混合式基因演算法應用於武器對目標指派問題

李仁鐘 [1] 李秋緣 [2] 蘇順豐*[3]

[1] 康寧護理專科學校資訊管理科
[2] 蘭陽技術學院資訊管理科
[*3] 國立台灣科技大學電機工程學系
Tel: (02)27376704; Fax (02) 27376699
E-mail: su@mig1.ee.ntust.edu.tw

摘要

本文提出使用混合式之基因演算法於武器對目標指派問題。文中混合式之基因演算法乃是採用免疫系統於基因演算法中。而武器對目標指派問題乃是 NP-Complete 類型問題，其目的乃是指派我方所有武器攻擊敵方目標並降低我方的損失。本文提出新的基因配對方法及免疫系統演算法則，以提高武器對目標指派問題的求解收斂速度，文中亦將使用知識之基因演算法的模擬結果與其他演算法則相比較，其數據顯示使用知識之基因演算法優於其他演算法則。

關鍵詞 ： 基因演算法則，武器對目標指派問題，免疫系統

A Hybrid Genetic Algorithm Applied to Weapon-Target Assignment Problem*

Zne-Jung Lee [1], Chou-Yuan Lee[2], and Shun-Feng Su[#3]

[1] Dept. of Information Management
Kang-Ning Junior College of Nursing

[2] Dept. of Information Management
Lang-Yang Institute of Technology

[#3]Dept. of Electrical Engineering
National Taiwan Univ. of Science and Technology
Tel: (02)27376704; Fax (02) 27376699
E-mail: su@mig1.ee.ntust.edu.tw

Abstract

In this paper, a hybrid genetic algorithm by combining a genetic algorithm with an immune system is proposed and applied to weapon-target assignment (WTA) problem. The WTA problem, known as a NP-Complete problem, is to find a proper assignment of weapons to targets with the objective of minimizing the expected damage of own-force asset. The used immune system serves as a local search mechanism for genetic algorithm. Besides, in our implementation, a new crossover operator called the elite preserving crossover is proposed to preserve good information contained in the chromosome. A comparison of the proposed algorithm with several existing search approaches shows that the proposed algorithm outperforms its competitors on all tested WTA problems.

Keywords—Optimization, Genetic Algorithm, Immune system, Weapon-Target Assignment.

1. Introduction

On modern battle fields, there are vague and uncertainties in determining the proper weapons while engaging with targets. However, it is an important task for a planner to make a proper weapon-target assignment (WTA) in front of threat targets. The WTA problem is to find a proper assignment of weapons to targets with the objective of minimizing the expected damage value of own-force asset. It has been shown that a WTA problem is a NP-compete problem, and it is difficult to solve this type of problems directly [4,37]. Various methods for solving optimization problems have been reported in the literature [3,28,29,38,40]. Those methods are based on graph search approaches and usually result in exponential computational complexities. As a consequence, it is difficult to solve this type of problems directly while the number of targets or weapons are large [16,28,40]. Recently, genetic algorithms (GAs) have been widely used as search algorithms in various applications and have demonstrated satisfactory performances [7,11,20,41,42]. GA simulates the evolution of individual structures for optimization inspired by natural evolution [8,23,39,44]. There are two genetic operators, crossover and mutation, giving chromosomes the opportunity to evolutionarily search for the optima in the space. If these two genetic operators were not carefully designed, the search might become inefficient or even random. This may cause certain degeneracy in the search performance [22,30]. That is why it is difficult to applied simple GAs into many complicated optimization problems successfully. GAs used in real-word optimization problems usually need to incorporate problem-specific knowledge into evolutionary operators [23]. In [43], we have employed simple GAs to solve WTA problems. Even though they could find the best solution in those simulated cases, the search efficiency seemed not good enough.

