Accelerating a GA Convergence by Fitting a Single-Peak Function

Ingu, Takeo
Graduate School, Kyushu Institute of Design

Hideyuki, Takagi
Department of Acoustic Design, Kyushu Institute of Design : Associate Professor

https://hdl.handle.net/2324/1670064

出版情報：IEEE International Fuzzy Systems Conference Proceedings. 1999, pp.1415-1420, 1999-08-22. IEEE
バージョン：
権利関係：

Accelerating a GA Convergence by Fitting a Single-Peak Function

Takeo INGU* Hideyuki TAKAGI**
Kyushu Institute of Design
*Graduate School and **Dept. of Acoustic Design
4-9-1, Shiobaru, Minami-ku, Fukuoka 815-8540, Japan
Tel&Fax: +81-92-553-4555, E-mail: takagi@kyushu-id.ac.jp

***Abstract*— This paper proposes an acceleration method of GA search that finds a new elite by fitting a single-peak function on GA search surface. The roughest approximation of a finite searching surface that has one global optimum would be a single-peak curved surface, and the vertex of the approximated single-peak function is expected to be near the global optimum of the original searching surface. We propose two data selection methods for the fitting, use a quadratic function as the single-peak function, and evaluate the proposed idea using seven benchmark functions. The experimental results have shown that the proposed method accelerates GA convergence.**

Keywords: *genetic algorithm, fast convergence, quadratic function, elite, data selection*

1 Introduction

Many applications requires accelerating a GA (genetic algorithm) search. Suppose we design a fuzzy controller using a GA. Applications of the obtained fuzzy controllers to the control process, which requires a fast convergence rather than a fast calculation of GA operations, take a long time to obtain fitness values for each generation. It is critical to have fewer search generations when GA's are used for human interface than for a normal GA search. For example, for an interactive GA, or an interactive EC (evolutionary computation) [13] in general form, a human operator evaluates all individuals in all generations and inputs his or her subjective feedback into an EC as fitness values. Generally, due to the fatigue of human operators, we can continue to search for the optimum solution in 10 or 20 generations at most [13].

There are several approaches to accelerate a GA search besides improving the basic GA operations or typically proposing new ones. One approach is to dynamically control the parameters of GA operations, such as population size, crossover rate, and mutation rate, by using GA [5, 2] or fuzzy rules [8]. Another approach to controlling GA parameters is to initialize poor individuals when a GA convergence becomes poor [6]. A priori knowledge of application tasks is also useful to accelerate a GA convergence when it is used for crossover and mutation [1]. Premature convergence of especially simple GAs is frequently noted. There are several proposals to avoid this problem by observing convergence characteristics and keeping the diversity of individuals. They control parameters to keep the balance of exploitation and exploration [9, 7] and to avoid a local minima by changing crossover and mutation rates using simulated annealing [11]. Combining classical optimization techniques with GA is also useful. Such proposals switch a GA search and a modified Powell method, one nonlinear optimization technique [10], or switch a GA search and a local search method [3].

There are several aspects to accelerating a GA search. Some tasks may give priority to the fast calculation of GA operations; some may give priority to the fast GA convergence to a precise global optimum; and some may give priority to fast convergence in early generations. For example, the interactive EC requires fast convergence in early generations but does not require high precision because human operators cannot continue search in many generations and cannot distinguish the slight differences of phenotypes near a global optimum.

We propose a GA acceleration method that approximates the GA search surface using a single-peak function in the next section. This method aims to speed up the GA convergence especially in the early generations, which is essential for the interactive EC [13].

2 New Elite by Fitting a Single-Peak Function on a Searching Surface

Searching surfaces is usually nonlinear and complex; this is why GA or other optimization methods take a long time to determine their global optima. If a complex surface is approximated by a simple surface, it is easier to find the global optimum of the simple surface than that of the complex surface, and the two global optima are expected to be close. Since the searching surface is bounded and usually has only one global optimum in the bounded space, the simplest approximation of any complex search surfaces should be a single-peak curved surface.

