total traveled distances of all salesmen. We recall the time-discrete model formulation from Stieber et al. [9]. This model is applicable to arbitrarily shaped trajectories. Thus, we generated non-linear trajectories based on polynomials, trigonometric functions and their combinations. Computational experiments are carried out with up to 16 trajectories and the results are compared to the ones obtained with linear trajectories.

**Keywords:** Dynamic traveling salesmen problem, moving targets, integer linear programming, non-linear trajectories.

## 1 Problem Description of the MTSPMT

The MTSPMT is a dynamic variant of the classical TSP and the time plays an important role here. The nodes (targets, objects) are not fixed as in the classical TSP, they move over time on arbitrary trajectories. Each target is associated with a certain speed value and a visibility time window. We consider hard time windows, so that a target can only be intercepted by a salesman within its respective time window. This variant also considers more than one salesman. Each salesman is assigned a certain speed value. All salesmen start their tours from an initial depot, located w.l.o.g. in the middle of the considered area or space. Each target has to be visited once and by exactly one salesman. The aim is to find a tour for each salesman in order to minimize the total traveled distance aggregated by all salesmen. If we restrict the number of salesmen to one, fix each target to a certain local position and extend their time windows to infinity, we obtain the classical TSP, which is NP-hard. Thus, the MTSPMT as a generalization of the TSP is also NP-hard.

Applications can be found in the defense sector, e.g., protection of an airport or a security zone (for details see Stieber et al. [9]) or in the logistic sector, e.g., supplying a fleet of boats or mobile ground units. For many such applications, MTSPMT should be treated as an online optimization problem, that is, the targets are not known before the optimization starts ("offline"), instead they occur afterwards. Still, a fast routine to solve the "offline" variant could serve as the backbone of an online solver with a moving horizon approach. Here new data is integrated into the offline algorithm at run-time, and a fast offline algorithm can be used to get a tentative decision to the "online" problem that is re-optimize anytime new targets emerge.

In the literature the MTSPMT is only addressed considering small instances and with many restrictions to the problem parameters, e.g., movement and speed of the targets or the one-dimensional case is considered, see for example [1,2,4,5,7]. The MTSPMT modeled with discrete time steps is also very similar to the asymmetrical equality generalized multiple depot TSP (E-GMDTSP), where targets are assigned to clusters and exactly one target from each cluster has to be visited by a salesman, see for example [6,8,10]. To the best of our knowledge, there is no exact algorithms to the asymmetrical E-GMDTSP.

## 2 Model formulation

We presented a mixed-integer linear programming formulation for the MTSPMT in Stieber et al. [9]. Therefor, the underlying graph is embedded in a time-expanded network and the MTSPMT is formulated as a multi-commodity flow problem. We recall the formulation in a concise way.

The set of salesmen is denoted by $\mathcal{W} = \{1, \ldots, w\}$ and the set of targets by $\mathcal{V} = \{1, \ldots, n\}$. All salesmen start their tour from the same depot location $o$, hence, $\mathcal{V}_o = \mathcal{V} \cup \{o\}$. Then we have the set of arcs (roads) as $\mathcal{A} \subseteq \mathcal{V}_o \times \mathcal{V}$. We consider a finite time horizon $[0, T]$. The distance for salesman $k$ traveling from target $i$ to target $j$ starting at time $s$ in $i$ and arriving at time $t$ in $j$ is given by the function $c_{i,j,k} : [0,T] \times [0,T] \to \mathbb{R}_+ \cup \{\infty\}$. Since each target $i \in \mathcal{V}$ is assigned a visibility time window $[\underline{t}_i, \overline{t}_i]$, we have

$$c_{i,j,k}(s,t) = \infty \quad \text{if } s \notin [\underline{t}_i, \overline{t}_i] \text{ or } t \notin [\underline{t}_j, \overline{t}_j] \text{ or } (t-s)\overline{v} < \|v_j(t) - v_i(s)\|_2,$$

where $v_i(s)$ and $v_j(t)$ are the respective locations of the targets at the times $s$ and $t$ and $\overline{v}$ is the maximum speed value of all salesmen. The arrival time of any salesman at a target is equal to his departure time at the same target, because waiting times are included in the traveling times. Thus, salesmen do not necessarily use the maximum speed $\overline{v}$.

