A Study of Parameter Optimization in Meta-Genetic Algorithm

Jun SHIRATAKI Takehiko TOMIKAWA
Dept. of Electrical & Electronic Engineering

Abstract

Genetic Algorithm is governed by its system parameters which are usually set in a heuristic manner at present. These parameters; the reproduction rate, the crossover rate, the mutation rate, and the population size are set to appropriate values depending on the characteristics of the target problem. This paper studies the behavior of the Meta-GA system which searches for the best parameter values for the problem oriented task-GA. Here the "number of generation cycles" is utilized to evaluate the performance of the task-GA. The experimental simulation was carried out to solve the standard functions of DeJong while the parameters of meta-GA are fixed.

Keywords: Genetic Algorithm, Meta-GA, Parameter Optimization

1. Introduction

Genetic Algorithm known as an evolutionary model in artificial life is assumed to be an effective search method for optimization problems. In this algorithm, appropriate parameter values have to be installed for GA activation though they depend strongly on the user's experience. There is research reported for optimizing the parameter values of a 2nd-GA (task-GA) by the 1st-GA(meta-GA)[1]. It is necessary to receive some performance index of the task-GA on whether optimization cycle progresses reasonably or not according to the given parameter values. For instance, Freisleben(1993) and Aiyoshi(1992) made some comparative study between Meta-GA search and Random-search, and both of them reported that the former was preferable than the latter in optimization performance[2][3]. They, however, both utilized the "fitness value" as a performance index of task-GA though under different experimental circumstances. On the other hand, our method utilizes the "number of generation cycles" of task-GA as a performance evaluation[4].

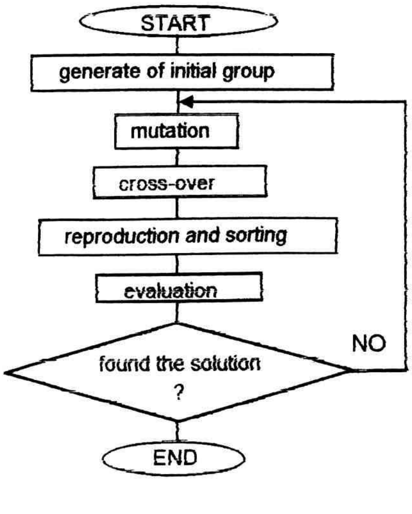


Fig. 1 Flow of Simple GA System

2. System Configurations

In GA to GA connected system (Meta-GA system), task-GA directly handles any given problems while meta-GA operates some performance index of task-GA as a feedback evaluation parameter. Therefore meta-GA becomes a kind of system controller to find suitable parameter values of task-GA based on the performance evaluations. The authors have used "the number of generation cycles" as a performance index of task-GA, and "the fitness value" as well to make some comparative study. The evaluation parameters of task-GA consist of four kinds; population size, reproduction rate, crossover rate, and mutation rate. A "Simple GA" used in this system is shown in Fig.1 as a process flow[5]. In this Meta-GA system trial, both meta-GA and task-GA are based on the processing procedure of "Simple GA system". That is, (1)creating initial population by generating random numbers, (2)doing reproduction and deletion according to the elite selection, (3)taking crossover, (4)reversing bits in mutation, and (5)evaluating by fitness function. Flow diagram of the Meta-GA connected system used in this trial is shown in Fig.2.

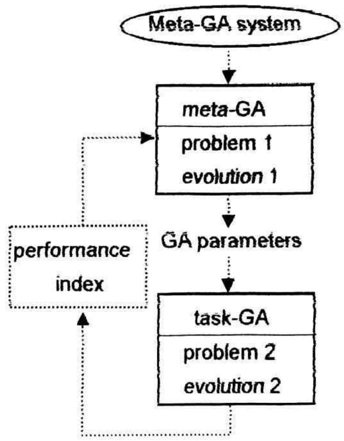


Fig. 2 Flow of Meta-GA system

3. Processing Procedure

In this experiments, prior preparations as well as activating procedure for the system are described below.

