DCGA : A Diversity Control Oriented Genetic Algorithm

Hisashi Shimodaira
Department of Information and Communication, Bunkyo University
1100 Namegaya, Chigasaki-City, Kanagawa 253, Japan
E-mail: f2610390@ca.aif.or.jp

Abstract

In order to attain the global optimum without getting stuck at a local optimum, the appropriate diversity of the structures in the population needs to be maintained. I propose a new genetic algorithm called DCGA (Diversity Control oriented Genetic Algorithm) to attain this goal. In the DCGA, the structures of the population for the next generation are selected from the merged population of parents and their offspring based on a selection probability, which is calculated using a hamming distance between the candidate structure and the structure with the best fitness value. Within the range of my experiments, the performance of the DCGA is remarkably superior to that of the simple GA and the DCGA seems to be a promising competitor of previously proposed algorithms.

1. Introduction

Genetic algorithms (GA's) are one of promising means for function optimization. Methods for function optimization are required to attain the global optimum without getting stuck at a local optimum. For multimodal functions, because the performance of the simple GA is poor in this point, various researches to improve the performance of the GA have been performed as follows.

Baker [1] observed that premature convergence (convergence to a local optimum) often occurs after an individual or a small group of individuals contributes a large number of offspring to the next generation. Booker [2] mentioned that a large number of offspring for one individual means fewer offspring for the rest of the population and when too many individuals get no offspring at all, the result is a rapid loss of diversity and premature convergence. What is needed to handle premature convergence is to prevent this situation. Baker and Booker explored the methods of selection and crossover, respectively, to achieve this end.

Goldberg [3] proposed a method of sharing function by which the issue is eased allowing the formation of species in niches. This mechanism modifies the reproduction probability of a population member by adjusting the fitness value of the structures according to how many population members occupy a niche of the solution space. The method proved superior, but is computationally expensive, because the distance calculation between the structures has a time complexity $O(N^2)$, where N is the size of the population.

In the paradigm of the simple GA, Srinivas [4] proposed the use of adaptive probabilities of crossover and mutation to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the GA. In his algorithm, the probabilities of crossover and mutation are varied depending on the fitness values of structures. High fitness structures are protected, while structures with subaverage fitness value are totally disrupted. In the simple GA, however, because the selection for reproduction is biased toward selecting the better performing individuals, premature convergence often occurs and is essentially inevitable as indicated in the results of Srinivas.

Eshelman [5] proposed an algorithm employing a highly destructive uniform crossover and the Population-Elitist Selection (PES) method which is cross-generational deterministic rank-based survival selection. In the reproduction stage, two candidate structures are selected for mating. In order to maintain diversity, the hamming distance between them is calculated, and if half that distance does not exceed a difference threshold, they are not mated and deleted from the population. Although the performance is remarkably superior to the simple GA, the algorithm is rather complicated.

In order to achieve the above goal, an essential method to maintain appropriate diversity of the structures in the population during the search so that local search and global search are performed in a balanced way, needs to be developed. In this paper, I propose a new genetic algorithm called DCGA (Diversity Control oriented Genetic Algorithm). In the DCGA, the structures for the next generation are selected from the merged population of parents and their offspring eliminating duplicates based on a selection probability, which is calculated using the hamming distance between the candidate structure and the structure with the best fitness value and is larger for structures with larger hamming distances. Within the range of my experiments, the DCGA outperformed the simple GA and seems to be a promising competitor of the previously proposed algorithms.

This paper describes the DCGA and presents the results of the experiments comparing with the simple GA.

1082-3409/97 $10.00 © 1997 IEEE

2. The simple GA

The outline of the simple GA is described to facilitate the later explanation. In the simple GA, the following processing is performed. (1) The number N of individuals in the population is constant and the population is initialized using random numbers. (2) In the reproduction stage, structures are selected from the present population P(t-1) and recombined to form the offspring population C(t), where t is the generation and structure the genotype of an individual. The selection for reproduction (select$_r$) is biased toward selecting the better performing individuals. The recombination is performed using crossover based on probability. A low rate of mutation is used in the recombination stage to maintain population diversity. (3) The selection for survival (select$_s$) is usually unbiased, typically replacing the entire parent population P(t-1) with the child population C(t).

