ISSN 0005-1179 (print), ISSN 1608-3032 (online), Automation and Remote Control, 2026, Vol. 87, No. 10, pp. 940–951.
© The Author(s), 2026 published by Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, 2026.
Russian Text © The Author(s), 2026, published in Avtomatika i Telemekhanika, 2026, No. 10, pp. 104–118.

Defense Planning for a Moving Target Against Multiple Attackers Using a Defender via the Monte Carlo Tree Search

E. A. Ryabushev[*,a] **and A. A. Galyaev**[*,b]

[*] *Trapeznikov Institute of Control Sciences, Russian Academy of Sciences, Moscow, Russia*
e-mail: [a] *forteamofdota@mail.ru,* [b] *galaev@ipu.ru*

Received June 24, 2026
Revised July 6, 2026
Accepted July 8, 2026

Abstract—The problem of protecting a moving target from an attack by multiple pursuing attackers using an interceptor defender is considered. A rigorous mathematical formulation of the problem is presented, quality criteria for defense plans are introduced, and mathematical properties of the problem are formulated. Based on these properties, heuristic rules and algorithms for their application are proposed in order to construct suboptimal solutions.

Keywords: mathematical optimization, operations research, Monte Carlo tree search (MCTS), local rules

DOI: 10.7868/S1608303226100055

1. INTRODUCTION

The world is undergoing major changes. Drones dominate on land and in the air, while unmanned surface vessels and underwater vehicles pose a danger at sea. Autonomous systems are capable of solving an increasing number of both military and civilian tasks without any human participation. This is most noticeable on the battlefield, where unmanned vehicles have already changed the nature of war and define new rules. Massed drone attacks on infrastructure and logistics facilities represent a significant and urgent problem. It is becoming obvious that economically feasible protection against this threat requires interceptors capable of sequentially destroying attacking drones one after another. An important role here is played by the choice of the best interception order, which is a problem close in meaning to the "dynamic traveling salesman problem" (DTSP) [1, 2] or Moving Targets TSP (MTTSP) [3, 4]. At present, there are no effective methods for the exact solution of DTSP, and the problem itself is at the initial stages of investigation [5]. Modern formulations of DTSP concentrate on conditions of uncertainty in information about the positions of targets and limited time windows for their observation [6, 7].

The traveling salesman problem and, as a consequence, many of its dynamic generalizations are NP-complete [8], which means that there are no effective exact solution algorithms for them. Building a defense requires decision-making under limited time, which necessitates the development of methods for fast suboptimal planning. Algorithms of this kind were already considered by the authors for the case of defending a stationary point in [9]; the goal of the present work is to generalize them to the problem of defending a moving point. For this purpose, a generalized formulation of the problem is proposed that includes both planning the order of visiting targets and constructing an evasion maneuver from attackers threatening the protected moving point. It is proposed to evaluate the quality of plans on the basis of vector criteria generalizing those from [10] for the defense of a static point to the dynamic case. The motion of the moving defense object is planned

on the basis of local rules similar to those used in [11], and the interception order of the attackers is chosen by means of Monte Carlo tree search (MCTS) [12, 13].

MCTS is a combinatorial optimization method that became known after its successful application in the AlphaGo project [12]. Classical MCTS applications assumed game formulations in finite games, but the algorithm has also proven itself for combinatorial optimization in non-game problems [14]. Monte Carlo tree search with PUCT estimates (Predictor + Upper Confidence bounds for Trees) is usually applied together with a neural network and is used for its training. However, MCTS is also capable of operating on the basis of local rules that can evaluate actions instead of a neural network [9]. Such an implementation of the algorithm not only allows it to be applied without preliminary training, but also makes it possible to use it for computing q-estimates for reinforcement learning. Thus, the adaptation of Monte Carlo tree search not only provides a ready-made algorithm for solving the planning problem, but also opens a path for effective training of neural-network models capable of solving this problem.

2. MATHEMATICAL MODEL, PROBLEM STATEMENT, AND SOLUTION METHOD

This section discusses the formulation of the problem of protecting a moving target by an interceptor defender against an attack by multiple attackers, which generalizes the problem formulations from [9, 10] to the case of a moving defense object.

2.1. Motion Model of the Protected Point, Attacking Targets, and Interceptor Defender

Suppose it is necessary to protect a moving target T, capable of moving with speed V_T, from an attack by n attackers that chase T with speed $V_A > V_T$. We assume that T must be defended by an interceptor defender D, which can pursue the attackers with speed $V_D > V_A$. Motion occurs in a plane; the target T and the defender D are located at the origin at the initial time, and the attackers are on the boundary of a circle of radius R in a layer of thickness $2\Delta R$.

The initial positions of the attackers are specified by radius vectors $\mathbf{r}_1^0, \ldots, \mathbf{r}_n^0$, and their velocities $\mathbf{V}_1, \ldots, \mathbf{V}_n$ are constantly directed toward T:

$$\mathbf{V}_i = V_A \frac{\mathbf{r}_T - \mathbf{r}_i}{|\mathbf{r}_T - \mathbf{r}_i|}, \tag{1}$$

where $\mathbf{r}_i$ and $\mathbf{r}_T$ are the current positions of the ith attacker and the target T. An attacker is considered missed if at some time its coordinates coincide with the position of T.

