Journal of Engg. Research **Vol. 5 No. (1) March 2017 pp.** *124-139*

التحسين التوافقي والمحاكاة لمحفظة نظام السلاح باستخدام خوارزمية التكييف الذاتي

شان لينغ يانغ و كيدي هوانغ

كلية نظام المعلومات والادارة، الجامعة الدولية لتكنولوجيا الدفاع ، تشانشا، جمهورية الصين

الخـلاصــة

تعتبر مشكلة محفظة نظام السلاح مقيدة باعتبارها الحل الأمثل لمشكلة التوافقية النموذجية بهدف تحقيق أقصى قدر من الضرر المتوقع من الأهداف المعادية. ونظراً لتعقيد الحساب وضيق الوقت يقترح منهجية اتخاذ القرار بناءً على التكيف الذاتي كبديل لمساعدة القادة العسكريين في اتخاذ القرارات المناسبة. في هذا الإطار، فإن خوارزمية التكيف الذاتي الجينية أدت بحث عالمي لمنع الحصر في الامثلية المحلية الذي فيه إمكانية الاجتياز وإمكانية التبديل الفجائي يمكن تعديلها بشكل ديناميكي وفقاً لدرجة البكور من تطور السكان. بالإضافة تستخدم الأحداث التاريخية لإجراء البحث المحلي وضبطه في فضاء الحل. وتعد حالة الدراسة لتوضيح الإجراء بأكمله والتحقق من أداء الخوارزمية المقترحة لدينا. تبين التجارب المقارنة بأن أنظمتنا تفوق منافسيها فيما يتعلق بنوعية الحل وحساب الوقت. بالإضافة إلى ذلك، فإن محاكاة السيناريوهات واسعة النطاق هي مقياس لإثبات قابلية الخوارزمية لدينا.

Combinatorial optimization and simulation for weapon system portfolio using self-adaptive Memetic algorithm

Shanliang Yang and Kedi Huang

College of Information System and Management, National University of Defense Technology, Changsha, China

Corresponding author: yangshanliang@nudt.edu.cn

ABSTRACT

The weapon system portfolio problem is considered as a typical constrained combinatorial optimization problem with the purpose of maximizing the expected damage of hostile targets. Considering the computation complexity and the strict time constraints, a decision-making methodology based on self-adaptive Memetic algorithm is proposed as an alternative to help military commanders in making appropriate decisions. In this framework, self-adaptive genetic algorithm performs global search to prevent trapping into the local optima, where the crossover probability and mutation probability could be adjusted dynamically according to the prematurity degree of evolving population. Furthermore, problem-specific heuristics are utilized to conduct local search and fine-tuning in the solution space. A case study is given to illustrate the entire procedure and verify the performance of our proposed algorithm. Comparative experiments show that our algorithm outperforms its competitors with regard to solution quality and computation time. In addition, very large-scale scenarios are also simulated to demonstrate the scalability of our algorithm.

Keywords: Combinatorial optimization; genetic algorithm (GA); local search method; self-adaptive Memetic algorithm; weapon system portfolio.

INTRODUCTION

With the rapid development of military technologies, high-tech weapon systems have been regarded as crucial to determine the outcomes of battles. Hence it is a critical decision for commanders to make efficient combinatorial applications of numerous weapon systems on the battlefield (Lee et al., 2010). The weapon system portfolio problem refers to the proper assignment of weapons to engage hostile targets with the objective of maximizing the expected damage of all targets (Bogdanowicz, 2009). In most of the previous researches, this problem is also known as weapon-target assignment (WTA) problem in the applications of military operations research (Lee *et al*., 2002; Lee, 2010). In fact, the weapon system portfolio problem can be categorized as a constrained combinatorial optimization problem, which is proved to be NP-hard (Bogdanowicz & Patel, 2015).

Various approaches have been introduced for the weapon system portfolio problem. There are global optimum methods, such as implicit enumeration algorithms and branch-andbound algorithms. Later, some scholars also tried to employ heuristic algorithms to solve the weapon system portfolio problem, including neural networks (Silven, 1992), auction algorithm (Bogdanowicz *et al*., 2013), simulated annealing (SA) (Lee *et al*., 2003), genetic algorithm (GA) (Wang *et al*., 2008a) and ant colony optimization (ACO) (Wang *et al*., 2008b). However, these algorithms have been presented with some restrictive conditions (e.g., targets can receive at most one weapon). More importantly, they have showed the premature convergence and the slower convergence speed in high-dimensional problems. Thus, more sophisticated search algorithms and heuristics are intensely desired to solve the weapon system portfolio problem effectively. Since only the enumeration-based methods could guarantee the global solution (Mehmet & Kemal, 2014), each "optimal solution" in this paper refers to a feasible solution of the highest possible quality, which satisfies all the constraints.