3. DCGA

In order to improve the performance of GA's, the algorithm needs to have ability to robustly explore the solution space to find out the best region containing the global optimum (global search) and to escape from a local optimum when being stuck at it. Attaching greater importance to only current better performing structures may result in premature convergence. On the other hand, a current worse solution may have a greater potential of evolving toward a better future solution to attain the global optimum. The idea motivating my research is to exploit these worse solution instead of discarding them by maintaining the diversity of structures in the population. In addition, it needs to exploit the best solution obtained so far (local search), because it may be in the region containing the global optimum. The DCGA is devised to achieve these twin goals.

The skeleton of the DCGA is shown in Fig.1. The number of structures in the population P(t) is constant and N. The population is initialized using uniform random numbers. In the selection for reproduction select$_r$, all the structures in P(t-1) are paired by selecting two structures without replacement to form P'(t-1). By applying mutation with probability p_m and always applying crossover to the structures of each pair in P'(t-1), C(t) is produced. The mutation rate p_m is constant for all the structures. The structures in C(t) and P(t-1) are merged and sorted in their fitness values order to form M(t). In the selection for survival select$_s$, the structures including the structure with the best fitness value are selected from M'(t) and the population for the next generation P(t) is formed.

The details of the selection for survival select$_s$ are as follows. (1) Duplicate structures in M(t) are eliminated and M'(t) is formed. (2) Structures are selected using the Cross-generational Probabilistic Survival Selection (CPSS) method and P(t) is formed from the structure with the best fitness value in M'(t) and the selected structures. In the CPSS method, structures are selected using random numbers based on a

```
begin;
  t=0;
  initialize population P(t);
  evaluate structures of P(t);
  while (termination condition not satisfied) do;
  begin;
    t=t+1;
    select_r P'(t-1) from P(t-1) by randomly pairing all
      structures without replacement;
    apply mutation with p_m and crossover to each pair
      of P'(t-1) and form C(t);
    evaluate structures in C(t);
    arrange structures of C(t) and P(t-1) in their fitness
      values order and form M(t);
    select_s N structures including the structure with the
      best fitness value from M(t) to form next popul-
      ation P(t) according to the following procedure;
      (1) eliminate duplicate structures in M(t);
      (2) select structures with CDSS or CPSS;
      (3) if the number of selected structures is
          smaller than N, introduce new structures;
  end;
end;
```

Fig.1 Skeleton of DCGA

selection probability defined by the following equation:

$$p_s = \{(1-c)h / L + c\}^{\alpha} \qquad (1)$$

where h is the hamming distance between the candidate structure and the structure with the best fitness value, L the length of the string representing the structure, c the shape coefficient, and α the exponent. If the generated random number is smaller than p_s calculated for a structure, then the structure is selected, otherwise it is deleted. The selection process is performed in the fitness values order of the structures in M'(t) except the structure with the best fitness value. (3) If the number of the structures in P(t) is smaller than N, then new structures generated using random numbers are introduced by the insufficient number. For the traveling salesman problem mentioned later, the hamming distance is calculated considering the order and the reverse order of the city in the structure, because the position of the city does not have a meaning.

The reasons why the above methods are employed in the DCGA are as follows.

Crossover and mutation may side-effectively destroy better performing schemata obtained so far. In the DCGA, because the structure with best performance obtained so far always survives intact into the next generation, the influence of this side-effect is small and large mutation rates can be used and crossover can be always applied. This results in producing offspring which are as different as possible from their parents and examining regions of the search space not yet explored. In fact, the best result was obtained when using the crossover rate equal to 1.0 in the examples mentioned later. On the other hand, in the simple GA, mutation is a background operator, assuring that the crossover has a full range alleles so that the

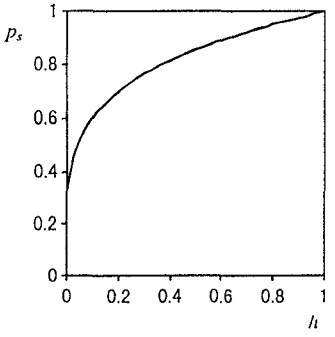

Fig.2 An example curve of Eq.(1) (α=0.225, c=0.0074)

adaptive plan is not trapped on a local optimum [6] and generally small mutation rates are used.

