International Journal of Innovative
Computing, Information and Control
Volume **8**, Number **5(B)**, May **2012**

ICIC International ©2012 ISSN 1349-4198
pp. **3769–3779**

A PERMUTATION ALGORITHM FOR LASER ANTIMISSILE STRATEGY OPTIMIZATION

Yong Sun, Maorui Zhang and Xiangyu Gao

Center for Control Theory and Guidance Technology
Harbin Institute of Technology
No. 92, West Da-Zhi Street, Harbin 150001, P. R. China
sunyong233@126.com

Received October 2010; revised March 2011

Abstract. *This paper focuses on a dynamical laser antimissile problem in real-time application. When faced with multiple attacking targets, it is important for the laser antimissile system to determine the attacking sequence in order to destroy more targets. Conceptually, this laser antimissile problem can be seen as a dynamical traveling salesman problem, which is much harder than a pure traveling salesman problem. It is extremely difficult to find the global optimal solution, especially when the number of the targets is greater than six. This paper introduces a fast algorithm for this problem based on two exchange strategies – the adjacent exchange and the symmetric exchange. With a given initial sequence, the adjacent exchange strategy searches for a local optimal sequence among neighbors fast. The symmetric exchange will search in a larger region so as to jump out from the local optimal solution. The combination of the two exchange strategies increases the probability of finding the global optimal solution. For illustration, a case with eight attacking targets is simulated. The results show the convergence of the proposed algorithm is very fast and it is a highly practical method. Interestingly, the local optimal solution obtained is, indeed, quite close to the global one.*
Keywords: Laser antimissile, Parallel navigation, Adjacent exchange, Symmetric exchange

1. **Introduction.** The antimissile system is very important for the country safety, which has been developed for a long time [1]. However, the laser antimissile system is laid out recently. The objective of the dynamic laser antimissile problem (DLAP) is to minimize the whole attacking time for intercepting all the missiles. In fact, DLAP is a dynamic traveling salesman problem (TSP), which is proved a NP-hard problem. All the targets are moving and the laser beam point needs to track the target and radiate the laser beam when the laser beam point meets with the missile. With the development of the laser technology, the laser can destroy or damage the missile within about 10 km. So the laser can engage the antimissile mission. Because there are many targets at the same time, it is necessary to determine which target should be attacked first. Even if every step gets the optimal target, the global optimality cannot be obtained.

The DLAP is similar with the traveling salesman problem, which has been recognized as the most famous combinatorial optimization problem. Given a finite set of cities $C = \{c_1, c_2, \ldots, c_k\}$ and the distance matrix D, the TSP asks for finding a tour through all of the cities, visiting each exactly once, and returning to the original city such that the total distance traveled is minimized. The distance matrix is constant in TSP, while the targets are all moving at some speed in DLAP. There is another difference between TSP and DLAP. The laser beam does not have to come back to the starting position in DLAP. These two differences result in some difficulty for solving DALP.

The TSP is regarded as one of the most challenging problems in the field of combinatorial optimization. It is almost forbidden to find the global optimal permutation with a long time for real-world applications. Even with a small number of targets, exact algorithms are not easy to be designed with moderate computational effort as can be seen from the complexity theory [2]. As a result, different types of heuristic methodologies have been applied in solving the combination problem. Differential evolution is used for the discrete problem, where the search space is augmented to improve the performance of the technique [3]. Although in principle differential evolution is used to find the optimal solution, the manner in which the space is stated and then searched is altered to improve the overall performance. Tian et al. [4] applied simulated annealing (SA) to permutation problems. They find that SA algorithm terminates with a global optimum if computation time is long enough. However, it is inapplicable in practical implementation, especially in DLAP. The heating strategy is introduced in SA algorithm and the results show it has the larger probability than that of the traditional SA algorithm to find the global optimal solution [5]. The method based on genetic algorithm (GA) is presented for solving DLAP with two essential crossover operators [6]. The combination of two crossover operators and the mutation operator can well deal with the premature problem and give the local optimal solution. The modified GA based on the simulated annealing is used to solve DLAP [7]. The SA algorithm can jump from the local optimal solution. Therefore, the combination of GA and SA has their merits. A novel approach for avoiding premature convergence to local minima is given by the introduction of diversity in the particle swarm optimization [8]. A modified particle swarm optimization is introduced for the optimization of mixed-variable problems [9]. In order to make use of the differences between any two permutations, the permutation distance is combined with GA for solving DLAP [10]. The fitness of the individual is modified by the niche technique based on the permutation distance. It prevents GA from being trapped in local optima of the search space [11]. However, GA usually needs long time to select a best solution from many generations. The much time consumption prohibits its application in DLAP.