Memetic algorithm is a population-based search method which combines global search and local refinements (Xue & Wang, 2015; Fraser *et al*., 2015). This marriage between global search and local search allows keeping high population diversity via strong mutation and increasing the convergence speed via the local search. The Memetic algorithm has been adopted to solve many optimization problems, such as the urban transit network optimization problem (Zhao *et al*., 2015), the patient transportation problem (Zhang *et al*., 2015), and the global path planning (Zhu *et al*., 2015). Therefore, it is very suitable to apply Memetic algorithm to solve the problem considered. In this paper, a self-adaptive Memetic algorithm is proposed to enhance the search capability for solving weapon system portfolio problem. Under the Memetic framework, genetic algorithm performs global search to prevent trapping into the local optima, while the problem-specific heuristics are utilized to conduct local search and fine-tuning in the solution space. The computational results indicate that our algorithm outperforms standard GA and self-adaptive GA in terms of execution time and solution quality. In addition, the experiments under different scales have validated the feasibility of the proposed algorithm. It can produce high-quality decisions in solving large-scale instances of weapon system portfolio.

The rest of this paper is organized as follows. In Section 2, a formulation of the mathematical model is presented for the weapon system portfolio problem. Section 3 describes the proposed self-adaptive Memetic algorithm. In Section 4, the simulation results are analyzed. Several existing approaches are also employed for comparison. The proposed method outperforms its competitors on all test cases. Finally, Section 5 concludes the paper.

PROBLEM DESCRIPTION AND MODELLING

The weapon system portfolio problem is a fundamental problem arising in the field of military operations research. The modeling issues about weapon system portfolio were first investigated in 1960s (Day, 1966). Since then, the mathematical models have been improved greatly. This problem involves assigning weapons to targets such that the total expected destroying effectiveness is maximized, subject to constraints. In fact, the weapon system portfolio models include various factors, such as defense strategies, features of weapons and targets, and actual combat environments. Different models may be constructed for different defense scenarios. For instance, Chen *et al*. (2009) proposed an asset-based optimization model with four kinds of constraints considered, including capability constraints, strategy constraints, resource constraints and engagement feasibility constraints. This asset-based model stresses on the protection of own-force assets, especially those important assets.

The scenarios considered in this paper are target-based and described as follows. At certain time, the defender detects N hostile targets with their objects of attack exposed, and there are M weapon systems available to intercept the offensive targets. Before the penetration of these targets, there is only a single stage in which the defender's weapons can engage with offensive targets. In order to formulate the weapon system portfolio problem, the following variables and notations are used.

M: The number of weapon systems.

N: The number of targets that are expected to be destroyed.

qi: The number of weapons for the ith weapon system.

wj: The threatening value of the jth target.

pij: The kill probability of the ith weapon system versus the jth target.

vij: The decision variable indicating the assigned number of weapons from the ith weapon system to the jth target.

The weapon system portfolio problem could be formulated as a non-linear integer programming model, with the objective of maximizing the total expected damage value of targets by using the above variables (Ni *et al.*, 2011)

$$max\ F(v)=\sum_{j=1}^{N} w_j\left[1-\prod_{i=1}^{M}(1-p_{ij})^{v_{ij}}\right] \tag{1}$$

$$Subject\,to\ \ \sum_{j=1}^{N} v_{ij}=q_i,\ i=1,2,\ldots,M \tag{2}$$

$$\sum_{i=1}^{M} v_{ij}\ \ldots 1,\ \ j=1,2,\ldots,N \tag{3}$$

$$v_{ij}\in \mathrm{int}^{+},\ i\in[1,M],\ j\in[1,N] \tag{4}$$

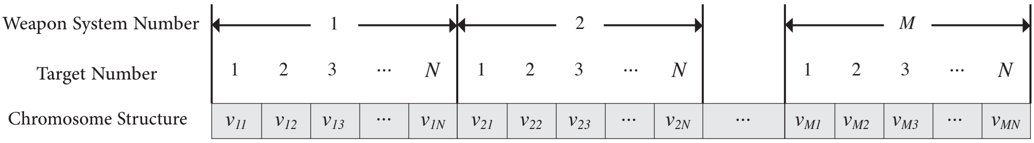
There are two assumptions, which are made for the formulation of weapon system portfolio problem. The first assumption is that the individual kill probability *pij* in Equation (1) is known for all $i \in M$ and $j \in N$. The kill probability is the weapon's probability of destroying the target if engaged. It depends on all aspects of engagement, such as the type of weapon and the type, state, and location (range, sector) of the target. In fact, the kill probability *pij* is a function of time t in the time window of the target *Nj*, during which the target *Nj* could be engaged with the weapon system *Mi*. However, in order to simplify the computation process, the static weapon system portfolio problem is discussed in this paper, i.e., *pij* is constant and the defensive weapons are assigned to these targets at a fixed time. The other assumption is that all the weapons of different weapon systems must be assigned to targets, as illustrated in Equation (2). Constraint set in Equation (3) guarantees that each hostile target would be assigned at least one weapon unit from all the weapon systems. The condition Equation (4) provides a constraint that the decision variable *vij* is nonnegative and discrete.

STRATEGY OF SELF-ADAPTIVE MEMETIC ALGORITHM

This paper presents a self-adaptive Memetic algorithm for optimization and simulation of weapon system portfolio, which can adjust crossover probability and mutation probability according to the observed performance. Moreover, local search method is introduced into this algorithm for the purpose of improving search capability. The detailed procedures of this method are described as follows.

CHROMOSOME CODE REPRESENTATION AND INITIALIZATION

Chromosome encoding is to design a genetic string for problem variables, on which genetic operators can be performed feasibly and effectively (Gallardo & Cotta, 2015). In general, the encoding rules have a high correlation with the natural characteristics of the problem considered. Therefore, according to the features of weapon system portfolio problem, a well-designed chromosome representation using integral numbers is brought forward, which is illustrated in Figure1.