Our basic idea is to fit a single-peak function on the complex GA searching surface, calculate the vertex of the single-peak function, and use the vertex vector as an elite parent for the next GA search generation (see Figure 1). In other words, we use the rough shape information of a searching space in addition to the usual GA search.

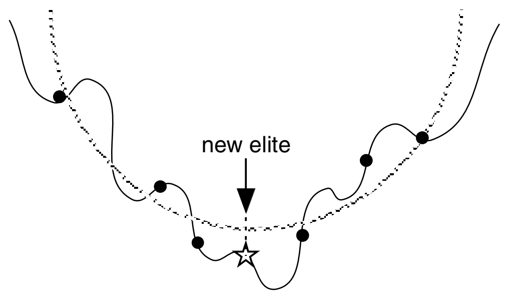


Figure 1: A single-peak function on a searching surface and a new elite.

The advantage of this method is *low risk and high return.* GA convergence becomes fast when the new elite obtained by the proposed method locates near the global optimum. Even if the new elite is poor, the GA convergence does not significantly worsen because only one of the many individuals is replaced by the new elite. For example, since individuals whose fitness is very poor are rarely selected by the roulette wheel selection, replacing the poorest individual by the new elite is not expected to be serious.

The disadvantage of the proposed method would be the calculation cost to obtain the new elite. It is not a problem when the calculation of fitness takes more time than that of a GA operation including the proposed method, such as a GA designing a fuzzy controller. When these calculations are equivalent, some variations of the method or those of its usage may shorten the calculation time.

When we actually use this method, several variations of the idea can be considered. For example, (1) fitting a single-peak function on the best m individual vectors in not only the current population but also the entire population in all past generations or (2) using this proposed method at every k-th generation or only in early generations to reduce calculation time. Replacing the poorest individual by the new elite obtained using the individual information in all past generation may be expected to be better than a conventional GA search.

In this paper, we use a quadratic function as a single-peak function, because when we suppose that a r-order polynomial approximates a searching surface, the simplest approximation is $r = 2$. As the single-peak curve fitting does not require an exact approximation to accelerate a GA search, a simple function such as the quadratic function is desirable from the viewpoint of computational cost and the necessary amount of data for fitting.

The dimension number of the quadratic function is the same as the number of genes. We do not use the $x_i x_j$ term which means rotation and simplify the fitting calculation. For example, when n parameters are encoded in a chromosome, i.e. n-dimensional searching space, the quadratic function that approximates the searching surface is expressed in Eq.(1).

$$f(x_1, x_2, ..., x_n) \approx \sum_{i=1}^{n} a_i (x_i - b_i)^2 + c \qquad (1)$$

where vector $(x_1, x_2, ..., x_n)$ is GA parameters, and $f(\mathbf{x})$ is a fitness value given to the vector.

3 Obtaining a Quadratic Function as a Simplified GA Searching Surface

3.1 Methods to determine function coefficients

The vertex of the quadratic function is given by b_i, $(i = 1, ..., n)$, and $(a_i,\ c)$ are not important for our proposed method, although we need to calculate them to obtain the vertex. We evaluate two methods—the LS (the least squares) method and the SubGA method using a sub GA in main GA—to obtain the coefficients, $(a_i,\ b_i,\ c)$.

(A) Quadratic function fitting by the LS method

Function fitting by the LS method is frequently used to obtain regularity in noisy data. This method minimizes the error between actual data and an approximation function, i.e., we calculate $(a_j,\ b_j,\ c)$ that minimizes $E = \sum_{j=1}^{m} \sum_{i=1}^{n} \{y_j - f(x_{ij})\}^2$ for individuals x_{ij} and fitness values y_j, where m is the number of saved data and n is the number of chromosome parameters.

The LS method requires the condition of $m \geq (2n + 1)$. This condition comes from the $(2n + 1)$ coefficients of Eq.(1) that minimizes the data size to solve the equations by the LS method. The LS method is not applied until accumulated past individuals exceed $(2n + 1)$ if the population size is less than the number.