The idea of local search is to find a better candidate nearby the current one before move to the next stage of search. There exist approaches that adopt the local search idea into their genetic algorithms. In the literature, a genetic algorithm equipped with a local search approach may be referred to as the memetic algorithm, the genetic local search, the hybrid genetic algorithm, or the cultural algorithm [2,6,30,25]. In those approaches, they viewed such algorithms as the evolution of ideas. The ideas, combined the genetic operators with heuristics, can quickly search the interested solution space to find better solutions [24,30,31,33]. Ideas can be recombined to create new ideas, and good ideas are more useful than weak ones. In this paper, we proposed to introduce a genetic algorithm with an immune system to improve the search efficiency. The immune system is a remarkable adaptive system and the corresponding concept

* This work was supported in part by the Chung-Shan Institute of Science Technology of Taiwan under the grant XU89A59P and in part by the National Science Council of Taiwan, R.O.C. under the grand NSC-89-2218-E-011-002.

can provide several important ideas in the field of computation [1,9,35]. Biologically, the function of immune systems is to protect a body from antigens. In combining with GAs, it is to improve search ability during the evolutionary process [22]. Besides, in our algorithm, a new crossover operator, the elite preserving crossover (EX) is also proposed. The idea of this operator is to preserve only good information contained in both parents for the next generation. Several cases are simulated in our study, and the results demonstrated nice efficiency of our approach while compared to other algorithms.

The paper is organized as follows. In section 2, a mathematical formulation of WTA problems is introduced. The pseudo algorithms of general genetic algorithms with local search are described in section 3. The proposed algorithm is presented and discussed in section 4. In section 5, the results of employing the proposed algorithm to solve WTA problems are presented. Several existing algorithms were also employed for comparison. The performance showed the superiority of our algorithm. Finally, section 6 concludes the paper.

2. The WTA Problem

It is an important task for battle managers to make a proper weapon-target assignment (WTA) in front of threat targets. For an example of anti-aircraft weapon (AAW) of naval battle force platforms, the threat targets may be enemy missiles launched form surface ships, aircraft and submarines. These missiles have different probabilities of killing to platforms which dependent on the missile types, target types, etc. In this situation, battle managers may use information gathered from radars to assign proper weapons to destroy these targets. But, that is not an easy task for planners to make a proper weapon-target assignment with such information. Thus, a decision-aided system for WTA problems is strongly desired in helping and training planners to make proper decisions on the battlefield [4].

To solve WTA problems, the following assumptions are made for the formulation of the WTA problem. The first one is that there are W weapons and T targets and all weapons must be assigned to targets. The second assumption is that the individual probability of killing (K_{ij}) by assigning the i-th target to the j-th weapon is known for all i and j. This probability defines the effectiveness of the j-th weapon to destroy the i-th target. The overall probability of killing (PK) value for a target (i) to damage the asset can be computed as:

$$PK(i)=\prod_{j=1}^{W}(1-K_{ij})^{X_{ij}} , \qquad (1)$$

where X_{ij} is a Boolean value indicating whether the j-th weapon is assigned to the i-th target. $X_{ij}=1$ indicates that the j-th weapon is assigned to the i-th target. The considered WTA problem is to minimize the following fitness function [43,37]:

$$C(\pi)=\sum_{i=1}^{T} EDV(i) *PK(i), \qquad (2)$$

subject to the assumption that all weapons must be assigned to targets; that is,

$$\sum_{i=1}^{T} X_{ij} = 1,\ j=1,2,\ldots,W. \qquad (3)$$

Here $EDV(i)$ is the expected damage value of the i-th target to the asset. π is a feasible assignment list and $\pi(j)=i$ indicates weapon j is assigned to target i.

3. Genetic Algorithms with Local Search

Since the WTA problem has been shown to be NP-Complete [10,37], it is difficult to solve this kind of problem directly while the number of targets and weapons are large. Based on the concept of natural selection, GAs search for the solution that optimizes a given fitness function. In GAs, the variables are represented as genes on a chromosome. Through natural selection and genetic operators, chromosomes with better fitness are found. However, if genetic operators were not carefully designed, it might be difficult to have effective search. In our previous study [43], we have also employed GAs to solve WTA problems and the search efficiency of that algorithm seemed not good enough.