Weapon System Number	1					2						M				
Target Number	1	2	3	···	N	1	2	3	···	N		1	2	3	···	N
Chromosome Structure	v_{11}	v_{12}	v_{13}	···	v_{1N}	v_{21}	v_{22}	v_{23}	···	v_{2N}	···	v_{M1}	v_{M2}	v_{M3}	···	v_{MN}

Fig.1 Chromosome encoding for weapon system portfolio problem

An individual chromosome represents an accurate solution to the weapon system portfolio problem. In the chromosome structure, *vij* identifies the allocated number of weapons from the ith weapon system with respect to the jth target, which is not greater than the available number of weapons *qi* of the ith weapon system. The entire length of a chromosome is *M*N*, and the encoding structure satisfies the three constraints expressed in Equations (2), (3) and (4) in order to make chromosomes feasible. Chromosomes in the initial population are all generated randomly under aforementioned conditions.

Fitness function

The fitness function is based on the degree of maximizing the overall destroying effectiveness of all targets. Consequently, the fitness function could be obtained by modifying the objective function, which is showed in Equation (5)

$$f(x) = \sum_{j=1}^{N} w_j \left[1 - \prod_{i=1}^{M} (1 - p_{ij})^{v_{ij}(x)}\right] \tag{5}$$

where x denotes a specific chromosome, and it is obvious that the fitness value *f(x)* of chromosome x reflects its performance among the population. In fact, the objective function in Equation (1) could be transformed into linear form or made less nonlinear by introducing auxiliary variables and constraints. However, the linearization process will only affect the arithmetical calculations of the fitness function expressed in Equation (5). And it has no influence on the main framework and the whole procedures of our proposed algorithm. Thus, we do not utilize any linearization techniques in our paper.

Self-adaptive genetic operators

Self-adaptive crossover and mutation operators

The self-adaptive crossover and mutation operators are employed in this paper. The crossover probability *pc* and mutation probability *pm* remain unchanged in traditional genetic algorithm *(He et al., 2015)*. However, these two parameters have strong influence on the overall performance of the algorithm, and they should be adjusted appropriately according to the current status of population. Thus, this paper puts forward a self-adaptive mechanism for modifying *pc* and *pm* dynamically. First of all, we define the prematurity degree of population as follows.

$$\beta = (f_{max} - f'_{avg}) / f_{max} \tag{6}$$

where β denotes the prematurity degree of population, *fmax* represents the maximum fitness value of all chromosomes, and *f'avg* is the average fitness value of those chromosomes, whose fitness values exceed the average fitness of the whole population. Then the self-adaptive method for *pc* and *pm* corresponding to the prematurity degree β is explained as.

$$p_c = 1/(1+exp(-k_1 \cdot \beta)) \tag{7}$$

$$p_m = 0.4 - 0.4/(1+exp(-k_2 \cdot \beta)) \tag{8}$$

where *k1* and *k2* are two adjustable nonnegative parameters based on the experience. The smaller the value of β is, the more convergent the population is. It is easy to find that the range of *pc* belongs to the interval (0.5, 1) and *pm* falls into the interval (0, 0.2). The value of β becomes smaller with the evolution of the population. According to Equation (7) and Equation (8), we can note that *pc* decreases and *pm* increases based on a decrease in the β value. It is reasonable to increase *pm* and decrease *pc* in order to expand search space along with the maturity of evolutional population. If *pc* is too large and the population updates too often, the individuals with high fitness values may be destroyed quickly. If *pm* is too small, it may be unable to produce new individuals to increase the population diversity.

A two-point crossover operator is introduced for exploring search space in this section. In order to satisfy the constraint expressed in Equation (2), the whole genetic part for the randomly selected weapon system is replaced by the part at the same position in another parent to generate offspring. In a same manner, the mutation operator is performed by re-assigning the number of weapons to all targets for the randomly selected weapon system. Figure.2 shows an example of crossover and mutation operators for chromosome *P1* and *P2*. The ith weapon system is selected randomly. Then the encoding parts for the ith weapon system in parents *P1* and *P2* interchange with each other. The mutation operator for parent *P2* is implemented by re-assigning the qi weapons to N targets in order to ensure that the offspring chromosomes are all feasible.

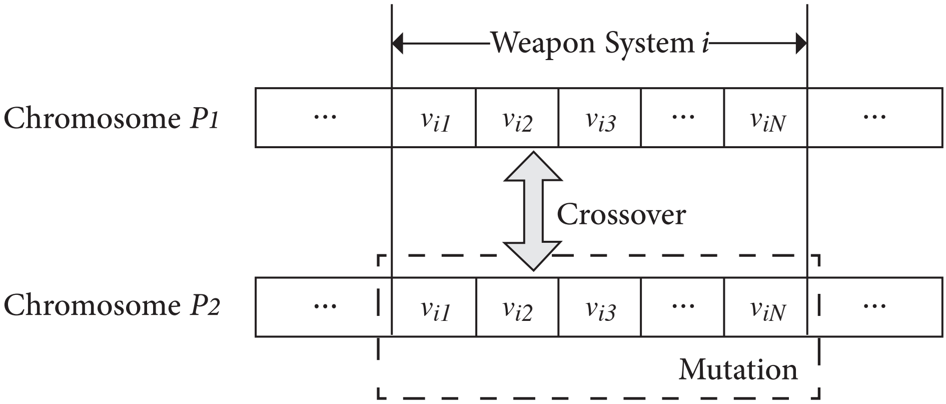


Fig. 2. An example of crossover and mutation operations

Elite preserving selection operator

The main idea of selection operator is that the better individuals with higher fitness values will have more opportunities to generate their offspring. Normally, roulette wheel selection is usually adopted as selection operator in most studies. However, at the early stage of genetic algorithm, the population has great dispersion, where differences among individuals are always large. Hence, the direct adoption of roulette wheel selection will cause a relatively small selection pressure and the convergence rate of algorithm will become very slow (Lu *et al*., 2015).