However, the above methods still occupy almost more than 0.1 second to get the local optimal solution. It is hardly used in real-time application. In order to reduce the computing time further, the efficient algorithm is necessary to be further studied. Therefore, the low efficiency of above methods motivate us to give a practical and fast algorithm to solve DLAP as quick as possible. In this paper, we propose a new heuristic algorithm made up of six steps to minimize the overall attacking time. The first step is to generate some initial solutions for the improvement of the next steps. The second step applies the adjacent exchange on the initial solutions and finds the better solution step by step. Another permutation called symmetric exchange is applied sometimes for jumping out from the local optimal solution. The best solution is chosen from all the improved local solutions in the final step.

The paper is organized as follows. Section 2 gives the problem formulation. The parallel navigation diagram is introduced in Section 3. In Section 4, the fast permutation algorithm is depicted. Two exchange strategies are shown in detail. The fast permutation algorithm is presented in Section 5. Section 6 gives an example with the case of eight targets. Section 7 concludes the paper and presents some advice in the future study.

2. **Problem Statement and Preliminaries.**

The antimissile system is a complicated system, including the early warning radar system, the ground director radar system, the command and control system and the intercepting weapon system. This is the conventional antimissile system. While the laser antimissile system is to protect a region such as missile site, high buildings and oil field. This system is located around the protected place.

When many missiles are coming at the same time from one direction, the radar system detects the targets and measures the positions and computes the velocities of them. The control center must make a decision on attacking sequence as soon as possible.

2.1. **Assumptions.** We map the three dimension space into the plane. The plane can be seen as the monitor screen, in which the targets trajectories can be predicted by the history information. The coordinate frame is built in this plane, the x-axis is along with the horizon and the y-axis is vertical. Then, the position of each target can be measured in this coordinate system. We make some assumptions about this problem.

(1) We only consider the movement of the targets in the monitor plane. The distance between the laser location and each target is ignored for simplifying the problem.

(2) When some targets are detected, no more targets will come into the monitor plane. The aim is to intercept all the targets with the minimum time.

(3) The direction and the magnitude of the velocity of every target are constant in the plane. Because the distance between the laser location and the target is usually about longer than one kilometer, the trajectory of the target can be regarded as a straight line in the monitor plane. Similarly, the magnitude of the velocity varies little in the monitor plane.

(4) The laser beam point is regarded as a point in the monitor plane denoted as L. The laser beam point needs to track the target according to someone navigation method. When the laser beam point L meets with the target, the target is damaged by the laser beam energy immediately. In fact, the laser beam point needs to keep with the target for a few seconds until the target is destroyed. In the future, this assumption can be discarded for real laser antimissile problem.

(5) The targets fly toward the protected place, the distance between the protected place and the target is long enough to complete the attack mission.

2.2. **Model description and cost function.** Based on the above assumptions, the dynamic laser antimissile problem can be considered as a dynamic TSP. However, the laser beam point does not necessarily come back to the original position. Each target is like the city, which is moving straightly at a constant velocity in the plane. The laser beam point in the plane can move at a constant velocity, whose direction is determined by the navigation method, such as parallel navigation, pursuit navigation and proportional navigation. Then, the laser beam point should meet with every target in a sequence. The problem can be presented as follows.

Consider N targets and set the initial time $t_0 = 0$. The initial position and the velocity of each target are $\mathbf{x}_i \in R^2$ and $\mathbf{v}_i \in R^2$ $(i = 1, \ldots, N)$ respectively. The targets are labeled by the nature number from 1 to N. The laser beam point L is at $\mathbf{x}_0 \in R^2$ and its maximum velocity is v_0. The permutation is denoted as $P = [p_1, \ldots, p_N]$, where $p_i \in \{1, \ldots, N\}$ $(i = 1, \ldots, N)$. The p_i of the permutation P represents that the target p_i should be attacked in the i-th attacking mission. For example, the second element of one permutation $P = [3, 1, 4, 2]$ represents that the target 1 should be attacked in the second mission.