Duplicate structures reduce the diversity of the structures in the population and often cause premature convergence, because the same structures can produce a large number of offspring with the same structure to the next generation. Therefore it is effective to eliminate duplicate structures in order to avoid premature convergence as shown in examples later.

Eq.(1) represents a curve which intersects the two points [h=0, p_s=c^{α}] and [h=L, p_s=1.0] as shown in Fig.2. The curvature of the curve is larger in the region of smaller h, whereas it becomes almost a straight line in the region of larger h. The smaller α becomes, the larger the curvature in the region of smaller h becomes. When α is equal to 1, it becomes a straight line. The larger c becomes, the larger p_s becomes and the curve approaches a horizontal straight line. The selection of structures based on Eq.(1) is biased toward selecting structures with larger hamming distance from the structure with the best fitness value. The degree of the bias is externally adjusted by the values of c and α. The appropriate values need to be explored by trial and error according to the problem. Eq.(1) is very suitable to control the diversity of the structures in the population so as to be in an appropriate condition by externally adjusting the values of c and α.

The structure with the best fitness value obtained so far always survives, because it may be in the region containing the global optimum. Thus the best fitness value for the population can increase monotonously. However, it may be in the region containing a local optimum and the increase of similar structures to it may result in the premature convergence. Therefore, the more similar to the structure with the best fitness value a structure is, the smaller selection probability is applied to it using the CPSS method and the increase of it is restricted. This mechanism may slow the convergence speed to the global optimum, whereas it can be compensated and in fact improved by preventing the solution from getting stuck at a local optimum and stagnating. In this sense, the convergence speed to the global optimum depends upon the diversity of structures in the population. In the DCGA, it can be controlled indirectly by the user through the constants α and c of Eq.(1).

The selection process is performed using the selection probability p_s in the fitness values order of the structures, not considering the fitness values itself. This gives more chances to survive and produce their offspring to current worse structures with the fitness value below the average. In the PES method [5], because the structures are deterministicly selected in their fitness values order, the diversity of structures is often rapidly lost and it results in the premature convergence. The CPSS method can avoid such a situation. The superiority of the CPSS method to the PES method will be demonstrated later.

The introduction of new structures occurs when the iteration proceeds and the diversity of the structures in the population happens to become smaller. This is equivalent to very large mutations introduced into the population and works effectively to restore the diversity automatically.

When the structure is represented by a bit string, binary coding or gray coding is usually used. Caruana [7] suggested that gray coding eliminates the "hamming cliff" problem that makes some transitions difficult when using a binary representation. With the gray coding, the hamming distance between two structures can represent better the degree of their similarity in the phenotype represented by decimal numbers. With the DCGA, because the performance with gray coding is superior to that with binary coding as demonstrated later, it is recommended to use gray coding.

The methods employed in the DCGA can work to escape from a local optimum or avoid premature convergence in the following way.

In the DCGA, structures which survived and the structure with the best fitness value obtained so far can always become parents and produce their offspring. The diversity of the structures in the population is maintained by eliminating duplicate structures and by the selection with the CPSS method. In addition, when the diversity is lost, it can be automatically restored by introducing new structures. Because large crossover and mutation rates are used and diverse structures are maintained in the population, variations of from small to large ranges are applied to each structure. When a small variation is applied to a structure, its neighborhood can be examined to result in the local search. When a large variation is applied to a structure, a region not yet explored can be examined to result in the global search. In such a way, local as well as global searches can be performed in parallel. The structure with the best fitness value obtained so far always survives as a promising candidate to attain the global optimum and its neighborhood can be examined by the local search. On the other hand, because current worse solutions can survive, even if the performance of a structure containing a schema concerning the global optimum is not so high in a stage, this gives the structure a chance by which it can produce an offspring with a fitness value near to the global optimum. This mechanism is

similar to that of the simulated annealing (SA) that can escape from a local optimum by accepting a solution based on a probability whose performance is worse than the present solution. In the DCGA, the solution can escape from a local optimum by the similar working to the SA.