With the purpose of maintaining the best solution to the terminal of algorithm and finding the global optimum, the elite preserving selection operator (Mahdavi *et al*., 2011) is employed in this paper. The pre-defined N_e best chromosomes of the current generation are selected into the next generation without any changes, while the rest of chromosomes will still be selected using roulette selection operator. The advantage of this method is that it will assure the best performance of the next generation will not be worse than that of the current generation. It is worth noting that the value of N_e should be identified carefully. If N_e is too large, the chromosomes with higher fitness values will occupy the vast majority in the next generation, leading to premature convergence.

Local search operation

In this section, the local search approach is utilized to enhance the search efficiency for solving weapon system portfolio problem. In this way, the problem-specific heuristics can be embedded into the search process to obtain better solutions. In order to perform local search operation, heuristic information must be extracted according to the prior knowledge. For the weapon system portfolio problem, the heuristic information Ψ can be defined as

$$\Psi = \{\psi_1, \psi_2, \ldots, \psi_M\} = \{(\max_{j} p_{ij} \mid j \in N)\} \tag{9}$$

Notation ψi, i∈*M* in Equation (9) represents the maximum kill probability of the ith weapon system to all targets. It is believed that, if a weapon system owns the highest kill probability to a specific target, the specific target should be assigned the most weapons from this weapon system. Based on this idea, the local search operation could be depicted in Figure.3.

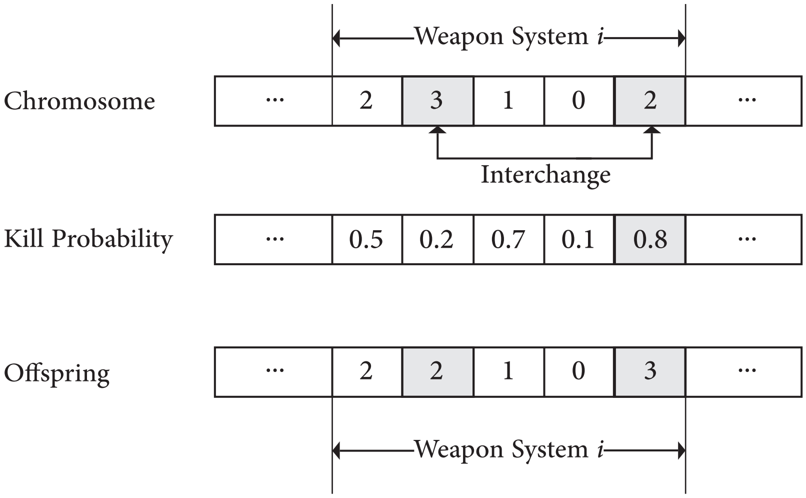


Fig. 3. An example of local search operation

The chromosome is chosen randomly to receive local search operation. Assume that the position of the ith weapon system is chosen randomly. We note that the heuristic knowledge of the ith weapon system ψi is 0.8. However, the corresponding target is not assigned the most weapon units from the ith weapon system. Hence, the local search operator is to exchange the assigned number of weapons at the ψi position with the highest value. This operation occurs within the same weapon system to satisfy the constraint in Equation (2).

Pseudo code for self-adaptive Memetic algorithm

Let Np be the size of population and Ne be the number of elites. Let Ng be the maximum generation limit and Fm be the highest fitness value of all individuals. Then, the pseudo code for the proposed self-adaptive Memetic algorithm can be given as in Table 1.

Table 1. Pseudo code for self-adaptive Memetic algorithm

```
Algorithm: Self-adaptive Memetic algorithm for weapon system portfolio problem
Begin
$N_p \leftarrow PopSize$ ;      % The number of individuals in the population
$N_g \leftarrow Max\_Gen$ ;     % The maximum generation limit
$N_e \leftarrow Elite\_Num$;    % The number of preserving elites
$t \leftarrow 1, k_1 \leftarrow 5, k_2 \leftarrow 2$ ;
$F_m \leftarrow 0$ ;            % The highest fitness value
Initialize all individuals of the population P(t);
Evaluate fitness values f(t) of all the individuals;
Extract heuristic information $\Psi$ from prior knowledge;
While $t,, N_g$ do
    $\beta(t) \leftarrow (f(t)_{max} - f'(t)_{avg}) / f(t)_{max}$ ;      % The convergence degree
    $p_c(t) \leftarrow 1/(1+exp(-k_1 \cdot \beta(t)))$ ;                  % Update $p_c$ and $p_m$
    $p_m(t) \leftarrow 0.4 - 0.4/(1+exp(-k_2 \cdot \beta(t)))$ ;
    Perform elitist selection to obtain elite population P'(t);
    Perform crossover operation with crossover probability $p_c(t)$;
    Perform mutation operation with mutation probability $p_m(t)$;
    Execute local search operation with heuristic information $\Psi$;
    Evaluate fitness values f(t) of all the individuals;
    If $F_m < max(f(t))$ Then
        $F_m \leftarrow max(f(t))$
    End IF
    $P(t+1) \leftarrow P(t) + P'(t)$ ;     % The number of individuals remains unchanged
    $t \leftarrow t+1$ ;
End While
End
```

