Angewandte Mathematik und Optimierung Schriftenreihe
Applied Mathematics and Optimization Series
AMOS # 18(2014)

Anke Stieber, Armin Fügenschuh, Maria Epp, Matthias Knapp, and Hendrik Rothe

The Multiple Traveling Salesmen Problem with Moving Targets

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

**Anke Stieber · Armin Fügenschuh ·
Maria Epp · Matthias Knapp · Hendrik
Rothe**

Abstract The multiple weapons to multiple targets assignment problem (MWMTAP) can be seen as a multiple traveling salesmen problem with moving targets (MTSPMT), where the weapons play the role of the salesmen, and the cities to be visited are the targets. Approaches in the literature for the MTSPMT include complexity results and approximation algorithms, where additional restrictions on the targets' trajectories and velocities are imposed. Our approach is based on a discretization of time, which leads to large-scale integer linear programming (ILP) problems, that need to be solved in very short time. Our computational studies on a set of test problems demonstrate, how many moving targets and salesmen can be handled, and how much solutions from a fast and simple first come, first served heuristic can be improved.

A. Stieber*
Tel.: +49-40-6541-2755
E-mail: anke.stieber@hsu-hh.de

A. Fügenschuh*
Tel.: +49-40-6541-3540
E-mail: fuegenschuh@hsu-hh.de

M. Epp*
Tel.: +49-40-6541-3351
E-mail: maria.epp@hsu-hh.de

M. Knapp*
Tel.: +49-40-6541-3351
E-mail: matthias.knapp@hsu-hh.de

H. Rothe*
Tel.: +49-40-6541-2723
E-mail: rothe@hsu-hh.de

*Helmut Schmidt University / University of the Federal Armed Forces Hamburg
Holstenhofweg 85, 22043 Hamburg, Germany

Keywords traveling salesmen problem · moving targets · integer linear programming · heuristics

1 Introduction

We consider the following multiple weapons to multiple targets assignment problem (MWMTAP). An area (for example, a military base or an airport) must be protected from hostile rocket, artillery, or mortar (RAM) fire. Simultaneous attacks from different firing positions can be considered. The flight time for a mortar shell is approximately 30 seconds, whereas the first 5-10 seconds are needed to radar-detect the shell and estimate its trajectory. Based on the estimated impact point the decision must be taken whether it needs to be destroyed. We assume that a battery of laser guns is deployed, where the guns are scattered over (or near) the protected area. Then it has to be decided which of the available guns to select for the countermeasure. The selected laser then aims at the target for a certain period of time to destroy it. The further away a target is, the longer a laser has to fire at the target to safely destroy it. For reasons of simplification we assume that all targets can be destroyed with the same amount of energy (each laser fires the same period of time) and we set it to one unit. Here it needs to be taken into account that a single laser gun cannot fire arbitrarily often without recharging the electric fuel cells. In our problem, the laser guns are electrified in a decentralized way, and the energy surplus at one laser gun cannot be shifted to another laser gun without a temporary shortage. Besides power issues, one tries to assign the closest laser gun to a target, where "close" does not refer to the physical distance, but to the angle the laser gun needs to traverse for aiming at the flying mortar shell. Furthermore, due to safety requirements, a laser should not shoot across the protected area. However, this rule will be neglected, if the destruction of the shell is impossible otherwise. The goal is to destroy all incoming shells, preferably as early as possible (i.e., in minimum time). For more details on the application we refer to Knapp and Rothe [10].

From a mathematical point, the MWMTAP is an online optimization problem, meaning that the complete data of the problem instance is not given in advance. A decision has to be made immediately in an absence of coming events. In our study we initially solve it as an offline problem. Having developed a solution algorithm for the offline problem, it can be adapted to solve the online problem by a moving horizon approach, that is, the data is introduced to the (offline) algorithm at runtime, and the algorithm now needs to decide how to insert the new data to the already constructed solution, which means that this solution can be partially reverted.

In literature only special cases of the MWMTAP are addressed and often restrictions on the movement of the targets are imposed as stated in Section 2. Our approach presented here does not impose such restrictions. We use a discretization of time by introducing a (small) time step. Then we are able to formulate the problem as a mixed-integer linear programming problem on a time-expanded graph. This formulation is of enormous size (in terms of number of variables and constraints), but still modern standard integer linear programming (ILP) solvers (such as IBM ILOG CPLEX or Gurobi) can deal with them to some extent. Furthermore we give heuristic solutions to the problem describing two variants of a first come, first served assignment strategy and a third parametrized version combining both variants. Computational results on a test set of problems show how much an ILP solver can improve the heuristic solutions within a real-time setting. Here we impose a time limit of 3 seconds, and report the ILP gap of the best available solution compared to the respective lower bound. As a more theoretical approach, we further let the solver run until a proven global optimal solution is computed. For larger instances, this would take up to several hours. Although not practically meaningful today, it reveals the potential of faster computers and more advanced solution techniques on this problem.

2 Literatur

The offline variant of the MWMTAP belongs to the class of NP hard problems, since it contains other well-known NP complete problems as subproblems. For instance, the case of a single laser gun and multiple shells that do not move in the sky can be seen as an instance of the classical traveling salesman problem (TSP), which is known to be theoretically difficult, see Garey and Johnson [3]. For a survey on the TSP we refer to Lawler et al. [11] or Reinelt [14].