The obtained b_i in Eq.(1) is used as a new elite parent for the next generation if its fitness value is better than the worst individual in the current population.

(B) Quadratic function fitting by the SubGA method

Since obtaining $(a_i,\ b_i,\ c)$ in Eq.(1) corresponds to their optimization problem, another GA can be used to determine the vector. Let us call it SubGA to distinguish it from the main GA. While the LS method calculates the exact values of (a_i, b_i, c) after a long calculation, the SubGA successively increases their precision. As mentioned before, we need not obtain the exact value of the vertex coordinate to use it as a new elite and accelerate its convergence. So, if a rough vertex coordinate obtained with a few SubGA computations has an equivalent performance to that obtained by the LS method, the GA fitting of a quadratic function is a better method. Furthermore, the calculation time may decrease if a quadratic function is adaptively fitted according to the main GA's searching situation, changed by the number of a SubGA's generations.

When a proper quadratic function is obtained by the SubGA, the worst individual of the main GA is replaced by the vertex coordinate of the function, b_i when the latter is better than the former.

3.2 Data selection methods to determine a quadratic function

It is necessary to collect data from past search points to determine the (a_i, b_i, c) of Eq.(1). The simplest method is to use all past individuals. Although this method might be suitable to obtain the whole shape of a searching surface, the calculation cost is in the population to generation number. Beside the calculation cost, we need to pay attention to the differences in the shapes between the quadratic function and the whole searching surface. Approximation of the quadratic function to the local area near the global optimum is frequently better than the whole searching surface. Since our interest is in the global optimum, the quadratic function fitting using better selected data not only decreases the calculation cost but may also provide a better elite.

We propose the following two methods to select n data for function fitting while considering the two points above.

(A) The best-n data selection based on fitness

This *best-n selection* method selects the best n individuals in the current and past generations. As the locations of the individuals that have higher fitness values are expected to be closer to the global optimum and/or local minima, the quadratic function fitting using these data may provide us with better new elite than using all of the past data. Furthermore, this method limits the amount of data to n, so that another advantage is that the calculation cost is not in the population to generation number.

(B) The nearest-n data selection based on the distance from an elite

The *nearest-n selection* method selects the nearest n individuals in the current and past generations from an elite in the current generation. When the local minima have similar fitness values to that of the global optimum and are located far from the global optimum, the vertex of a quadratic function locates the center between the global optimum and the local minima. The searching surface in Figure 2 is such a case. The *nearest-n selection* method is a better way to solve such cases.

Concentrating the global optimum area is a feature of the *nearest-n selection* method, while less computation is that of the *best-n selection* method. Figure 2 illustrates which data are selected by these methods.

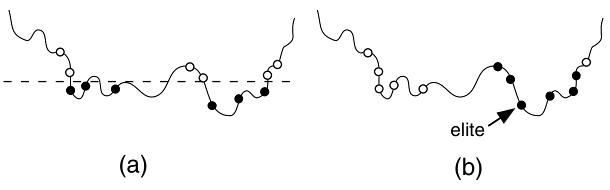


Figure 2: Two data selection methods that fit a quadratic function on a searching surface. Black and white dots on a searching surface are the selected and unselected individuals, respectively, by (a) the *best-n selection* method that selects the best n individuals in the current and past generations, and (b) the *nearest-n selection* method that selects the nearest n individuals from a current elite.

4 Evaluation experiments

We evaluate how the proposed methods accelerate the GA convergence using De Jong's five functions [4] and Schaffer's two functions [12]. Practically, GA convergence characteristics versus computation time should be evaluated at last. However, several variations to shorten computation time are considered. In this paper, we examine the convergence characteristics versus generation as the first stage of the evaluation. Such variations to shorten calculation time are, for example, to apply the proposed method to every k-th generation, or to apply the proposed method only to the early generations to quickly find a rough location of the global optimum. Or, when the method is applied to the interactive EC, the computation time is insignificant because the thinking or waiting time of human operators is quite slower than that of a computer calculation.