The time horizon is discretized into $m+1$ equidistant time steps $\mathcal{T} = \{0, \ldots, m\}$ with step length $\Delta t$, hence $T = m\Delta t$. We evaluate $c$ only at these:

$$c_{i,j,k}^{p,q} := c_{i,j,k}(p\Delta t, q\Delta t).$$

In the time-expanded network arcs go from one time layer to a later time layer, hence the time-dependent set of arcs is denoted by $\tilde{\mathcal{A}}$ and an arc is specified by $(i,p,j,q)$ with $i \in \mathcal{V}_o, j \in \mathcal{V}$ and $p, q \in \mathcal{T}$. This means the length of an arc (i.e., distance) is dependent on the departure and arrival times. We introduce a family of binary decision variables $x_{i,j,k}^{p,q} \in \{0,1\}$, where $x_{i,j,k}^{p,q} = 1$ represents the decision of sending salesman $k$ from target $i$ to $j$, departing in $i$ at time step $p$ and arriving in $j$ at time step $q$. Then the optimization problem is given by

$$\sum_{k \in \mathcal{W}} \sum_{(i,p,j,q) \in \tilde{\mathcal{A}}} c_{i,j,k}^{p,q} \, x_{i,j,k}^{p,q} \quad \to \quad \min . \tag{1}$$

$$\text{s.t.} \sum_{k \in \mathcal{W}} \sum_{(i,p,q):(i,p,j,q) \in \tilde{\mathcal{A}}} x_{i,j,k}^{p,q} = 1, \quad \forall\, j \in \mathcal{V}. \tag{2}$$

$$\sum_{(i,j,q):(i,p,j,q) \in \tilde{\mathcal{A}}} x_{i,j,k}^{p,q} \leq 1, \quad \forall\, k \in \mathcal{W}, p \in \mathcal{T}. \tag{3}$$

$$\sum_{(i,p):(i,p,j,q) \in \tilde{\mathcal{A}}} x_{i,j,k}^{p,q} \geq \sum_{(i,p):(j,q,i,p) \in \tilde{\mathcal{A}}} x_{j,i,k}^{q,p}, \qquad \forall\, j \in \mathcal{V}, q \in \mathcal{T}, k \in \mathcal{W}. \tag{4}$$

$$x \in \{0,1\}^{\tilde{\mathcal{A}} \times \mathcal{W}}. \tag{5}$$

The objective function (1) is the sum of all traveled distances of all salesmen. Constraints (2) ensure, that every target is reached once. Inequalities (3) guarantee, that a tour is not split up and (4) are the flow conservation constraints. The presented model is not restricted to particular shapes of the target trajectories. Thus, it can handle linear and non-linear trajectories.

## 3 Instance generation

The operating space for our test instances is a square of 500 length units. A test instance is specified by the number of targets, the number of salesmen and the discretization level. The discretization level is a measure of how dense the discretization is done. We used 3 different levels D32, D16 and D8. The first one is based on a discretization every 32 length units (arc length) on each trajectory, the other discretization levels use a step size of 16 and 8 length units respectively. The targets are assigned a constant speed value of 32 length units per time step and the salesmen can travel with at most 200 length units per time step.

For the non-linear trajectories we used polynomial functions and trigonometric functions and a combination by sum and product. We created 16 non-linear trajectories. Then, instances for 6, 8, 10, 12, 14, and 16 targets were created in a way, that we started with 6 trajectories and gradually added two more until we had 16. See the left picture of Figure 1 for their visualization. In the linear case we used randomly generated trajectories. For reasons of visibility we avoided the straight lines from intercepting each other. We created 5 instances per setting. The trajectories were added the same way as for the nonlinear ones when the number of targets rises. All generated trajectories have a length between 100 and 400 length units and time steps are distributed in a way that all instances are solvable instances. An instance with 16 linear trajectories is visualized in the right picture of Figure 1 as an example.

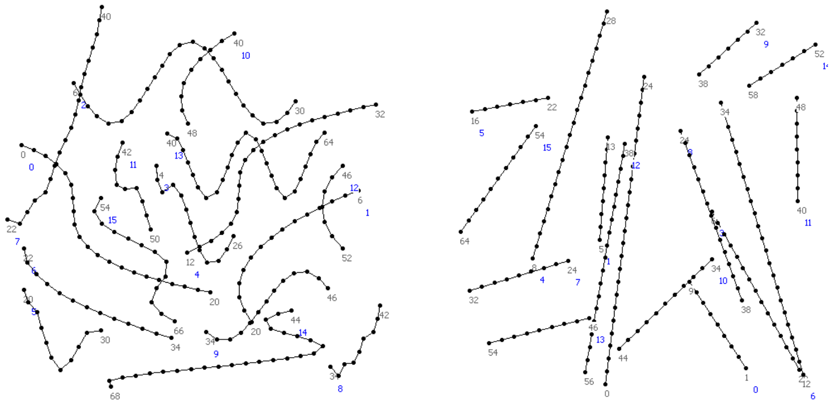

**Fig. 1.** Generated trajectories. Left picture: 16 nonlinear trajectories. Right picture: 16 linear trajectories (one out of 5 instances). The trajectory number is given in blue and the first and last time step of the trajectory is given in black. All trajectories were generated with the discretization level D16.

