Proceedings of the 6th World Congress on Intelligent Control
and Automation, June 21 - 23, 2006, Dalian, China

An Improved Genetic Algorithm for Target Assignment Optimization of Naval Fleet Air Defense

Houqing Lu, Hongjun Zhang and Xiaojuan Zhang
Engineering Institute of Engineering Corps PLA Univ. of Sci. & Tech.
Nanjing, Jiangsu Province, China
luhouqing@yahoo.com.cn

Ruixin Han
Naval War Gaming Center
Naval Command College
Nanjing, Jiangsu Province, 210016, China
ruixinsun@126.com

***Abstract* - As a weapon-target assignment problem (WTA), the fire-allocation of naval fleet air defense was studied and an optimal target assignment model was established. By combining the features of naval fleet anti-air combat, several classical assignment algorithms were discussed. To find global optimal results efficiently, an effective global search method i.e. genetic algorithms (GAs) was applied. However, this method still has disadvantages of slow convergence and poor stability in practical engineering. To solve these problems, an improved GA was proposed in terms of uniform creation of initial population, selection based on fitness scaling and self-adaptive parameters for genetic operators, etc. In the end, the steps for solving the optimal model were put forward, and satisfactory results have being obtained. It is shown that the present method can serve as a scientific and effective support for a decision maker in command automation of the naval fleet air defense combat.**

Index Terms* - *Naval fleet. Air defense. Weapon-target assignment. Genetic algorithm. Optimization.

I. INTRODUCTION

The main mission of a naval fleet air defense is to destroy the attack aircraft, pilotless aircraft and head off enemy's cruise missile, anti-fleet missile and anti-radiation missile, etc. The objective of fire-allocation is to find an optimal assignment of weapons to aerial targets [1,2], and head off aerial attackers efficiently.

Naval fleet air defense WTA is a typical Integer Programming and belongs to 0-1 Programming problem. There are some classic methods for solving such NP-complete problems [3]. Enumerate method is not practicable, due to the numerous feasible integer combinations when many variables exist. Implicit Enumeration can reduce the operation times and increase the optimization speed by adding the flitting constraint and only part of the variable combinations need to be checked. Branch and Band Method is also an Implicit Enumeration method. Other methods such as Hungary Method proposed by Kuhn in 1955, citing the zero in matrix theorem of Hungarian mathematician D.Konig; Shortest Path presented by Dijkstra and the Dynamic Programming for multi-stage decision, etc. These traditional non-linear programming optimization methods are based on graph search approaches. When the problem size is large, those methods usually result in exponential computational complexities. For the considered combination problem [4,5], the optimal solution can be found only for very small problems.

A GA based search is capable of solving large and difficult problems [6,7]. GAs provides a general purpose robust search methodology [8,9], which uses principles inspired by natural genetics "survival of the fittest, excellent win and inferior discard". In modern naval battle, naval fleet faced numerous aerial targets. The solution space of air defense WTA is in great size. According to the general principle and optimal criterion of WTA, a target assignment model is established and an improved GA is proposed.

II. PROBLEM FORMULATION

After obtaining exact tracks of the targets, the naval fleet air defense command control system make an optimal assignment of weapons to targets according to the target-assignment model or the artificial intelligent system model. For convenience of calculation, the studied WTA is simplified to an assignment problem for fire-units in one fire-turnover cycle.

1) Tactics background: Suppose that there are n weapons and m aerial-targets, every weapon only head off one target and every targets might be assigned several weapons. P_{ij} stands for the individual probability of killing by assigning the j-th target to the i-th weapon. This probability defines the effectiveness of the i-th weapon to destroy the j-th target.

2) Assignment matrix: $\boldsymbol{V}=(v_{ij})_{n\times m}$, where v_{ij} represents whether the i-th weapon is assigned to the j-th target or not. v_{ij} is a Boolean value, 1 means the i-th weapon is assigned to the j-th target, and 0 is otherwise.

3) Kill expectation: The overall probability of killing value for j-th target to damage the asset is computed as

$$p_j = 1 - \prod_{i=1}^{n} \left(1 - p_{ij} v_{ij}\right).$$

The considered WTA is to maximize the expectation of destroyed targets [1,10]. So, the object function (M) is

$$(M) \quad \max \sum_{j=1}^{m} r_j \left(1 - \prod_{i=1}^{n} \left(1 - p_{ij} v_{ij}\right)\right). \tag{1}$$

Subject to the assumption that all weapons must be assigned to targets, i.e.

$$s.t. \quad \sum_{j=1}^{m} v_{ij} = 1, \; v_{ij} = 0 \;\; or \;\; 1$$
$$r_i, p_{ij} \in [0,1] \tag{2}$$
$$i = 1,2\cdots,n; \; j = 1,2\cdots,m$$

In which r_j is the expected damage value of the j-th batch targets to the asset.