We compare three groups of methods: the proposed method whose quadratic function is determined by the LS method, the proposed method whose quadratic function is determined by a SubGA, and the conventional GA. The former two proposed methods are categorized into three based on data selection methods mentioned in section 3.2. They are shown in Table 1.

Table 1: GAs used in convergence experiment. The GA symbols in the left column are used in the convergence graphs in Figure 3. The number, 5, in the symbol means that up to $5 \times (2n+1)$ data are used as data to determine a quadratic function, where n is the number of dimension.

L_{all}	Proposed method with a quadratic function formed by the LS method using all past data.
L_{f5}	Proposed method with a quadratic function formed by the LS method using data selected by the *best-n selection* method.
L_{d5}	Proposed method with a quadratic function formed by the LS method using data selected by the *nearest-n selection* method.
G_{all}	Proposed method with a quadratic function formed by SubGA using all past data.
G_{f5}	Proposed method with a quadratic function formed by SubGA using data selected by the *best-n selection* method.
G_{d5}	Proposed method with a quadratic function formed by SubGA using data selected by the *nearest-n selection* method.
N	Conventional GA without our proposed method.

Table 2: GA conditions.

main GA	
coding	real coding
# of generation	100
population size	30
selection	roulette wheel selection elite strategy
crossover	two-point crossover
crossover rate	90 %
mutation	Gaussian base
mutation rate	5% per chromosome

SubGA	
coding	binary coding
population size	20
# of generation	5
selection	roulette wheel selection elite strategy
crossover	two-point crossover
crossover rate	90 %
mutation	unique random base
mutation rate	5% per bit

The GA convergence experiment is conducted with 30 different initial random values of 100 generations each. The average convergence curves of the best individual of 30 trials for 7 benchmark functions are shown in Figure 3. The GA conditions are shown in Table 2.

5 Discussion

We statistically test whether or not our proposal method accelerates GA convergence. We suppose that the distribution of GA convergence data in our simulation approximates that of Gaussian distribution and test if the difference between two distribution means is significant in each generation. Since 30 simulations were tried, each GA in Table 1 has 30 samples in each generation, and their averages are the convergence curve in Figure 3. Since we assumed the 30 samples belong to a Gaussian distribution, we can test to see if any two convergence curves of the seven GAs in Figure 3 are significantly separated by using averages and variances of the 30 data.

The statistical test has shown that the LS and SubGA methods accelerate GA convergence for all seven benchmark tests; there was no case that both methods do not have significant effect.

Both the LS and SubGA methods were effective for DeJong F_1 and F_5, and Schaffer F_1 and F_2. There was no significant difference between the two proposed methods. Since DeJong F_1 is a 3-D quadratic function, it is a matter of course that the LS method exactly determines the vertex of a fitted quadratic function as it converges to the global optimum in the second generation. The SubGA method takes a little longer generations than the LS methods because it tries to find a 3-D quadratic function using a population size of 20; however, its convergence is faster than that of a conventional GA because it uses the information of the whole searching surface. DeJong F_5 has 25 peaks, and the peaks become high in order. The proposed methods might work well because the 25 peaks roughly form a 2-D quadratic function whose vertex is at the edge of the 2-D space. Schaffer F_2 can be roughly approximated by a quadratic function, which resulted in the same convergent effect. A glance at Schaffer F_1 shows the difficulty in approximating a quadratic function due to its wide plate. But, when many individuals are uniformly spread on the plane part and a few individuals locate in the center searching area, the vertex of the quadratic function that is forcibly fitted should be located near the center, i.e. near the global optimum. This may be the reason for a good result.

For DeJong F_2, three SubGA methods — G_{all}, G_{f5}, and G_{d5} — converged significantly faster than the conventional GA after the 45th generation, while there was no significant difference between the results of the conventional GA and the three LS methods — L_{all}, L_{f5}, and L_{d5} —. It is conceivable that the LS method adopted the exact center point of the F_2 on a convex part as a new elite, and a new elite determined by the SubGA method might be biased due to its rough search after only five generations. However, this estimated result requires further study.