It can be observed from pseudo code that the elitist scheme is utilized in this paper. The best *Ne* individuals of each population are copied unaltered to the next generation, while the remaining individuals are selected based on their fitness values (Srinivasa *et al*., 2007). The usage of elitist scheme ensures that the best individual of any generation is at least as good as that of the previous generation, which is helpful in achieving global convergence.

Experimental Results and Analysis

In order to verify the performance of our proposed hybrid Memetic algorithm, we performed a series of tests on problem instances that some were generated randomly. Regarding the programs used in the computational experiments, all algorithms are coded in Matlab in a Windows 7 environment. With the purpose of making a fair comparison, all executions for different algorithms are performed on the same machine (ThinkPad SL400 computer, Intel(R) Core(TM) 2 T6670

2.2GHz processors, 3GB RAM). In this section, a ground-based air defence scenario is considered. Suppose that there are 5 hostile aircraft on their way to attack our radar station. The radar station has 4 surface-to-air missile (SAM) launchers to defend itself. Assume that the number of missiles for each SAM launcher is 4, 3, 2 and 5, respectively. And the threatening value vector for offensive aircrafts is [3, 1, 5, 4, 8]. In addition, the kill probability of each type of SAM versus aircrafts is listed in Table. 2.

Table 2. The kill probability for each type of SAM versus aircrafts

SAM	Aircraft				
	1	2	3	4	5
1	0.2	0.6	0.1	0.9	0.7
2	0.8	0.5	0.5	0.7	0.3
3	0.3	0.7	0.8	0.6	0.5
4	0.7	0.5	0.4	0.2	0.9

In our first scenario, the following parameters are set as default values: N_p=100, N_g=500, N_e=2, k_1=5, k_2=2. After several trials of our algorithm, the best found solution value is discovered to be 20.736 with an average computation time of 4.62s, where the ideal objective value is 21. In fact, it is verified that this is indeed the global optimal value by using a computer program in LINGO software. Furthermore, the corresponding allocated matrix for SAMs is V = [0,2,0,2,0|2,0,1,0,0|0,0,2,0,0 |1,1,0,0,3], as illustrated in Figure 4. The designed self-adaptive crossover and mutation operators can be explained in Figure 5. Note that the crossover probability p_c decreases with the rise of average fitness value, while the adjustment of mutation probability pm coincides with the variation trend of average fitness value at the early stage of population evolution. Then the values of pc and p_m hover around 0.54 and 0.193, respectively. Obviously, it is reasonable to increase p_m and decrease p_c to expand search space with the convergence of population.