3.1 Sample Problems in Task-GA

The task-GA is used to solve some sample problems. The standard functions which DeJong had reported[6] are chosen for the optimizing targets in task-GA. Three of the functions used in this experiments are: Parabola (f_1), Rosenbrock's saddle (f_2), Step function (f_3), as

Parabola : $f_1(x_i)=\sum_{1}^{3} x_i^2 \quad (-5.11 \leqq x_i \leqq 5.12)$

Rosenbrock's addle : $f_2(x_i)=100\left(x_1^2-x_2\right)^2+(1-x_1)^2$

$(-2.047 \leqq x_i \leqq 2.048)$

Step function : $f_3(x_i)=\sum_{1}^{5} \mathrm{int}\, x_i \quad (-5.11 \leqq x_i \leqq 5.12)$

3.2 Chromosome Structure

Figure 3(a) shows the chromosome structure in meta-GA consisting of four separate genes. The allocated genes correspond to the four parameters mentioned previously, and each of them must satisfy a certain defined range of the given functions. The decimal point is considered to be shifted in the given parameter values, as shown in the example of Fig.3(b).

population size	reproduction	crossover	mutation

(a) Structure

100	0.42	0.06	0.06

↓

1111	1110	0010	0001

(b) Coding

Fig. 3 Chromosome in Meta-GA

The chromosome structure in task-GA is shown in Fig.4(a). Within this structure the appropriate values in the function above can be installed. Since each of these functions has its own definition region, the length of the gene must be intrinsic. This appearance is shown in Fig.4(b) as an example. The value of (xi) in **3.1** is updated by (-T + s x t) one after another to calculate the given function. Where, (T) indicates the defined domain of the given function, and (s) and (t) indicate the conversion coefficient and the width of updated step within definition region, respectively.

X_1	X_2	X_3	X_4	X_5

(a) Structure

-2.175	-1.791

↓

000001000000	000010000000

(b) Coding

Fig. 4 Chromosome inTask-GA

3.3 Fitness Function

First of all, the fitness function of meta-GA is described. It is necessary to return the performance index of task-GA to meta-GA, so that the objective target of meta-GA would be system parameters of task-GA. The previous methods utilized the "fitness value" of task-GA as a performance index of Meta-GA system. Here the "number of generation cycles" which task-GA needs to solve any given problem is handed to meta-GA as a performance index of task-GA. The "number of generation cycles"(GEN) as well as the "fitness value"(FIT) of task-GA is studied in comparison with each other. Here (1)(GEN) or (2)(FIT) in either case can be referred by meta-GA in consecutive executing flow. And the system iterations may progress step by step with the value (GEN) becoming smaller in (1) or larger in (2) up to the highest level. Where, (FIT) mentioned here indicates the average value of the fitness values of task-GA. We also consider another "fitness value" as a performance index yet the best score among individuals of task-GA, i.e. (FIT1).

4. Experiments

The retrieved value of (GEN) or (FIT) from task-GA can be examined to determine the system performance. Here the problems to be solved in task-GA are three DeJong's functions (f1, f2 and f3). The experiments here aim to do the simulated evaluation of Meta-GA performance on workstation(SUN-16MIPS) witten in C-language. According to the authors' intuitions, the four parameters of (population size 50, reproduction rate 0.5, crossover rate 0.3, and mutation rate 0.01) for the meta-GA are fixed and remain unchanged throughout the simulation. On the other hand, other initial parameters generated randomly should be updated together with the progress of the system iterations. In this simulation, the evaluation of the obtained parameter value and the end criterion are both noted to be used to check the effectiveness of the Meta-GA system. The validity of the obtained parameter values must be confirmed by inputting them into a stand-alone single GA system. The (GEN), and the (FIT) as well, will halt its system iterations at around the best evaluation score of task-GA, i.e. within 90% levels.

The variation ranges of the four parameters described above are; the population size(10-100: 6 steps), reproduction rates(0.03-0.45: 0.03 steps), crossover rates(0.03-0.45: 0.03 steps), and mutation rates(0-0.9: 0.06 steps). The fixed length of 16 bits (four genes of four bits each) is allocated per chromosome. An integer to/from decimal conversion must be applied to adjust these numerical values for gene. If the task-GA has come to its upper limit of generation cycles (50 times) without any progress from the local optimum, it should be forced to end at this point regardless of either (GEN), (FIT), or (FIT1).