On the other hand, for DeJong F_3 and F_4, the LS method converged significantly faster than the conventional GA, while there was no significance between the conventional GA and the SubGA method. We use small SubGA whose population size is only 20 in consideration of the computational cost. The DeJong F_3 and F_4 have 5- and 30-dimensional searching spaces, respectively, and it seems hard for the SubGA to find better 5-D and 30-D quadratic functions using 20 individuals. Observing the experimental results of seven test functions, 2-D or 3-D may be the applicable limits of SubGA with a population size of 20. This problem can be solved by increasing the population size, but at the expense of increasing the computational cost.

All experimental results have shown that there was no significant difference among three data selection methods, which concludes that the best-n selection method yields the best performance per computational cost.

Our proposed method works well especially in early generations. To shorten calculation time for a single-peak function fitting as mentioned, it is effective to apply the proposed method to only early generations, to every generation, and to every subsequent k-th generation, or some adaptation of this method.

6 Conclusion

We have proposed a GA acceleration method that fits a single-peak function on a searching surface and uses the vertex of the fitted function as a new elite. Experimental results have shown that the method is effective for all seven benchmark test functions.

The method replaces only one of many individuals with the new elite. So that, even in the worst case, if the new elite is very poor, its performance is almost the same as a GA without our method, and it works just as well with a better elite. Our experimental results matched these characteristics, which is another advantage of this method.

References

[1] Berger, J., Salois, M., and Begin R., "A hybrid genetic algorithm for the vehicle routing problem with time windows," Advances in Artificial Intelligence. 12th Biennial Conf. of the Canadian Society for Computational Studies of Intelligence (AI'98), Vancouver, BC, Canada, pp.114–127 (June 1998).

[2] Bramlette, M.F., "Initialization, mutation, and selection methods in genetic algorithms for function optimization," 4th Int'l Conf. on Genetic Algorithms (ICGA'91), San Diego, CA, USA, pp.100–107, Morgan Kaufmann (July, 1991).

[3] Chu, K. C., and Gang, F., "Accelerated genetic algorithms: combined with local search techniques for fast and accurate global search," IEEE Int'l Conf. on Evolutionary Computation, Perth, WA, Australia, vol.1, pp.378–383 (Nov.–Dec.,1995).

[4] De Jong, K.A., "An analysis of the Behavior of a calaa of genetic adaptive systems," Doctoral Dissertation, Univ. of Michigan, Univ. Microfilms No.76-9381 (1975).

[5] Grefenstette, J.J., "Optimization of control parameters for genetic algorithms," IEEE Trans. on System, Man, and Cybernetics, vol.SMC-16, pp.122–128 (1986).

[6] Kawanishi, H.; Hagiwara, M., "A shape detection method using improved genetic algorithm," Trans. of the Institute of Electrical Engineers of Japan, Part C, vol.116-C, no.8, pp.891-897 (Aug. 1996) (*in Japanese).*

[7] Kim, B.M. , Kim, Y.B. , and Oh, C.H. , "A study on the convergence of genetic algorithms," (1996 ICCC & IC, South Korea,) Computers & Industrial Engineering, vol.33, no.3-4, pp.581-588 (Dec., 1997)

[8] Lee, M.A. and Takagi, H., "Dynamic control of genetic algorithms using fuzzy logic techniques," 5th Int'l Conf. on Genetic Algorithms (ICGA'93), Urbana-Champaign, IL, USA, pp.76–83, Morgan Kaufmann (July, 1993).

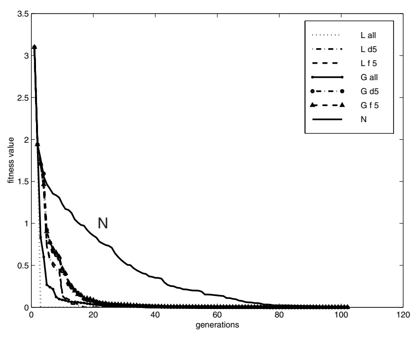
[9] Li, T.-H., LucAsius, C.B., and Kateman, G., "Optimization of calibration data with the dynamic genetic algorithm," Analytia Chimica Acta, vol.268, pp.123–134 (1992).

