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

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 local optima. For multimodal functions, the power of the traditional GA is poor in this point. In order to achieve this goal, the appropriate diversity in the structures of the population needs to be maintained during the search so that local search and global searches are performed in a balanced way. In this paper, I propose a new genetic algorithm called DCGA (diversity control oriented genetic algorithm). In the DCGA, the structures of the population for the next generation are selected from a merged population of parents and their children eliminating duplicates based on a selection probability, which is a function of a hamming distance between the candidate structure and the structure with the best fitness values and is larger for structures with lager hamming distances. Within the range of my experiments, the performance of the DCGA is remarkably superior to that of the traditional GA and conjectured 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 local optima. For multimodal functions, because the performance of the traditional GA is poor in this point, various researches 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 the following. Since populations are finite, a large number of offspring for one individual means fewer offspring for the rest of the population. 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.

Srinivas [3] recommended 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 are totally disrupted. In the traditional GA, however, because the selection for reproduction is biased toward selecting the better performing individuals, premature convergence often occurs and it is substantially inevitable as indicated in the results of Srinivas.

Eshelman [4] proposed an algorithm employing a highly destructive uniform crossover and population-elitist selection which is cross-generational deterministic rank-based 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 child population. Although the performance is remarkably superior to the traditional GA, the algorithm is rather complicated.

Shimodaira [5] proposed an algorithm employing large mutation rates (in some cases, they are controlled by a decreasing function of generation) and population-elitist selection. In his algorithm, large mutation rates are used to maintain diversity in the structures. Within the range of small examples, the performance is remarkably superior to the traditional GA.

In order to achieve the above goal, the substantial method to maintain the appropriate diversity in the structures of 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 a merged population of parents and their children 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 lager hamming distances. Within the range of my experiments, the performance of the DCGA is remarkably superior to the traditional GA and conjectured to be a promising competitor of previously proposed algorithms.

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

2. The traditional GA

I describe the outline of the traditional GA to facilitate the later explanation. Fig.1 shows the framework of the traditional GA (based on Eshelman [4], but modified), where t is generation, structure the genotype of an individual, and P(t) and C(t) populations of generation t. In the traditional GA, the following processing is performed. (1) The number N of individuals of the population is constant and the population is initialized using random numbers. (2) The selection for reproduction (select_r) is biased toward selecting

Genetic Algorithms in Engineering Systems: Innovations and Applications, 2-4 September 1997, Conference Publication No. 446, © IEE, 1997

```
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);
    recombine structures of P'(t-1) and form C(t);
    evaluate structures of C(t);
    select_s P(t) from C(t) and P(t-1);
  end;
end;
```

Fig.1 Skeleton of the traditional GA

the better performing individuals. (3) The recombination is performed using crossover and mutation based on probability. A low rate of mutation is used in the recombination stage to maintain population diversity. (4) The selection for survival ($select_s$) is usually unbiased, typically replacing the entire parent population P(t-1) with the child population C(t). When an elitist strategy is employed, the individual with the best performance always survives intact into the next generation [6].

3. DCGA

While the traditional GA shows good performance for optimization of unimodal functions, it has a problem with multimodal functions that the solutions are apt to being stuck at local optima and difficult to escape from them. Therefore, in order to improve the performance of GA's, the algorithm needs to have ability to robustly explore the solution space (global search) in order to find out the best region containing the global optimum and to escape from local optima when being stuck at them. 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 goals.