With the simple GA, better performing structures can produce multiple offspring. Therefore, schemata for a dominating local optimum can increase rapidly and eventually dominate the population. On the other hand, with the DCGA, the chance for each structure to become a parent is one time in spite of its performance. In addition, the same structures are eliminated and the number of structures similar to the best performing one is restricted by the selection with the CPSS method. Because these can prevent a structure (especially the structure with the best fitness value) or a small group of structures from contributing a large number of offspring to the next generation, the DCGA can avoid premature convergence.

The comparison of the time complexity on the simple GA and the DCGA is as follows. The number of function evaluations in the reproduction selection stage is N times per a generation on both the methods. The amount of computations for the selection for reproduction is almost the same on both the methods. In the survival selection stage, that on the DCGA is much larger than that on the simple GA by the amount of the computations for the processes (1), (2) and (3) in Fig.1. The time complexity on the check for the identity of structures in the process (1) is not so large, because the check needs to be performed among the structures having the same fitness value. That on the distance calculation between the candidate structure and the structure with the best fitness value is $O(N)$. With the CHC [5], that is $O(N/2)$. With the sharing function method [3], that is $O(N^2)$. Therefore, the DCGA requires much smaller computational cost for maintaining the diversity of structures than the sharing function method.

With the simple GA, the parameters to be tuned are N, p_m and p_c, whereas with the DCGA, N, p_m, c and α.

The originality of the DCGA is to have presented a new genetic algorithm in the generation replacement type GA combining the following ideas and to have experimentally proved their effectiveness in attaining the global optimum: in the selection for survival, (1) duplicate structures are eliminated, (2) structures for the next generation are selected based on Eq.(1), and (3) new structures are introduced, if the number of selected structures is smaller than N. It has a salient feature that the diversity of structures in the population (therefore the convergence speed to the global optimum) is externally controlled through the constants of α and c in Eq.(1) so as to be in an appropriate condition according to the objective problem. I believe that these ideas, the CPSS method defined Eq.(1) among others, have not been presented in previous researches as far as I know. The major difference between the DCGA and the CHC [5] is in that the former employs the CPSS method and the latter the PES method. With the PES method, structures for the next generation are deterministicly selected from the merged population of parents and their offspring in their fitness values order. The superiority of the former to the latter will be demonstrated later. Comparing the DCGA with the sharing function method [3], the purpose and the method of realizing the diversity of structures are essentially different. The purpose of the former is to attain the global optimum efficiently, whereas that of the latter is the parallel investigation of many peaks of a multimodal function. The method of the latter is in the paradigm of the simple GA and modifies the reproduction probability of a population member by adjusting the fitness value of the structures according to how many population members occupy a niche of the solution space. In addition, the time complexity on the distance calculation for the former is much smaller than that for the latter as mentioned above.

4. Experimental results

The performance of the DCGA has been tested on various benchmark problems and compared with that of the simple GA, because the simple GA has been treated as the standard measure for comparisons on performance of various GA's. With simple problems, the differences of the performance were small, whereas with complex problems, the differences were large. Therefore, I present the results for the deceptive functions [5], the multimodal functions f6 [8] and the 30-city TSP [9] which are difficult for GA's to optimize.

Each problem has only one global optimum and it was searched by GA's. For both the DCGA and the simple GA, two-point crossover was used. For the DCGA, crossover was always applied to each pair in P'(t-1). For the simple GA, the performance of (1) the roulette selection method using the elitist strategy with or without the fitness scaling, (2) the pure selection method with the fitness scaling proposed by Grefenstette [11], and (3) the stochastic remainder selection method without replacement [4,10] with or without the fitness scaling, was compared. The results by the first one without the fitness scaling (De Jong's standard GA) [11] which showed the best performance are described in the following. For the DCGA, the following three cases of computation conditions were tested in order to examine the effect of the methods (1) and (2) in Fig.1 employed in the survival selection. Case-1: noneliminating duplicate structures and the PES method. Case-2: eliminating duplicate structures and the PES method. Case-3: eliminating duplicate structures and the CPSS method. With the PES method, structures for the next generation are deterministicly selected from M(t) or M'(t) in their fitness values order in the process (2) in Fig.1.