Supposing the permutation $P = [p_1, \ldots, p_N]$, the intercepting mission is carried out according to P. At the initial time t_0, L begins to track the target p_1. If the point L and the target p_1 meet at time t_1, the target p_1 is destroyed at time t_1. The time duration between the t_0 and t_1 be denoted by $\Delta t_0 = t_1 - t_0$. Now, L turns its mission to track the target p_2. If L and the target p_2 meet at time t_2 at which the target p_2 is destroyed, and let the time duration between t_2 and t_1 be denoted by $\Delta t_1 = t_2 - t_1$. The process continues according to the sequence specified by the perturbation $P = [p_1, \ldots, p_N]$, until the target p_N is destroyed at time t_N. To sum up, a total of N interceptions are carried out during

the whole mission, and the N time durations are Δt_i, $i = 0, 1, \ldots, N-1$. Therefore, once the permutation P is chosen, the overall attacking time from the initial position to the last attacking can be predicted based on the current positions and the velocities. The overall attacking time is denoted by $T(P)$. Summing up these time durations, the overall time $T(P)$ to finish interceptions is:

$$T(P) = \sum_{i=0}^{N-1} \Delta t_i (p_i, p_{i+1}, t_i), \tag{1}$$

where $t_k = \sum_{i=0}^{k-1} \Delta t_i$, $p_i \in \{1, \cdots, N\}$, t_k is the time at which the laser beam point meets the target p_i and this target is damaged immediately. It is necessary to note that the point p_0 is the initial position of laser beam point. $\Delta t_i (p_i, p_{i+1}, t_i)$ represents the time duration from the position of target p_i to meet with the target p_{i+1}.

Remark 2.1. *The cost function is (1) and the decision variable is the permutation P. The aim is to optimize the permutation to minimize the overall time.*

Remark 2.2. *The cost function (1) does not have the gradient with respect to the decision variables, so it can not be solved by the general nonlinear programming methods. In order to solve it effectively, we develop a heuristic fast permutation algorithm based on the exchange strategies.*

3. **Parallel Navigation Method.** The laser beam point L must track the target first until L meets with the target at some place. There are many methods for guiding the laser beam point L to the target. The parallel navigation method is used here because this method causes a straight line for the laser beam point in the monitor plane. In this section, some properties are derived under the parallel navigation method.

For illustration, one target M is moving with the velocity $\mathbf{v}_M$. The laser beam point L moves at the speed of $\mathbf{v}$ toward point B, at which they will meet. The key idea of the parallel navigation is that the sight line LM at any time is parallel with the sight line at the initial time t_0. The diagram of their relative position is shown in Figure 1.

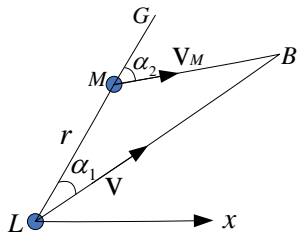

FIGURE 1. Parallel navigation diagram

Therefore, their velocities projections vertical with the sight line LM are equal and the relationship is expressed as:

$$\|\mathbf{v}_M\| \sin \alpha_2 = \|\mathbf{v}\| \sin \alpha_1 \tag{2}$$

The relative position $\mathbf{r}$ can be calculated by their positions and the angle α_2 can be derived from the definition of the inner product.

$$\alpha_2 = \cos^{-1} \frac{(\mathbf{r}, \mathbf{v}_M)}{\|\mathbf{r}\| \, \|\mathbf{v}_M\|}. \tag{3}$$

From Equation (2),

$$\alpha_1 = \sin^{-1} \frac{\|\mathbf{v}_M\| \sin(\alpha_2)}{\|\mathbf{v}\|}. \tag{4}$$

Based on the relative distance and the velocity, the time from the point L to B can be calculated by

$$t_M = \frac{\|\mathbf{r}\|}{\|\mathbf{v}\| \cos \alpha_1 - \|\mathbf{v}_M\| \cos \alpha_2}. \tag{5}$$

Thus the attacking time t_M is calculated by Formula (5) with the parallel navigation. In order to get the attacking time t_M, the initial positions of the target and the laser beam point and the target velocity must be measured.