The dynamics of the protected target T is described by the simple motion model

$$\begin{aligned} \dot{x}_T &= V_T \cos(u), \\ \dot{y}_T &= V_T \sin(u), \end{aligned} \tag{2}$$

where x_T and y_T are the coordinates of T, and $u \in [0, 2\pi]$ is the control of the direction of motion.

The velocity vector of the defender D is determined by the attacker that D is intercepting at the given time:

$$\mathbf{V}_D = V_D \frac{\mathbf{r}_i - \mathbf{r}_D}{|\mathbf{r}_D - \mathbf{r}_i|}, \tag{3}$$

where $\mathbf{r}_D$ and $\mathbf{r}_i$ are the positions of D and of the attacker whose interception D is moving toward. An attacker is considered destroyed if at some time its coordinates coincide with the position of D.

2.2. Concept of a Defense Plan, Its Quality Criteria, and Problem Statement

An interception plan $\pi = [\pi_1, \ldots, \pi_n]$ is a sequence of targets, where π_i is the number of the target that D visits in the ith turn. Let $\tau^*(\pi)$ be the execution time of π; then on $[0, \tau^*(\pi)]$ it is required to construct a control $u(t)$ of the motion of T according to (2). To simplify the problem, it is proposed to specify the motion of the point T in discrete time with step Δt. We introduce the following

Definition 1. A defense plan Π is a pair (π, ρ), where π is the sequence of target interceptions by the defender D, and $\rho = \{\mathbf{r}_T(t_k)\}_{k=0}^K$ is the set of points that T occupies at times $t_k = k\Delta T$. Moreover, the following constraints are imposed on ρ:

$$|\mathbf{r}_T(t_{k+1}) - \mathbf{r}_T(t_k)| = V_T \Delta t,$$

which express the reachability conditions for the points ρ; without loss of generality, we always assume that $\mathbf{r}_T(0) = \mathbf{r}_D(0) = (0, 0)$.

We introduce the loss vector $L(\Pi) = (m_0, \ldots, m_5)$, where m_0 is the number of missed targets, and m_i is the number of targets intercepted at a distance of at least R_i from T, where $R_1 < R_2 < \cdots < R_5$. Then we formulate the problem.

Problem 1. Given the initial positions $\mathbf{r}_1^0, \ldots, \mathbf{r}_n^0$ of the attacking targets, construct a defense plan Π according to Definition 1 with the smallest possible loss vector $L(\Pi)$, where comparison is performed in lexicographic order, i.e.,

$$(m_0, \ldots, m_5) < (m_0', \ldots, m_5') \to \exists k : \forall i < k : m_i = m_i' \wedge m_k < m_k'.$$

In practice, finding an optimal plan Π is practically impossible, and it is required to find a suboptimal Π with an acceptable $L(\Pi)$. In [10], the following is proved

Theorem 1. *For any initial arrangement of attackers, in the presence of a dynamic advantage $(V_D > V_A)$ of the defender over the attackers, for a stationary protected target there exists an interception plan π that ensures interception of all attackers without misses.*

Corollary 1. *For any initial arrangement of attackers, in the presence of a dynamic advantage $(V_D > V_A)$ of the defender over the attackers, for a moving protected target there exists an interception plan π that ensures interception of all attackers without misses.*

Proof. Indeed, since a dynamic target is able to remain stationary, its ability to move cannot worsen the result of Theorem 1, and therefore there always exists a defense plan without missed attackers.

Thus, a reasonable requirement for the sub-optimality of a defense plan is $m_0 = 0$, i.e., the absence of missed targets.

2.3. Positional Properties of Attackers and Local Decision Rules

Suppose that at time t the defender D and the protected target T occupy positions $\mathbf{r}_I(t)$ and $\mathbf{r}_T(t)$, $\mathbf{V}_T(t)$ is the velocity of T, and $\mathbf{r}_i(t)$ are the radius vectors of the attackers. For further discussion, the following quantities are required:

$$\tau_i(t) = \frac{V_A|\mathbf{r}_T(t) - \mathbf{r}_i(t)| + \mathbf{V}_T(t)[\mathbf{r}_T(t) - \mathbf{r}_i(t)]}{V_A^2 - V_T^2},$$

where $\tau_i(t)$ is the time required by the ith attacker to hit T, provided that T continues moving in a straight line with velocity $\mathbf{V}_T$, and

$$\theta_i(t) = \frac{V_I|\mathbf{r}_T(t) - \mathbf{r}_i(t)| + \mathbf{V}_i(t)[\mathbf{r}_T(t) - \mathbf{r}_i(t)]}{V_I^2 - V_A^2}, \tag{4}$$

where $\theta_i(t)$ is the time required by the defender D to intercept the ith attacker, provided that the latter moves in a straight line with velocity $\mathbf{V}_i$ determined according to (1) from the current values of $\mathbf{r}_i(t)$ and $\mathbf{r}_T(t)$.