## 4 Computational results

All computational experiments were carried out on a 2014 Apple Mac mini computer with an Intel Core i7 CPU running at 2.6 GHz on 4 cores, and 16 GB 1600 MHz RAM. The model was implemented in C++ and instances were solved with the MILP solver IBM ILOG CPLEX 12.7.0 [3]. The computations were performed on a single thread, the CPLEX parameter for the MIP gap was set to 0.0. All other CPLEX parameters were used with their default values.

The computational results are listed in Table 1. Here, the first two columns define the number of targets (nbt) and the number of salesmen (nbs), columns 3 to 5 contain the running times for different discretization levels (dl) and column 6 and 7 the objective function values (ofv) for the nonlinear instances. The running times for the linear instances are given in columns 8 to 10 (for different discretization levels). Since the run-time values for the linear trajectories are averaged values over 5 different instances, we do not provide objective function values. All values in Table 1 are rounded to one digit after point. The results show for both nonlinear and linear trajectories, that the instances become more complex, when the discretization level rises. But there is another effect, that is apparent in the results. For bigger instances (for nonlinear trajectories 10 targets and greater with D16 and D8, for linear trajectories 12 targets and greater with D16 and D8) the running times for a salesman number of 2 is much higher than for 4 and 6 salesmen. The run-time for 2 salesmen is more than twice as much as the run-time for 4 or 6 salesmen in the nonlinear case. Also, in the linear case most of the instances follow this behavior. Considering the smaller instances with 6 and 8 targets, the behavior is completely reversed, the run-times increase when the salesman number rises. In the linear case this behavior is similar but not so apparent as for the nonlinear trajectories, because of the instances with 8 targets. The optimal solutions of the trajectories visualized above are given in Figure 2.

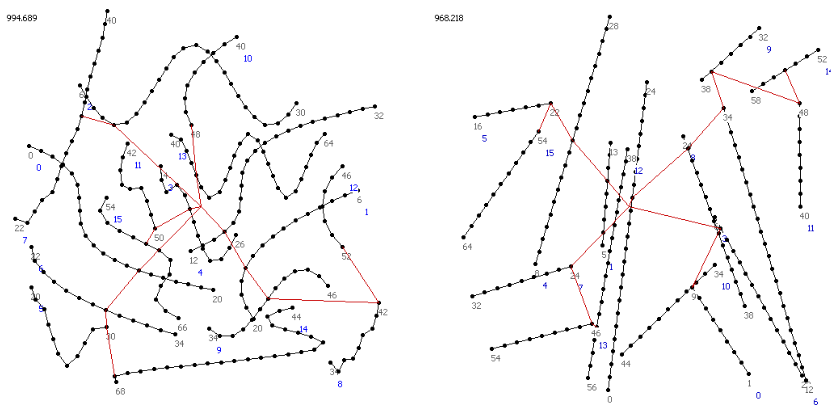

**Fig. 2.** Solution tours with 16 targets and 6 salesmen with D16 is given by a red line. Left picture: nonlinear trajectories. Right picture: linear trajectories.

## 5 Conclusion

We considered the MTSPMT with a model as a multi-commodity flow problem. In the literature and in [9] only linear trajectories were considered. Here, we solved instances with nonlinear trajectories and compared them with linear ones. From the computational results we can conclude, that the running time is not dependent on the shape of the trajectories and that instances with 2 targets are often more complex than with 4 or 6 targets.

## References


1. C. Helvig, G. Robins, and A. Zelikovsky. Moving-Target TSP and Related Problems. In A.P.G. Bilardi, G. F. Italiano, and G. Pucci, editors, *Proceedings of the European Symposium on Algorithms*, volume 1461 of *Lecture Notes in Computer Science*, pages 453–464. Springer Verlag, Berlin, 1998.
2. C. Helvig, G. Robins, and A. Zelikovsky. The moving-target traveling salesman problem. *Journal of Algorithms*, 49(1):153–174, 2003.

