Efficiently solving general weapon-target assignment problem by genetic algorithms with greed eugenics

Article *in* IEEE TRANSACTIONS ON CYBERNETICS · February 2003

DOI: 10.1109/TSMCB.2003.808174 · Source: PubMed

3 authors:

Zne-Jung Lee
Huafan University
88 PUBLICATIONS **2,790** CITATIONS

SEE PROFILE

Shun-Feng Su
National Taiwan University of Science and Technology
172 PUBLICATIONS **2,894** CITATIONS

SEE PROFILE

Chou-Yuan Lee
Lan Yang Institute of Technology
28 PUBLICATIONS **818** CITATIONS

SEE PROFILE

Efficiently Solving General Weapon-Target Assignment Problem by Genetic Algorithms With Greedy Eugenics

Zne-Jung Lee, Shun-Feng Su, *Member, IEEE,* and Chou-Yuan Lee

***Abstract*—A general weapon-target assignment (WTA) problem is to find a proper assignment of weapons to targets with the objective of minimizing the expected damage of own-force asset. Genetic algorithms (GAs) are widely used for solving complicated optimization problems, such as WTA problems. In this paper, a novel GA with greedy eugenics is proposed. Eugenics is a process of improving the quality of offspring. The proposed algorithm is to enhance the performance of GAs by introducing a greedy reformation scheme so as to have locally optimal offspring. This algorithm is successfully applied to general WTA problems. From our simulations for those tested problems, the proposed algorithm has the best performance when compared to other existing search algorithms.**

***Index Terms*—Evolutionary optimization, genetic algorithm (GA), greedy algorithm, local search, weapon-target assignment (WTA).**

I. Introduction

THE weapon-target assignment (WTA) problem is to find a proper assignment of weapons to targets with the objective of minimizing the expected damage of own-force assets. This is an NP-complete problem [1], [2]. Various methods for solving optimization problems have been reported in the literature [3]–[7]. These methods are based on graph search approaches and usually result in exponential computational complexities. As a consequence, it is difficult to solve these types of problems directly while the number of targets or weapons are large [1], [4], [6], [8], [31]. Recently, genetic algorithms (GAs) have widely been used as search algorithms in various applications and have also demonstrated satisfactory performances [9]–[20]. In our previous work [21], [51], we have employed general GAs to solve WTA problems. Even though those approaches could find the best solution in those simulated cases, the search efficiency did not seem good enough. In this paper, we propose a novel approach, in which a gene reformation for WTA problems is introduced. Such a reformation scheme is referred to as a *eugenic process for offspring*.

Manuscript received November 19, 2001; revised March 10, 2002. This work was supported in part by the Chung-Shan Institute of Science Technology of Taiwan under Grant XU89A59P and in part by the National Science Council of Taiwan, R.O.C., under Grant NSC-89-2218-E-011-002. This paper was recommended by Associate Editor H. Takagi.

Z.-J. Lee is with the Department of Information Management, Kang-Ning Junior College, Taipei, Taiwan, R.O.C. (e-mail: johnlee@mis.knjc.edu.tw).

S.-F. Su and C.-Y. Lee are with the Department of Electrical Engineering, National Taiwan University of Science and Technology, Taipei, Taiwan, R.O.C. (e-mail: su@orion.ee.ntust.edu.tw)

Digital Object Identifier 10.1109/TSMCB.2003.808174

The concept of eugenic is to find a better candidate nearby the current one before moving to the next stage of search. Traditionally, those eugenic mechanisms are called *local search mechanisms* in various systems. Many local search approaches can be found in various search algorithms, such as ant colonies optimization (ACO) [22], [23], greedy randomized adaptive search procedure (GRASP) [24], etc. There also exist approaches employing local search into GAs. In the literature, a GA equipped with a local search approach is referred to as the memetic algorithm, the genetic local search, the hybrid GA, or the cultural algorithm [19], [25]–[27]. Those algorithms emulate cultural evolution or the evolution of ideas [19], [28]–[30]. In those algorithms, a population of ideas can be maintained. Ideas can be recombined to create new ideas. Good ideas are more useful than weak ones. The unit of information in those algorithms is referred to as a meme and it can be improved by problem-specific heuristics. In our paper, local search is viewed as a gene reformation process in evolutionary algorithms. In order to have fruitful reformation, instead of using a random trial process as is usually used in the literature, the proposed algorithm is to greedily reform the current chromosome. Such a process is called the *greedy eugenics*. Furthermore, the elite preserving crossover (EX) is also proposed to improve the evolution performance for WTA problems. Several examples are used in our paper and the results have shown better search efficiency of our approach than that of other existing algorithms. In order to show the effectiveness of our approach, we also used the quadratic assignment problems (QAPs) [53], [54] as examples. The simulation results again showed the superiority of our approach.

The paper is organized as follows. In Section II, a mathematical formulation of WTA problems is introduced. The general GA for WTA problems is described in Section III. The GA with eugenic mechanisms is presented and discussed in Section IV. In Section V, the results of employing the proposed algorithm to solve general WTA problems are presented. Besides, QAP is also employed as an xample. Several existing algorithms were also employed for comparison. The performance showed the superiority of our algorithm. Finally, Section VI concludes the paper.

II. Weapon-Target Assignment (WTA) Problems

On modern battlefields, it is an important task for battle managers to make a proper WTA to defend own-force assets. As an example in considering anti-aircraft weapons of

naval battle force platforms, threat targets may be launched from surface ships, aircrafts, or submarines. These targets have different probabilities of killing to platforms which are dependent on the target types. Thus, a WTA decision-aided system is strongly desired in helping and training planners to make proper decisions on the battlefield. In this research, we first considered the above weapons to targets assignment problem as an optimization problem, such that the optimal assignment for a scenario of attack can be found. Hopefully, such findings can bring some insight information for battlefield planners. Besides, if a constraint about exhaustive and exclusive assignments is imposed [51], it becomes a QAP, except with a different form of cost function. Actually, WTA problems can also be abstractly viewed as the resource allocation problems [6], [52], which can be seen in various fields. The approaches discussed in this paper can directly be used for those problems.