5. Considerations

The experimental results for the three functions(f1, f2, and f3) are shown in Fig.5(a)-(c), respectively. Where, three curves can be seen in each figure for the return values(GEN, FIT, and FIT1) of the task-GAs. These curves are taken to the averaged scores from ten different random starting populations with the end criterion of 100th generation in meta-GA and 50th generation in task-GA. Left and right vertical axes indicate the evaluations of meta-GA for (GEN) and (FIT), respectively. The higher the plotted position the better the evaluation score in either of the vertical axes, while the horizontal axis in each case indicates the generation cycles in meta-GA. All curves in the figures were gradually in the tendency to proceed higher with the system iterations. For instance, in Fig.5(a) for the case of (GEN) the tendency to saturated at about the 20th generation cycle can be seen.

The parameter values which were obtained in each Fig.5(a), (b), or (c) were installed into the stand-alone single GA to verify whether the obtained values are reasonable enough or not. Figure 6 shows the results of the implemented single GA using these parameters, where, (a), (b), and (c) are for the functions f1, f2, and f3, respectively. The vertical and horizontal axes indicate the evaluation and the generation cycles, respectively.

6. Conclusions

We have proposed to utilize the "number of generation cycles" as a return value of task-GA to meta-GA for a performance index. We had comparative experiments of the Meta-GA system using two different performance indexes (1) the "number of generation cycles" (GEN) and (2) the "fitness score" (FIT). As a result, the differences between (GEN) and (FIT) can not be remarkable in this experiments. Also we can not simply say that the desirable performance index in Meta-GA system is the "fitness score" in any case.

Areas needing further investigation include finding (1) other desirable variables as a performance index, (2) the optimal parameter value of meta-GA, and (3) the proper end criterion of the task-GA. Lastly, the actual application to the image data optimization is also left as a problem for further study.

Acknowledgement
The authors gratefully acknowledge Mr.H.O.Absaloms at Kanagawa Inst.of Tech. for his final check of this paper.

References

[1] Grefenstette J. J, 1986, Optimization of Control Parameters for Genetic Algorithms, IEEE, Vol.SMC-16, No.1, 122-128.

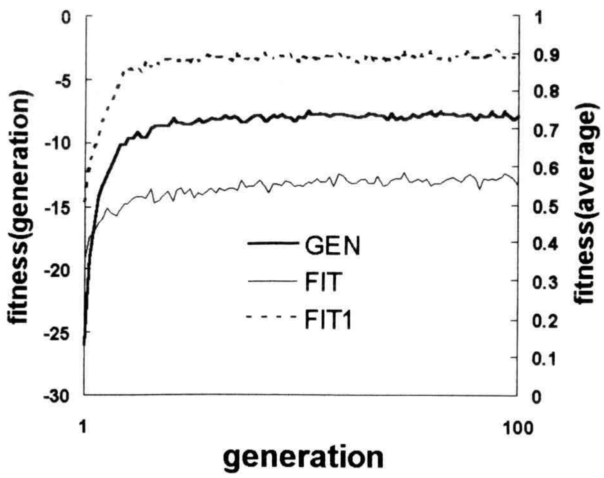
[2] Freisleben B,Hartfelder M, 1993, Optimization of Genetic Algorithms by Genetic Algorithms, Proc.Int'l Conf.Austria, Springer Verlag, 392-399.

[3] Aiyoshi E. and Mimuro N., 1992, A Meta-Optimization Problem for Global Optimization and Its Solution by the Genetic Algorithm, Vol.28, No.8, 999-1006.

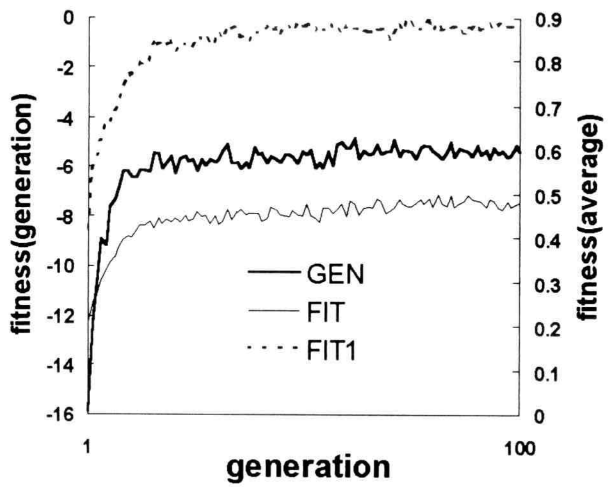
[4] Hattori M, Shirataki J, Tomikawa T, 1996, A Study of Fitness Function in Meta-GA, ICICE, Rec. of Tokyo Conf., 21.