Recently, genetic algorithms with local search have been considered as good alternatives for solving optimization problems [14,18,32,34]. In general, GA with local search consists of five steps. The first step is to initialize the population by randomly selecting a set of chromosomes encoding possible solutions. The second step is to evaluate the fitness of each chromosome in the population. The third step is to create new chromosomes by mating current chromosomes with the use of recombination and mutation operations. The fourth step is to apply local search to create new chromosomes, and then evaluate the new chromosomes. The fifth step is to select the chromosomes with the best fitness from the population. It should be noticed that the algorithm becomes a general GA if the fourth step is omitted.

The general structure of the GA with local search is shown in the following where *P(t)* and *C*(*t*) are parents and offspring in generation *t* [23, 24,30].

```
Procedure:    GA with local search
    Begin
        $t \leftarrow 0$;
        Initialize $P(t)$;
        Evaluate $P(t)$;
        While (not match the termination conditions) do
            Recombine $P(t)$ to yield $C(t)$ with possible mutation;
            Apply local search for $C(t)$;
            Select $P(t+1)$ from $P(t)$ and $C(t)$ based on the fitness;
            $t \leftarrow t+1$;
        End
    End
```

In our study, a chromosome represents a list of weapons to targets assignment, where the value of the *i*-th gene indicates to which target is assigned to the *i*-th weapon. Take *W*=4 and *T*=4 as an example, a chromosome (2 4 1 3) represents a feasible assignment list where target 2 is assigned to weapon 1, target 4 is assigned to weapon 2, and so on. In GA, two genetic operators, crossover and mutation, must be implemented. Traditionally, the one-cut-point (OCP) operator is employed as the crossover. The one-cut-point operator is to randomly generate one cut-point and swap the cut parts of two parents to generate offspring [21,23]. Mutation operation is performed to perturb genes *M* times within permissive integers from 1 to *T*, where *M* (<*W*) is randomly generated [23]. Local search approach has the advantage of providing a variety for possible solutions in a local manner. Furthermore, a selection operator is employed to select chromosomes in the population for the next generation. The better the chromosome is, it is more likely selected for reproduction in the next generation. After those new chromosomes are selected, GA with local search is again conducted and the consequent processes follow until a stop criterion is satisfied.

Local search approaches incorporated into GA algorithms have produced excellent results, and these approaches have played a key role in solving real world optimization problems [2,19,25]. The general idea of local search is shown in the following pseudo code [30]:

```
Procedure:    General local search
    Begin
        While (local search is stopped) do
            Generate a neighborhood solution $\pi'$
            If $C(\pi') < C(\pi)$ then $\pi = \pi'$
        End;
    End;
```

General local search starts from the found feasible solution and repeatedly tries to improve the current assignment by local changes. If a better assignment is found, then it takes place of the current assignment and the algorithm searches from the new assignment again. These introduced steps are repeated until a criterion is satisfied. The used criterion usually is to perform *N* times of local changes, where *N* (<*W*) is randomly generated. Usually, the neighborhood solution is defined as a chromosome obtained by randomly swapping two positions in the current chromosome [13,30].

Simulated annealing (SA) takes advantage of search strategies in which cost-deteriorating neighborhood solution may be accepted to search the optimal solutions [13]. In SA, in addition to better-fitness neighbors are always accepted, worse-fitness neighbors may also be accepted according to a probability that is gradually decreased in the cooling process. With the stochastic nature, SA enables asymptotic convergence to optimal solution and has been widely used for solving optimization problems [12,15,27]. In SA, if a modified solution is found to have better fitness than its ancestor then the modified solution is retained and the previous solution is discarded. If the modified solution is found to have less fitness than its ancestor, the modified solution may be still retained with a probability related to the current temperature. As the process continues and the temperature decreases, unsatisfactory solutions are less likely accepted. By using this approach, it is possible for the SA algorithm to move out of local minima, and more likely that good solutions will not be discarded. The pseudo code of the SA algorithm is described as follow [15,36,26].

```
Procedure: SA algorithm
    Begin
        Define the initial temperature $T_1$ and the coefficient $\gamma(0<\gamma<1)$;
        $\lambda \leftarrow 1$;
        While (SA has not been frozen) do
            $\sigma \leftarrow 0$; $\phi \leftarrow 0$;
            While (equilibrium is not approached sufficiently closely) do
```

```
                Generate new solution from the current solution;
                $\triangle F$= fitness of current solution –fitness of new solution;
                $P_r$ = exp(-$\triangle F/T_\lambda$);
                If $P_r \geq$ random[0,1] then
                    Accept new solution;
                    current solution$\leftarrow$new solution;
                    $\phi \leftarrow \phi$+1;
                End
                $\sigma \leftarrow \sigma$+1;
            End
            Update the maximum and minimum fitness;
            $T_{\lambda+1} \leftarrow T_\lambda * \gamma$;
            $\lambda \leftarrow \lambda$+1
    End
End
```