In our WTA problems, the following assumptions are made. The first one is that there are W weapons and T targets and all weapons must be assigned to targets. It is worth noting that in our previous work [21], [51], this assumption is further restricted to $W = T$ and that all targets must also be assigned. Such a restricted assumption, in fact, has largely reduced the search space, but also may restrict the diversity of solutions. In our paper, a WTA problem with the assumption stated here is referred to as the *general WTA problem.* If the restricted assumptions as those in [51] are used, it is called the *constrained WTA problem*. The second assumption is that the individual probability of killing (K_{ij}) by assigning the jth weapon to the ith target is known for all i and j. This probability defines the effectiveness of the jth weapon to destroy the ith target. It is easy to verify that the overall probability of the ith target being destroyed is $\prod_{j=1}^{W}(1-K_{ij})^{X_{ij}}$. Then, the considered WTA problems are to minimize the following cost function [1], [21], [51]:

$$C(\pi)=\sum_{i=1}^{T} EDV(i)^{*}\left[1-\prod_{j=1}^{W}(1-K_{ij})^{X_{ij}}\right] \tag{1}$$

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

$$\sum_{i=1}^{T} X_{ij}=1,\ \text{for } j=1,2,\ldots,W \tag{2}$$

where X_{ij} is a Boolean value indicating whether the jth weapon is assigned to the ith target. $X_{ij}=1$ indicates that the jth weapon is assigned to the ith target. Here, $EDV(i)$ is the expected damage value of the ith target to the asset. Thus, (1) simply summarizes the overall damage for all targets. π is a feasible assignment list and $\pi(j)=i$ indicates that the jth weapon is assigned to the ith target. If a constrained WTA problem is considered, the above constraints become ($W=T=N$)

$$\begin{aligned}\sum_{j=1}^{N} X_{ij}&=1, \quad i=1,2,\ldots,N\\ \sum_{i=1}^{N} X_{ij}&=1, \quad i=1,2,\ldots,N.\end{aligned} \tag{3}$$

III. Genetic Algorithms (GAs) for Weapon-Target Assignment (WTA) Problems

GAs have been considered as a class of general-purpose search strategies for optimization problems. A general GA is shown as follows [15]. Let $P(t)$ and $C(t)$ be parents and offspring in generation t.

```
Procedure: General GA
Begin
 t ← 0;
 Initialize P(t);
 Evaluate P(t);
 While (not matched with the termination
conditions) do
  Recombine P(t) to yield C(t);
  Evalute C(t);
  Select P(t+1) from P(t) and C(t);
  t ← t+1;
 End
End
```

The GA starts with a set of randomly selected chromosomes as the initial population that encodes a set of possible solutions. Variables of a problem are represented as genes in a chromosome, and chromosomes are evaluated according to their fitness values, which are obtained by evaluating the considered fitness function or cost function such as (1). Recombination typically involves two genetic operators: 1) crossover and 2) mutation. Genetic operators alter the composition of genes to create new chromosomes referred to as offspring. The selection operator is an artificial version of natural selection, a Darwinian survival of the fittest among populations, to create populations from generation to generation. Chromosomes with better fitness have higher probabilities of being selected in the next generation. After several generations, GA can converge to the best solution. The used chromosomes and genetic operations for our WTA problem are briefly introduced in the following.

The encoding of WTA solutions is straightforward. The permutation π is encoded as a vector of targets and the value of the jth component (gene) indicates to which target the jth weapon is assigned. Take $W=4$ and $T=3$ as an example. The chromosome (2 3 1 3) represents an assignment list where weapon 1 is assigned to target 2, weapon 2 is assigned to target 3, and so on. It is noted that there are exact W genes in a chromosome and their corresponding values are integers between 1 and T.

The one-cut-point (OCP) crossover, inversion mutation, and roulette selection operators are employed as the basic operations in GA [15], [32]. The OCP operator is to randomly generate OCP and to swap the cut parts of two parents so as to generate offspring. However, if the considered problem is a constrained WTA problem [21], a repair process must be used after swapping to ensure the feasibility of new chromosomes.

In this paper, we propose a novel crossover operator (the (EX) operator). The concept of EX is to construct offspring with possibly good genes from parents. EX is modified from CX (also called UX) in [19]. In the CX operator, the information contained in both parents is preserved. EX also adopts the similar

concept, except it preserves only those genes supposed to be "good." In our implementation, when the value of the jth gene is i and if $K_{ij}{}^{*}EDV(i)$ is the highest value among all targets assigned to the jth weapon, it is called a "good" gene. Note that the problem considered in [19] has constraints for their solutions and then a repair algorithm must be employed to make new chromosomes feasible. Thus, if constrained WTA problems are considered, the same repair algorithm must also be employed. The EX operation is described in the following steps.

Step 1) Find genes with the same values i in both parents.
Step 2) Inherit "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) Go to Step 1 until a stop criterion is satisfied.
Step 6) Return the solution.

To see the procedure, consider two parents

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

First, the genes of 1, 3, 5, and 7 are of the same values in both parents. The values of $K_{ij}{}^{*}EDV(i)$ among these weapon-target pairs of the first, third, fifth, and seventh weapons ($j = 1, 3, 5,$ and 7) to all targets ($i = 1$–T) are evaluated. Assumed is that these values of $K_{11}{}^{*}EDV(1)$, $K_{23}{}^{*}EDV(2)$, and $K_{85}{}^{*}EDV(8)$ are the highest values among those corresponding weapon-target pairs. Then, these "good" genes are inherited from parents. Thereafter, two positions, e.g., the fourth and the seventh positions, are selected at random in parents and then genes are exchanged to generate the offspring as

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

These steps are repeated until a stop criterion is satisfied. The used criterion is simply that the process is repeated M_c times, which is randomly generated and $M_c < W$. Note that the crossover operation will not be performed if the number of available genes for selection is less than two.

Mutation operations are to randomly change a gene in the chromosome M_m times [15], where $M_m < W$ is randomly generated. After offspring are generated, it needs to select chromosomes as the next generation population. A set of chromosomes as $P(t+1)$ is selected from the pool of parents and offspring to reduce the population to its original size. The used selection strategy is referred to as the $(u+\lambda)$-ES (evolution strategy) survival [19], [33], where u is the population size and λ is the number of offspring created. The process simply deletes redundant chromosomes and then retains the best u chromosomes in $P(t+1)$. After those new chromosomes are selected, the process is repeated until a stop criterion, such as enough generations found or after a fixed time of running, is satisfied.