The case of a single laser gun and multiple moving shells corresponds to a problem that was more recently described in the literature as the “traveling salesman problem with moving targets” (TSPMT). Several variants of this problem are addressed in Helvig et al. [5,6]. The authors only consider targets moving with constant velocity. An approximate algorithm is given in case the number of targets is sufficiently small and for the one dimensional TSPMT. For the TSPMT with resupply, where salesmen must return to the origin for resupply after intercepting each target an exact algorithm is proposed assuming the targets move to the origin and are far away from the origin or move slowly. The authors also describe a variant with multiple salesmen at the same maximum speed. An exact algorithm is given for the case when all targets have the same speed. However, these algorithms are not applicable to the general case of TSPMT.

Jiang et al. [8] present a solution approach based on genetic algorithms for the TSPMT with a fixed number of targets (cities), that means all targets are visible during the whole considered time horizon. Further assumptions are made to the movement of the targets, such as constant velocities and a restriction to the two dimensional space. The authors develop a genetic algorithm and compare two different crossover operators on a randomly generated test set.

Jindal et al. [9] also address a special case of the TSPMT. They impose the following assumptions. All targets start from their starting position and move away from an origin, they traverse on a straight line (one dimension) and with constant velocity. The authors propose an algorithm for this one dimensional case of TSPMT and consider two distinct objective functions, minimum total time and minimum total distance. The algorithm is evaluated with test sets of two and three targets.

The TSPMT with resupply, where salesmen must return to the origin after intercepting each target, is also a common application in the literature. Liu [12] discusses this case of the TSPMT with the restriction, that all targets move towards the origin with constant velocities. Additionally, an algorithm is proposed, where one target traverses away from the origin and has positive acceleration.

To solve the offline version of the MWMTAP we present a computational approach that does not impose restrictions on the movement of the targets. The moving targets can move in three dimensions, and also with varying acceleration and on arbitrary trajectories (although parabola-shaped trajectories would be sufficient for our application). The targets also do not necessarily need to travel from or to the same origin or destination, as in some of the existing approaches. Furthermore, we can handle the case of one as well as several salesmen (i.e., laser guns).

3 A Mathematical Model

We introduce some notation to formulate the MTSPMT as an optimization problem. We assume a finite time horizon $[0, T]$. Let $\mathcal{V} = \{1, \ldots, n\}$ be a set of nodes (cities in the MTSPMT, targets in the MWMTAP) and $\mathcal{A} \subseteq \mathcal{V} \times \mathcal{V}$ be a set of arcs (roads in the MTSPMT, gun movements in the MWMTAP). Let $\mathcal{W} = \{1, \ldots, w\}$ be a set of salesmen (weapons in the MWMTAP). The time (or distance) for traveling from node i to node j for salesman k at time t is given by the function $c_{i,j,k} : [0, T] \to \mathbb{R}_+ \cup \{\infty\}$. That is, $c_{i,j,k}(t)$ denotes the travel time (distance) between i and j when salesman k arrives in j at time t. In the MWMTAP the targets are only visible within a certain time window, and thus we have $c_{i,j,k}(t) = \infty$ for all t outside this window. The goal is to assign exactly one salesman from $\mathcal{W}$ to each node from $\mathcal{V}$ such that the sum of all traveled distances of all salesmen (that is, the movements of the weapons for aiming at the targets) is minimal.

To give a time-continuous formulation of the MTSPMT we introduce binary tour assignment variables $x_{i,j,k} \in \{0, 1\}$ and tour flow variables $t_{i,j,k} \in [0, T]$ as used by Gavish and Graves [4]. Here $x_{i,j,k} = 1$ describes the decision of sending salesman k from i to j within the considered time horizon, while $t_{i,j,k}$ represents the arrival time of k traveling from i to j. In addition continuous variables $\bar{t}_{i,k} \in [0, T]$ are introduced to model the time, at which salesman k ends his tour. Then this problem is formulated as a bi-level flow problem, with a flow of the salesmen and a flow of the arrival times. Both flows are combined using a capacity constraint.

The objective function is to minimize the total distances of all salesmen (total movements of all lasers):

$$\sum_{k\in\mathcal{W}}\sum_{(i,j)\in\mathcal{A}} c_{i,j,k}(t_{i,j,k})x_{i,j,k} \quad \rightarrow \quad \min. \tag{1}$$

The following constraints are derived from the application. Each node must be visited exactly once by exactly one salesman. Therefore we formulate the constraint that each node must be entered once:

$$\sum_{k\in\mathcal{W}}\sum_{i:(i,j)\in\mathcal{A}} x_{i,j,k} = 1, \quad \forall\, j\in\mathcal{V}. \tag{2}$$

Each salesman can do at most one tour:

$$\sum_{j\in\mathcal{V}} x_{i,j,k} \leq 1, \quad \forall\, i\in\mathcal{V}, k\in\mathcal{W}. \tag{3}$$

The following constraint ensures the flow conservation of the salesmen flow. Each salesman continues his tour after arriving at a node or he ends his tour, then the node can be regarded as the sink of the salesman flow:

$$\sum_{i\in\mathcal{V}} x_{i,j,k} \geq \sum_{i\in\mathcal{V}} x_{j,i,k} \quad \forall\, j\in\mathcal{V}, k\in\mathcal{W}. \tag{4}$$

In order to couple both flows the following capacity restriction is given. If salesman k does not travel from i to j, the corresponding arrival time cannot take a positive value:

$$t_{i,j,k} \leq T\cdot x_{i,j,k} \quad \forall\,(i,j)\in\mathcal{A}, k\in\mathcal{W}. \tag{5}$$