On the basis of $\tau_i(t)$ and $\theta_i(t)$, we introduce the following definitions.

Definition 2. The danger of the ith attacker is understood as the quantity

$$K_i(t) = \frac{1}{\tau_i(t)},$$

the reciprocal of the hit time $\tau_i(t)$.

Danger shows how strongly the attacker threatens T: the less time it needs to hit T, the higher its danger.

Definition 3. The convenience of the ith attacker is understood as the quantity

$$U_i(t) = \frac{1}{\theta_i(t)},$$

the reciprocal of the interception time $\theta_i(t)$.

Convenience shows how easy it is for the defender to intercept the attacker: the less time required for this, the higher its convenience.

Definition 4. The advantage over the ith attacker is understood as the quantity

$$A_i(t) = \tau_i(t) - \theta_i(t),$$

the difference between the hit time $\tau_i(t)$ and the interception time $\theta_i(t)$.

Advantage shows the time reserve for interception: the higher the advantage, the longer the decision on assigning this target can be postponed.

When planning the motion of T, it is reasonable to primarily evade attackers that pose the greatest threat in order to gain time for their interception by D. The degree of threat from the ith attacker is estimated by its danger $K_i(t)$; therefore, to find the velocity vector of the target, it is logical to use the rule

$$\mathbf{V}_T(t) \sim \sum_{i=1}^{n} \frac{K_i(t)}{\sum\limits_{j=1}^{n} K_j(t)} \frac{\mathbf{r}_T(t) - \mathbf{r}_i(t)}{|\mathbf{r}_i(t) - \mathbf{r}_T(t)|}, \tag{5}$$

where the proportionality sign means that the quantity on the right determines the direction of the velocity $\mathbf{V}_T(t)$, while its magnitude coincides with V_T. This rule corresponds to priority escape from attackers with the greatest danger $K_i(t)$. An important caveat should be made here: the use of this rule deprives Corollary 1 of force, since now the protected target cannot stop moving, which means that there may be initial conditions under which no interception plan without misses exists, but such configurations are extremely rare.

In addition, the instantaneous characteristics of the attackers make it possible to formulate local rules for the defender D:

$$P_Q(i) = \frac{Q_i(t)}{\sum\limits_{j=1}^{n} Q_j(t)}, \tag{6}$$

where instead of $Q_i(t)$ one substitutes the danger $K_i(t)$ or the convenience $U_i(t)$ of the ith target. The quantities $P_Q(i)$ can be used both for deterministic assignment of attackers to the defender by $\operatorname{argmax} P_Q(i)$ and for constructing a stochastic policy, where $P_Q(i)$ is treated as the probability distribution of assignments.

2.4. Concept of a Partial Plan and Properties of the Planning Problem

When rule (5) is used for the motion of the target T, its trajectory is completely determined by the initial positions and the interception plan π. For $\pi = [\pi_1 \dots \pi_n]$, we introduce the sequence of times $\mathbf{t} = [t_1 \dots t_n]$ required by D to service the first k attackers from π. Here, servicing an attacker means either its successful interception or its hitting T before capture. Then the motion of T up to time t_k depends only on the first k elements of the plan π, which motivates the following definitions.

Definition 5. A partial plan $\pi = [\pi_1, \dots, \pi_k]$ is an ordered set of $k \leqslant n$ distinct targets specifying the order of their interception, to which the intermediate state $S(\pi)$ corresponds. By $S(\pi)$ we mean the coordinates $\mathbf{r}_T(t_k)$, $\mathbf{r}_D(t_k)$, and $\mathbf{r}_j(t_k)$, where t_k is the execution time of the partial plan π, $\mathbf{r}_j(t_k)$ are the positions of those attackers that were not intercepted by D and have not yet hit T by the time π is completed, together with the current loss vector $L(\pi) = (m_0(t_k), \dots, m_5(t_k)$, where $m_0(t_k)$ is the number of missed targets, and $m_i(t_k)$ are the numbers of attackers intercepted in the corresponding zone by time t_k.

For a partial plan, we introduce the estimate $l(\pi) = L(\pi) + Z(S)$, where $L(\pi)$ is the current loss vector of the plan π, and $Z(S) = (0, z_1, \dots, z_5)$, where z_i is the number of targets located in the corresponding loss zone in the intermediate state $S(\pi)$. Then the following lower-bound theorem holds.

Theorem 2. *The final loss vector $L(\pi^*)$ for any continuation of the plan π is always no less than $l(\pi)$, i.e., $L(\pi^*) \geqslant l(\pi)$, if π^* is a continuation of π.*

Proof. Indeed, since the speed $V_A > V_T$, the distance $d = |\mathbf{r}_i(t) - \mathbf{r}_T(t)|$ between the ith attacker and the target T decreases monotonically with increasing time t. Therefore, the loss vector $L(\pi^*)$ of any continuation of the partial plan π will include both the loss vector $L(\pi)$ and the losses for not yet intercepted attackers, which cannot be stopped at a distance greater than the current distance between them and T.