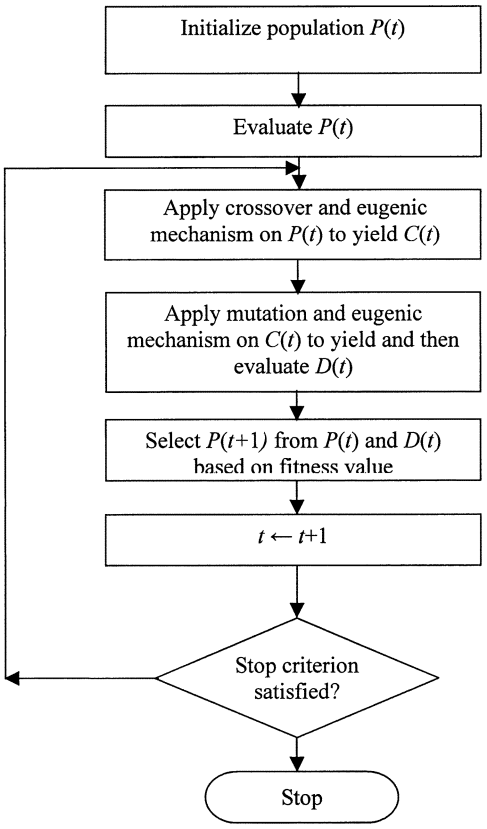


Fig. 1. Flow diagram of GA with a eugenic mechanism.

IV. Genetic Algorithm (GA) With Eugenic Mechanisms

Recently, local search approaches combined with GAs have been considered as good alternatives for solving optimization problems [18], [24], [25], [34]–[38]. Local search approaches have the advantage of exploring the neighborhood of a solution to provide a variety for possible solutions in a local manner. Usually, such local search mechanisms are applied to the obtained offspring $C(t)$ to generate a set of better $C(t)$. From the evolutionary viewpoint, local search can be viewed as a eugenic process, which may involve the use of domain knowledge in defining good genes. From our simulations, it can be seen that if specific-problem heuristics are embedded into local search approaches, the search performance can be significantly improved. Thus, in our research, we call local search approaches *eugenic processes*. Recent work suggests that eugenic mechanisms at best be integrated into genetic operators [49], [50]. Thus, in this paper, the eugenic mechanism is performed after genetic operators and the diagram of GA with eugenic mechanisms is shown in Fig. 1. Various eugenic mechanisms are used to demonstrate our idea about using the greedy concept in the place of eugenic processes. They are simple eugenics, simulated annealing, immune operator, and greedy eugenics. We shall introduce them in the following subsections.

A. Simple Eugenics

The concept of eugenics is illustrated by the following simple algorithm [19]:

```
Procedure: Simple eugenics
 Begin
  While (the eugenic process has not been
stopped) do
   Generate a neighborhood solution π′
  If C(π′) < C(π) then π = π′
  End
 End
```

The process of eugenics starts from a found feasible solution and repeatedly tries to improve the current assignment by local changes. If a better assignment is found, then it replaces the current assignment and the algorithm searches from the new assignment again. These steps are repeated until a criterion is satisfied. The used criterion usually is to perform M_g times of local changes, where $M_g(< W)$ is randomly generated. Usually, the neighborhood solution is defined as a chromosome obtained by randomly swapping two positions in the current chromosome [19], [39].

B. Simulated Annealing (SA) Algorithm

Simulated annealing (SA) algorithms are widely used for solving optimization problems [40]–[42]. SA is an analogy to the annealing process of solids, which is the process of healing up a solid to a high temperature and then cooling down gradually. With the capability of escaping from local optimums by incorporating a probability function in accepting or rejecting new solutions, SA enables asymptotic convergence to the optimal solution. In our approach, SA serves as an alternative to simple eugenics and is employed as eugenic mechanism to take advantages of search strategies in which cost-deteriorating neighborhood solution may possibly be accepted in searching for the optimal solutions. The SA algorithm is as follows [41]–[44]:

```
Procedure: SA algorithm
 Begin
  λ ← 1
  Define the initial temperature T_1 and
the coefficient γ(0 < γ < 1);
  While (SA has not been frozen) do
   σ ← 0;  π ← 0;
   While (equilibrium is not approached
sufficiently close) do
    Generate a new solution from the cur-
rent solution;
    ΔF   = fitness of new solution − fit-
ness of current solution;
    P_r = exp(−ΔF/T_λ);
    If P_r ≥ random[0, 1] then
     Accept the new solution; current so-
lution ← new solution;
     φ ← φ + 1;
    End
    σ ← σ + 1;
   End
   Update the maximum and minimum fitness;
   T_{λ+1} ← T_λ*γ,
   λ ← λ + 1
  End
 End
```

The initial temperature is [20]

$$T_1 = \ln\left(F^{\text{elitist}} + 1\right) \tag{4}$$

where F^{elitist} is the elitist fitness value from the beginning of the running. A way of generating new solutions is to inverse two randomly selected positions in the current solution. The new generated solution will be regarded as the next solution only when $\exp(-\Delta F/T_\lambda) \geq \text{random}[0,1]$, where random$[0,1]$ is a random value generated from a uniform distribution in the interval [0,1]. It is easy to see that when the generated solution is better, ΔF is negative and $\exp(-\Delta F/T_\lambda)$ is always greater than one. Thus, the solution is always updated. When the new solution is not better than its ancestor, the solution may still replace its ancestor in a random manner. New solutions are repeatedly generated until the equilibrium state is approached sufficiently close. The used equilibrium state is defined similarly to that used in [45] and is

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

where

- σ number of new solutions generated;
- ϕ number of new solutions accepted;
- Γ maximum number of generations;
- Φ maximum number of accepted new solutions.

This algorithm is repeated until it enters a frozen situation, which is

$$\frac{(F^{\max} - F^{\min})}{F^{\max}} \leq \varepsilon \text{ or } T_\lambda \leq \varepsilon_1. \tag{6}$$

$F^{\max}$ and $F^{\min}$ are the maximum and minimum fitness values, respectively, and ε and ε_1 are prespecified constants. $\varepsilon = 0.001$ and $\varepsilon_1 = 0.005$ are used in our simulations.