The flow conservation of the arrival time flow ensures that for each salesman the arrival time at a node j of the tour and the travel time from j to the next node i cannot be greater than the real arrival time at i. In fact it may be less since waiting is permitted for all $j\in\mathcal{V}$. If salesman k finishes the tour at j the inequality reduces to the fact, that the arrival time in j cannot be greater than the finishing time $\bar{t}_{j,k}$ of the tour. $\bar{t}_{j,k}$ takes positive values if and only if k ends the tour in j. These constraints are formulated as follows:

$$\sum_{i\in\mathcal{V}} t_{i,j,k} + \sum_{i\in\mathcal{V}} c_{j,i,k}(t_{j,i,k})x_{j,i,k} \leq \sum_{i\in\mathcal{V}} t_{j,i,k} + \bar{t}_{j,k} \quad \forall\, j\in\mathcal{V}, k\in\mathcal{W}. \tag{6}$$

$$\bar{t}_{j,k} \leq T\cdot\left(1-\sum_{i\in V} x_{j,i,k}\right) \quad \forall\, j\in\mathcal{V}, k\in\mathcal{W} \tag{7}$$

Summing up, we aim to solve the following optimization problem:

$$\begin{aligned}\min\{(1)\mid\ & (2),(3),(4),(5),(6),(7),\\ & x\in\{0,1\}^{\mathcal{A}\times\mathcal{W}}, t\in[0,T]^{\mathcal{A}\times\mathcal{W}}, \bar{t}\in[0,T]^{\mathcal{V}\times\mathcal{W}}\}.\end{aligned} \tag{8}$$

The presented time-continuous formulation of the MTSPMT is complex and no appropriate solver is available. In order to apply a standard ILP solver we give an ILP formulation of the MTSPMT. Therefore we introduce a time discretization. To this end, let m be an integer number. Then the step size is defined by $\mathit{\Delta} := T/m$. The set of all time steps is $\mathcal{T} := \{0, \dots, m\}$. The distance function c is only evaluated at the discrete points in time defined by $t_\ell := \ell \cdot \mathit{\Delta}$. As abbreviation we set $c^\ell_{i,j,k} := c_{i,j,k}(t_\ell)$. The travel time is aligned to discrete time steps by $\delta^\ell_{i,j,k} := \lceil c^\ell_{i,j,k}/\mathit{\Delta} \rceil$. To refer to the energy consumption restriction denote by L_k the number of nodes that can be visited within each h_k consecutive time steps by $k \in \mathcal{W}$.

We introduce a family of binary decision variables $x^\ell_{i,j,k} \in \{0,1\}$. Here $x^\ell_{i,j,k} = 1$ shall represent the decision of sending traveling salesman k from i to j, arriving in j at time step ℓ (i.e., at time t_ℓ).

The objective is to minimize the total distance for all movements of all salesmen:

$$\sum_{k \in \mathcal{W}} \sum_{(i,j) \in \mathcal{A}} \sum_{\ell \in \mathcal{T}} c^\ell_{i,j,k} x^\ell_{i,j,k} \quad \to \quad \min. \tag{9}$$

As in the formulation above, each node must be visited exactly once by exactly one salesman:

$$\sum_{k \in \mathcal{W}} \sum_{i:(i,j) \in \mathcal{A}} \sum_{\ell \in \mathcal{T}} x^\ell_{i,j,k} = 1, \quad \forall j \in \mathcal{V}. \tag{10}$$

The following flow conservation constraints ensure the compatibility of the time flow, where each node at which a salesman ends his tour can be regarded as the sink of the flow:

$$\sum_{i:(i,j) \in \mathcal{A}} x^\ell_{i,j,k} \geq \sum_{i:(j,i) \in \mathcal{A}} x^{\ell + \delta^\ell_{j,i,k}}_{j,i,k}, \quad \forall j \in \mathcal{V}, k \in \mathcal{W}, \ell \in \mathcal{T}. \tag{11}$$

At each point in time, each salesman can do at most one trip:

$$\sum_{(i,j) \in \mathcal{A}} x^\ell_{i,j,k} \leq 1, \quad \forall k \in \mathcal{W}, \ell \in \mathcal{T}. \tag{12}$$

If energy consumption is crucial, then we can limit the number of visits per salesman within a certain time period:

$$\sum_{(i,j) \in \mathcal{A}} \sum_{\ell = t}^{t + h_k} x^\ell_{i,j,k} \leq L_k, \quad \forall k \in \mathcal{W}, t \in \mathcal{T}, t \leq |\mathcal{T}| - h_k. \tag{13}$$

Summing up, we aim to solve the following optimization problem:

$$\min\{(9) \mid (10), (11), (12), (13), x \in \{0,1\}^{\mathcal{A} \times \mathcal{W} \times \mathcal{T}}\}. \tag{14}$$

This problem is NP-hard, since it is a generalization of the classical TSP, which is NP-hard. NP-hardness means that the corresponding decision problem is NP-complete. It is unlikely that a polynomial-time exact algorithm can be developed to solve (14).

4 Heuristic Solutions to the Model

We solve model (14) by addressing it from two sides. First, we develop a primal heuristic that assigns targets to salesmen on a first come, first served basis. Then we compute a dual bound by solving the linear programming relaxation of (14). The dual bound is further improved by adding cutting planes and a branch-and-bound procedure. The primal solution here serves as a first initial upper bound, which is replaced, in case further integer feasible solutions with better objective values are found. For more details on the general solution process we refer to Nemhauser and Wolsey [13] or to the survey of Fügenschuh and Martin [2]. Our heuristic assigns targets to the salesmen on a first come, first served (or first-in, first-out, FIFO) strategy. We implemented this basic strategy in three variants.