The outline of the DCGA is as follows. The skeleton of the DCGA is shown in Fig.2. The number of structures of 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 of P(t-1) are paired by selecting two structures without replacement to form P'(t-1). By applying mutation with probability p_m and crossover to the structures of each pair, C(t) is produced. The mutation rate p_m is constant for all structures or changed for each structure based on individual probabilisticly sizing mutation rate method (IPMR). In the IPMR method, the size of probability for each structure is determined using random number of which upper limit is the prescribed mutation rate. The structures of C(t) and P(t-1) are merged and arranged in their fitness values order to form M(t). In the selection for survival $select_s$, N structures including the structure with the best fitness values are selected from M(t) to form the population for the next generation P(t) according to the following procedures. (1) Duplicate structures in M(t) are eliminated. (2) Structures are selected using cross-generational deterministic survival selection method (CDSS) or cross-generational probabilistic survival selection method (CPSS). (3) If the number of the selected structures is smaller than N after the procedure in (1) or the CPSS method in (2), new structures are introduced by the insufficient number.

```
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.2 Skeleton of DCGA

The details of the selection for survival $select_s$ are as follows. N structures are selected from M(t) and the population for the next generation P(t) is formed according to the following procedures. (1) Duplicate structures in M(t) are eliminated. (2) The structure with the best performance always survives intact into the next generation. Other structures are selected using the CDSS method or the CPSS method. In the CDSS method, structures are selected in their fitness values order. In the CPSS method, structures are selected using random numbers based on a selection probability calculated 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 the p_s calculated for the structure, then the structure is selected, otherwise it is deleted. The selection process is performed in the fitness values order. (3) If the number of selected structures is smaller than N after the procedure of (1) or the CPSS method in (2), new structures are introduced by the insufficient number. The new structure is generated using random number.

For the TSP, the hamming distance is calculated considering the 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.

In the traditional GA, mutation is a background operator, assuring that the crossover has a full range alleles so that the adaptive plan is not trapped on local optima [7] and generally small mutation rates are used. On the other hand, in the DCGA, mutation is applied with a considerably large probability to pairs of P'(t-1) and crossover is always applied to them. This is for the purpose of producing offspring that are as different as possible from their parents and examining regions of the search space not yet explored. Crossover and

mutation may side-effectively destroy better performing schemata obtained so far. In the DCGA, because the structure with best performance 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.

The mutation rate p_m is constant for all structures or changed for each structure with the IPMR method. With IPMR, when a large mutation is applied to a structure, global search for the structure is performed, whereas when a small mutation is applied to a structure, local search for the structure is performed.

Duplicate structures reduce the diversity in the structures of 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.

Eq.(1) represents a curve which intersects the two points [h=0, p_s=c^{α}] and [h=L, p_s=1.0]. The curvature of the curve is larger in the region of smaller h, whereas it becomes almost a straight line in the region of lager h. The smaller α becomes, the lager 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. The degree of the bias is externally adjusted with the values of c and α. By adjusting the values of c and α, the diversity in the structures of the population can be externally controlled to be an appropriate condition. The appropriate values needs to be explored by trial and error according to the problem.

The structure with the best fitness value may be the most significant and dominating one. However it may be in the region containing the global optimum or in the region containing a local optimum. Therefore although it is always preserved, it is prevented from contributing a large number of offspring to the next generation with the CPSS method.

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

When the structure is represented by a bit string, binary coding or gray coding is usually used. Caruana [8] suggested that gray coding eliminates the "hamming cliff" problem that makes some transitions difficult when using a binary representation. In the DCGA, it is recommended to use gray coding, because the performance with gray coding is superior to that with binary coding as shown later.