C. Immune Operator

In the above approaches, eugenic processes are domain independent. The immunity, however, provide another concept in which problem-specific heuristics are used for eugenic mechanisms. The immunity is a remarkable adaptive system [46]–[48]. Biologically, the function of immunity is to protect a body from antigens. When incorporated into GAs, immunity can use local information to improve the search capability during the evolutionary process. In [20], an immune operator was proposed to solve the travel salesman problem. The algorithm consists of two main operations: 1) the vaccination used for reducing the current fitness and 2) the immune selection used for preventing deterioration [20]. The process of vaccination is to modify genes of the current chromosome with heuristics so as to possibly gain better fitness. The immune selection includes two steps. The first one is called the *immune test* and the second one is called

the *annealing selection*. The immune test is used to test the vaccinated genes and the annealing selection is used to accept these genes with a probability according to their fitness.

Similar to that used in [20], the vaccination operation is applied to keep "good" genes and modify other genes. In our implementation, when the jth weapon is assigned to the ith target with $K_{ij}{}^{*}EDV(i)$ being the highest among all targets, this gene is defined as a "good" gene. If it is not a "good" gene, the gene is replaced by a randomized integer between one to T. In the immune test, modified genes with better fitness values are always accepted and those with worse fitness values may also be accepted according to the annealing selection. In our implementation, the selection probability for the jth modified gene is calculated as

$$P^{j} = \frac{e^{-F_t^{ij}/T_i}}{\sum_{j=1}^{W} e^{-F_t^{ij}/T_i}} \tag{7}$$

$$T_t = \ln\left(\frac{F^{\text{elitist}}}{t} + 1\right) \tag{8}$$

where F_t^{ij} is the value of $K_{ij}{}^{*}EDV(i)$ at generation t. It is noted that the annealing selection is similar to that used in [20].

D. Greedy Eugenics

In the above three approaches, eugenics is considered as a mechanism used for enhancing the search capability for GA. In this research, we viewed the search in GA as a mechanism providing a main portion of diversity in search and the eugenics as a major role nailing down a local optimum for the solution provided by GA. Thus, in this paper, instead of using random mechanisms or some powerful global search mechanisms to find a neighbor, we propose to use a greedy reformation scheme in the place of eugenics. In other words, the proposed eugenic scheme is to greedily reform the current chromosome. Even though greedy algorithms may have a great possibility to be trapped into a local optimum, due to crossover and mutation operations and the parallel search nature used in GAs, the search can easily escape from local optima. Hence, if a greedy eugenics is used, it can find locally best solutions quickly and will not be trapped in local optima. As a result, the proposed algorithm can have a more effective search than other algorithms do.

The problem-specific heuristics used in our greedy algorithm is similar to that used for the immunity operator such that the comparison between them is fair. The idea is to greedily assign targets to weapons that have the highest $EDV(i)^{*}K_{ij}$. The flow diagram of the proposed greedy eugenics is shown in Fig. 2. In the first phase, for those targets existing in the current chromosome define their $EDV(i)$ as $EDV_{\text{new}}(i)$ and sort $EDV_{\text{new}}(i)^{*}K_{ij}$ in a descending order. Weapon j with the highest $EDV_{\text{new}}(i)^{*}K_{ij}$ is selected and assigned to the ith target in the new chromosome. After this assignment, the corresponding $EDV_{\text{new}}(i)$ is updated as $EDV_{\text{new}}(i)^{*}(1 - K_{ij})$. This weapon selection process continues until all currently existing targets have been assigned in the chromosome. Since there are no constraints in our WTA problems, there may exist unassigned weapons in the new chromosome. Then, in the second phase, those unassigned weapons then greedily select the targets with the highest $EDV_{\text{new}}(i)^{*}K_{ij}$. Note that in this process, all targets, including those not in the current chromosome, are considered. In the first phase, the process attempts to maintain certain properties in the original chromosome. In this algorithm, the used property is the set of targets in the original chromosome. In the second phase of the assignment, since the target set property has been satisfied, the assignment process become truly greedy for all targets. After the new chromosome is generated, it is evaluated. In our implementation, in order to keep diversity under greedy eugenics, the population of the next generation will include the original chromosome generated by GA and the best chromosome generated in the eugenic process.

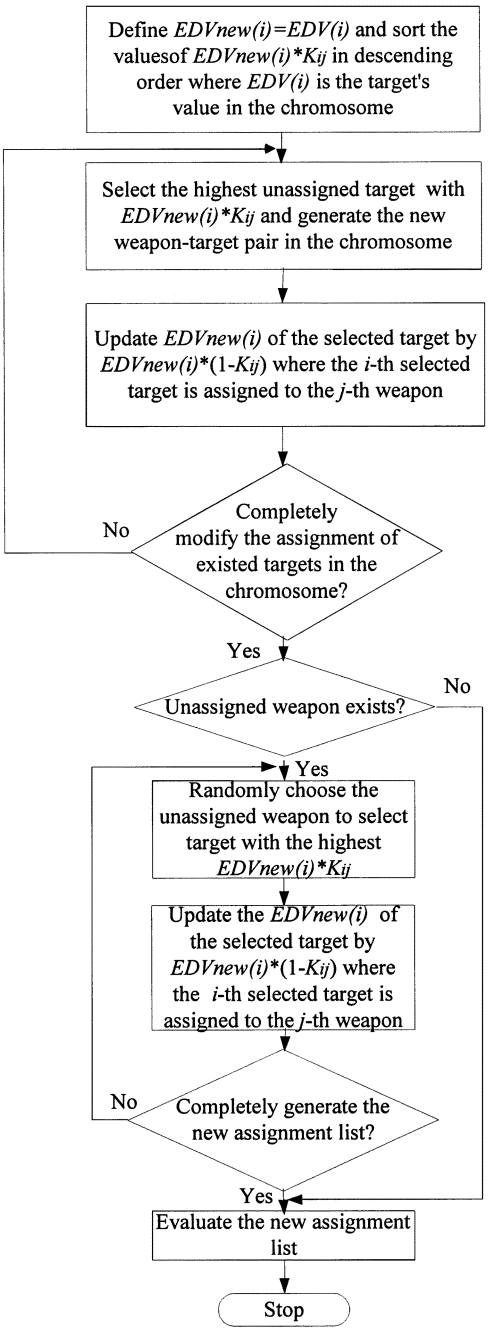


Fig. 2. Flow diagram of greedy eugenics for general WTA problems.