Least time. Each target is defined by a lower and an upper bound in time. It has to be visited by exactly one of the salesmen in that time period. From the list of unassigned targets we select the one that emerges first. Then we compute the distances and travel times from all salesmen to that target. We assign the target to that particular salesman that is able to be at the target at the earliest possible point in time, even if he needs to travel a longer distance compared to other salesmen nearby. Additionally, this particular salesman is not allowed to violate the energy consumption restriction. If this regularization is fulfilled the target is assigned to this salesman. As long as the salesman has not reached his destination target he is prohibited from further assignments.

Least distance. In this variant, we select the target that emerges earliest in each step of the heuristic. We then check for each salesmen if he can intercept the target at all, and at what point in time he would have to travel the least distance to meet the target. We then assign the target to that particular salesman that has to travel the least distance and fulfills the energy consumption restriction. In this case the salesmen are allowed to wait to use the least distance to catch the target rather than intercept it as early as possible.

Since both heuristics are very fast, we can allow to run them both. It turns out that the first strategy (least time) is more robust in terms that it more often finds a feasible solution. The second strategy (least distance) in several cases is not able to find a feasible solution. It focuses on the traveled distances too much, so that the salesmen wait too long to travel to a target with least distance and, as a result, are not able to catch targets, that emerges only slightly later. However, if it does find a feasible solution, it in principle is better (objective function value) than the one found by the 'least time' strategy. The objective function values of both heuristic variants are computed as the sum of traveled distances of all salesmen. Obviously, the distance variant is advantageous and hence results in better starting solutions.

To overcome the drawbacks of both heuristics we combine both variants to generate a third heuristic, which is robust and yields good starting solutions.

We applied a parametrized heuristic described by Fügenschuh [1]. This method is based on a parametrized scoring function, that is linear in the parameters.

Parametrized heuristic. In each iteration we select the unassigned target, that emerges first. Then for each available salesman it is checked if the target can be reached and for which time steps. The parametrized scoring function $s_{i,j}$ is a mapping $s_{i,j} : \mathbb{Q}^2 \to \mathbb{Q}$, where $i \in \mathcal{W}$ is a salesman and $j \in \mathcal{V}$ is the selected target. In each step of the heuristic the salesman, who fulfills the energy consumption restriction and minimizes the parametrized scoring function

$$s_{i,j}(\lambda_1, \lambda_2) := \lambda_1 d_{i,j}^{t_l} + \lambda_2 t_l, \tag{15}$$

where $d_{i,j}^{t_l}$ denotes the distance from the current position of salesman i to the location of target j at time t_l, is assigned to the selected target. The first part of the scoring function stems from the heuristic variant based on least distances while the second part is taken from the least time heuristic.

Since the local selection of variables and hence the entire heuristic solution depends on the choice of $\lambda \in \mathbb{Q}^2$, we have to find an appropriate vector λ to yield a reasonable good heuristic. As we are looking for a robust and good heuristic λ describes a compromise between the first two variants. Let $z(\lambda)$ be the objective function value for the heuristic solution with parameter vector λ. Now we are faced with the problem to find a λ with $z(\lambda) \leq z(\mu)$ for all $\mu \in \mathbb{Q}^2$. In [1] it is shown that the parameter domain is bounded to the surface of the unit sphere (in our case to the circumference of the unit circle) in the euclidean space. Since a straightforward or randomized sampling over the surface turns out to be inefficient the author used an algorithm called *improving hit-and-run* (IHR for short), to find the λ vector. IHR was introduced by Zabinsky et al. [15] and is a randomized Monte-Carlo algorithm for global optimization problems. It is proven that for the class of positive definite quadratic programs the expected number of function evaluations is polynomial ($O(n^{5/2})$).

We also applied this algorithm to determine the weights λ controlling the parametrized scoring function. Within this procedure the third heuristic is called to obtain the corresponding objective function value $z(\lambda)$. The basic steps executed by the IHR are the following: In the first iteration $k := 0$ an initial λ_0 is chosen and the corresponding $z(\lambda_0)$ is computed. The next step is to generate a new candidate point w_{k+1} on the boundary of the unit circle. Therefore we randomly choose a direction d_k on the circle surface, which means "go right" or "go left". With a random selection of a distance $t_k > 0$ we obtain the candidate point $w_{k+1} := \lambda_k + t_k \cdot d_k$ so that w_{k+1} lies on the part of the circle boundary chosen by d_k. If the candidate point is improving, i.e. $z(w_{k+1}) < z(\lambda_k)$ then $\lambda_{k+1} := w_{k+1}$, otherwise $\lambda_{k+1} := \lambda_k$. The steps are repeated until a stopping criterion is met.

The best solution (objective function) of the first two described heuristic variants and the parametrized one with IHR is then used as the starting solution in the branch-and-bound procedure. Having a feasible solution early in the branch-and-bound search helps to reduce the overall solution time. Any such feasible solution is an upper bound on the optimal value of (14). When the objective function at some node of the branching tree is already greater or equal than an available feasible solution, the tree can be pruned at that node, see [13] or [2].