It is conjectured that the methods employed in the DCGA can work to escape local optima or avoid premature convergence in the following way, although it needs to be verified experimentally and theoretically.

In the DCGA, structures which survived and the structure with the best fitness value can always become parents and produce their offspring. The diversity in the structures of the population is adjusted by eliminating duplicate structures and selection by the CPSS method. In addition, when the diversity is lost, it can be automatically restored by introducing new structures. Because crossover and mutation rates are large 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 and a structure with the best fitness value is always preserved as a promising candidate to reach the global optimum. 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 near to the global optimum. This mechanism is similar to that of the simulated annealing (SA) that can escape from local optima by accepting a solution based on a probability whose performance is worse than the present solution. It is conjectured, from the similarity between the DCGA and the SA, that escaping from local optima could be prompted by such a way.

In the traditional 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, in the DCGA, the chance for each structure to become a parent is one time in spite of its performance and the number of structures with fitness values near to the best one is restricted by 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 amount of computations on the traditional 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 traditional GA by the amount of the computations for procedures (1), (2) and (3) in Fig.2.

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

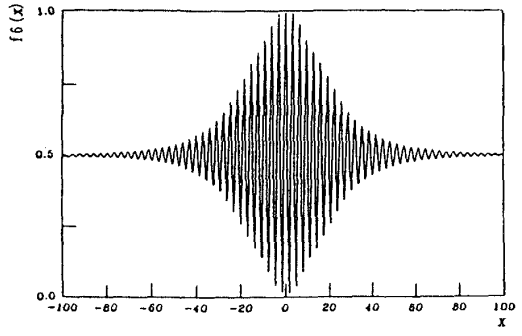
The originalities of the DCGA are to have presented a new genetic algorithm combining the following ideas and to have experimentally proved their effectiveness in attaining the global optimum. In the selection for survival in the generation replacement type GA, (1) eliminate duplicate structures, (2) select structures based on Eq.(1), and (3) introduce new structures, if the number of selected structures is smaller than N after the procedure in (1) or in (2). I believe that these ideas have not been presented in previous researches as far as I know.

4. Experimental results

The performance of the DCGA has been tested on various problems and compared with that of the traditional GA, because the traditional GA has been treated as the standard measure for comparisons on performance of various GA's.

Table 1 Definitions of major symbols in Tables

G	Gray cording
B	Binary cording
N	Population size
p_m	Mutation 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



Fig,3 Section of f6 function

With simple problems the differences of the performance were small, whereas with multimodal functions and traveling salesman problems (TSP) the differences were wide. Next I present the results for the multimodal functions f6 [9] and the 30-city TSP [10] which have been often used as benchmarks and difficult for GA's to optimize.

Each problem has only one global optimum and it was searched by GA's. I examined the combination of best-performing parameter values including the population size changing their values little by little. For the traditional GA, the roulette selection method and the elitist selection method were used but the scaling of fitness value was not used referring to the standard parameter values by De Jong [6] by which better performance was obtained [5]. 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 500000. The performance was evaluated by the number of instances out of 20 trials that the GA succeeded in attaining the global optimum value (convergence) and the average number of function evaluation times in those trials where the global optimum was attained. (An algorithm performs better on a function if it attains the global optimum more often, or if it finds the global optimum 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.

Table 1 shows the definitions of major symbols used in the tables.

Table 2 The best result for f6 function on traditional GA

Cord	N	p_m	p_c	*NCV*	*AVFE*	*SDFE*	*AVEL*
G	50	0.001	0.6	7	109800	147974	4.26
B	50	0.001	0.6	9	26067	16500	5.07

Table 3 The best result for f6 function on DCGA with case-1 (constant mutation rate and CDSS method with duplicates)

Cord	N	p_m	*NCV*	*AVFE*	*SDFE*	*AVEL*
G	92	0.091	20	25746	6382	8.94
B	100	0.080	18	67467	186642	8.25

Table 4 The best result and variations for f6 function on DCGA with case-2 (constant mutation rate and CDSS method without duplicates)

Cord	N	p_m	*NCV*	*AVFE*	*SDFE*	*AVEL*
G	94	0.079	18	24359	4421	8.25
		0.080	20	22814	6639	8.94
		0.090	19	28071	4646	8.60
	90	0.080	19	31244	39772	8.60
	200		20	40720	8630	8.94
B	110	0.100	18	88251	94872	8.25

4.1 Multimodal function