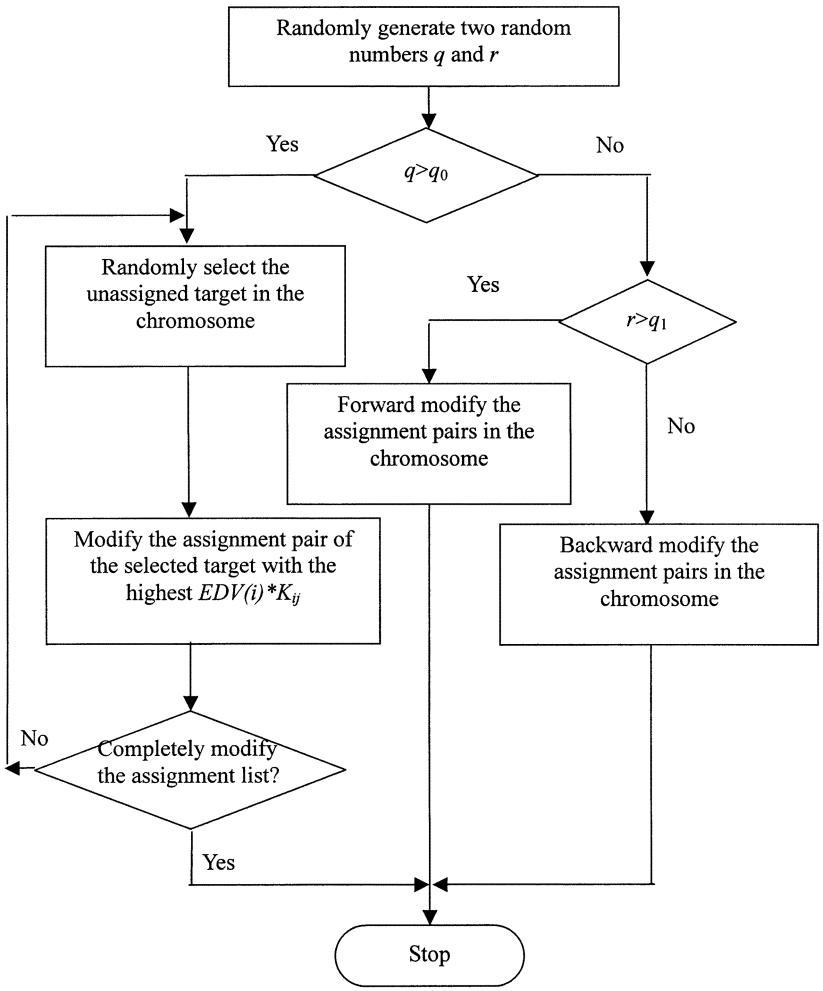


Fig. 3. Flow diagram of greedy eugenics for constrained WTA problems.

In our paper, we also consider another greedy idea, of which the assignment of weapons to targets should be as diverse as possible. An extreme case becomes the constrained WTA problem. When the constrained WTA problem is considered, the flow diagram of the used greedy eugenics is shown in Fig. 3. In this algorithm, the assignment is performed based on either weapons or targets depending on whether a random number q is larger than a prespecified q_0. When the assignment is based on weapons, another random number r is used to decide to forward or backward modify the assigned target-weapon pairs in the chromosome. From later simulations, it is found that this approach can have very efficient search due to its search space is much smaller than that of general WTA problems. However, when the number of weapons and targets are large, due to its constraints the found solution may not be the optimal solution.

V. Simulations and Results

In simulations, we need to identify a set of parameters. We performed simulations of five scenarios with various values for those parameters. The size of the initial population for GA is all set as the maximum number of targets or weapons considered. The tested values are: the crossover probability $P_c = 0.1$, 0.3, 0.6, 0.8, 0.9; the mutation probability $P_m = 0.05$, 0.1, 0.4, 0.9; and $\gamma = 0.1$, 0.5, 0.9 for SA. The results are all similar. We keep the following values as default: $P_c = 0.8$, $P_m = 0.4$, and $\gamma = 0.5$, because under these values, the results are fair in all algorithms.

First, a simple case consisting of randomized data for ten targets and weapons is used to investigate the performances for various crossover operators. Simulations are conducted for ten trials and the average results are reported. All simulations use the same initial population, which are randomly generated. The maximum number of generations is set as $\max\_\text{gen} = 2000$ and experiments were run on PCs with a Pentium 1-GHz processor. The results for using OCP, modified CX (MCX), and EX are listed in Table I. MCX is similar to CX adopted in [19]. MCX keeps genes with the information contained in both parents and the other genes are randomly selected from parents. In other words, MCX is CX without the repairing process. In Table I, we report the best fitness value, the standard deviation in parentheses, the percentage of convergence in ten trials, averaged converged generation, and averaged converged CPU time. Also in the table, 100% means that all ten tests converge to the best fitness. It is marked as "N/A" if the optimal value is not found within the maximum generation in any trials. It is clearly evident that the EX operator can find better fitness values with less CPU time than other operators do in both general GAs and GAs with greedy eugenics. Since the EX operator has the best

TABLE I
SIMULATION RESULTS FOR RANDOMIZED DATA OF $W = 10$ AND $T = 10$. RESULTS ARE AVERAGED OVER TEN TRIALS

Algorithm	Operator	Best fitness value	Percentage of convergence	Converged generation	CPU time (sec)
General GA	OCP	92.7668 (14.1685)	10 %	N/A	N/A
	MCX	75.9367 (9.6214)	20%	N/A	N/A
	EX	69.3263 (6.8752)	40 %	N/A	N/A
GA with greedy eugenics	OCP	58.4774 (0)	100%	1501	76.55 (8.39)
	MCX	58.4774 (0)	100%	724	50.68 (7.92)
	EX	58.4774 (0)	100%	338	20.28 (7.15)

N/A-Not Available

TABLE II
BEST FITNESS VALUES OF RANDOMIZED SCENARIOS OBTAINED BY VARIOUS SEARCH ALGORITHMS AFTER TWO HOURS OF RUNNING. RESULTS ARE AVERAGED OVER TEN TRIALS