5 Computational Results

We applied our method described above to a set of randomly generated test problems. We created a total of 36 instances of different sizes, which can be found in Table 1. An instance is generated by specifying the number of lasers/salesmen (column 'ns' in Table 1) and the number of targets (column 'nt'). Motivated by our application, we chose values between 1 and 3 for the salesmen and 6 to 24 for the moving targets. The targets are assumed to have a constant traveling speed (column 'st'). The salesmen have a maximum traveling speed (column 'ss'), and their actual traveling speed can be any value less or equal to the maximum speed. In particular, waiting is also permitted for the salesmen. We tested three different combinations of target and salesmen speed: 20-200 (slow targets, fast salesmen), 40-100 (slightly faster targets, slightly slower salesmen), and 60-60 (targets and salesmen traveling at the same speed). The values refer to the number of length units that are traversed per time unit. In all instances, the salesmen start their tours in an initial “home” location, based at the center of the operating space, which is a square of size 500 length units. Due to visualization reasons the trajectories of the targets are straight lines in a two dimensional space of random lengths between 100 and 400 length units. The method, however, is capable to handle also non-linear trajectories in a three dimensional space.

The size of the resulting ILP models is shown in the next three columns of Table 1: 'var' refers to the number of (binary) decision variables, 'cons' shows the number of linear constraints, and 'nz' gives the number of nonzero entries in the constraint matrix of the problem. We apply the three variants of the first come, first served heuristic assignment strategy to all of our test instances. If we get more than one feasible solution for an instance, we will take the one with the lowest objective function value. For the parametrized heuristic, we applied the IHR algorithm, which evaluates the parametrized scoring function in each iteration. As stopping criterion we set a time limit of one second. Depending on the selected problem instance from our test set the number of IHR iterations ranges from a size of $3 \cdot 10^5$ to $4 \cdot 10^6$. To find the best incumbent solution of this heuristic, by far less iterations are needed. For half of the 36 instances the solution was found within the first 100 iterations. The highest iteration number has a size of 10^4, that means the number of iterations can be reduced at least to this size. The resulting components of the vector λ for all test instances are visualized in Figure 1. Most of the values concentrate in the top center of the picture, where $\lambda_1 \approx 0.05$ and $\lambda_2 \approx 1.0$. Nevertheless, several instances have their best values significantly deviating from this point. We apply all three heuristic variants to the test set of 36

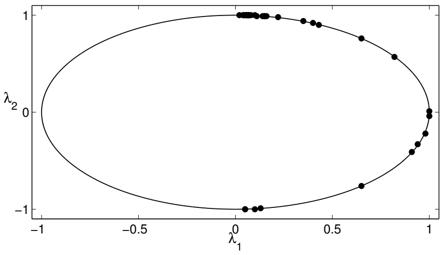

Fig. 1 IHR result of the vector λ.

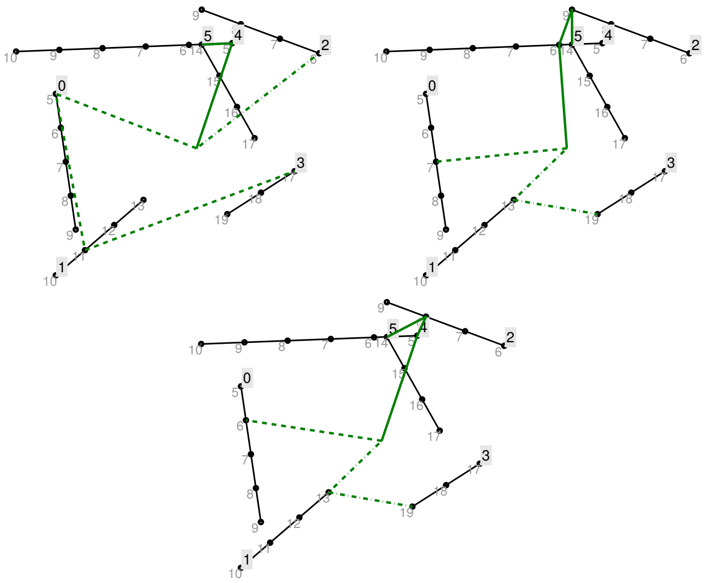

Fig. 2 Instance 27. First: Least time heuristic solution, second: Least distance heuristic solution, third: Parametrized heuristic solution.

problem instances. It turns out that the parametrized heuristic yields the best incumbent solutions with respect to the objective function for all test instances. To directly contrast all three variants, we use instance 27 as one example MTSPMT instance to compare the three solutions. Figure 2 visualizes the heuristic solutions for this instance consisting of 3 salesmen (maximum speed: 60) and 6 targets (speed: 60). The trajectories are presented as straight lines with points for each time step (the corresponding time values are given in grey numbers). All salesmen start their tours in the center and the different tours are visualized by a solid, dashed and dash-dotted green line respectively. The corresponding objective function values are 1288.86 (least time), 726.18 (least distance) and 718.91 (parametrized). The sum of traveled distances in the solution of the parametrized heuristic is only slightly better than in the least distance solution. It can be seen, that the salesman, who intercepts only one target (dashed line) travels a shorter distance in the second figure, but the salesman, who intercepts three targets (solid line) moves a shorter distance in the third graphic. The decision of the latter salesman was more influenced by the least time variant and results in a better objective function value.

The values of all three heuristics are given in the columns 'fcfsT' (least time), 'fcfsD' (least distance) and 'fcfsP' (parametrized) of Table 1. The resulting value used as the starting MIP value is shown in column 'fcfs' of Table 2. In all of our 36 instances the IHR algorithm with the parametrized heuristic variant provides the minimum objective function values.