I examined the combination of best-performing parameter values including the population size changing their values little by little. I performed 20 trials per a parameter set changing seed values for the random number generator to initialize

the population. The same 20 seed values were used for the trials with each parameter set. The trial was continued until the global optimum was attained by at least one structure (I call this the convergence) or until the maximum number of function evaluation times was reached. The maximum number of function evaluation times was 50,000. The performance was evaluated by the number of instances out of the 20 trials in which the GA converged and the average number of function evaluation times in those trials which converged. (An algorithm performs better on a function if it converged more often, or if it converged the same number of times as its competitor but in fewer evaluation times.). The reason why the number of function evaluation times was used is because the amount of computations for function evaluations is generally larger than that of GA itself. In the following, I present the best result obtained in each case. Table 1 shows the definitions of major symbols used in the following tables.

4.1 Deceptive functions

The Goldberg's order-3 deceptive functions were used. Its structure consists of a 30-bit binary string and the value of the function is the sum of 10 3-bit subfunctions. The subfunctions are defined as in Table 2. The tightly ordered and loosely ordered functions were tested. In the case of the tightly ordered deceptive function, the bits of the subfunction are adjacent (1, 2, 3 for the first subfunction; 4, 5, 6 for the second subfunction, and so forth). In the case of the loosely ordered deceptive function, the first subfunction is located at positions 1, 11, and 21, the second subfunction is located at positions 2, 12, 22, and so forth. The global maximum is 300.

Table 3 shows the best result for the tightly ordered function on the simple GA. Tables 4, 5 and 6 show the best results for the tightly ordered function on the DCGA with case-1, case-2 and case-3, respectively. Table 7 shows the best result for the loosely ordered function on the simple GA. Tables 8, 9 and 10 show the best results for the loosely ordered function on the DCGA with case-1, case-2 and case-3, respectively.

Table 1 Definitions of major symbols in Tables

B	Gray cording
G	Binary cording
N	Population size
p_m	Mutation rate
p_i	Inversion rate
p_c	Crossover rate
α	Exponent for probability function, Eq.(1)
c	Shape coefficient for probability function, Eq.(1)
NCV	Number of convergence
AVFE	Average number of function evaluation times
SDFE	Standard deviation of function evaluation times
AVEL	Average number of error logarithm (*EL*)
AVBF	Average number of best fitness values
AVNS	Average number of new structures introduced

Table 2 Goldberg's order-3 deceptive function

f(000)=28	f(001)=26	f(010)=22	f(011)=0
f(100)=14	f(101)=0	f(110)=0	f(111)=30

4.2 Multimodal function

The f6 function [8] is as follows:

$$f6 = 0.5 + \frac{0.5 - \sin^2\sqrt{x^2 + y^2}}{[1 + 0.001(x^2 + y^2)]^2} \qquad (2)$$

This function is cylindrically symmetric about the z axis and has the maximum value 1.0 at the origin. Fig.3 shows a section for y=0 including the x and z axes. The points in the search space were coded as Cartesian x and y values in the range -100 to +100 with 22-bit code, respectively. The comparison of gray coding and binary coding was performed. The maximum value to be searched by GA's in the discrete space is the number with eight 9's below the floating point. The error of the best fitness values (f_{max}) obtained in each trial is calculated by the following equation.

$$EL = -\log_{10}(1 - f_{max}) \qquad (3)$$

For the exact global optimum $EL \cong 8.94$.

Table 11 shows the best results on the simple GA. For the DCGA, Tables 12, 13 and 14 show the best results for case-1, case-2 and case-3, respectively.

4.3 Traveling salesman problem

The Euclidean symmetric 30-city TSP [9] of which global optimum is 420 was tested. The structure was expressed by the path representation. For both the DCGA and the simple GA, the order crossover [9] was used. For the mutation method, the performance with the order-based mutation [14], the position-based mutation, and the inversion was compared. The results by the last one which showed the best performance are described in the following.