Corollary 2. *If for some partial plan π the loss estimate $l(\pi) \geqslant L(\pi^*)$ exceeds the losses for some complete plan π^*, then none of the continuations of π can be optimal.*

Corollary 3. *If for some partial plan π it is true that the loss estimates $l(\pi')$ of all its continuations of fixed length $\mathrm{len}(\pi') = k$ exceed the estimate of some complete plan π^* (i.e., $\forall \pi' : \mathrm{len}(\pi') = k \to l(\pi') \geqslant L(\pi^*)$), then none of the continuations of π can be optimal.*

These corollaries make it possible to reject obviously nonoptimal search directions when searching for an optimal or suboptimal solution. Moreover, using the fact that the existence of a plan that misses no attacker is known in advance, the vector $l^* = [0, n, 0, 0, 0]$, corresponding to the worst estimate without misses, can be used as the baseline value of the inadmissible estimate $l(\pi)$.

In addition, the interception plan π can be characterized by a set of local properties of its intermediate states. We introduce the definitions.

Definition 6. The complexity of the plan π is understood as the maximum time

$$\theta(\pi) = \max_{k=1\dots n} \theta(t_k)$$

of interception of an individual target within this plan. Here the quantities $\theta(t_k)$ are understood in the sense of (4) with respect to the attacker that was intercepted at time t_k from the vector $\mathbf{t}(\pi)$.

Definition 7. The reliability of the plan π is understood as the minimum advantage

$$A(\pi) = \min_{k=1\ldots n} A(t_k)$$

over the target from the plan π. Here $A(t_k)$ is understood in the sense of Definition 4 with respect to the attacker that was intercepted at time t_k from the vector $\mathbf{t}(\pi)$.

The reliability and complexity of a plan are its global properties, which can be used to study the dependence of the loss function $L(\pi)$ on the characteristics of π.

3. CONSTRUCTION OF AN INTERCEPTION PLAN BY MONTE CARLO TREE SEARCH

This section discusses the algorithm for constructing suboptimal plans by the MCTS method.

3.1. Monte Carlo Tree Search Tree

We introduce the concept of a search tree.

Definition 8. A search tree Ξ is understood as a set of partial plans Π^* equipped with an incidence relation: $\pi \to \pi'$, if π' differs from π by adding exactly one target to the end of the plan. For a partial plan π, the set $C(\pi) = \{\pi' \in \Pi^* : \pi \to \pi'\}$ of all continuations incident to it is called its children. The root of the tree Ξ is the empty plan $\pi_0 = []$.

The MCTS algorithm performs an intelligent search over the tree Ξ, which is guided by the policy (6) and by the q-estimates of its nodes. Each plan π is marked with the number p_π of continuations constructed for this plan and with the average estimate q_π achieved on them. Here, the average q-estimate is understood as the average scalar estimate of the plans constructed from this node (the loss vector $L(\pi)$ is transformed into a scalar estimate using a linear function that assigns the estimate -1.0 to the maximum possible losses and $+1.0$ to the minimum ones). The basic loop of the algorithm consists of two procedures.

Algorithm 1. Tree descent

1. $\pi \leftarrow \pi_0$.
2. While $p_\pi > 0$: $\pi \leftarrow \arg\max\limits_{\pi':\pi\to\pi'} q_{\pi'} + \dfrac{P_Q(\pi')\sqrt{n_\pi}}{1+n_{\pi'}}$.
3. Return π.

For the found node π, a random continuation π^* is constructed according to the local rule (6) and the loss vector $L(\pi^*)$ is computed, on the basis of which the q-estimate of the plan π is formulated and then propagated up to the root.

Algorithm 2. Backpropagation of the estimate.

1. $r \leftarrow L_q(L(\pi^*))$, where $L_q(L(\pi^*))$ is the q-estimate of the loss vector $L(\pi^*)$.
2. While $\pi = [\pi_1 \ldots \pi_k]$ is not π_0: update p_π and q_π on the basis of r (the number of visits is increased by one, and the average q-estimate is recalculated taking into account the addition of r), $\pi \leftarrow \pi' = [\pi_1 \ldots \pi_{k-1}]$.

Thus, this algorithm updates the statistics accumulated over the tree nodes taking into account the new solution π^*, which was constructed from the tree node selected according to the tree descent Algorithm 1. The general scheme of the algorithm is shown in Fig. 1; more details can be found in [9].