There are no CPU times reported for the heuristic runs, because the least time and least distance heuristics solve each problem in virtually no time and for the Parametrized variant we set the time limit to one second. Our computational experiments were carried out on a Mac mini computer running the MacOS operating system with an Intel Core i7 running at 2.6 GHz on 4 cores, 6 MB L3 cache, and 16 GB 1600 MHz DDR3 RAM. As solver for the ILPs we used CPLEX 12.5.1 ([7]). The relative duality gap (i.e., the difference between upper and lower bound divided by the primal bound) was set to 0.0%. Other than that we used the solver's default settings.

We use the heuristics' results as starter for the ILP branch-and-cut process. We solve each model twice, once with a three seconds time limit and once with unlimited time. The objective function value of the best solution found after three seconds is given in column 'sol3s' of Table 2, and the remaining ILP gap can be found in column 'gap3s'; its value is computed as gap3s $=$ (sol3s $-$ lb3s)/sol3s $\cdot$ 100%, where lb3s is the dual (lower) bound value after three seconds. The exact time in seconds used to solve the instances with a limit of three seconds is given in column 'et3s'. As one can see, in 33 of the 36 instances, the ILP solver was able to find a better solution that requires less movements from the salesmen. In 30 out of 36 instances, the solver was even able to prove that its solutions were indeed global optimal. Often the solver finds the global optimal solution relatively fast, but it takes a long time to prove optimality.

For curiosity reasons, we raised the time limit to infinity and solved each instance to proven optimality. The objective function values of the global optimal solutions are shown in column 'opt' of Table 2, and the corresponding computation times can be found in column 'time'. It turns out that those instances where the targets are slow and the salesmen are fast are particularly difficult for the ILP solver. These instances have the longest total time horizon, and due to the potential speed of the salesmen the largest amount of possible tours for the salesmen. This results in problem instances with a high number of binary variables, and leaves a higher burden in finding an optimal combination to the ILP solver.

As far as energy consumption is concerned all salesmen act identically. To test the behavior of our instances according to the restriction we applied different energy consumption parameters L_k and h_k (limit of nodes L_k, that can be visited by each salesman in time period h_k). According to our application

Table 1 Instances, problem sizes and heuristic solutions.

inst	ns	nt	st	ss	var	cons	nz	fcfsT	fcfsD	fcfsP
1	1	6	20	200	1481	98	12677	1430.23	failed	1319.33
2	1	6	40	100	809	97	9599	1398.48	1045.43	970.90
3	1	6	60	60	299	128	4474	1462.78	1198.78	1198.78
4	1	12	20	200	8546	212	137751	3171.25	failed	2189.07
5	1	12	40	100	2285	209	39649	3645.92	2744.64	2712.51
6	1	12	60	60	1673	294	35572	3368.57	failed	2985.78
7	1	18	20	200	16331	309	347228	5411.92	failed	4314.78
8	1	18	40	100	6759	346	140895	6493.76	3702.97	3689.38
9	1	18	60	60	3280	421	70909	5260.21	3923.58	3765.81
10	1	24	20	200	27731	419	575744	7693.21	failed	6516.21
11	1	24	40	100	9508	421	203828	7321.97	5547.79	5407.57
12	1	24	60	60	6057	598	139795	7254.23	failed	5018.45
13	2	6	20	200	2878	168	11394	1257.68	675.75	618.14
14	2	6	40	100	1382	136	5444	1509.01	702.78	702.78
15	2	6	60	60	510	130	4602	1573.27	797.56	720.60
16	2	12	20	200	16226	364	184782	3428.24	failed	1825.18
17	2	12	40	100	4172	262	58888	3704.98	1504.72	1425.74
18	2	12	60	60	3104	352	61810	3101.40	failed	1907.65
19	2	18	20	200	31670	508	507976	3923.79	2778.70	2679.61
20	2	18	40	100	12940	482	265290	5031.06	2270.65	2236.21
21	2	18	60	60	6232	504	130058	4271.83	2240.77	2240.77
22	2	24	20	200	53952	662	926338	6195.71	failed	4118.06
23	2	24	40	100	18364	550	363994	7045.67	4001.87	3587.76
24	2	24	60	60	11648	692	254872	6184.46	3927.99	3532.10
25	3	6	20	200	4323	249	17115	902.98	599.16	599.16
26	3	6	40	100	1551	180	6087	967.57	614.81	575.34
27	3	6	60	60	537	144	2070	1288.86	726.18	718.91
28	3	12	20	200	23808	504	142068	2764.59	failed	1408.41
29	3	12	40	100	6033	357	84285	2666.85	1455.93	1186.11
30	3	12	60	60	4455	414	76065	2632.83	1236.92	1236.92
31	3	18	20	200	45963	693	505197	3471.88	2279.59	1872.53
32	3	18	40	100	18591	618	367719	4964.80	1830.64	1784.35
33	3	18	60	60	8877	591	180324	4274.46	2419.63	2323.75
34	3	24	20	200	78555	903	1187280	6006.59	failed	3549.14
35	3	24	40	100	26727	705	507111	6653.59	2369.77	2359.37
36	3	24	60	60	17166	840	358611	6080.76	2478.73	2277.00

this means, that a laser gun k can destroy at most L_k targets in each time period h_k, the free time is used to recharge the battery. Table 3 shows the objective function values and the corresponding computing times for two selected instances. It can be seen, that for small values of L_k and high values of h_k in most cases the instances are getting more difficult and their computing times rise. In the case $L_k = 6$ and $h_k = 6$ the energy consumption restriction is redundant. The corresponding objective function value is equal to the objective function value of the problem without consideration of energy consumption. For our computations we set the limit of nodes $L_k = 5$ and the time period $h_k = 9$. For laser guns, that are currently used in the described application a 15 minutes recharging time is needed after 60 seconds of firing.