The initial temperature is set as [22]:

$$T_1 = \ln(F^{elitist} / \lambda + 1) , \qquad (4)$$

where $F^{elitist}$ is the elitist fitness in the beginning. New solutions are generated by inversing two randomly selected positions in the solution. The new generated solution is regarded as the next solution only when exp(-$\Delta F/T_\lambda$) ≥ random[0,1], where random[0,1] is a uniformly random value in [0,1]. If the generated solution is better, ΔF is negative and exp(-$\Delta F/T_\lambda$) is always greater than 1. Thus, the solution is always updated. If the new solution is not better than its ancestor, the solution may still take place of its ancestor in a random manner. The process is repeated until the equilibrium state is approached sufficiently close. The equilibrium state is defined as follows [45]:

$$\left\{\begin{matrix}(\sigma \geq \Gamma) \text{ or } (\phi \geq \Phi) \\ \Gamma = 1.5W, \Phi = W\end{matrix}\right\} \qquad (5)$$

where σ is the number of new solutions generated, ϕ is the number of new solutions accepted, Γ is the maximum number of generation, and Φ is the maximum number of acceptance. This algorithm is repeated until it enters a frozen situation, which is:

$$(F^{\max} - F^{\min}) / F^{\max} \leq \varepsilon \text{ or } T_\lambda \leq \varepsilon_1 \qquad (6)$$

where $F^{\max}$ and $F^{\min}$ are the maximum and minimum fitness, ε and ε_1 are pre-specified constants (ε=0.001, ε_1=0.005).

4. The Proposed Algorithm

In the above approaches, the local search approach separates from genetic operators. However, recent work has suggested that local search approach can be integrated into genetic operators [5]. In this paper, we employ local search integrated into genetic operators to provide the best solution in a local manner. In this research, we viewed the search in GA as the global sense, which provides a main portion of diversity in search, and the local search plays a major role to nail down a local optimum for the current solution. Thus, we proposed to use an immune system in place of a local search approach. The immune system has two main features: one is vaccination used for reducing the current fitness and the other is the immune selection used for preventing deterioration [22]. Vaccination is used to modify current solution with heuristics so as to find better solution. In this process, it should repair the wrong genes and keep the good genes of current solution by the information abstracted from the problem itself. In our implementation, the j-th good gene is an assignment of the j-th weapon to the target with the highest K_{ij}*EDV(i), where i=1 to T. The value of a wrong gene is repaired by a randomized integer from 1 to T. After vaccination, the immune selection is performed to obtain solutions for the next step. The immune selection includes two steps: the immune test and the annealing selection. In the immune test, repaired genes with better fitness are always accepted and genes with worse fitness are also accepted according to the annealing selection. In our implementation, the selection probability for the j-th repaired gene is calculated as follows:

$$P^j = \frac{e^{F_\lambda^{ij} / T_1}}{\sum_{j=1}^{W} e^{F_\lambda^{ij} / T_1}} \qquad (7)$$

where F_λ^{ij} is the value of K_{ij}*EDV(i), which is the damage of the j-th weapon assigned to the randomly generated target (i) at iteration λ.

Beside of using the traditional one-cut-point operator, we also proposed another crossover operator, called the elite preserving crossover (EX) operator, to enrich a more effective search. EX is adopted from CX [30], in which the genes shared by both parents are all preserved and then the remaining genes are then randomly swapped to generate offspring. Since the problem considered in [30] has constraints for their solutions, a repair algorithm is employed to make new chromosomes feasible. EX also adopts the similar concept, except preserves only those genes which are possible good genes. The EX operator is described as follows:

Step 1: Find the genes with the same values (targets) in both parents.
Step 2: Inherited good genes from both parents.
Step 3: Randomly select two genes that are not inherited from parents.
Step 4: Exchange the selected genes in both parents to generate offspring.
Step 5: Update the best solution and repeat to *step 1* until a stop criterion has satisfied.
Step 6: Return the best solution.