4. **Fast Exchange Strategies.** Two exchange strategies, which are called the adjacent exchange and the symmetric exchange are proposed to improve the efficiency for solving DLAP. The fast exchange algorithm is presented with two exchange strategies in this section.

The number of the complete permutations of N targets is $N!$. When the N is bigger than six, the value is very large, which makes it impossible within a permitted time to compute all the cost. Therefore, we can start from one permutation P_0 and apply the exchange strategies and get the better permutation. This section includes three steps to get the local optimal permutation. They are initial permutation selection, permutation exchange and the final comparison.

4.1. **Initial permutation selection.** Because the exchange strategies only can find the local optimal solution of DLAP, some initial permutations are chosen to get several local optimal solutions. In this paper, the initial four permutations are selected at random. So four local optimal solutions are compared and the best one is applied in the application.

4.2. **Permutation exchange strategy.** In order to improve the solution, the initial permutation needs be transformed by some strategies. The crossover of GA exchanges a segment of the chromosome with each other and the broad space can be searched. However, this operation makes the convergence slow, which limits the real-time application in DLAP. In order to make the solution converge fast, two strategies are introduced, which are adjacent exchange and the symmetric exchange.

The adjacent exchange of the permutation can keep the merit and search the better permutation within its neighbors. While the symmetric permutation can reach the far permutation and inherit some better segment of the permutation, it can search for the better solution in a wide region. These two exchange strategies are applied in the algorithm.

4.2.1. *Adjacent exchange.* The adjacent exchange of a permutation P_0 is to change the two adjacent elements of the permutation each other and get the new permutation P. The operator of the adjacent exchange is denoted as E and the adjacent exchange of the permutation P_0 is denoted as $E(P_0) = P$. For N targets, there are N operators, E_i $(i = 1, \ldots, N)$. The operator E_i $(i = 1, \ldots, N-1)$ is defined to exchange p_i and p_{i+1}. And the operator E_N is defined to exchange p_N and p_1. For example, $P_0 = [2, 4, 1, 5, 3]$, $E_2(P_0) = [2, 1, 4, 5, 3]$, $E_5(P_0) = [3, 4, 1, 5, 2]$.

Based on the adjacent exchange definition, the neighbors of the permutation can be defined as follows. Let V_N be the set of complete permutations with N targets. Every permutation $P \in V_N$ is regarded as a point in the N dimension space. The number of all the points is $N!$. If $E_k(P_i) = P_j$, the point P_j is called the k-th neighbor of the point P_i and an edge is linked from P_i to P_j. Therefore, every point P has N neighbors, which are $E_1(P), \ldots, E_N(P)$. There are $N \times N!/2$ edges among all the points. Let A_N be the

set of all edges. So the sets V_N and set A_N consist of the graph $G(V_N, A_N)$. There is an interesting property, if P_j is called the k-th neighbor of P_i, P_i is also the k-th neighbor of P_j.

The path from one point to another by the adjacent exchange is along the edges. So we should find the neighbor whose cost function is less than the current point one by one until there is no better neighbor to be found. At each point P_0, one neighbor of P_0 is chosen to check its cost function. If the cost of this neighbor is less than that of P_0, we let this neighbor replace P_0. Otherwise, the next neighbor is checked in the same way until the cost of all the neighbors of P_0 are worse than that of P_0 and the search stops and outputs the local optimal solution P_0. Because each point has N neighbors, the continuous failures is no more than N. This can be selected as the criteria of stopping the search.

Although this strategy can find the local optimal solution very fast, it may fall into the local optimal solution within a small region. In order to jump out of the local optimal solution, the symmetric exchange strategy is introduced next.

4.2.2. *Symmetric exchange.* The symmetric exchange strategy is not like the adjacent exchange, which always exchange two adjacent elements. This strategy exchange p_i ($i = 1, \ldots, N$) and its symmetric position p_{N-i+1} of the permutation P. This operator is denoted as D_i ($i = 1, \ldots, N$). The operator D_i ($i = 1, \ldots, N$) is defined to exchange p_i and p_{N-i+1} of P. For example, $P_0 = [2, 4, 1, 5, 3]$, $D_2(P_0) = [2, 5, 1, 4, 3]$.