For one of the instances in our test set, we graphically compare the solutions found by the best heuristic and the exact method. The upper image in Figure 3

Table 2 Instances, start heuristic and exact solutions.

inst	ns	nt	st	ss	fcfs	sol3s	gap3s	et3s	opt	time
1	1	6	20	200	1319.32	905.60	0.00	0.15	905.60	0.14
2	1	6	40	100	970.90	962.86	0.00	0.02	962.86	0.02
3	1	6	60	60	1198.77	1142.19	0.00	0.01	1142.19	0.01
4	1	12	20	200	2189.06	1828.25	0.00	2.66	1828.25	2.92
5	1	12	40	100	2712.50	2664.37	0.00	0.07	2664.37	0.08
6	1	12	60	60	2985.78	2729.20	0.00	0.06	2729.20	0.06
7	1	18	20	200	4314.77	3185.62	23.02	3.02	3004.44	13.93
8	1	18	40	100	3689.38	3503.40	0.00	0.26	3503.40	0.27
9	1	18	60	60	3765.81	3652.00	0.00	0.12	3652.00	0.12
10	1	24	20	200	6516.21	6516.21	41.81	3.03	4403.24	16.96
11	1	24	40	100	5407.57	5174.88	0.00	0.45	5174.88	0.49
12	1	24	60	60	5018.44	4703.86	0.00	0.22	4703.86	0.22
13	2	6	20	200	618.13	576.63	0.00	0.07	576.63	0.07
14	2	6	40	100	702.77	697.70	0.00	0.03	697.70	0.03
15	2	6	60	60	720.59	634.96	0.00	0.02	634.96	0.02
16	2	12	20	200	1825.18	1100.37	0.00	2.34	1100.37	2.35
17	2	12	40	100	1425.73	1315.64	0.00	0.08	1315.64	0.08
18	2	12	60	60	1907.65	1307.79	0.00	0.04	1307.79	0.04
19	2	18	20	200	2679.60	1997.15	32.02	3.03	1578.16	20.41
20	2	18	40	100	2236.20	1584.84	0.00	0.34	1584.84	0.44
21	2	18	60	60	2240.76	2108.52	0.00	0.14	2108.52	0.14
22	2	24	20	200	4118.05	4118.05	47.68	3.03	2615.41	255.30
23	2	24	40	100	3587.76	2890.91	0.00	1.88	2890.91	1.86
24	2	24	60	60	3532.10	3147.53	0.00	0.71	3147.53	0.66
25	3	6	20	200	599.16	516.44	0.00	0.07	516.44	0.07
26	3	6	40	100	575.33	570.61	0.00	0.03	570.61	0.02
27	3	6	60	60	718.91	668.74	0.00	0.02	668.74	0.01
28	3	12	20	200	1408.40	1099.27	0.00	2.17	1099.27	2.14
29	3	12	40	100	1186.10	1111.46	0.00	0.19	1111.46	0.18
30	3	12	60	60	1236.92	1058.12	0.00	0.06	1058.12	0.06
31	3	18	20	200	1872.52	1425.85	10.15	3.01	1372.44	6.97
32	3	18	40	100	1784.35	1392.47	0.00	0.78	1392.47	0.74
33	3	18	60	60	2323.75	1963.98	0.00	0.44	1963.98	0.45
34	3	24	20	200	3549.14	3549.14	99.99	3.08	1963.97	4714.07
35	3	24	40	100	2359.36	1968.44	0.00	2.46	1968.44	2.47
36	3	24	60	60	2276.99	1785.85	0.00	0.48	1785.85	0.47

Table 3 Objective function values (ofv) and computing times (ct) for the energy consumption restriction with different parameters.

	$h_k = 6$		$h_k = 9$		$h_k = 12$	
inst4	ofv	ct	ofv	ct	ofv	ct
$L_k = 3$	1971.44	(12.49)	2431.70	(1.73)	failed	(1.33)
$L_k = 4$	1808.49	(3.30)	1961.42	(9.31)	2209.87	(2.26)
$L_k = 5$	1788.88	(2.59)	1828.25	(2.74)	1861.00	(2.26)
$L_k = 6$	1788.88	(2.48)	1805.90	(2.07)	1828.25	(3.11)
inst19	ofv	ct	ofv	ct	ofv	ct
$L_k = 3$	1805.80	(578.87)	2056.18	(2786.67)	3278.73	(29.47)
$L_k = 4$	1667.97	(31.19)	1744.93	(170.18)	2002.52	(4374.31)
$L_k = 5$	1563.21	(7.12)	1578.16	(19.39)	1781.50	(1320.31)
$L_k = 6$	1531.42	(11.92)	1554.58	(15.98)	1735.18	(618.54)