The f6 function [9] 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 z axis. 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 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 the 20 trials is calculated by the following equation.

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

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

For both the DCGA and the traditional GA, two-point crossover was used and the comparison of gray and binary coding was performed. For the DCGA, the following three cases of computation conditions were tested. Case-1: constant mutation rate, noneliminating duplicate structures and the CDSS method. Case-2: constant mutation rate, eliminating duplicate structures and the CDSS method. Case-3: constant mutation rate, eliminating duplicate structures and the CPSS method.

Table 2 shows the best results obtained using the best parameter settings on the traditional GA. For the DCGA, table 3 shows the best results for case-1. Tables 4 and 5 show the best results and the variations of the solutions when a parameter value was changed for case-2 and case-3, respectively. Although the IPMR method was tested, the performance was not improved.

The major characteristics of the results are as follows. In all cases, the performance of the DCGA is superior to that of the traditional GA. For the traditional GA, the performance with binary coding is rather superior to that with gray coding.

Table 5 The best result and variations for f6 function on DCGA with case-3 (constant mutation rate and CPSS method without duplicates)

Cord	N	p_m	α	c	NCV	$AVFE$	$SDFE$	$AVEL$	$AVNS$
G	12	0.0	0.51	0.235	20	57884	71733	8.94	1508.5
		0.005			20	30422	27694	8.94	652.7
		0.010			20	23987	20528	8.94	416.5
		0.015			20	16604	12703	8.94	229.6
		0.030			20	25553	13731	8.94	136.1
		0.078			20	162689	75998	8.94	129.5
		0.079			19	177880	80030	8.91	154.4
	12	0.015	0.34	0.235	19	104198	143003	8.60	836.1
			0.35		20	80964	96524	8.94	566.8
			0.45		20	24486	27036	8.94	241.5
			0.75		20	51721	24304	8.94	899.2
			1.00		20	145428	114617	8.94	3028.9
	12	0.015	0.51	0.000	20	68926	32299	8.94	943.4
				0.100	20	29115	15185	8.94	422.5
				0.300	20	17364	18409	8.94	194.7
				0.380	20	82642	115057	8.94	664.9
				0.390	19	74056	98706	8.25	932.9
	6	0.015	0.51	0.235	20	30578	32557	8.94	968.0
	8				20	24419	30029	8.94	556.7
	16				20	29328	15185	8.94	213.4
	20				20	59796	46167	8.94	223.7
B	12	0.015	0.51	0.235	20	65529	101282	8.94	259.1

For the DCGA, the performance with gray coding is remarkably superior to that with binary coding. For the case-1, the performance is not robust to the changes of the parameter values. For the case-2, the performance is robust to the change of the population size, whereas not robust to the change of mutation rate. For the CDSS method with gray coding,, the performance with case-2 (eliminating duplicate structures) is significantly superior to that with case-1 (noneliminating duplicate structures). The results with case-3 is remarkably better than that with case-2. It should be noted that the best population size is only 12 and the ranges of parameter values with which the global optimum is attained in all trials are very wide and the performance is robust to the changes of the parameter values.

Table 6 The best result for 30-city TSP on traditional GA

N	p_m	p_c	NCV	$AVFE$	$SDFE$	$AVBF$
56	0.009	0.61	10	162428	115786	425.6

Table 7 The best result and variations for 30-city TSP on DCGA with case-1(constant mutation rate and CDSS method with duplicates (D)) and case-2 (constant mutation rate and CDSS method without duplicates (ND))

Method	N	p_m	NCV	$AVFE$	$SDFE$	$AVBF$
D	100	0.058	8	57800	21772	425.8
ND	100	0.057	17	93706	102245	423.0
		0.058	20	78605	34345	420.0
		0.059	18	108589	92221	421.7

Table 8 The best result and variations for 30-city TSP on DCGA with case-3 (constant mutation rate and CPSS method without duplicates)

N	p_m	α	c	NCV	$AVFE$	$SDFE$	$AVBF$	$AVNS$
18	0.0095	0.195	0.0075	18	64072	34277	420.8	0.85
	0.0096			20	71936	55459	420.0	1.20
	0.0097			20	40132	20363	420.0	0.65
	0.0098			20	52445	29430	420.0	0.65
	0.0099			19	94469	65635	420.5	1.00

4.2 Traveling salesman problem

The Euclidean symmetric 30-city TSP [10] of which global optimum is 420 was tested. The structure was expressed by the path representation. For both the DCGA and the traditional GA, the order-based mutation [11] (two city are selected at random and their positions are interchanged) and order crossover [10] were used. For the DCGA, the following four cases of computation conditions were tested. Case-1: constant mutation rate, noneliminating duplicate structures and the CDSS method. Case-2: constant mutation rate, eliminating duplicate structures and the CDSS method. Case-3: constant mutation rate, eliminating duplicate structures and the CPSS method. Case-4: the IPMR method, eliminating duplicate structures and the CPSS method.

Table 6 shows the best results obtained using the best parameter settings on the traditional GA. For the DCGA, Table 7 shows the best results for case-1 and the best results and the variations of the solutions when a parameter value was changed for case-2. Tables 8 and 9 show the best results and the variations of the solutions when a parameter value was changed for case-3 and case-4, respectively.