It is noted that there are no constraints on the assignment in our WTA definition. Our implementation of EX has omitted the repair process as used for CX [30]. To see the procedure, consider two chromosomes:

A= $\underline{1}$ 2 $\underline{2}$ 4 $\underline{8}$ 6 $\underline{3}$ 8 9
B= $\underline{1}$ 4 $\underline{2}$ 9 $\underline{8}$ 2 $\underline{3}$ 6 8

First, the genes of 1, 3, 5, and 7 with the same values in both parents are found. The values of K_{ij}*$EDV(i)$ among these weapon-target pairs of the 1st, the 3rd, the 5th, and the 7th weapons (j=1, 3, 5, and 7) to all targets (i=1 to T) are evaluated. It is assumed that these values of K_{11}*$EDV(1)$, K_{23}*$EDV(2)$, and K_{85}*$EDV(8)$ are the highest values among these weapon-target pairs of the genes of 1, 3, 5, and 7. Then, they are called good genes and inherited from parents. Thereafter, two positions, e.g., the 4th and the 7th positions, are selected at random in those two chromosomes and then genes are exchanged to generate the offspring as:

A'= 1 2 2 $\underline{3}$ 8 6 $\underline{9}$ 8 9
B'= 1 4 2 $\underline{3}$ 8 2 $\underline{4}$ 6 8

These steps are repeated until a stop criterion, which is randomly generated M (<W) times, is satisfied. The crossover operation will not be performed if the number of wrong gene is less than two.

Selection is performed after the offspring have been evaluated and generated. It chooses the best chromosome from the pool of parents and offspring to reduce the population to its original size. In order to select the best chromosomes from the parents and offspring, selection is referred to as (u + λ) –ES (evolution strategy) survival where u corresponds to the population size and λ refers to the number of offspring created [17,30].

5. Simulation and Results

In the following simulations, the default following values are used: the size of the initial population for GAs is the same as the maximum number of targets or weapons considered, the crossover probability P_c=0.8 and the mutation probability P_m=0.4, and γ=0.5 for SA. In this paper, first scenario is to test the performance for crossover operators. It consists of randomized data for 10 targets and weapons. All algorithms consistently use the same initial population when they are randomly generated. In this study, the effects for crossover operator of one-cut-point (OCP), CX, and EX are studied. The maximum generation is 2000, and experiments were run on PCs with Pentium 1GHz processor. The simulation is conducted for 10 trials and the average results are reported here. The simulation results are listed in Table 1. From Table 1, it is evident that the EX operator can have the best performances among all operators in both general GA, and GA with an immune system. Since the EX operator can result in better performance, in the following simulations EX is used as the crossover operator.

Several scenarios are considered to compare the performances of existing algorithms. These algorithms include general GA, simulated annealing (SA), GA with general local search, GA with SA as local search, GA with an immune system as local search (the proposed algorithm). Since those algorithms are search algorithms, it is not easy to stop their search in a fair basis from the algorithm itself. In our comparison, we simply stopped these algorithms after a fixed time of running. Experiments were also run on PCs with Pentium 1GHz processor, and were stopped after two hours of running. The results are listed in Table 2. From Table 2, it is easy to see that algorithms with local search approaches have better performance than algorithms without them. It also shows that the proposed algorithm can always find the best solution among these algorithms; meanwhile other algorithms may not find the best solution all the time.

Furthermore, we want to test the converged performance for algorithms with local search approaches. This scenario consists of randomized data for 120 weapons and 100 targets. In our study, we stopped those algorithms when the best fitness of all algorithms converged to a solution. The results are listed in Table 3. From Table 3, it is shown that the proposed algorithm could efficiently find the converged solution among those algorithms. It concludes that the proposed algorithm has also better long-term search capability than other algorithms do.