While if N is odd, when $i = (N + 1)/2$, the symmetric point of p_i is itself. In this case, the exchange does not change the permutation.

Based on the symmetric exchange definition, we define the friends of the permutation. If $D_k(P_i) = P_j$, the point P_j is called the k-th friend of the point P_i and a bridge is built from P_i to P_j. As the definition, P_i has N friends, but there are about a half of friends is the same with others since $D_i(P) = D_{N-i+1}(P)$. If N is even, every point P has $N/2$ friends and $(N - 1)/2$ friends with N odd. Let B_N be the set of all bridges. And the new graph can be denoted as $G(V_N, B_N)$. Because the number of friends of P is smaller than that of its neighbors, it should begin to search within neighbors first and friends for the deep search. The deep search is the characteristic of the symmetric exchange, which extends the search region in a wide area.

4.2.3. *Exchange strategy property.* The above two strategies have some useful properties. First, the period of the adjacent exchange in sequence is shown.

Lemma 4.1. *Given a permutation of N elements, $P_0 = [p_1, \ldots, p_N]$, the adjacent exchange operator is applied in sequence as follows:*

$$E_k(P_{i-1}) = P_i \tag{6}$$

where $k = \mathrm{mod}(i, N)$, and if $k = 0$, let $k = N$. Then, $P_{N(N-1)} = P_0$.

Proof: Since $P_0 = [p_1, p_2, p_3, \ldots, p_N]$, then

$$P_1 = E_1(P_0) = [p_2, p_1, p_3, \ldots, p_{N-1}, p_N],$$
$$\ldots$$
$$P_{N-1} = E_{N-1}(P_{N-2}) = [p_2, p_3, p_4, \ldots, p_N, p_1].$$

From the $N - 1$ steps permutations the final permutation is P_{N-1}. This course can be regarded as the first element p_1 of P_0, is moved into the N-th position after $N - 1$ steps of adjacent exchange. Then the next $N - 1$ times adjacent exchange are applied and we

have

$$P_N = E_N(P_{N-1}) = [p_1, p_3, p_4, \ldots, p_N, p_2],$$
$$\cdots$$
$$P_{2(N-1)} = E_{N-2}(P_{2(N-1)-1}) = [p_3, p_4, \ldots, p_N, p_1, p_2].$$

The above process is repeated with N times and the final permutation is

$$P_{N(N-1)} = E_N(P_{N(N-1)-1}) = [p_1, p_2, p_3, \ldots, p_N] = P_0.$$

Theorem 4.1. *Given the initial permutation P_0, only the adjacent exchange is applied as Formula (6) and each permutation P_i has a cost T_i. Then, the maximum steps of continuous descending of the cost is $N(N-1)-1$.*

Proof: The result can be obtained by Lemma 4.1 easily.

Remark 4.1. *The procedure generates a loop from P_0 to the itself after $N(N-1)$ steps in sequential as defined by (6). If the cost T_i of every permutation P_i after sequential adjacent exchange always descends, the cost will reach the minimum $T_{N(N-1)-1}$ after $N(N-1)-1$ steps in sequential. In fact, this case happens rarely in practice. Therefore, it usually needs less than $N(N-1)-1$ steps to meet a failure of cost descending. And the next neighbor is checked to find the better solution. This theorem shows that the solution converges very fast within the finite steps only by the adjacent exchange.*

The property of the adjacent exchange has the well local search capacity and converges fast but it falls into the local optimal solution easily. Because the symmetric exchange can find the far away solution effectively, the symmetric exchange is combined with the adjacent exchange to extend the search region. When the cost descending fails after the adjacent exchange, the symmetric exchange is applied. If this exchange is also a failure, the next adjacent exchange is applied in sequential. If this exchange gets a better solution, this solution replaces the original permutation and the next adjacent exchange continues.

4.2.4. *Criteria of stopping iteration.* The criteria of stopping iteration is very essential for the application in DLAP. We set two criteria for stopping the iteration. The first is the number of iteration is no more than n_{Max}, such as $n_{Max} = 100$. This criterion mainly limits the maximum of computer time for the real-time application. The second criterion is based on the property of the exchange strategies. Since every point P has N neighbors, the maximum number of failure from P by the adjacent exchange is N. Therefore, if the number of continuous failures is N, then the iteration stops and the local optimal solution is P.