Algorithms	Scenarios			
	W=50 T=50	W=80 T=80	W=100 T=80	W=120 T=80
SA	290.4569 (50.8541)	392.5413 (50.8137)	290.564 (43.8612)	175.631 (35.1352)
General GA	282.65 (47.6715)	351.7641 (52.9217)	279.558 (50.4279)	140.1381 (38.2817)
Repeating randomly made chromosome with greedy eugenics	238.9935 (15.7814).	359.6623 (35.7891)	280.8347 (20.1576)	170.7835 (13.8778)
GA with simple eugenics	226.335 (30.7316)	348.4352 (60.7415)	197.8655 (37.1495)	106.4778 (12.9815)
GA with SA as eugenics	230.5209 (27.981)	347.1698 (55.215)	203.586 (30.9173)	105.5135 (14.571)
GA with immunity as eugenics	171.8513 (25.8734)	282.2294 (30.2742)	161.3612 (19.0139)	98.8892 (8.1775)
GA with greedy eugenics	**138.671** (19.1175)	**248.2979** (25.973)	**136.7236** (15.8916)	**88.1155** (8.731)

performances among those operators, it is employed as the default crossover operator in the following simulations.

Next, the performances of various search algorithms are investigated. These algorithms include general GA, SA algorithm, GA with general eugenics, GA with SA as eugenics, GA with immunity as eugenics, repeating randomly made chromosome with greedy eugenics, and GA with greedy eugenics. The algorithm of repeating randomly made chromosome with greedy eugenics is to randomly generate feasible solutions, then apply greedy eugenics to those solutions. Since these algorithms are search algorithms, it is not easy to stop their search in a fair basis from the algorithm itself. Since the issue considered in this research is the search efficiency of algorithms, in our comparison, we simply stopped these algorithms after a fixed time of running. Those simulations are to see which algorithm can find a better solution in a fixed period of running. Experiments were also run on PCs with a Pentium 1-GHz processor and were stopped after two hours of running. Since we need to run ten tests for each algorithm, it may not be feasible to run too long. If the running time is too short, the results may not be significant. To run algorithms for two hours is only a handy selection. The results of averaged best fitness and the standard deviation

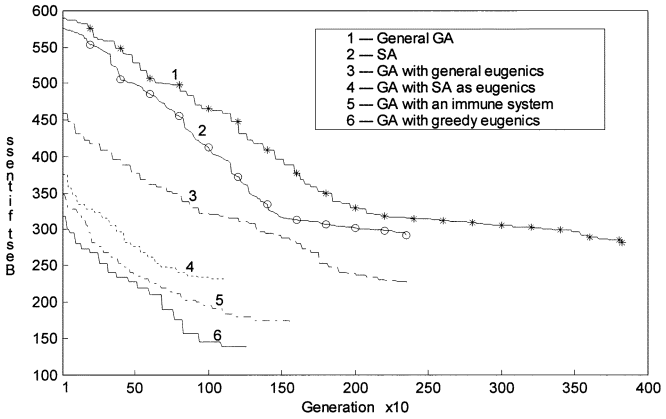


Fig. 4. Fitness curves of various algorithms for $W = 50$ and $T = 50$.

TABLE III
SIMULATION RESULTS FOR RANDOMIZED DATA OF $W = 120$ AND $T = 100$

Algorithm	Best fitness value	CPU time (minute)
GA with simple eugenics	173.3241	335.3
GA with SA as eugenics	173.3241	316.5
GA with immunity as eugenics	173.3241	285.4
GA with greedy eugenics	173.3241	**210.7**

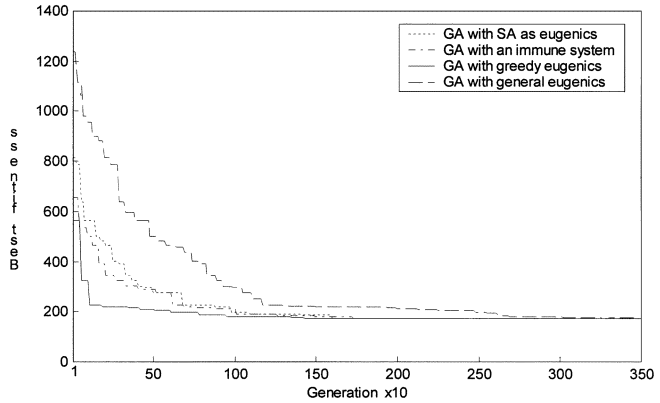


Fig. 5. Fitness curves for $W = 120$ and $T = 100$.

in parentheses are listed in Table II, and the fitness curves by generations are shown in Fig. 4. Note that the computational times required for one generation are different for different algorithms. From Table II, it is easy to see that the GA with eugenic mechanisms can find better fitness values than algorithms without them do. It also shows that the GA with greedy eugenics can always outperform other algorithms.

Furthermore, we want to test the converged performance for algorithms with eugenic mechanisms. We are aware that the solutions found in Table II may be local optima [51]. We also run one example for 12 h to see whether those algorithms will be trapped in a local optimum. This scenario consists of randomized data for 120 weapons and 100 targets. We want to see when these algorithms reach the best fitness value. The reached best fitness value and when these algorithms reach that fitness value are listed in Table III. The fitness curves by generations are shown in Fig. 5. Evidently, in our simulation, those curves

TABLE IV
BEST FITNESS VALUES OBTAINED BY USING GA WITH GREEDY EUGENICS APPLIED TO WITH AND WITHOUT CONSTRAINED ASSIGNMENT. RESULTS ARE AVERAGED OVER TEN TRIALS

Algorithms	Scenarios			
	W=50 T=50	W=80 T=80	W=100 T=100	W=120 T=120
GA with greedy eugenics with constrained assignment	133.3829 (15.6152)	252.8929 (25.7657)	321.2934 (35.917)	427.7887 (30.7813)
GA with greedy eugenics without constrained assignment	**138.671** (19.1175)	**248.2979** (25.973)	**256.9861** (28.5617)	**378.6415** (30.1763)

TABLE V
COMPARISONS OF THE BEST SOLUTIONS OBTAINED BY VARIOUS ALGORITHMS FOR QAP. RESULTS ARE AVERAGED OVER TEN TRIALS WITH A CONSTANT COMPUTATION TIME OF ONE HOUR

Algorithms	Nugent (15)	Nugent (20)	Nugent (25)	Nugent (30)
GA with simple eugenics	1150 (0)	2590 (15.6)	3842 (32.8)	6295 (47.3)
GA with SA as eugenics	1150 (0)	2570 (0)	3816 (30.4)	6264 (43.5)
GA with immunity as eugenics	1150 (0)	2570 (0)	3744 (0)	6208 (20.2)
GA with greedy eugenics	**1150 (0)**	**2570 (0)**	**3744 (0)**	**6124 (0)**

are similar and can all converge to the same fixed value. This is because they all use the same GA framework. From Table III, it is evident that the proposed algorithm used the least time to converge to the optimal solution among those algorithms. Next, the greedy eugenics with the constrained assignment are used. From Table IV, it can be found that when the numbers ($W = T = 50$) are small, the constrained assignment can find a better solution due to its efficient search capability. However, when the numbers increase, the found solutions become worse than those without the constrained assignment.