Table 15 shows the best results on the simple GA. For the DCGA, Tables 16, 17 and 18 show the best results for case-1,

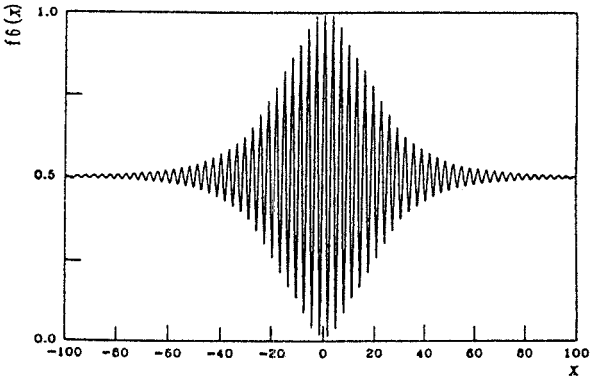

Fig.3 Section of f6 function

Table 3 The best result for tightly ordered deceptive function on simple GA

N	p_m	p_c	NCV	$AVFE$	$SDFE$	$AVBF$
80	0.003	0.6	3	38640	2401	295.4

Table 4 The best result for tightly ordered deceptive function on DCGA with case-1

N	p_m	NCV	$AVFE$	$SDFE$	$AVBF$
90	0.095	20	16880	4782	300.0

Table 5 The best result for tightly ordered deceptive function on DCGA with case-2

N	p_m	NCV	$AVFE$	$SDFE$	$AVBF$
100	0.035	20	6640	2918	300.0

Table 6 The best result for tightly ordered deceptive function on DCGA with case-3

N	p_m	α	c	NCV	$AVFE$	$SDFE$	$AVBF$	$AVNS$
4	0.008	0.49	0.33	20	5018	1940	300.0	242.4

Table 7 The best result for loosely ordered deceptive function on simple GA

N	p_m	p_c	NCV	$AVFE$	$SDFE$	$AVBF$
110	0.0034	0.6	8	75460	14748	297.6

Table 8 The best result for loosely ordered deceptive function on DCGA with case-1

N	p_m	NCV	$AVFE$	$SDFE$	$AVBF$
8	0.085	13	29220	8835	299.1

Table 9 The best result for loosely ordered deceptive function on DCGA with case-2

N	p_m	NCV	$AVFE$	$SDFE$	$AVBF$
4	0.08	20	24928	11001	300.0

Table 10 The best result for loosely ordered deceptive function on DCGA with case-3

N	p_m	α	c	NCV	$AVFE$	$SDFE$	$AVBF$	$AVNS$
4	0.045	0.37	0.8	20	14511	5385	300.0	5.5

Table 11 The best result for f6 function on simple GA

Cord	N	p_m	p_c	NCV	$AVFE$	$SDFE$	$AVEL$
G	70	0.001	0.6	5	30352	8261	3.77
B	30	0.001	0.6	7	7924	2054	4.07

Table 12 The best result for f6 function on DCGA with case-1

Cord	N	p_m	NCV	$AVFE$	$SDFE$	$AVEL$
G	92	0.091	20	25746	6382	8.94

Table 13 The best result for f6 function on DCGA with case-2

Cord	N	p_m	NCV	$AVFE$	$SDFE$	$AVEL$
G	94	0.08	20	22814	6639	8.94

Table 14 The best result for f6 function on DCGA with case-3

Cord	N	p_m	α	c	NCV	$AVFE$	$SDFE$	$AVEL$	$AVNS$
G	12	0.014	0.51	0.235	20	18368	8611	8.94	261.0
B	12	0.014	0.51	0.235	12	20260	9912	7.02	247.8

Table 15 The best result for 30-city TSP on simple GA

N	p_m	p_c	NCV	$AVFE$	$SDFE$	$AVBF$
56	0.09	0.61	7	81048	14866	424.6

Table 16 The best result for 30-city TSP on DCGA with case-1

N	p_i	NCV	$AVFE$	$SDFE$	$AVBF$
110	0.206	10	20174	9830	422.2

Table 17 The best result for 30-city TSP on DCGA with case-2

N	p_i	NCV	$AVFE$	$SDFE$	$AVBF$
120	0.196	20	20298	6029	420.0

Table 18 The best result for 30-city TSP on DCGA with case-3

N	p_i	α	c	NCV	$AVFE$	$SDFE$	$AVBF$	$AVNS$
22	0.097	0.19	0.0	20	19643	6798	420.0	1.1

case-2 and case-3, respectively.