6. Conclusions

In this paper, we presented a hybrid genetic algorithm by including domain specific knowledge into the crossover operator and an immune system for solving weapon-target assignment (WTA) problems. From simulations, it can be found that the proposed algorithm can indeed have best search efficiency among existing genetic algorithms.

7. References

[1] A. Gasper and P. Collard, "From GAs to artificial immune systems: improving adaptation in time dependent optimization", *Proceedings of the 1999 Congress on Evolutionary Computation, CEC 99*, pp. 1999 –1866, vol. 3, 1999.

[2] A. Kolen and E. Pesch, "Genetic local search in combinatorial optimization," *Discrete Applied Mathematics and Combinatorial Operation Research and Computer Science*, vol. 48, pp. 273-284, 1994.

[3] A. M. H. Bjorndal, *et al.*, "Some thoughts on combinatorial optimisation," *European Journal of Operational Research*, pp.253-270, 1995.

[4] A. William, Meter, and Fred L. Preston, *A Suite of Weapon Assignment Algorithms for a SDI Mid-Course battle Manager*, AT&T Bell Laboratories, 1990.

[5] C. R. Reeves, "*Genetic algorithms and neighborhood search*," In T. C. Forgarty (ed) *Evolutionary computing: AISB Workshop, leeds, UK, April 1994: Selected Papers,* no. 865 in Lecture Notes in Computer Science, Springer-Verlag: Berlin, 1994.

[6] C.-T. Lin and C. S. George Lee, *Neural Fuzzy Systems*, Prentice-Hall Inc., 1995.

[7] C. Y. Ngo and V. O. K. Li, "Fixed channel assignment in cellular radio networks using a modified genetic algorithm," *IEEE Trans. Vehicular Technology*, vol. 47, no. 1, 1998.

[8] D. A. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning,* Addision-Wesley: Reading, MA, 1989.

[9] D. Dasgupta and N. Attoh-Okine, "Immunity-based systems: a survey," *IEEE International Conference on Systems, Man, and Cybernetics, 1997. Computational Cybernetics and Simulation*, vol. 1, pp. 369 –374, 1997.

[10] D. L. Pepyne, *et al.*, "A decision aid for theater missile defense," *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC '97)*, 1997.

[11] D. M. Tate and A. E. Smith, "A genetic approach to the quadratic assignment problem," *Computers Operation Research,* vol. 22, no. 1, pp.73-83, 1995.

[12] E. H. L Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*, John Wiley & Sons Inc., 1989.

[13] E. H. L Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*, John Wiley & Sons Inc., 1997.

[14] E. K. Burke and A. J. Smith, "Hybrid evolutionary techniques for the maintenance scheduling problem," *IEEE Trans. On Power Systems*, vol. 15, pp. 122-128, 2000.

[15] F.-T. Lin, C.-Y. Kao, and C.-C. Hsu, "Applying the genetic approach to simulated annealing in solving some NP-hard problems," *IEEE Trans. On Systems, Man and Cybernetics,* vol. 23, no. 6, pp.1752-1767, 1993.

[16] H. A. Eiselt and G. Laporate, "A combinatorial optimization problem arising in dartboard design", *Journal of The Operational Rsearch Society,* vol. 42, pp. 113-181, 1991.

[17] H.-P. Schwefel, *Numerische Optimierung von Computer-Modellen mittels der Evolutionsstrategie*, Basel: Birkäuser Verlag, vol. 26, Interdisciplinary Systems Research, 1977.

[18] J. D. Knowles and D.W. Corne, "M-PAES: a memetic algorithm for multi objective optimization," *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 325-332, 2000.

[19] J. Miller, W. Potter, R. Gandham, and C. Lapena, "An evaluation of local improvement operators for genetic algorithms," *IEEE Trans. On Systems, Man and Cybernetics,* vol. 23, no. 5, pp. 1340-1341, 1993.

[20] J.-T. Horn, C.-C. Chen, B.-J. Liu, and C.-Y. Kao, "Resolution of quadratic assignment problems using an evolutionary algorithm," *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 2, pp. 902-909, 2000.

[21] L. Davis, editor *Handbook of Genetic Algorithms*, Van Nostrand Reinhold, New York, 1991.