[10] Okamoto, M., Nonaka, T., Ochiai, S. and Tominaga, D., "Nonlinear numerical optimization with use of a hybrid genetic algorithm incorporating the modified Powell method," Applied Mathematics and Computation, vol.91, no.1, pp.63-72 (April 1998).

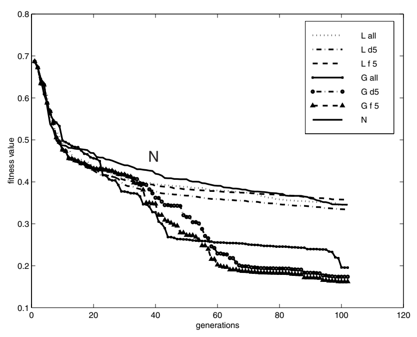
[11] Sefrioui, M. , Periaux, J., and Ganascia, J.-G. , "Fast convergence thanks to diversity," 5th Annual Conf. on Evolutionary Programming. San Diego, CA, USA, MIT Press., pp.313–327 (Feb.-March, 1996).

[12] Schaffer, J.D., Caruana, R.A., Eshelmanm, L.J., and Das, R., "A study of control parameters affecting online performance of genetic algorithms for function optimization," 3rd Int'l. Conf. on Genetic Algorithms (ICGA'89), Morgan Kaufmann Publishers, San Mateo, California, 1989.

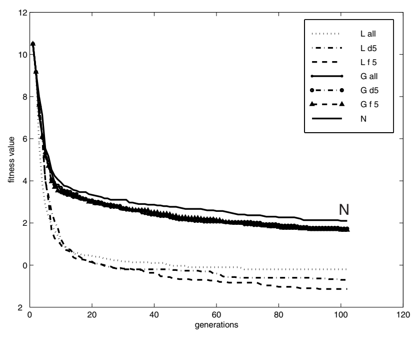
[13] Takagi, H., "Interactive Evolutionary Computation, – Cooperation of computational intelligence and human *KANSEI* –," 5th Int'l Conf. on Soft Computing and Information/Intelligent systems (IIZUKA'98), pp.41–50, Iizuka, Fukuoka, Japan, World Scientific (Oct., 1998).