In model (M), as a typical combine optimization problem, there are $(m+1)^n$ assignment solutions. Thus, the solving algorithm must have high searching speed and must have the ability of finding satisfactory solutions.

III. IMPROVED GENETIC ALGORITHM

GAs have been widely applied in many different fields such as computer science, information science, artificial intelligence research and operations research since its introduction by Holland (1975). Goldberg (1989) summarized these algorithms and achieved a simple genetic algorithm (SGA), which is the basis of GAs research [11,12].

SGA involves the following procedures: creation of population, evaluation of cost, mate selection, reproduction and mutation. Using the procedures of mate selection, reproduction and mutation, a new population is created and the whole circle is repeated for a certain number of times or until a proper stop criterion is fulfilled.

Though GAs has being widely used in complex engineering optimization problem, its slow convergence speed and poor computing stability becomes a difficult problem in its application [13,14]. To enhance its performance, an improved genetic algorithm (MGA) is proposed.

A. Encode

1) Decimal encode strategy: GAs code a solution as a finite-length string over some finite alphabet. If there are n weapon units, the length of the string is n and every code is an integer. The string can be presented by $S=C_1C_2...C_i..C_n$, C_i is the code of the target assigned to the i-th weapon unit, here 1•C_i•m. The string S of the individual is also referred to as chromosome, and C_i means a gene.

2) Generate of uniform initial population: The genes are selected by means of random integers from appropriate regions where they are supposed to exist. Therefore, the initial population is randomly generated within pre-established ranges of the genes $C_1, C_2, \ldots, C_n$, by using a random seed which is different in each run of the genetic algorithm.

To cover homogeneously the whole solution space, and to avoid the risk of having too much individuals in the same region [15], extended Hamming distance is cited as the judgment of generating scheme.

*Definition.*1. Give two strings of the same length and with the same basis of , the different number of the gene in correspondence location is called extended Hamming distance of the two strings, presented by *.

Assuming that the individual is a string based on and its length is n, extended Hamming distance of individual i and j is performed as:

$$\rho^*_{ij} = \sum_{k=1}^{n} \mathrm{sgn}\left|S_i(C_k) - S_j(C_k)\right|, \quad i \neq j.$$

ρ^*_{ij} among every individuals in the selected population must be satisfied by:

$$\rho_{ij}^* \geq (n-b), \qquad i \neq j. \tag{3}$$

Where $i,j=1,2,\ldots,N$; b is a constant, and in decimal encoding, b can value from b•2.

This method must compute extended Hamming distance among every initial individual and make sure that the distance is more than a given distance range.

B. Initial population and fitness function

All the potential weapon-assignments compose the solution space (also called population). Each feasible solution may be a parent. The fitness function evaluates strings by quantifying the fitness of the strings or solutions. The fitness value of an individual is always positive number and determines the quality of the solution. In this case, higher expectations imply better fitness, and the fitness is maximized during the optimization process.

The proposed fitness function is identical to the objective function of the formulation (M) shown in (1).

C. Selection

1) Roulette wheel selection: The selection method of individuals has a significant influence on driving the search towards a promising area and finding a good solution within a short time. The roulette wheel selection (also referred to as fitness-proportional selection) proposed by Holland [16,17] is the most famous, and the selection probability for each individual is proportional to its fitness value. Hence, the individual i is selected with the probability:

$$p_{Si} = f_i \Big/ \sum_{i=1}^{N} f_i. \tag{4}$$

Where f_i is the fitness of the individual i, p_{si} is the probability of individual selection; and N is the number of individuals in the population.

2) Fitness scaling: Whitley [18] pointed out two weaknesses for the fitness proportional selection, which are stagnation and premature convergence of the search. When the relative difference between fitness values of individuals is small the search process stagnates. On the other hand, when the relative difference between the fitness values of individuals is large, the fittest individuals dominate the creation of the next generation. Consequently, the search prematurely converges to a solution.