[22] L. Jiao and L. Wang, "Novel genetic

algorithm based on immunity," *IEEE Transactions on Systems, Man and Cybernetics, Part A,* vol. 30, no. 5, pp. 552 –561, 2000.

[23]M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design,* John Wiley & Sons Inc., 1997.

[24]N. J. Radcliffe and P. D. Surry, "Formal memetic algorithms," in *Evolutionary Computing: Selected Papers from the AISB Workshop*, pp. 1-16, 1994.

[25]N. L. J. Ulder, E. H. L. Aarts, H. J. Bandelt, P. J. M. van laarhoven, and E. Pesch, "Genetic local search algorithms for the traveling salesman problem," in *Parallel Problem solving from nature-Proc. 1st Workshop, PPSN I*, H. P. Schwefel and R. Männer, Eds., vol. 496, Lecture notes in Computer Science, pp. 109-116, 1991.

[26]N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A. Teller, "Equation of state calculations by fast computing machines, " Journal of Chemical Physics, vol. 21, pp. 1087-1092, 1953.

[27]P. J. M. Van Laarhoven and E. H. L. Arts, *Simulated Annealing: Theory and Applications,* Kluwer Academic Publishers, 1992.

[28]P. L. Hammer, "Some network flow problems solved with pseudo-boolean programming," *Operation Research,* 13, pp.388-399, 1965.

[29]P. L. Hammer, P. Hansen, and B. Simeone, "Roof duality, complementation and persistency in quadratic 0-1 optimization," *Mathematical Programming,* 28, pp.121-155, 1984.

[30]P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for quadratic assignment problem," *IEEE Trans. On Evolutionary Computation,* vol. 4, no. 4, pp. 337-352, 2000.

[31]P. Moscato and M. G. Norman, "A memetic approach for the traveling salesman problem. Implementation of computational ecology on message passing systems," in *Parallel Computing and Transporter Applications,* IOS Press: Amsterdam, pp. 187-194. 1992.

[32]R. Cheng and M. Gen, "Parallel machine scheduling problems using memetic algorithms," *IEEE International Conference on Systems, Man and Cybernetics*, vol. 4, pp. 2665 –2670, 1996.

[33]R. Dawkins, *The Selfish Gene*, Oxford University Press, 1976.

[34]R. J .W. Hodgson, "Memetic algorithms and the molecular geometry optimization problem," *Proceedings of the 2000 Congress on Evolutionary Computation*, vol. 1, pp. 625 –632, 2000.

[35]S. A. Frank, "The design of natural and artificial adaptive systems," Academic Press, New York, M. R. Rose and G.V. Lauder edition, 1996.

[36]S. Kirkpatrick, C. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing, " *Science,* vol. 220, pp. 671-680, 1983.

[37]S. P. Lloyd and H. S. Witsenhausen, "Weapon allocation is NP-Complete," *IEEE Summer Simulation Conference*, Reno, Nevada, 1986.

[38]S. Sahni and T. Gonzales, "P-complete approximation problem," *ACM Journal,* vol. 23, pp. 556-565, 1976.

[39]T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the History and current state," *IEEE Trans. On Evolutionary Computation,* vol. 1, no. 1, 1997.

[40]T. Ibarraki and N. Katoh, *Resource allocation Problems,* The MIT Press: Cambridge, Massachusetts, 1988.

[41]W. K. Lai and G. G. Coghill, "Channel assignment through evolutionary optimization," *IEEE Trans. Vehicular Technology*, vol. 45, no. 1, pp.91-96, 1998.

[42]Y. Sun and Z. Wang, "The genetic algorithm for 0-1 programming with linear constraints," *Proceedings of the First IEEE Conference on Evolutionary Computation*, 1994.

[43]Z.-J. Lee, C.-Y. Lee, and S.-F. Su, "A fuzzy-genetic based decision-aided system for the naval weapon-target assignment problems," *2000 R. O. C. Automatic Control Conference*, pp. 163-168, 2000.

[44]Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, Springer-Verlag: Berlin, 1994.

Table 1. The simulation results for randomized data of *W*=10 and *T*=10. Results are averaged over 10 trials.