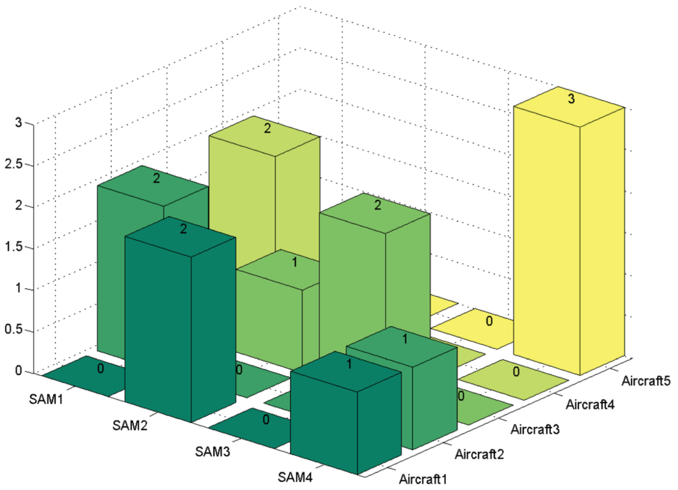


Fig. 4 , The computational result for air defense scenario

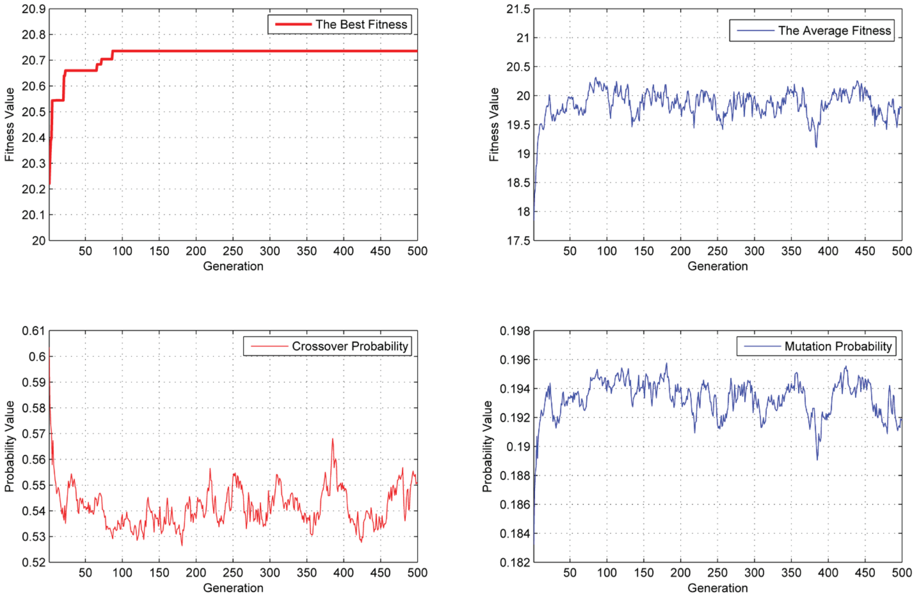


Fig. 5. The variation of pc and pm with fitness curves

Sensitivity analysis

The influence of preserving elite number *Ne* on the convergence rate of population has been highlighted in the procedures of proposed self-adaptive Memetic algorithm. It is worth noting that the value of *Ne* should be identified carefully. If Ne is too large, the chromosomes with higher fitness values will occupy the vast majority in the next generation, leading to premature convergence. If *Ne* is too small, the global optimal solution might not be found within a limited time. The sensitivity analysis of preserving elite number *Ne* is conducted and the results are shown in Figure 6.

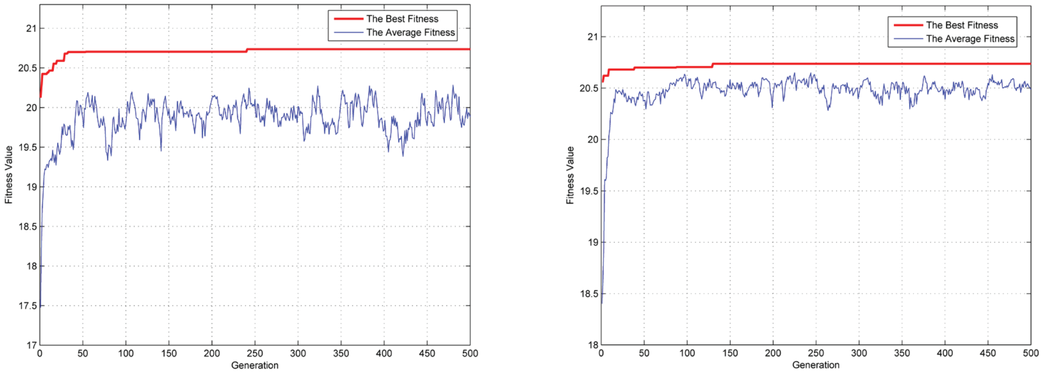


Fig. 6 (a). Convergence curves for elites number $Ne = 2$; (b) elites number $Ne = 8$

The default parameters remain unchanged, while the elite number Ne is set to be 2 and 8, respectively. It is evident that the convergence rate accelerates apparently along with the increase of elite number. Although the global optimal solution is discovered in both trials, the elite number Ne is identified to be 2 cautiously in our experiments. This will ensure the algorithm is not likely to be trapped into the local optima.

Comparative analysis

A comparative analysis of our algorithm with standard Genetic Algorithm (SGA) and self-adaptive Genetic Algorithm (SA_GA) is conducted to show the superiority of proposed self-adaptive Memetic algorithm (SA_Memetic). Here, SA_GA is implemented by integrating the self-adaptive strategy proposed in this paper with SGA. Note that the crossover probability pc and mutation probability pm are fixed values in SGA ($pc = 0.75, pm = 0.1$). The same scenario is used in this experiment. The best fitness curves by generations of these three algorithms are depicted in Figure 7 (a). It is worth noting that the computational times required for each generation are different for different algorithms. It can be observed that those curves can all converge to the best fitness value. However, the convergence rate of SA_Memetic is obviously faster than the other two algorithms.

In addition, 100 trials are simulated for each algorithm within limited generations to observe whether the algorithm can find the global optimal solution. The maximum number of generations is set to 500, and the statistical results are shown in Figure 7 (b). The successful numbers for SGA, SA_GA and SA_Memetic are 38, 46 and 79, respectively. Evidently, the search efficiency of the proposed self-adaptive Memetic algorithm is obvious, and our algorithm outperforms SGA and SA_GA greatly in terms of convergence rate and solution robustness.

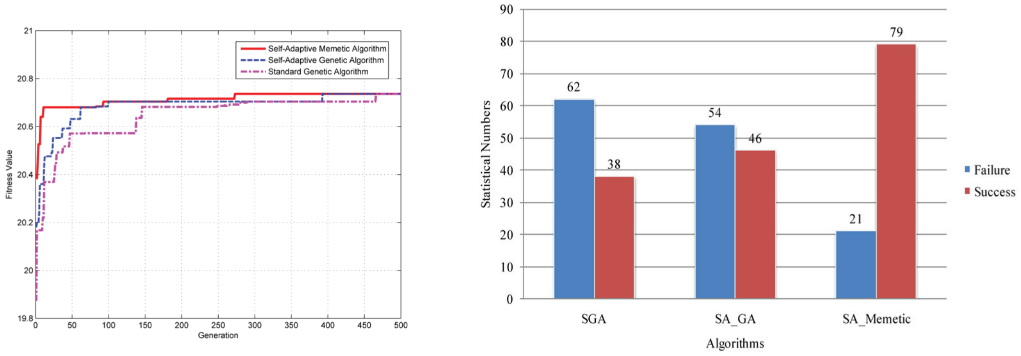


Fig. 7(a) The optimal fitness curves for the three comparative algorithms; (b) the statistics of successes and failures for discovering global optimal solution