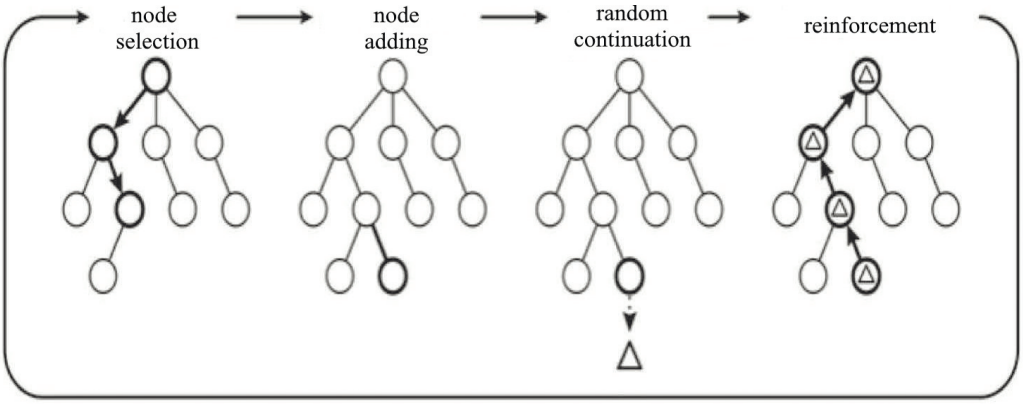


Fig. 1. Principal scheme of the Monte Carlo tree search algorithm. The algorithm consists of alternating two phases: descent through the tree to a new node and backpropagation of the estimate of this node back to the root.

3.2. Construction of Random Continuations

In the previous subsection, it was discussed that the MCTS algorithm relies on random continuations of partial plans to obtain estimates of tree nodes. This subsection describes in detail the methodology for constructing such continuations and the method for computing q-estimates.

Suppose it is required to evaluate a tree node with a partial plan π, to which the intermediate state $S(\pi)$ corresponds. A random continuation is constructed as a solution to the planning problem from the state $S(\pi)$ on the basis of the local rules (5) and (6), where rule (6) is used as a stochastic policy for assigning the defender to intercept attackers.

Algorithm 3. Construction of a random continuation.

1. $\pi^* \leftarrow \pi$.

2. While there are free targets (i.e., not intercepted and not missed): $\pi^* \leftarrow i$, which is chosen according to the probability distribution $P_Q(i|S)$ for the current intermediate state $S(\pi^*)$.

Once a complete plan π^* has been constructed according to this algorithm, the loss vector $L(\pi^*)$ is computed for it. Then $L(\pi^*)$ is transformed into a scalar q-estimate in the range $[-1, 1]$ using the linear function $L_q(L(\pi^*))$, which acts according to the rule

$$L_q([n, 0, 0, 0, 0, 0]) = -1,$$
$$L_q([0, 0, 0, 0, 0, n]) = 1,$$

i.e., assigns the estimate -1 to maximum losses and $+1$ to minimum losses. This estimate is then used in the backpropagation Algorithm 2.

3.3. Pruning of Nonoptimal Nodes and Construction of the Final Plan

In addition, the results of the MCTS search algorithm can be substantially improved on the basis of Theorem 2 and its corollaries. For this purpose, the algorithm stores at each step the best complete plan $\pi_{\max}$ that was constructed during the evaluation of one of the tree nodes over the entire running time, as well as its loss vector $L(\pi_{\max})$. A tree node with a partial plan π is considered nonoptimal if its partial state is nonoptimal with respect to $L(\pi_{\max})$ according to Corollary 2, or if all of its child nodes have been recognized as nonoptimal.

During tree descent in Algorithm 1, the choice is made exclusively among nodes that are not nonoptimal. If a node is reached that must be recognized as nonoptimal, the algorithm returns to its parent node and repeats the search process taking this additional information into account.

When the computational limit is exhausted, the algorithm returns the best plan $\pi_{\max}$ that was constructed during its entire running time.

4. EXPERIMENTAL RESULTS

The Monte Carlo tree search algorithm together with the heuristics for the motion of T and the assignment of targets to D by target danger was used to construct defense plans against attacks by 10 to 30 targets. In this case, the construction of effective defense plans on average requires from 1 to 10 s of algorithm operation depending on the problem size. Examples of characteristic defense plans are given in Figs. 2 and 3.

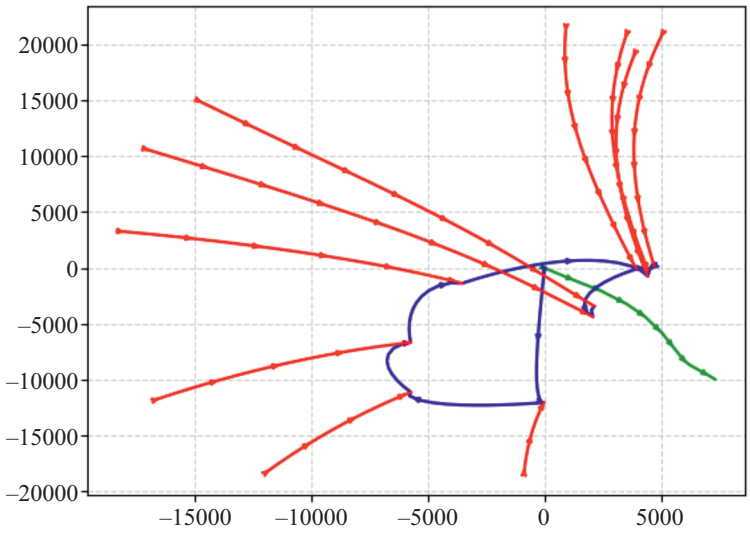


Fig. 2. Defense plan against an attack by 10 attackers. The green line shows the trajectory of the protected target, the blue line that of the interceptor defender, and the red lines those of the attackers. The arrows indicate the direction of motion and the relative arrangement of the objects on equal-time slices.