4.3. **Comparison among the local optimal solutions.** The initial permutations are transformed by the exchange strategies and arrive at the local solutions. The best solution is selected from the obtained local optimal solutions as the final permutation. Based on the above three steps, the algorithm can be organized as follows.

5. **Fast Algorithm for Solving DLAP.** As many targets are flying toward the protected place, the ground radar system detects the positions and the velocities of the targets. And the label is given for each target with the nature number from 1 to N. Then the fast algorithm is organized as follows.

Algorithm 5.1.

Step 1: Select K initial permutations at random, P_0^j $(j = 1, \ldots, K)$. Set the maximum number of iteration $n_{Max} = 30$ and $j = 1$.

Step 2: If $j > K$, go to Step 6. Calculate the cost T_0^j of P_0^j; Set $k = 0$, $flag = 0$, $i = 1$.

Step 3: $k = \mathrm{mod}(i, N)$. If $k = 0$, let $k = N$. Get the permutation $P_i^j = E_k\ (P_0^j)$ and compute its cost T_i^j. Then compare the cost T_i^j and T_0^j. If $T_i^j < T_0^j$, then let $T_0^j = T_i^j$ and $P_0^j = P_i^j$, $i = i + 1$, $flag = 0$, return to Step 3, else, go to Step 4.

Step 4: $P_i^j = D_k\ (P_0^j)$ and compute its cost T_i^j. Then compare T_i^j and T_0^j. If $T_i^j < T_0^j$, then let $T_0^j = T_i^j$ and $P_0^j = P_i^j$, $i = i + 1$, $flag = 0$, return to Step 3, else, $i = i + 1$, $flag = flag + 1$, go to Step 5.

Step 5: If $i > n_{Max}$ or $flag = N$,then output the j-th local optimal solution, $j = j + 1$, and go to Step 2, else, go to Step 3.

Step 6: Select the minimum among K local solutions and output the best permutation.

Although the solution is not global, it can be used in real-time application as a better attacking sequence. In next section an experiment is presented as an example.

6. **Experimental Results.** As an example, we select eight targets in the monitor plane. The position coordinates of targets and the velocities are measured in the monitor plane. The physical units are neglected. The initial position coordinate of the laser beam point is [10,12] and the velocity magnitude is 5. The initial parameters of targets are shown in Table 1.

Table 1. Initial parameters of targets

Target	Position	Velocity	Target	Position	Velocity
1	[4,19]	[0.6,–1.1]	5	[15,18]	[0,–1]
2	[10,18]	[–0.2,–1]	6	[14,12]	[–0.5,–0.8]
3	[8,12]	[–0.3,–0.8]	7	[5,10]	[0.5,–0.7]
4	[18,10]	[–0.7,–0.4]	8	[2,14]	[0.8,–0.7]

Four initial permutations are generated at random and the maximum number of iteration, $n_{Max} = 30$. The proposed fast algorithm is executed and the simulation results are shown in Table 2. The attacking time profile with exchange times from each initial permutation is shown from Figure 2 to Figure 5, respectively.

Table 2. Simulation results

Initial number	Initial permutation	Optimal permutation	Minimum time	Exchange times	Computer time
P_0^1	[8,7,5,6,4,2,3,1]	[3,7,6,4,5,2,1,8]	6.39 s	17	0.00727 s
P_0^2	[2,3,5,6,4,7,8,1]	[3,2,5,4,6,7,8,1]	6.72 s	12	0.00564 s
P_0^3	[5,8,7,6,1,3,2,4]	[5,4,6,7,3,8,1,2]	6.28 s	25	0.01184 s
P_0^4	[2,6,3,5,8,1,4,7]	[3,7,8,1,2,5,4,6]	6.46 s	20	0.00822 s

The third local optimal permutation is the best solution from Table 2. The minimum attacking time is 6.28 s, while the global optimal time is 6.19 s, which is gat from computing all the permutations of eight targets during the time of 24.17 s. The local optimal permutation is $P_{opt} = [5, 4, 6, 7, 3, 8, 1, 2]$ and the trajectories of targets and the laser beam point are shown in Figure 6.

From Figure 2 to Figure 5, the maximum number of iteration is 25 and the local optimal solution is reached within 17 steps. And the whole computer time is 0.03172 s, which proves the fact that the optimal solution converges fast and this permutation algorithm can be applied in practical applications.