4.4 Summary of the results and discussions

The followings have been experimentally confirmed by these three problems. The performance with case-2 (eliminating duplicate structures) is superior to that with case-1 (noneliminating duplicate structures). In the case of the deceptive functions and the 30-city TSP, the improvement of the performance is remarkable. Therefore, it is obvious that eliminating duplicate structures is very effective to improve the performance for most problems. With the deceptive functions and the f6 function, the performance with case-3 is remarkably superior to that with case-2. With the 30-city TSP, the performance with case-3 is slightly superior to that with case-2. Therefore, it is obvious that the CPSS method is superior to the PES method. According to the results for the f6 function on the DCGA, the performance with gray coding is remarkably superior to that with binary coding. Therefore, it is recommended to use gray coding for the DCGA. It should be noted that with the DCGA, the optimum population size is extremely small and the performance is robust to the changes of the parameter values and the ranges of parameter values with which the global optimum is attained in all trials are considerably wide.

The best results with case-3 show obviously that there exist an optimum population size and an optimum diversity of the structures in the population according to the objective problem. The value of p_{s0} which is p_s for $h = 0$ represents the magnitude of the selection probability and the smaller p_{s0} is, the higher diversity of the structures in the population can be realized. The value of p_{s0} on the best results is 0.58 for the tightly ordered deceptive function, 0.92 for the loosely ordered deceptive function, 0.48 for the f6 function, and 0.0 for the 30-city TSP. It seems that the more difficult the problem is to optimize, the larger the optimum population size becomes. It should be noted that the global optimum can be attained by extremely small structures, if their diversity is appropriately maintained.

In all these three problems, the performance of the DCGA is remarkably superior to that with the simple GA.

It is interesting that how well the DCGA performs comparing with the previous leading methods. The computation conditions in the previous researches are different from each other and some computation conditions and results are not described. Therefore, although exact comparisons of the performance are not impossible, the best results are described in the following for the purpose of conjecturing the differences of the performance.

The results for the deceptive functions are as follows. With the Srinivas's method [4] using the population size of 100 and the upper limit of function evaluation times of 20,000, although the kind of the function used was not described, the 21 trials out of the 30 trials converged. With the CHC [5] using the population size of 50 and the upper limit of function evaluation times of 50,000 for the tightly ordered function, all the 50 trials converged and the average, maximum and minimum numbers of function evaluations to converge were 20,960, 36,297 and 9,933, respectively. With the DCGA using the population size of 4 and the upper limit of function evaluation times of 50,000 for the tightly ordered function, all the 20 trials converged and the average, maximum and minimum numbers of function evaluations to converge were 5,018, 7,684 and 1,834, respectively. With the DCGA using the population size of 4 and the upper limit of function evaluation times of 50,000 for the loosely ordered function, all the 20 trials converged and the average, maximum and minimum numbers of function evaluations to converge were 14,511, 30,216 and 9,173, respectively.

The results for the f6 function are as follows. With the Srinivas's method using a population size of 100, the upper limit of function evaluation times of 20,000 and the convergence threshold value of 0.999 for the function value, the 24 trails out of the 30 trials converged. With the CHC using the population size of 50 and the upper limit of function evaluation times of 50,000, although other computation conditions such as the length of the structure are not described, all the 50 trials converged and the number of function evaluations to converge was on the average 6,496. With the Whitley's method using the population size of 100, the number of 9's below the floating point almost presents a peak at the function evaluation times 4,000 and the maximum value is smaller than 3 [12]. With the DCGA using the population size of 12, gray coding, and the upper limit of function evaluation times of 50,000, all the 20 trials converged and the average, maximum and minimum numbers of function evaluations to converge

were 18,368, 42,685 and 6,322, respectively.

The results for the 30-city traveling salesman problem are as follows. With the Srinivas's method using a very large population size of 1,000 and the upper limit of 100,000 function evaluations, the only 7 trials out of the 10 trials converged. With the CHC using the population size of 50 and the upper limit of 50,000 function evaluations, the 29 trials out of the 50 trials converged (convergence rate = 58%) and the number of function evaluations to converge was on the average 24,866. According to this result, it seems that the CHC suffers from the loss of diversity of structures and premature convergence. With the Whitley's method using a very large population size of 1,000 and the upper limit of the function evaluation times of 30,000, all the 30 trials converged [13]. With the DCGA using the population size of 22 and the upper limit of function evaluation times of 50,000, all the 20 trials converged and the average, maximum and minimum numbers of function evaluations to converge were 19,643, 33,838 and 10,362, respectively.