Fitness scaling has two purposes: (i) Sustain reasonable variable of the relative fitness proportion among the chromosomes; and (ii) in the early stage of optimization, weaken the competition by constraining the extensively fast convergence of some super chromosomes, and in the later speedup the competition. The convert function can be divided into linear scaling [19], scaling with truncation [20], power law scaling and logarithmic scaling, etc. Here sigma truncation scaling is employed, that is

$$F' = F - (\overline{F} - c\sigma). \tag{5}$$

Where, $\overline{F}$ is average fitness value; c is a constant and is the standard variance of individual fitness, it is computed as

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - f_{avg})^2}. \tag{6}$$

Here, N means population size, f_i is the fitness of the i-th individual and f_{avg} is the average value of all the individual fitness. indicates the discrete level of individual fitness. The bigger the more discrete. This method can avoid the appearance of minus.

To compute faster, decrease the calculation of , set:

$$c' = c/\sqrt{N} \quad , \sigma' = \sum_{i=1}^{N} \left| f_i - f_{avg} \right| .$$

Let c' and $'$ replace c and , then

$$c' \sigma' = c\sqrt{\frac{1}{N}\sum_{i=1}^{N}(f_i - f_{avg})^2} = c\,\sigma .$$

3) Steady-state reproduction: The temporary new population is composed through selection and genetic operator from the current generation. The new individuals may be superior or inferior to existing individuals in the current population. We apply elitism that replaces bad individuals of next population with good individuals of current population. Some individuals are moved to next generation not through genetic operator.

Set the replace percentage R_p, in the combination of new population there must have R_p inferior ancestor individuals replaced by R_p newly-generate elitism individuals. Thus, the best chromosomes are preserved, and offspring is selected due to its survival probability. This reproduce method allowing ancestor to attend competition is called overlapping population. It is helpful in sustaining the variety of population and avoiding premature convergence.

4) Crossover: Adopt two-point crossover operation. Set two crossover points randomly in the coding mating string and then change the genes with the same indexes, as shown in Fig.1. The generated child chromosomes can inherit good characters of parents.

A: 3 6 3 5 2 1 6 1   B: 4 6 5 2 4 3 1 7

A': 3 6 5 2 4 3 6 1   B': 4 6 3 5 2 1 1 7

Fig. 1 Two-point crossover operator

Randomly select two crossover points 3rd and 6th in parents A and B. After crossover operation, offspring chromosome A' and B' are created.

4) Mutation: Employ swap mutation operator (Gen and Cheng, 1997). Select two positions within a chromosome at random, then exchange the two genetic values on the two positions. In a mutation process, the 3rd and the 7th positions are randomly selected. If the mutation operation is meant to be performed, then the offspring become:

A : 3 6 5 2 4 3 6 1

A′ : 3 6 6 2 4 3 5 1

Fig. 2 Swap mutation operator

To overcome the premature convergence and stagnancy in the evolution of SGA, many different adaptive GAs have been proposed [21-23]. Adaptive techniques have been suggested to adjust parameters in the process of running the genetic algorithm. These include adapting mutation probability (P_m), crossover probability (P_c), population size (N) [24,25], and even the crossover operator [26]. According to the distribution character of the fitness value of every evolution population, Srinivas M and Patnaik L M dynamically adjusted [27] the probability of mutation p_c and p_m:

$$Pc = \begin{cases} 0.7\left(f_{\max} - f_{big}\right)/\left(f_{\max} - f_{avg}\right) & f_{big} \ge f_{avg} \\ 1.0 & f_{big} < f_{avg} \end{cases} . \tag{7}$$

$$P_m = \begin{cases} 0.5\left(f_{\max} - f_i\right)/\left(f_{\max} - f_{avg}\right) & f_i \le f_{avg} \\ 1.0 & f_i < f_{avg} \end{cases} . \tag{8}$$

where, f_i is the fitness value of the i-th individual, f_{big} is the bigger one of two cross bred individuals; f_{max} and f_{avg} is the maximum and average fitness value of current population.

GAs is sometimes trapped into a local optimum. Using larger mutation rate in (8) can overcome this difficulty, by enhancing the diversity of individuals, preceding the intensification.