Finally, we also use the greedy eugenics to test QAP [53], [54]. QAP, one of the classical NP-hard problems, is a combinatorial optimization problem found in the optimal assignment of facilities to allocations. The greedy eugenics are similar to that used for constraint WTA problems stated above; that is, to forward or backward modify the assigned facility-to-location pairs in the chromosome. In our paper, the test problems of QAP are taken from QAPLIB [53]. The results are listed in Table V. It is easy to see that GA with greedy eugenics also outperforms other algorithms in QAP, as expected.

VI. CONCLUSIONS

In this paper, a GA with greedy eugenics is proposed to solve general WTA problems. Even though various GAs with eugenic mechanisms have been proposed in the literature, those eugenic mechanisms are considered as mechanisms used for enhancing the search capability for GA. In our research, we viewed the search in GA as a mechanism providing a main portion of diversity in search and the eugenics as a major role nailing down a local optimum for the solution provided by GA. Thus, in this paper, the proposed eugenic scheme is to greedily reform the current chromosome. Even though greedy algorithms may have a great possibility of being trapped into a local optimum, due to crossover and mutation operations and the parallel search nature used in GAs, the search can easily escape from local optima. When compared to existing search algorithms, including general GA, SA, GA with general eugenics, GA with SA as eugenics, GA with immunity as eugenics, and repeating randomly made chromosome with greedy eugenics, the proposed algorithm obviously outperforms those algorithms.

REFERENCES

[1] S. P. Lloyd and H. S. Witsenhausen, "Weapon allocation is NP-complete," in IEEE Summer Simulation Conf., Reno, NV, 1986.

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

[3] A. M. H. Bjorndal *et al.*, "Some thoughts on combinatorial optimization," *Eur. J. Oper. Res.*, pp. 253–270, 1995.

[4] P. L. Hammer, "Some network flow problems solved with pseudo-boolean programming," *Oper. Res.*, vol. 13, pp. 388–399, 1965.

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

[6] T. Ibarraki and N. Katoh, *Resource Allocation Problems*. Cambridge, MA: MIT Press, 1988.

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

[8] H. A. Eiselt and G. Laporate, "A combinatorial optimization problem arising in dartboard design," *J. Oper. Res. Soc.*, vol. 42, pp. 113–181, 1991.

[9] W. K. Lai and G. G. Coghill, "Channel assignment through evolutionary optimization," *IEEE Trans. Veh. Technol.*, vol. 45, pp. 91–96, Jan. 1998.

[10] D. M. Tate and A. E. Smith, "A genetic approach to the quadratic assignment problem," *Comput. Oper. Res.*, vol. 22, no. 1, pp. 73–83, 1995.

[11] C. Y. Ngo and V. O. K. Li, "Fixed channel assignment in cellular radio networks using a modified genetic algorithm," *IEEE Trans. Veh. Technol.*, vol. 47, Jan. 1998.

[12] Y. Sun and Z. Wang, "The genetic algorithm for 0-1 programming with linear constraints," in *Proc. 1st IEEE Conf. Evolutionary Computation*, 1994.

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

[14] T. Bäck, U. Hammel, and H.-P. Schwefel, "Evolutionary computation: Comments on the history and current state," *IEEE Trans. Evol. Comput.*, vol. 1, pp. 3–17, Apr. 1997.

[15] M. Gen and R. Cheng, *Genetic Algorithms and Engineering Design*. New York: Wiley, 1997.

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

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

[18] L. Davis, *Genetic Algorithms and Simulated Annealing*. San Mateo, CA: Morgan Kaufmann, 1987.

[19] P. Merz and B. Freisleben, "Fitness landscape analysis and memetic algorithms for quadratic assignment problem," *IEEE Trans. Evol. Comput.*, vol. 4, pp. 337–352, Nov. 2000.

[20] L. Jiao and L. Wang, "Novel genetic algorithm based on immunity," *IEEE Trans. Syst., Man, Cybern. A*, vol. 30, pp. 552–561, Sept. 2000.

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

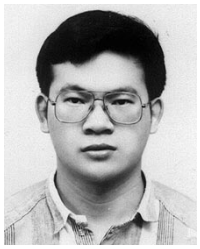
[22] M. Dorigo and G. Di Caro, "Ant colony optimization: A new meta-heuristic," in *Proc. 1999 Congr. Evolutionary Computation*, vol. 2, 1999, pp. 1470–1477.

[23] E. Bonabeau, M. Dorigo, and G. Theraulaz, *Swarm Intelligence From Natural to Artificial Systems*. London, U.K.: Oxford Univ. Press, 1999.

[24] T. A. Feo and M. G. C. Resende, "Greedy randomized adaptive search procedure," *J. Global Optim.*, vol. 6, pp. 109–113, 1995.

[25] A. Kolen and E. Pesch, "Genetic local search in combinatorial optimization," *Discr. Appl. Math. Combin. Oper. Res. Comput. Sci.*, vol. 48, pp. 273–284, 1994.