According to the above, the DCGA seems to be especially suitable for problems of which the global optimum is isolated as the deceptive function. It seems that the DCGA outperforms the Srinivas's method for all these three problems. It seems that the DCGA outperforms the CHC remarkably for the deceptive functions and the 30-city TSP. However, it seems that the latter is superior to the former for the f6 function. It should be noted that the optimum population size for the DCGA is extremely small comparing with the these previous methods.

5. Conclusion

Within the range of the above experiments, the following conclusions can be drawn. The methods employed in the DCGA is effective to attain the global optimum. The DCGA has a salient feature that the diversity of structures in the population is externally controlled so as to be in a appropriate condition according to the objective problem. In addition, the optimum population size is extremely small and its performance is robust to the changes of the parameters. The performance of the DCGA is remarkably superior to that of the simple GA. The DCGA may be a promising competitor to the GA's proposed in the previous researches. However, further evaluations of the DCGA on various problems is required before firm conclusions may be drawn. In additional the theoretical analysis of the convergence process of the DCGA is required.

References

[1] Baker, J.E., "Adaptive Selection Methods for Genetic Algorithms", *Proc. of an International Conference on Genetic Algorithms and Their Applications*, Lawrence Erlbaum Associates, 1985, pp.101-111.

[2] Booker, L., "Improving Search in Genetic Algorithms", *Genetic Algorithms and Simulated Annealing*, Morgan Kaufmann, 1987, pp.61-73.

[3] Goldberg, D. E. et al, "Genetic Algorithms with Sharing for Multimodal Function Optimization", *Proc. of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, 1987, pp.41-49.

[4] Srinivas, M. et al, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms", *IEEE Transactions on Systems, Man and Cybernetics*, Vol.24, No.4, 1994, pp.656-667.

[5] Eshelman, L.J., "The CHC Adaptive Search Algorithm: How to Have Safe Search When Engaging in Nontraditional Genetic Recombination", *Foundation of Genetic Algorithms*, Morgan Kaufmann, 1991, pp.265-283.

[6] Holland, J.H., "*Adaptation in Natural and Artificial Systems*", MIT Press, 1992, p111.

[7] Caruana, R. et al, "Representation and Hidden Bias: Gray vs. Binary Coding for Genetic Algorithms", *Proc. of 5th Int. Conf. on Machine Learning*, Morgan Kaufman, 1988, pp.153-161.

[8] Schaffer J.D. et al., "A Study of Control Parameters Affecting Online Performance of Genetic Algorithms for Function Optimization", *Proc. of the Third International Conference on Genetic Algorithms*, Morgan Kaufmann, 1989, pp.51-60.

[9] Oliver, I.M. et al, "A Study of Permutation Crossover Operators on the Traveling Salesman Problem", *Proc. of the Second International Conference on Genetic Algorithms*, Lawrence Erlbaum Associates, 1987, pp.224-230.

[10] Goldberg, D.E., "Genetic Algorithms in Search, Optimization & Machine Learning", Addison Wesley, 1989.

[11] Grefenstette,J.J., "Optimization of Control Parameters for Genetic Algorithms", *IEEE Transactions on Systems, Man, and Cybernetics*, Vol.SMC-16, No.1, 1986, pp.122-128.

[12] Davis L., "*Handbook of Genetic Algorithms*", Van Nostrand Reinhold, 1991.

[13] Starkweather, T. et al, "A Comparison of Genetic Sequencing Operators", *Proc. of the Fourth International Conference on Genetic Algorithms*, Morgan Kaufmann, 1991, pp.69-76.

[14] Shimodaira, H., "DCGA: A Divesity Control Oriented Genetic Algoritm", *Proc. 2nd IEE/IEEE Int. Conf. Genetic Algorithms in Engineering Systems*, 1997, to appear.