III. Implementation of Proposed MGA

To carry out the proposed MGA, programs were coded in MATLAB 7.0 and experiments were run on PCs with Pentium Ⅳ 2.6GHz processor. The flow chart is shown in Fig. 3:

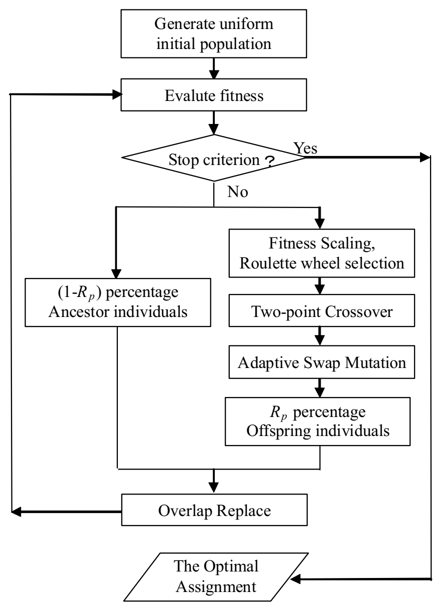

Fig.3 Flow chart of the improved genetic algorithm

The processes of crossover, selection and replacement are repeated until a termination criterion is met. The simplest criterion is a pre-specified maximum number of generations. In this method, the process is terminated after 100 generations are performed. Tests showed that 100 generations were sufficient for convergence.

IV. Example Application

Give 7 batches targets and 11 weapons. The target values r_j is listed in the below Table •. The kill possibilities p_{ij} of every weapon unit is listed in Table II.

TABLE Ⅰ
Importance Weight Of Target

Target batch	1	2	3	4	5	6	7

Target value	0.7	0.5	1	0.9	0.7	0.7	0.5

To confirm the higher efficiency of the proposed algorithm, the random search method, SGA and the MGA have been carried out separately.

TABLE II
KILL PROBABILITY OF WEAPON TO TARGET

Weapon	Kill Probability p_{ij} to Target						
	1	2	3	4	5	6	7
1	0	0.5	0	0	0.7	0.5	0
2	0.7	0	0.5	0.8	0.8	0.7	0.5
3	0.5	0.7	0.9	0	0.7	0	0
4	0	0	0.7	0.6	0	0.5	0.8
5	0.6	0	0.5	0.7	0.5	0	0.7
6	0.8	0.5	0	0	0.7	0.5	0.5
7	0.5	0	0.7	0.6	0	0.8	0.5
8	0	0.7	0.7	0.6	0	0.5	0.7
9	0.5	0.7	0	0.8	0.7	0.6	0
10	0.5	0.5	0.5	0.7	0	0	0.5
11	0	0.7	0.8	0.5	0.5	0	0.7

In this study, the control parameter of the algorithm is selected as: population size N=50, number of generations T=100, truncation constant c'=0.66, replace proportion R_p=0.25. The crossover possibility P_c and mutation possibility P_m are adjusted dynamically.

V. RESULTS AND DISCUSSION

The evaluation of performance for such searching algorithm is generally considered as the efficiency and quality of explored solutions. The efficiency is to compare the exhausted time for the same feasible solution, and the quality is to compare the feasible solution reached in same time. The comparison of performance is shown in Fig. 4.

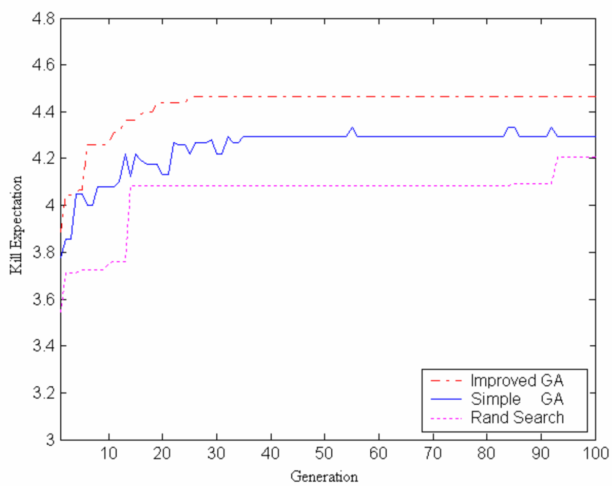

Fig.4 Evolution process of three search methods

With regard to the convergence speed, repeat the trial for 1000 times and compare the average kill expectation of every generation. It is shown in Fig. 5 that MGA can reach the optimal solutions in 30 generations. However to do the same work 80 generations is needed for SGA. Compared with SGA, it is evident that MGA has excellent performance. Meanwhile the random search algorithm has lowest convergence speed.