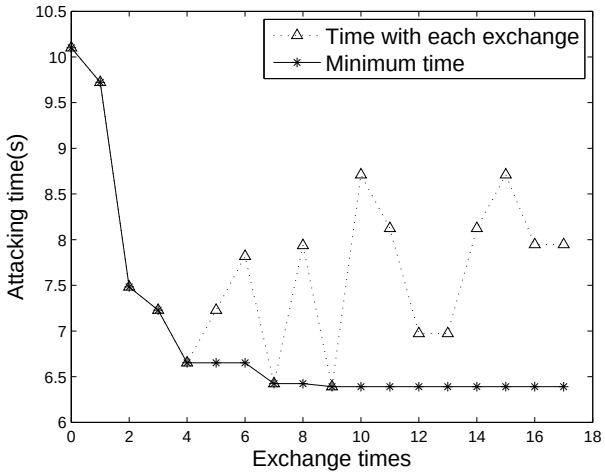

FIGURE 2. Cost profile with the first initial permutation

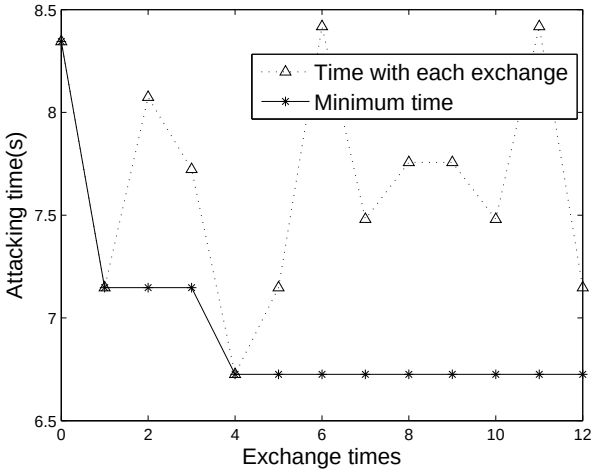

FIGURE 3. Cost profile with the second initial permutation

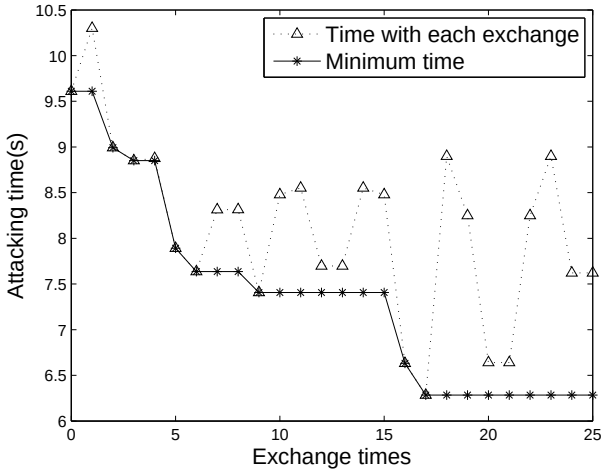

FIGURE 4. Cost profile with the third initial permutation

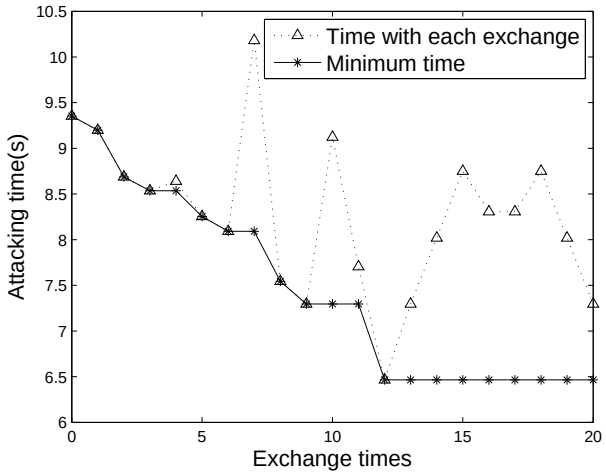

FIGURE 5. Cost profile with the fourth initial permutation

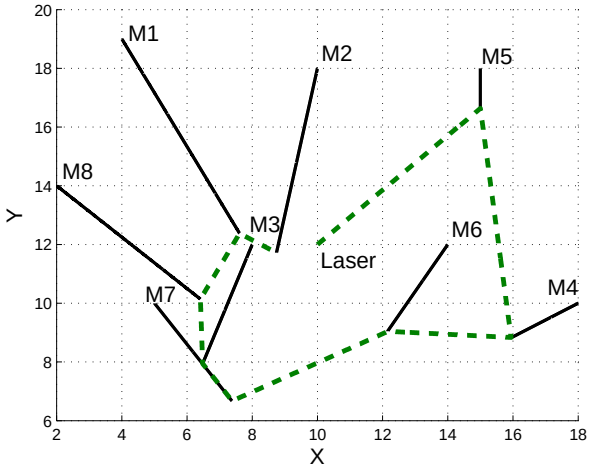

FIGURE 6. Trajectories with the optimal permutation

7. **Conclusion.** The fast permutation algorithm based on the exchange strategies is presented for solving the dynamic laser antimissile problem. Two essential exchange strategies are introduced in the algorithm. The adjacent exchange can inherit the merit of the permutation and the symmetric exchange has the capacity of the deep search in a wide region. They are combined to search in the both near and far regions, which can generate the better local optimal solution. The experimental results show that the algorithm converges fast and the local solution is very close to the global optimal solution.

The laser antimissile problem is hard to get the global optimal solution in the permitted time. Although this algorithm can get a good local optimal solution, there are some questions requiring attention in the future. When the laser beam point meets with the target, it requires some time to damage the target. This time should be taken into account in the model. In fact, more targets may come into the monitor plane as the laser attacking the previous targets according to someone permutation. This case with varying targets should be considered in practical implementation. Therefore, the other fast algorithms need be further studied for solving DLAP.

Acknowledgment. This work is supported by National Nature Science Foundation under Grant 60710002. The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation. Especially, the authors would like to thank Prof. Kok Lay Teo for suggestions on this algorithm and congratulations on his 65^{th} birthday.

REFERENCES