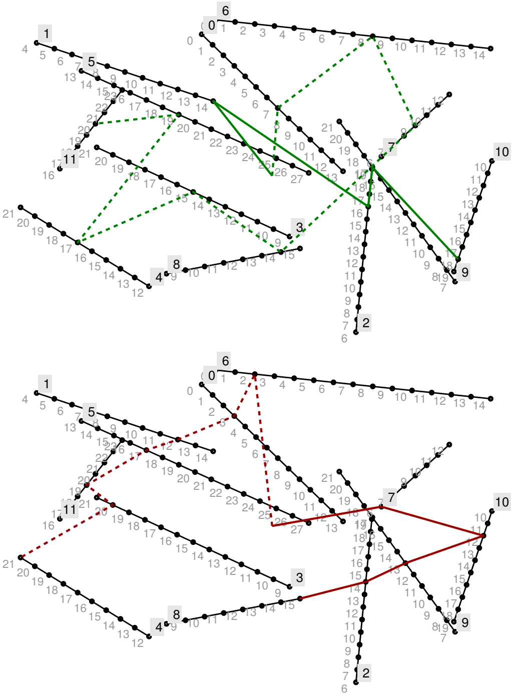

Fig. 3 Instance 16. Top: heuristic solution, bottom: exact solution.

shows the result of the third first come, first served assignment strategy. The two salesmen are initially located at the center of the image, from where they start their tours, visualized as solid and dashed lines. For the lasers in the application that would mean that both lasers are located in the same position, and initially point in the same direction. The tours of the salesmen in these images correspond to angles of the lasers pointing in the sky. Note that our methods can still be applied, if the lasers are installed in different positions and point in different directions. According to Table 2 the heuristic has an objective function value of 1825.18 units. In the lower image of this figure, the global optimal solution is depicted. It was found after 2.34 seconds and has an objective function value of only 1100.37 units. So the necessary movements could be reduced in this particular case by almost 40%.

6 Conclusion

We gave an integer programming formulation for the multiple traveling salesmen problem with moving targets that is based on a discretization of time. It turns out that modern ILP solvers are able to solve instances of relevant size for a weapon-to-target assignment problem in reasonable short time (less than three seconds). The solutions often are already global optimal. In many cases, the best feasible solution that was found after three seconds improves the one that was found by the parametrized first come, first served heuristic with IHR. For future research the model can be extended to include additional restrictions. In our application it could be very useful that more than one laser gun aim at one target to save energy or to destroy the target faster. In addition, security restrictions should be considered, that means the lasers are not allowed to fire across the protected area to prevent damage. In this case only a part of the available lasers are permitted to aim at a target depending on the direction from which the target comes from. This restriction would reduce the number of possible solutions. As a third extension the energy consumption restriction should be mentioned. Different firing times related to different laser-target distances can be included into our model as well.

References

1. Fügenschuh, A.: Parametrized greedy heuristics in theory and practice. In: M. Aguilera, C. Blum, A. Roli, M. Sampels (eds.) Hybrid Metaheuristics, Second International Workshop, HM 2005, Barcelona, pp. 21 – 31 (2005)
2. Fügenschuh, A., Martin, A.: Computational integer programming and cutting planes. In: R.W. K. Aardal G. Nemhauser (ed.) Handbook on Discrete Optimization, Series Handbooks in Operations Research and Management Science, vol. 12, pp. 69 – 122. Elsevier (2005)
3. Garey, M.R., Johnson, D.S.: Computers and Intractability: A Guide to the Theory of NP-Completeness. W. H. Freeman and Company, New York (1979)
4. Gavish, B., Graves, S.: The traveling salesman problem and related problems. Tech. Rep. OR 078-78, Operations Research Center, Massachusetts Institute of Technology (1978)
5. Helvig, C., Robins, G., Zelikovsky, A.: Moving-Target TSP and Related Problems. In: A.P. G. Bilardi G. F. Italiano, G. Pucci (eds.) Proceedings of the European Symposium on Algorithms, *Lecture Notes in Computer Science*, vol. 1461, pp. 453 – 464. Springer Verlag, Berlin (1998)
6. Helvig, C., Robins, G., Zelikovsky, A.: The moving-target traveling salesman problem. Journal of Algorithms **49**(1), 153 – 174 (2003)
7. IBM ILOG CPLEX: Information available at `http://www.ibm.com/software/integration/optimization/cplex/` (2013)
8. Jiang, Q., Sarker, R., Abbass, H.: Tracking moving targets and the non-stationary traveling salesman problem. Complexity International **11**, 171 – 179 (2005)
9. Jindal, P., Kumar, A., Kumar, S.: Multiple Target Intercepting Traveling Salesman Problem. International Journal on Computer Science and Technology **2**(2), 327 – 331 (2011)
10. Knapp, M., Rothe, H.: Concept for Simulating Engagement Strategies for C-RAM Systems using Laser Weapons. In: Proceedings of the DMMS (2012)
11. Lawler, E., Lenstra, J., Rinnooy, A., Shmoys, D.: The Traveling Salesman Problem: a Guided Tour of Combinatorial Optimization. John Wiley and Sons, Chichester, New York (1985)
12. Liu, C.H.: The Moving-Target Traveling Salesman Problem with Resupply. Tech. rep., The National Chung Cheng University Library, http://ccur.lib.ccu.edu.tw/handle/987654321/7877 (2013)
13. Nemhauser, G., Wolsey, L.: Integer and Combinatorial Optimization. Wiley Interscience, New York (1988)
14. Reinelt, G.: The Traveling Salesman: Computational Solutions for TSP Applications. Springer Verlag, Berlin (1994)
15. Zabinsky, Z.B., Smith, R.L., Mcdonald, J.F., Edwin, H., Kaufman, D.E.: Improving hit-and-run for global optimization. Journal of Global Optimization **3**, 171–192 (1993)