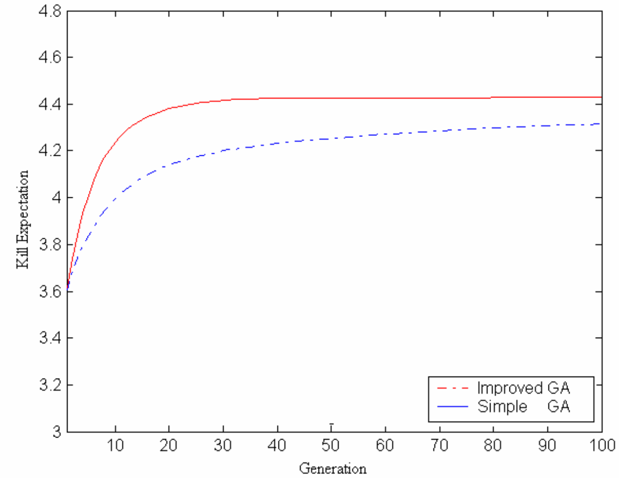

Fig.5 Averaged kill expectation of MGA and SGA

It is found that MGA yields optimal solutions in nearly 20 generations, and SGA in 60 generations while random search cannot obtain satisfied solutions in 100 generations.

Fig. 6 gives kill expectation distribution of optimal target assignments of MGA and SGA in 1000 trials. The algorithms obtain the optimal solution when the maximum evolution generation comes to 100, and compute kill expectation of the obtained optimal solution with the object function (M).

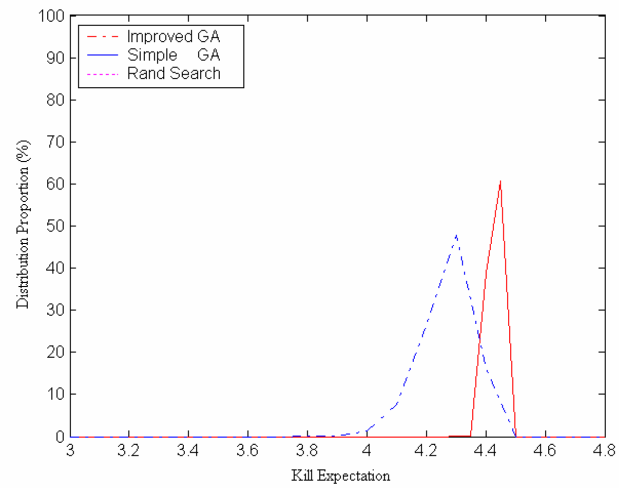

Fig.6 Expectation distribution of optimal target assignment

It can be seen that MGA exhibits higher efficiency in handling weapon-target assignment problems, and can yield solutions better than those given by SGA search method.

As shown in Fig. 6 and listed in Table III, the probability of the global solution or satisfied solution, that is kill expectation is more than 4.407, obtained by MGA is as high as 99.7%, the minimum object value is 4.345, with the average expectation of 4.455. The probability of the global solution or satisfied solution obtained by SGA is only 16.2%, the minimum object value is as low as 3.836, with the average expectation of 4.319.

TABLE III
EXPECTATION DISTRIBUTION OF KILL PROBABILITY FOR OPTIMUM

Algorithm	Distribution of Kill Probability(%)

	<4.1	≥4.1	≥4.2	≥4.3	≥4.4	≥4.45	≥4.5
SGA	1.6	7.8	26.6	47.8	14.7	1.5	0.0
MGA	0.0	0.0	0.0	0.3	38.9	60.8	0.0

Table Ⅳ shows the achieved global optimal assignment, and the overall kill possibility is 4.463.

TABLE Ⅳ
GLOBAL OPTIMAL TARGET-ASSIGNMENT

Target	1	2	3	4	5	6	7
Weapon	5,6	8,11	3	9,10	1,2	7	4

As can be seen, the assignment of every target is optimal and the weapons of high killing possibility have been assigned to 3rd and 4th target, in accord with the general principle and optimal criterion of WTA.

VI. CONCLUSION

In practical application, pre-mature convergence appears and the pure ability of local searching optimal solution in GAs. Aiming at this disadvantage, some improvements is made in the generation of initial population and the operation of genetic operator. Both the convergence speed and computing stability have been enhanced.

Computation results of naval fleet air defense WTA problem indicating that the MGA can find solutions of perfect quality and high searching efficiency in contrast to the existing search algorithms.

REFERENCES

[1] S.P. Lloyd and H.S. Witsenhausen, "Weapon allocation is NP-complete," in Proc. of the IEEE Summer Simulation Conference, 1986.