Scalability analysis for large-scale cases

In this section, 10 test cases are used to illustrate the scalability performance of proposed self-adaptive Memetic algorithm. The number of weapon systems M and the number of targets N vary from 10 to 100 in these cases. In other words, the test data includes the following cases:

M=10, N=10(No.1), M=20, N=20(No.2), M=30, N=40(No.3), M=40, N=50(No.4)

M=50, N=50(No.5), M=60, N=70(No.6), M=70, N=80(No.7), M=80, N=90(No.8)

M=90, N=100(No.9), M=100, N=100(No.10)

For each case, the kill probability pij, i$\in M, j\in N$ are generated randomly in the interval (0,1), the available number of weapons for each weapon system qi, i$\in M$ is a random integral number generated from the interval (1,5), and the threatening values of every target wj, j$\in N$ are produced in the interval (1,10) at random. Our algorithm runs 10 times for the decision making on each instance. In addition, in order to report results in tables and figures uniformly, we calculate the relative percentage fitness (RPF) of the solution obtained from the best result for the corresponding instance, defined as $\mathrm{Sol}_{\mathrm{optimal}}\times 100/\mathrm{Sol}_{\mathrm{Target}}$. The numerical results are presented in Table.3, including the ideal target value, the optimal fitness value, RPF, and computation time plus its corresponding standard deviation. Figure 8 (a) shows the results of these experiments as box plots for the RPF from best solutions of results in each case. The minimum and maximum time cost for each case by using our algorithm is shown in Figure 8 (b).

Table 3. Statistical results on scalability test

No.	Target value	Optimal value	RPF(%)	Computation time(s)
1	69	68.93	99.90	6.37±0.14
2	114	113.81	99.84	12.55±0.27
3	226	220.13	97.40	25.84±0.08
4	300	294.51	98.17	39.00±0.26
5	282	281.03	99.66	48.02±0.22
6	391	382.65	97.86	74.41±0.49
7	385	378.14	98.22	99.08±0.36
8	461	453.78	98.43	148.47±0.67
9	551	543.20	98.58	200.28±0.65
10	541	535.17	98.92	226.47±0.45

According to the statistical results, it can be observed that our algorithm could find a good feasible solution after a fixed times of iteration. The relative percentage fitness (RPF) obtained for each instance exceeds 97.40 and it is believed that this result could be still improved, if abundant decision-making time is available. As seen from Figure 8 (a), the solutions for each case remain stable within the interval (0.96, 1), indicating the robustness of our algorithm. When solving the instances No.1~No.7, the computation time required is less than 2 min, which is acceptable for military commanders. However, with the increasing of problem scales, the time cost grows greatly, which may be unbearable for practical real-time decision making problems. In the instance No.10

with 100 weapon systems and 100 targets, the algorithm took almost 4 min to make a final decision. Although the time cost is disappointing for very large-scale instances in the test, it can be reduced by the utilization of more advanced computing platforms. Hence, the self-adaptive Memetic algorithm proposed in this paper is competent for solving weapon system portfolio problems.

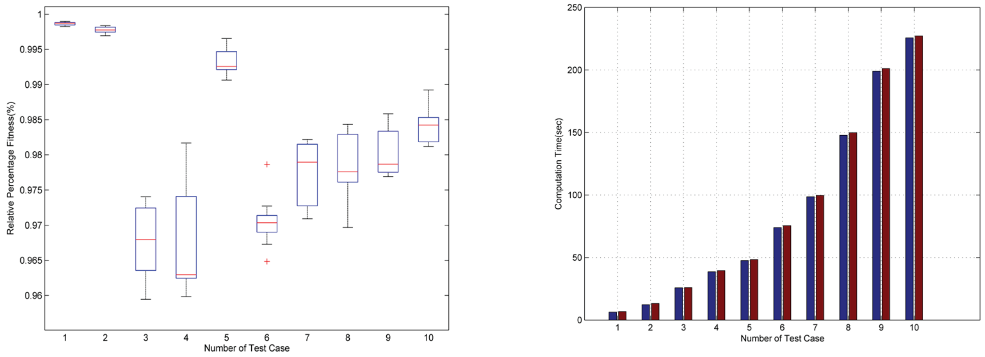


Fig. 8(a) Box plots for RPF values; (b) the minimum and maximum computation time

CONCLUSION

In this work, a self-adaptive Memetic algorithm is presented to solve the weapon system portfolio problem incorporating resources constraints. The objective of this Memetic framework is to assign weapons to targets in such a way that the total expected damage is maximum. Our algorithm consists of two phases. In Phase 1, the genetic algorithm performs global search to prevent trapping into the local optima. In Phase 2, the problem-specific heuristics are utilized to conduct local search and fine-tuning in the solution space.