[26] 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 *Proc. 1st Workshop PPSN*, vol. 496, Lecture Notes in Computer Science, Schwefel and Männer, Eds., 1991, pp. 109–116.
[27] C.-T Lin and C. S. G. Lee, *Neural Fuzzy Systems: A Neural Fuzzy Synergism to Intelligent Systems*. Englewood Cliffs, NJ: Prentice-Hall, 1995.
[28] N. J. Radcliffe and P. D. Surry, "Formal memetic algorithms," in *Proc. Evolutionary Computing: Selected Papers From the AISB Workshop*, 1994, pp. 1–16.
[29] P. Moscato and M. G. Norman, "A memetic approach for the traveling salesman problem. Implementation of computational ecology on message passing systems," in *Parallel Comput. Transport. Applicat.*, 1992, pp. 187–194.
[30] R. Dawkins, *The Selfish Gene*. London, U.K.: Oxford Univ. Press, 1976.
[31] D. L. Pepyne *et al.*, "A decision aid for theater missile defense," in *Proc. IEEE ICEC*, 1997.
[32] *Handbook of Genetic Algorithms*, L. Davis, Ed., Van Nostrand, New York, 1991.
[33] H.-P Schwefel, *Numerische Optimierung von Computer-Modellen Mittels der Evolutionsstrategie*. Basel, Germany: Birkäuser Verlag, 1977, vol. 26, Interdisciplinary Systems Research.
[34] E. K. Burke and A. J. Smith, "Hybrid evolutionary techniques for the maintenance scheduling problem," *IEEE Trans. Power Syst.*, vol. 15, pp. 122–128, Feb. 2000.
[35] R. J. W. Hodgson, "Memetic algorithms and the molecular geometry optimization problem," in *Proc. Congr. Evolutionary Computation*, vol. 1, 2000, pp. 625–632.
[36] J. D. Knowles and D. W. Corne, "M-PAES: A memetic algorithm for multi objective optimization," in *Proc. Congr. Evolutionary Computation*, vol. 1, 2000, pp. 325–332.
[37] R. Cheng and M. Gen, "Parallel machine scheduling problems using memetic algorithms," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, vol. 4, 1996, pp. 2665–2670.
[38] J. Miller, W. Potter, R. Gandham, and C. Lapena, "An evaluation of local improvement operators for genetic algorithms," *IEEE Trans. Syst., Man, Cybern.*, vol. 23, pp. 1340–1341, Sept./Oct. 1993.
[39] E. H. L. Aarts and J. K. Lenstra, *Local Search in Combinatorial Optimization*. New York: Wiley, 1997.
[40] P. J. M. Van Laarhoven and E. H. L. Arts, *Simulated Annealing: Theory and Applications*. Norwell, MA: Kluwer, 1992.
[41] 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. Syst., Man, Cybern.*, vol. 23, pp. 1752–1767, Nov./Dec. 1993.
[42] E. H. L. Aarts and J. Korst, *Simulated Annealing and Boltzmann Machines*. New York: Wiley, 1989.
[43] S. Kirkpatrick, C. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.
[44] N. Metropolis, A. Rosenbluth, M. Rosenbluth, and A. Teller, "Equation of state calculations by fast computing machines," *J. Chem. Phys.*, vol. 21, pp. 1087–1092, 1953.
[45] J.-S. R. Jang, C. -T. Sun, and E. Mizutani, *Neuro-Fuzzy and Soft Computing*. Englewood Cliffs, NJ: Prentice-Hall, 1997.
[46] S. A. Frank, *The Design of Natural and Artificial Adaptive Systems*. New York: Academic, 1996.
[47] D. Dasgupta and N. Attoh-Okine, "Immunity-based systems: A survey," in *Proc. IEEE Int. Conf. Syst., Man, Cybern., Computational Cybernetics and Simulation*, vol. 1, 1997, pp. 369–374.
[48] A. Gasper and P. Collard, "From GAs to artificial immune systems: Improving adaptation in time dependent optimization," in *Proc. Congr. Evolutionary Computation*, vol. 3, 1999, pp. 1999–1866.
[49] C. R. Reeves, "Genetic algorithms and neighborhood search," in *Evolutionary Computing: AISB Workshop*, Selected Papers, no. 865 in Lecture Notes in Computer Science, T. C. Forgarty, Ed., Leeds, U.K., Apr. 1994.
[50] D. Hausmann, B. Korte, and T. A. Jenkyns, "Worst case analysis of greedy-type algorithms for independence system," *Math. Programming Study*, vol. 12, 1980.
[51] Z.-J. Lee, S.-F. Su, and C.-Y. Lee, "A genetic algorithm with domain knowledge for weapon-target assignment problems," *J. Chin. Inst. Eng.*, vol. 25, no. 3, pp. 287–295, 2002.
[52] Z.-J. Lee, S.-F. Su, C.-Y. Lee, and Y.-S. Hung, "A heuristic genetic algorithm for solving resource allocation problems," Int. J. Knowl. Inf. Syst..
[53] R. E. Burkard, S. E. Karisch, and F. Rendl, "QAPLIB-A quadratic assignment problem library," Technical Univ. Graz, Austria, Tech. Rep. 287, 1994.
[54] V. Nissen, "Solving the quadratic assignment problem with clues from nature," *IEEE Trans. Neural Networks*, vol. 5, pp. 66–72, Jan. 1994.

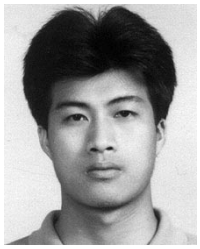


Zne-Jung Lee was born in Taiwan, R.O.C., in 1963. He received the B.S. degree in electronic engineering from Feng-Chia University (FCU), Taiwan, in 1986, the M.S. degree in automatic control engineering from FCU in 1988, and the Ph.D. degree in electrical engineering from National Taiwan University of Science and Technology, Taipei, in 2002.

From 1988 to 1999, he was an Assistant Research at the Chung-Shan Institute of Science and Technology, Taiwan.

Shun-Feng Su (S'89–M'91) received the B.S. degree in electrical engineering from the National Taiwan University, Taipei, Taiwan, R.O.C., in 1983, and the M.S. and Ph.D. degrees in electrical engineering, from Purdue University, West Lafayette, IN, in 1989 and 1991, respectively,

He is currently a Professor in the Department of Electrical Engineering, National Taiwan University of Science and Technology. His current research interests include neural networks, fuzzy modeling, machine learning, virtual reality simulation, data mining, and intelligent control.



Chou-Yuan Lee was born in Taiwan, R.O.C., in 1966. He received the B.S. and M.S. degrees in automatic control engineering from Feng-Chia University, Taiwan, in 1989 and 1991, respectively. He is currently pursuing the Ph.D. degree in electrical engineering at the National Taiwan University of Science and Technology, Taipei.

His current research interests are neural networks, fuzzy systems, and genetic algorithms.