[2] 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] Z.J. Lee and W.L. Lee, "A Hybrid Search Algorithm of Ant Colony Optimization and Genetic Algorithm Applied to Weapon-Target Assignment Problems," Springer-Verlag Berlin Heidelberg 2003, pp. 278-285.

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

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

[6] M. Dorigo and G. D. Caro: "Ant colony optimization: A new meta-heuristic," in Proc. of the 1999 Congress on Evolutionary Computation, vol. 2, pp. 1470-1477, 1999.

[7] E. Bonabeau, M. Dorigo, and G. Theraulaz, "*Swarm Intelligence From Natural to Artificial Systems*," Oxford University Press, 1999.

[8] Z.J. Lee, S.F. Su, and C.Y. Lee, "A Genetic Algorithm with Domain Knowledge for Weapon-Target Assignment Problems," *Journal of the Chinese Institute of Engineers*, vol. 25, no. 3, pp. 287-295, 2002.

[9] Z.J. Lee, S.F. Su, and C.Y. Lee, "An Immunity Based Ant Colony Optimization Algorithm for Solving Weapon-Target Assignment Problem," *Applied Soft Computing*, vol. 2, pp. 39-47, 2002.

[10] 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. of the 2000 ROC Automatic Control Conference, 2000, pp. 163–168.

[11] Z. Michalewicz, "*Genetic Algorithms CData StructuresD Evolution Programs*," 2nd ed., Springer, 1996.

[12] D.E. Goldberg, "*Genetic Algorithms in Search, Optimization, and Machine Learning*," Addison-Wesley, 1989.

[13] C.R. Reeves, "*Modern Heuristic Techniques for Combinatorial Problems*," Blackwell Scientific Publications, Oxford, 1993.

[14] P. Merz and B. Freisleben, "A comparison of memetic algorithms, tabu search, and ant colonies for the quadratic assignment problem," in Proc. of the 1999 Congress on Evolutionary Computation, vol. 3, pp. 2063-2070, 1999.

[15] M. Gen and R. Cheng, "*Genetics algorithms and engineering optimization*," Wiley, New York, 2000.

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

[17] D. Goldberg and R. Lingle, "Alleles, loci and the traveling salesman problem," in Proc. of the first International Conference on Genetic Algorithm, Lawrence Erlbaum, Hillsdale, NJ, 1985, pp. 154–159.

[18] D. Whitley, "The GENITOR algorithm and selection pressure: Why rank-based allocation of reproductive trials is best." in Schaffer JD, editor. Proc. of the Third International Conference on Genetic Algorithms, 4–7 June 1989, Mason University. Los Altos, CA: Morgan Kaufmann Publishers; pp. 116–21, 1989.

[19] J.J. Grefenstette and J. *Baker,* "How genetic algorithms work: acritical look at implicit parallelism," in Schaffer[564], pp. 20-27.

[20] S. Forrest, "Documentation for prisoners dilemma and norms programs that use the genetic algorithm," Ph. D. dissertation, University of Michigan-Ann Arbor, 1985.

[21] T.Y. Chen and J.C. Chung. "Improvement of simple genetic algorithm in structural design," *Int J Numer Meth Eng*, vol. 40, pp. 1323–34, 1997.

[22] S.Y. Mahfouz. "Design optimization of structural steelwork," PhD thesis, UK: University of Bradford; 1999.

[23] M.G. Sahab. "Cost optimization of reinforced concrete flat slab buildings," PhD thesis, UK: Univeristy of Bradford, 2002.

[24] M. Annuziato and S. Pizzuti, "Adaptive parameterization of evolutionary algorithms driven by reproduction and competition," atti di ESIT2000 European Symp. on Intelligent Techniques, Aachen (Germania), September 2000.

[25] A.E. Eibon, R. Hinterding, and Z. Michalewicz, "Parameter control in evolutionary algorithm," *IEEE Trans. Evolution. Comput.* 1999, pp. 124-141.

[26] W.M. Spears, "Adapting crossover in evolutionary algorithms," in Proc. of 4th Annu. Conf. on Evolutionary Computing, Ed Fogel, MIT Press, Cambridge, MA, 1995.

[27] M. Srinivas and L.M. Patnaik, "Adaptive probabilities of crossover and mutation in genetic algorithms," *IEEE Trans. Systems, Man Cybernet.* vol. 24, no. 4, pp. 656–667, 1994.