[5] Goldberg, D. E., 1989, Genetic Algorithms in Search Optimization and Machine Learning, Addison Wesley.

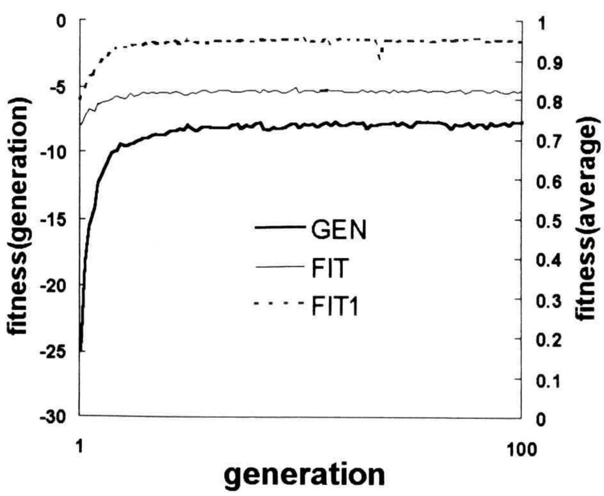
[6] Iba H., 1994, Fundamentals of Genetic Algorithm, Ohm Pub.



(a) f_1

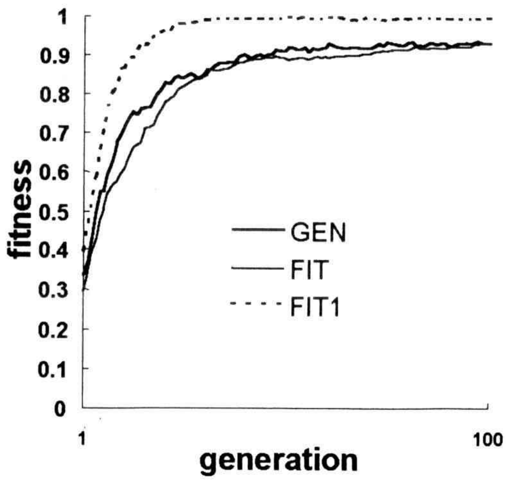


(b) f_2

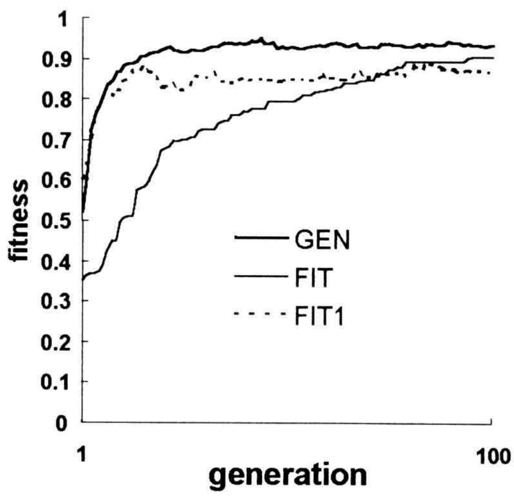


(c) f_3

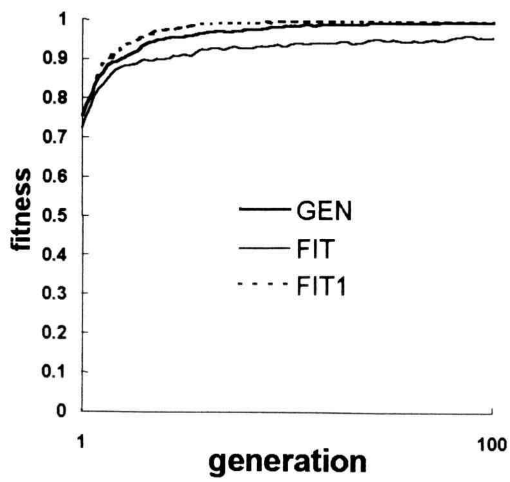
Fig. 5 Experimental Results of Fitness



(a) f_1



(b) f_2



(c) f_3

Fig. 6 Experimental Results of Single GA