The major characteristics of the results are as follows. In all cases except case-1, the performance of the DCGA is superior to that of the traditional GA. For the DCGA with the CDSS method, the performance with case-2 (eliminating duplicate structures) is remarkably superior that with case-1 (noneliminating duplicate structures). The performance with the CPSS method is remarkably superior to that with the CDSS method. Although the best performance is obtained in case-3, the performance with case-3 is not robust to the change of mutation rate. By using the IPMR method (case-4), however, the performance becomes stable and robust to the change of parameter values. It should be noted that the best population size is only 20 and the ranges of the parameter values by which the global optimum is attained in all trials are relatively wide.

4.3 Summary of the results and discussions

Table 9 The best result and variations for 30-city TSP on DCGA with case-4 (IPMR method and CPSS method without duplicates)

N	p_m	α	c	NCV	$AVFE$	$SDFE$	$AVBF$	$AVNS$
20	0.0037	0.225	0.0074	19	70976	56264	420.3	12.5
	0.0038			20	68912	48847	420.0	9.9
	0.0104			20	43647	25533	420.0	2.9
	0.0122			20	86329	86948	420.0	2.4
	0.0123			19	64010	54976	420.0	1.5
20	0.0104	0.194	0.0074	19	40142	33587	421.2	3.4
		0.195		20	55553	47968	420.0	4.9
		0.243		20	71477	55556	420.0	4.8
		0.244		19	78784	88403	420.2	3.9
20	0.0104	0.225	0.0068	19	64827	59456	420.7	4.6
			0.0069	20	49440	31853	420.0	2.8
			0.0090	20	47939	55167	420.0	3.0
			0.0109	20	59521	48143	420.0	3.2
			0.0110	19	50105	30737	421.3	2.4
14	0.0104	0.205	0.0070	20	57793	34795	420.0	6.3
18		0.205		20	49348	40698	420.0	3.2
22		0.220		20	48609	36135	420.0	1.4
24		0.230		20	66101	48972	420.0	2.2

For the DCGA, the following have been confirmed by the two problems. Eliminating duplicate structures and the CPSS method is remarkably effective to attain the global optimum. Although with the 30-city TSP the IPMR method is remarkably effective to improve its performance, with the f6 function it is not true. The reason is conjectured as follows. With the f6 function, because many new structures are introduced during the search, the action of variations by the mutation probability is relatively small. On the other hand, with the 30-city TSP because very few new structures are introduced, variations caused by the IPMR method work effectively to improve its performance.

The results of previous researches for the f6 function are as follows. With Srinivas [3] using a population size of 100 and a maximum of 20000 function evaluations, the threshold value of 0.999 (not the exact global maximum) is reached in 24 instances out of 30 trials. With the GALME [5], the global optimum is attained in all 20 trials and the average number of function evaluation times is 22315. With the steady-state genetic algorithm [11], the number of 9's below the floating point almost presents a peak at the function evaluation times 4000 and the maximum value is smaller than 3. Although the result of Eshelman [4] is very good, the computation condition is not obvious.

The results of previous researches for the 30-city TSP are as follows. With Srinivas [3] using a very large population size of 1000 and a maximum of 100000 function evaluations, the global optimum is attained in only 7 instances out of 30 trials. With Eshelman [4] using a population size of 50 and a maximum of 50000 function evaluations, the global optimum is attained in 29 instances out of 50 trials. With GENITOR [12] using a very large population size of 1000, the global optimum is attained is in all 30 trials.

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 optimum population size for the DCGA is very small and its performance is robust to the changes of the parameters. The performance of the DCGA is remarkably superior to that of the traditional GA. The DCGA may be a promising competitor to GA's proposed in the previous researches. However further evaluations of the DCGA for 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*, pp.101-111, 1985.

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

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

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

[5] Shimodaira, H.: A New Genetic Algorithm Using Large MutationRateandPopulation-Elitist Selection (GALME), *Proc. of Eighth IEEE Int. Conf. Tools with Artificial Intelligence*, pp.25-32, 1996.

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

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

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

[9] 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*, pp.51-60, Morgan Kaufmann, 1989.

[10] 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*, pp.224--230, Lawrence Erlbaum Associates, 1987.

[11] Davis L.: Handbook of Genetic Algorithms, Van Nos trand Reinhold, 1991.

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