**Table 1.** Running times in seconds. Values for linear instances averaged over 5.

| instance | | run-time nonlinear | | | ofv nonlinear | | run-time linear | | |
|---|---|---|---|---|---|---|---|---|---|
| nbt | nbs | D32 | D16 | D8 | D32 | D8 | D32 | D16 | D8 |
| 6 | 2 | 0.1 | 0.2 | 0.9 | 449.2 | 431.1 | 0.0 | 0.1 | 0.6 |
| 6 | 4 | 0.1 | 0.5 | 1.3 | 437.6 | 422.0 | 0.0 | 0.1 | 0.7 |
| 6 | 6 | 0.2 | 0.8 | 2.2 | 437.6 | 422.0 | 0.0 | 0.3 | 0.9 |
| 8 | 2 | 0.1 | 0.5 | 3.4 | 584.3 | 521.9 | 0.1 | 0.7 | 5.0 |
| 8 | 4 | 0.1 | 0.8 | 3.8 | 531.6 | 484.0 | 0.1 | 0.4 | 1.3 |
| 8 | 6 | 0.1 | 1.3 | 4.8 | 531.6 | 484.0 | 0.1 | 0.6 | 2.8 |
| 10 | 2 | 0.4 | 3.9 | 44.3 | 849.3 | 818.3 | 0.1 | 0.9 | 16.1 |
| 10 | 4 | 0.4 | 1.7 | 4.5 | 716.1 | 666.2 | 0.2 | 1.2 | 13.6 |
| 10 | 6 | 0.6 | 1.7 | 9.1 | 716.1 | 666.2 | 0.2 | 1.9 | 18.7 |
| 12 | 2 | 0.5 | 6.9 | 34.3 | 998.2 | 965.5 | 0.3 | 6.2 | 77.9 |
| 12 | 4 | 0.6 | 1.7 | 6.5 | 865.1 | 813.2 | 0.2 | 3.0 | 15.3 |
| 12 | 6 | 0.9 | 2.6 | 10.9 | 865.1 | 813.2 | 0.4 | 3.2 | 22.7 |
| 14 | 2 | 0.8 | 13.0 | 80.5 | 1187.3 | 1136.9 | 0.6 | 16.0 | 3757.5 |
| 14 | 4 | 0.8 | 3.3 | 10.2 | 955.1 | 911.8 | 0.4 | 2.9 | 37.5 |
| 14 | 6 | 0.4 | 3.2 | 14.6 | 948.6 | 898.3 | 0.5 | 3.9 | 63.9 |
| 16 | 2 | 1.0 | 12.8 | 1763.4 | 1321.4 | 1276.7 | 0,5 | 14,3 | 4939,1 |
| 16 | 4 | 1.8 | 5.6 | 88.3 | 1071.8 | 1022.0 | 0,6 | 4,2 | 31,0 |
| 16 | 6 | 1.7 | 5.9 | 31.1 | 1039.1 | 982.5 | 0,6 | 4,6 | 41,4 |

3. IBM ILOG CPLEX. User's Manual, 2017. Online available at short URL `https://goo.gl/TpwkRq`; retrieved 5.7.2017.
4. Q. Jiang, R. Sarker, and H. Abbass. Tracking moving targets and the non-stationary traveling salesman problem. *Compl. Internat.*, 11:171–179, 2005.
5. P. Jindal, A. Kumar, and S. Kumar. Multiple Target Intercepting Traveling Salesman Problem. *Intern. J. on Comp. Sc. and Techn.*, 2(2):327–331, 2011.
6. G. Laporte, H. Mercure, and Y. Nobert. Generalized travelling salesman problem through n sets of nodes: the asymmetrical case. *DAM*, 18(2):185–197, 1987.
7. C.H. Liu. The Moving-Target Traveling Salesman Problem with Resupply. Technical report, The National Chung Cheng University Library, 2013. Online available at URL `http://ccur.lib.ccu.edu.tw/handle/987654321/7877`; retrieved 5.7.2017.
8. C.E. Noon and J.C. Bean. A lagrangian based approach for the asymmetric generalized traveling salesman problem. *Operations Research*, 39(4):623–632, 1991.
9. A. Stieber, A. Fügenschuh, M. Epp, M. Knapp, and H. Rothe. The multiple traveling salesmen problem with moving targets. *Opt. Let.*, 9(8):1569–1583, 2014.
10. K. Sundar and S. Rathinam. Generalized multiple depot traveling salesmen problem - polyhedral study and exact algorithm. *COR*, 70:39–55, 2016.