[1] J. Simpson, Boeing: 2009 shoot-down 'key to suvival' for airbone lasster program, *Inside the Air Force*, vol.19, no.13, pp.10-12, 2008.

[2] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, Freeman, New York, 1979.

[3] V. Viktor, G. K. Pierens and Q. M. Tieng, A variant of differential evolution for discrete optimization problems requiring mutually distinct variables, *International Journal of Innovative Computing, Information and Control*, vol.7, no.2, pp.897-914, 2011.

[4] P. Tian, J. Ma and D. M. Zhang, Application of the simulated annealing algorithm to the combinatorial optimization problem with permutation property: An investigation of generation mechanism, *European Journal of Operational Research*, vol.118, no.1, pp.81-94, 1999.

[5] M. R. Zhang, Y. Sun, W. W. Liu and X. Y. Gao, Modified simulated annealing algorithm for laser antimissile problem, *International Conference on Artificial Intelligence and Education*, Hangzhou, 2010.

[6] M. R. Zhang, Y. Sun and X. Y. Gao, Laser antimissile system strategy optimization based on genetic algorithm, *The 3rd International Symposium on Computational Intelligence and Design*, Hangzhou, pp.64-67, 2010.

[7] Y. Sun et al., Genetic algorithm with simulated annealing for laser antimissile optimization, *The 2nd International Conference on Information Science and Engineering*, Hangzhou, pp.1229-1232, 2010.

[8] M. Rashid and A. R. Baig, A genetic programming based adaptable evolutionary hybrid particle swarm optimization, *International Journal of Innovative Computing, Information and Control*, vol.6, no.1, pp.287-296, 2010.

[9] C. Sun, J. Zeng and J.-S. Pan, A modified particle swarm optimization with feasibility-based rules for mixed-variable optimization problems, *International Journal of Innovative Computing, Information and Control*, vol.7, no.6, pp.3081-3096, 2011.

[10] Y. Sun et al., Genetic algorithm based on permutation distance for laser antimissile problem, *Applied Mechanics and Materials*, vol.40, pp.488-493, 2011.

[11] W. W. Liu et al., Niche genetic algorithm for control strategy optimization in laser antimissile problem, *International Journal Sensing, Computing & Control*, vol.1, no.1, pp.1-10, 2011.