The computational results indicate that our algorithm outperforms standard GA and self-adaptive GA in terms of execution time and solution quality. In addition, the experiments under different scales have validated the feasibility of the proposed algorithm. It can produce high-quality decisions in solving large-scale instances of weapon system portfolio. The time cost in scalability test may be unacceptable for some real-time military applications. Consequently, future works could apply some effective heuristic methods within our approach to reduce computation time. Furthermore, some advanced mechanisms and techniques (e.g., parallel computing) may be employed to enhance our algorithm's efficiency and scalability.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interests regarding the publication of this paper.

ACKNOWLEDGEMENTs

This research was supported by National Natural Science Foundation of China under Grant No.61074108 & 61374185.

REFERENCES

Lee, J., Kang, S.H., Jay, R. & Kim, S.B. 2010. A hybrid approach of goal programming for weapon systems selection, Computers & Industrial Engineering, **58**:521-527.

Bogdanowicz, R.Z. & Patel, K. 2015. Quick collateral damage estimation based on weapons assigned to targets, IEEE Transactions on Systems, Man, and Cybernetics: Systems, **45**:762-769.

Lee, Z.J., Su, S.F. & Lee, C.Y. 2002. A genetic algorithm with domain knowledge for weapon-target assignment problems, Journal of the Chinese Institute of Engineers, **25**:287-295.

Bogdanowicz, R.Z. 2009. A new efficient algorithm for optimal assignment of smart weapons to targets, Computers and Mathematics with Applications, **58**:1965-1969.

Lee, M.Z. 2010. Constrained weapon-target assignment: enhanced very large scale neighborhood search algorithm, IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans, **40**:198-204.

Silven, S. 1992. A neural approach to the assignment algorithm for multiple-target tracking, IEEE Journal of Oceanic Engineering, **17**: 326-332.

Bogdanowicz, R.Z., Antony, T., Patel, K. & Coleman, P. 2013. Optimization of weapon-target pairings based on kill probabilities, IEEE Transactions on Cybernetics, **43**:1835-1844.

Lee, Z.J., Su, S.F. & Lee, C.Y. 2003. Efficiently solving general weapon-target assignment problem by genetic algorithms with Greedy Eugenics, IEEE Transactions on Systems, Man, and Cybernetics-Part B: Cybernetics, **33**:113-120.

Wang, W., Cheng S. & Zhang Y. 2008a. Research on approach for a type of weapon target assignment problem solving by genetic algorithm, Systems Engineering and Electronics, **30:**1708-1711.

Wang, Y., Qian L., Guo Z. & Ma L. 2008b. Weapon target assignment problem satisfying expected damage probabilities based on ant colony algorithm, Journal of Systems Engineering and Electronics, **19**:939-944.

Mehmet, A.S. & Kemal, L. 2014. Approximating the optimal mapping for weapon target assignment by fuzzy reasoning, Information Sciences, **255**:30-44.

Xue, X. & Wang, Y. 2015. Optimizing ontology alignments through a Memetic algorithm using both MatchFmeasure and Unanimous Improvement Ratio, Artificial Intelligence, **223**:65-81.

Fraser, G., Arcuri, A. & McMinn, P. 2015. A Memetic algorithm for whole test suite generation, The Journal of Systems and Software, **103**:311-327.

Zhao, H., Xu, W. & Jiang, R. 2015. The Memetic algorithm for the optimization of urban transit network, Expert Systems with Applications, **42**:3760-3773.

Zhang, Z., Liu, M. & Lim, A. 2015. A Memetic algorithm for the patient transportation problem, Omega, **54**:60-71.

Zhu, Z., Wang, F., He, S. & Sun, Y. 2015. Global path planning of mobile robots using a Memetic algorithm, International Journal of Systems Science, **46**:1982-1993.

Day, R.H. 1966. Allocating weapons to target complexes by means of non-linear programming. Operations Research, **14**:992–1013.

Chen, J., Xin, B., Peng, Z., Dou, L. & Zhang, J. 2009. Evolutionary decision-makings for the dynamic weapon-target assignment problem. Science in China Series F: Information Sciences. **52(11)**:2006–2018.

Ni, M., Yu, Z., Ma, F. & Wu, X. 2011. A lagrange relaxation method for solving weapon-target assignment problem, Mathematical Problems in Engineering, **2011**: 1-10, Article ID: 873292, doi:10.1155/2011/873292.

Gallardo, J. & Cotta, C. 2015. A GRASP-based Memetic algorithm with path relinking for the far from most string problem, Engineering Applications of Artificial Intelligence, **41**:183-194.Name is different in text

He C., Xing J., Li, J., Yang, Q., Wang, R. & Zhang, X. 2015. A new optimal sensor placement strategy based on modified modal assurance criterion and improved adaptive genetic algorithm for structural health monitoring, Mathematical Problems in Engineering, **10**:1-10.

Lu, H., Wen, X., Lan, L., An, Y. & Li, X. 2015. A self-adaptive genetic algorithm to estimate ja model parameters considering minor loops, Journal of Magnetism and Magnetic Materials, **374**:502-507.

Mahdavi, I., Movahednejad, M. & Adbesh, F. 2011. Designing customer-oriented catalogs in E-CRM using an Effective Self-adaptive Genetic Algorithm, Expert Systems with Applications, **38**:631-639.

Srinivasa, K.G., Venugopal, K.R. & Patnaik, L.M. 2007. A self-adaptive migration model genetic algorithm for data mining applications, Information Sciences, **177**:4295-4313.