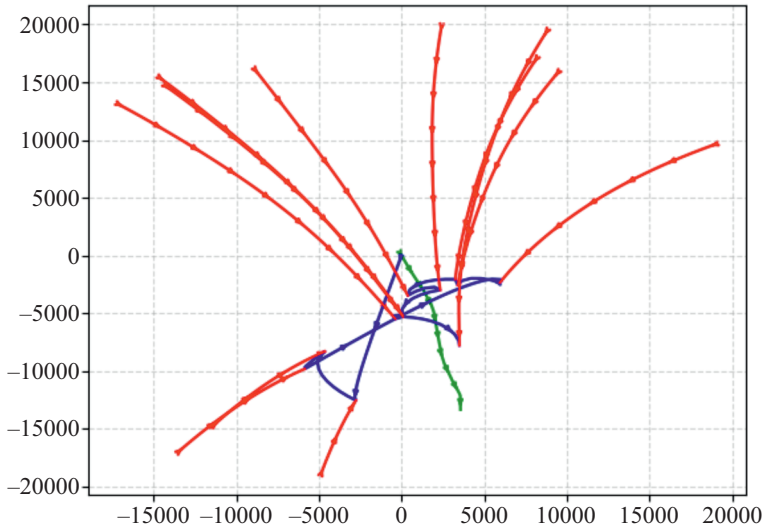


Fig. 3. Defense plan against an attack by 12 attackers. The green line shows the trajectory of the protected target, the blue line that of the interceptor defender, and the red lines those of the attackers. The arrows indicate the direction of motion and the relative arrangement of the objects on equal-time slices.

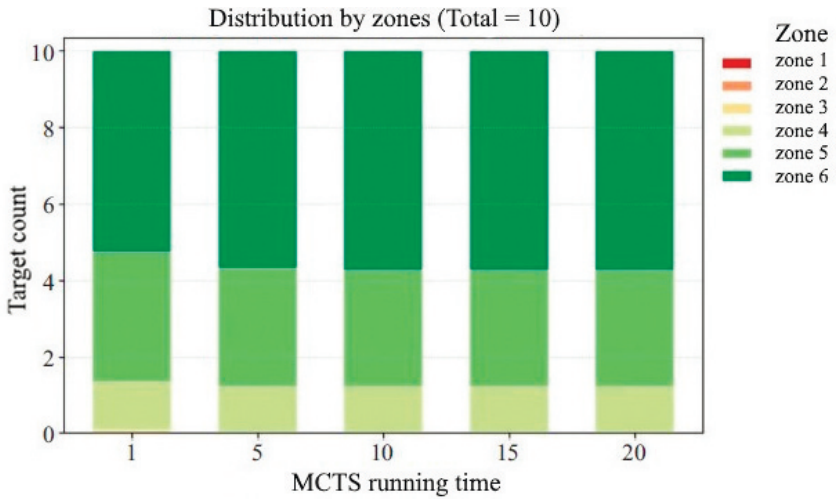


Fig. 4. Distribution of interception points by proximity zones as a function of MCTS running time for problems with $n = 10$ attackers.

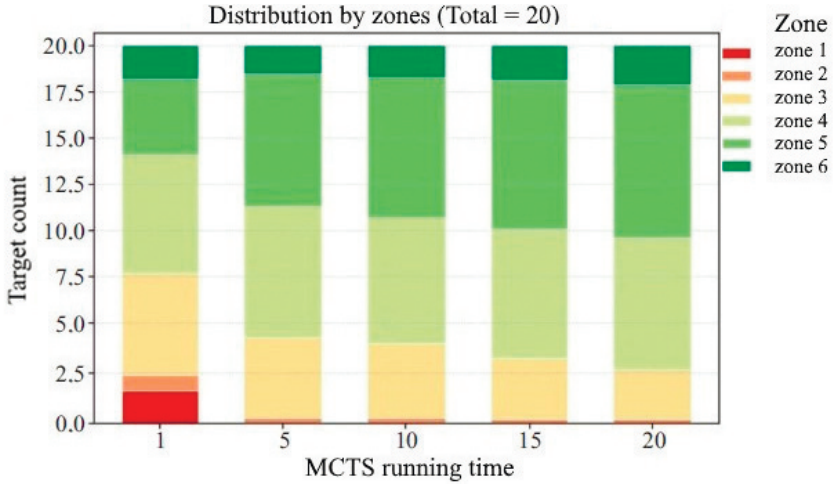


Fig. 5. Distribution of interception points by proximity zones as a function of MCTS running time for problems with $n = 20$ attackers.