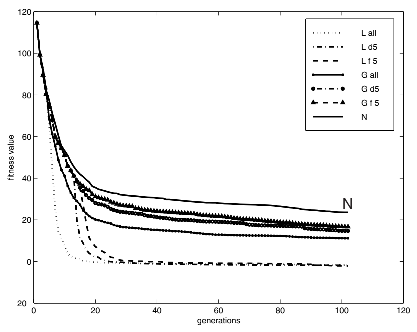
(a1) DeJong F_1



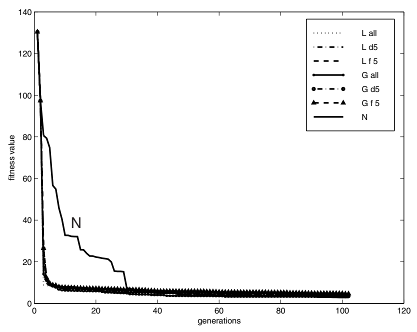
(a2) DeJong F_2



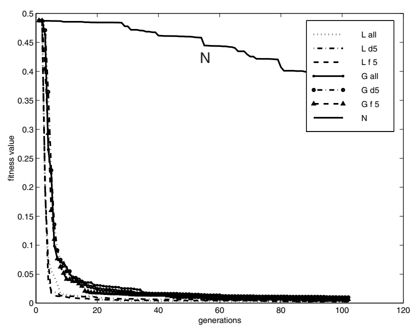
(a3) DeJong F_3



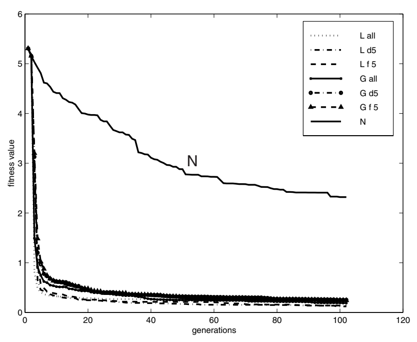
(a4) DeJong F_4



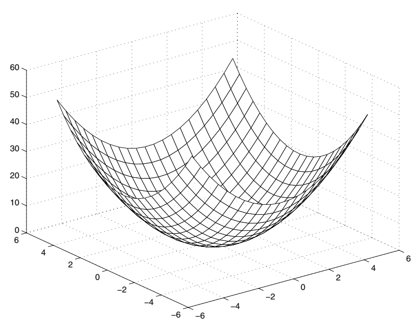
(a5) DeJong F_5

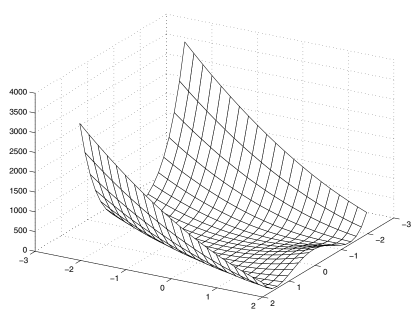


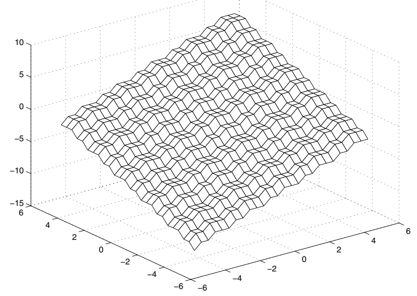
(a6) Schaffer F_1

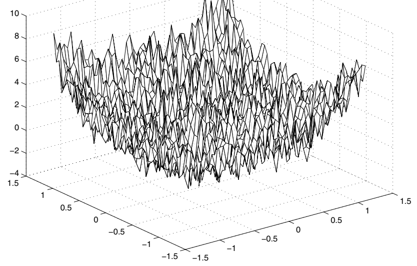


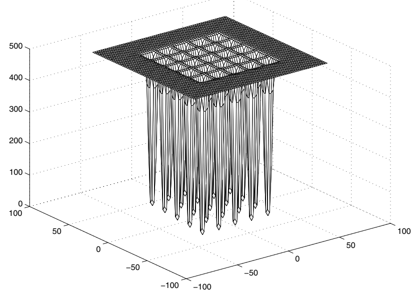
(a7) Schaffer F_2

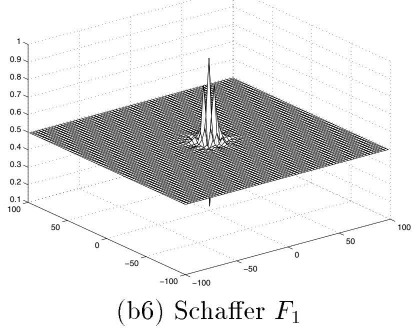

(b1) DeJong F_1


(b2) DeJong F_2


(b3) DeJong F_3


(b4) DeJong F_4


(b5) DeJong F_5


(b6) Schaffer F_1

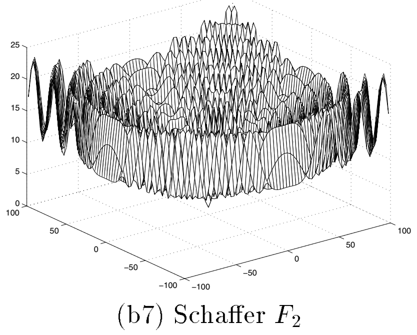

(b7) Schaffer F_2

Figure 3: (a) The convergence characteristics and (b) the shapes of seven benchmark test functions. Symbol 'N' is attached near the convergence curve of conventional GA to distinguish from our proposed methods. See Table 1 for symbols in convergence graphs.