Algorithm	Operator	Best fitness	Percentage of convergence	Converged generation	CPU time (sec)
General GA	OCP	92.7668	10 %	N/A	N/A
	CX	75.9367	20%	N/A	N/A
	EX	69.3263	40 %	N/A	N/A
GA with immune system	OCP	61.1183	80%	N/A	N/A
	CX	58.4774	100%	733	82.29
	EX	58.4774	100%	465	51.15

N/A-Not Available

Table 2. Compare the best fitness of randomized scenarios obtained by various search algorithms. Results are averaged over 10 trials.

Algorithms	*W*=50 *T*=50	*W*=80 *T*=80	*W*=100 *T*=80	*W*=120 *T*=80
SA	290.4569	352.5413	290.564	175.6310
General GA	282.6500	351.7641	279.558	140.1381
GA with general local search	226.335	348.4352	197.8655	106.4778
GA with SA as local search	230.5209	347.1698	203.586	105.5135
The proposed algorithm	171.8513	282.2294	161.3612	98.8892

Table 3. The simulation results for randomized data of W=120 and T=100.

Algorithm	Best fitness	CPU time (minute)
GA with general local search	173.3241	335.3
GA with SA as local search	173.3241	316.5
The proposed algorithm	173.3241	285.4

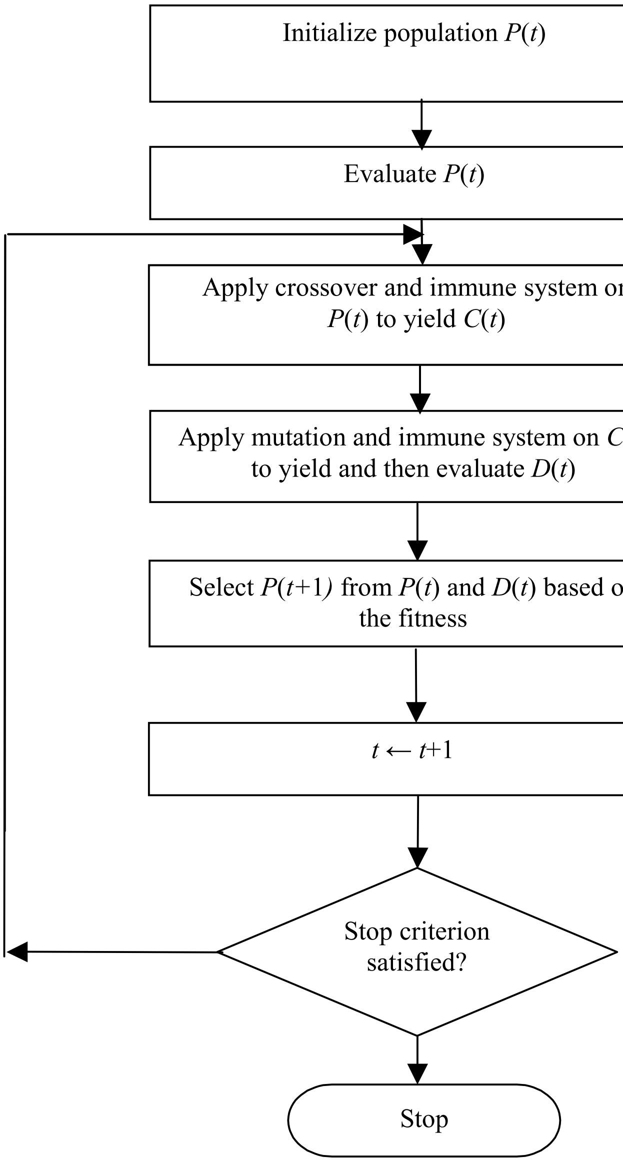
Initialize population $P(t)$
Evaluate $P(t)$
Apply crossover and immune system on $P(t)$ to yield $C(t)$
Apply mutation and immune system on $C(t)$ to yield and then evaluate $D(t)$
Select $P(t+1)$ from $P(t)$ and $D(t)$ based on the fitness
$t \leftarrow t+1$
Stop criterion satisfied?
Stop

Figure 1. The flow diagram of the proposed algorithm