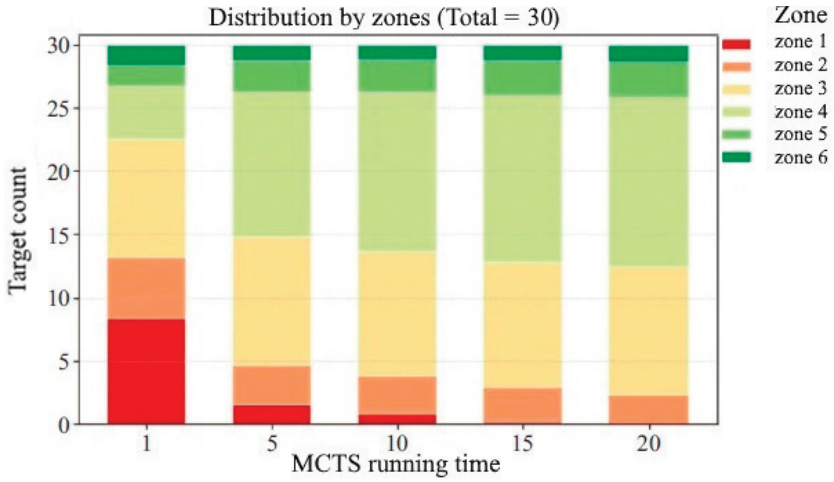


Fig. 6. Distribution of interception points by proximity zones as a function of MCTS running time for problems with $n = 30$ attackers.

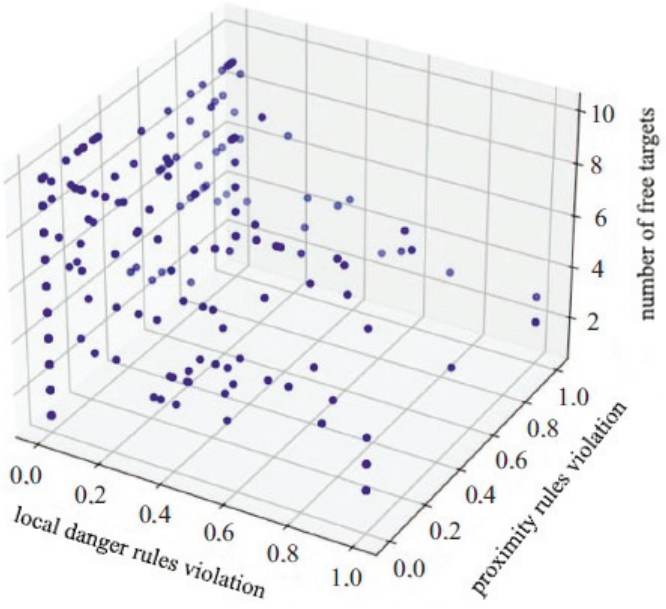


Fig. 7. Violation of the local danger and proximity rules in the optimal plan as a function of the number of free targets in the intermediate state.

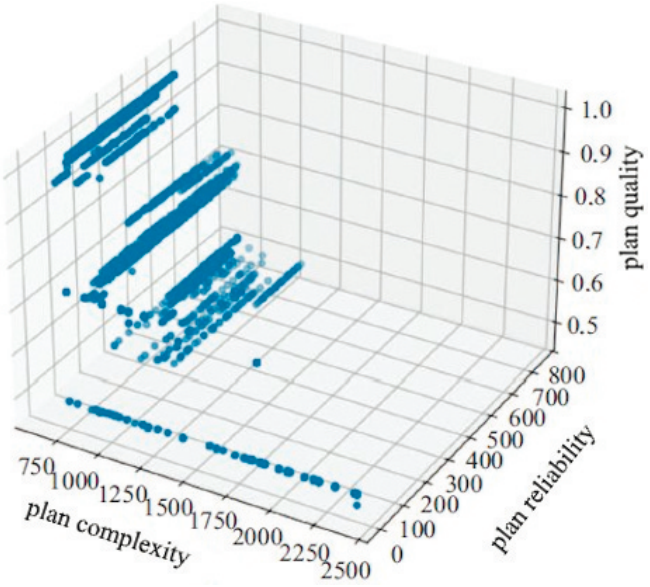


Fig. 8. Dependence of plan quality on their reliability and complexity.

The dependence of the quality of the interception plan on the algorithm running time for different numbers of attackers is given in Figs. 4–6; the data for constructing these figures were collected on a sample of 30 random initial positions with a given number of attackers. Thus, one can see the characteristic times for constructing suboptimal plans for problems of the corresponding size.

In addition, studies were conducted on the quality of the local rules (3) in comparison with the optimal plan, the results of which are given in Fig. 7. Here, the violation of a rule is understood as an estimate that varies linearly from 0 to 1, where an estimate of 0 means that the choice of target in the optimal rule is fully consistent with the heuristic, and 1 means fully inconsistent (i.e., the optimal choice corresponds to the attacker that is least attractive from the heuristic viewpoint).

Also, for completeness, we present the distribution of plans by quality depending on their global properties of complexity and reliability, constructed for the set of all admissible plans for a position with $n = 8$ attackers.

5. CONCLUSION

This work was devoted to the problem of protecting a dynamic target from an attack by multiple attackers using an interceptor defender. The main results of the work can be briefly presented in the following points.

1) The problem of protecting a dynamic target is an NP-hard problem, for which, however, effective algorithms for constructing suboptimal solutions can be proposed.

2) The Monte Carlo tree search algorithm shows good results on input examples with numbers of targets up to $n = 30$ in reasonable computational time.

3) As reference local rules for MCTS, it is logical to use heuristics based on danger and proximity, defined according to the probability distribution 6.

The MCTS algorithm constructs a higher-quality heuristic on the basis of the initial heuristics through intelligent additional search over variants. The distributions of visit numbers of child nodes at the root of the tree can be interpreted as an improved local policy, which can then be used for initial training of neural-network models, opening prospects for applying the results obtained in this work to the construction of large neural-network models for this problem. On the other hand, the possibility of effectively solving the defense problem without involving neural-network training is valuable in itself, since neural-network approaches are not always more effective than simple heuristic solvers [15, 16] in DTSP.

REFERENCES

1. Sikharulidze, G.G., On one generalization of the traveling salesman problem. I, *Autom. Remote Control*, 1971, no. 8, pp. 116–123.

2. Sikharulidze, G.G., On one generalization of the traveling salesman problem. II, *Autom. Remote Control*, 1971, no. 10, pp. 142–147.

3. Picard, J.C. and Queyranne, M., The time-dependent traveling salesman problem and its application to the tardiness problem in one-machine scheduling, *Oper. Res.*, 1978, vol. 26, no. 1, pp. 86–110. https://doi.org/10.1287/opre.26.1.86

4. Helvig, C.S., Robins, G., and Zelikovsky, A., The moving-target traveling salesman problem, *J. Algorithms Comput. Technol.*, 2003, vol. 49, no. 1, pp. 153–174. https://doi.org/10.1016/S0196-6774(03)00075-0

5. Li, C., Yang, M., and Kang, L., A new approach to solving dynamic traveling salesman problems, in *Lect. Notes Comput. Sci.*, 2006, vol. 4247, Berlin, Heidelberg, Springer.

6. Stieber, A., Dealing with time in the multiple traveling salesmen problem with moving targets, *Cottbus Math. Preprints*, 2019, COMP. 4.

7. Maskooki, A. and Kallio, M., A bi-criteria moving-target travelling salesman problem under uncertainty, *Eur. J. Oper. Res.*, 2023, vol. 309, no. 1, pp. 271–285. https://doi.org/10.1016/j.ejor.2023.01.009

8. Garey, M.R. and Johnson, D.S., *Computers and Intractability: A Guide to the Theory of NP-Completeness*, San Francisco, Calif., W. H. Freeman & Co., 1979.

9. Ryabushev, E.A. and Galyaev, A.A., Search for a suboptimal solution of the dynamic traveling salesman problem by the Monte Carlo method, *Autom. Remote Control*, 2024, no. 2, pp. 103–119.

10. Galyaev, A.A., Yakhno, V.P., Berlin, L.M., et al., Optimization of the interception plan for straight-line moving targets, *Autom. Remote Control*, 2023, no. 10, pp. 18–36.

11. Galyaev, A.A. and Ryabushev, E.A., Local dynamic planning for protecting a moving target from an attack by a high-speed interceptor with limited field of view, *Proc. Inst. Math. Mech. Ural Branch RAS*, 2026, vol. 32, no. 3.

12. Silver, D., Huang, A., Maddison, C., et al., Mastering the game of Go with deep neural networks and tree search, *Nature*, 2016, vol. 529, no. 7587, pp. 484–489. https://doi.org/10.1038/nature16961

13. Schadd, M.P.D., Winands, M.H.M., van den Herik, H.J., et al., Single-player Monte-Carlo tree search, in *Comput. Games. CG 2008. Lect. Notes Comput. Sci.*, 2008, vol. 5131, Berlin, Heidelberg, Springer.

14. Marco, K., Beyond games: A systematic review of neural Monte Carlo tree search applications, 2023. https://arxiv.org/abs/2303.08060

15. Xuanhao, P., Chenguang, W., Chaolong, Y., et al., Rethinking the heatmap + Monte Carlo tree search paradigm for solving large scale TSP, 2025. https://openreview.net/forum?id=TMHOHRR0FA

16. Xia, Y., Xianliang, Y., Zichuan, L., et al., Position: Rethinking post-hoc search-based neural approaches for solving large-scale traveling salesman problems, *Proc. 41st Int. Conf. Mach. Learn.*, 2024, vol. 235, pp. 54178–54190. https://proceedings.mlr.press/v235/xia24f.html

This paper was recommended for publication by A.A. Lazarev, a member of the Editorial Board