A Comparative Study of Gene-Duplicated GAs Based on PfGA and SSGA

Hidefumi Sawai and Susumu Adachi
Intelligent Communications Division,
Communications Research Laboratory, MPT,
4-2-1, Nukui-Kita-machi, Koganei, Tokyo, 184-8795 Japan,
e-mail: {sawai, sadachi}@crl.go.jp, Phone number: +81 42 327 7518

Abstract

Gene-Duplicated GAs (GDGAs) based on two different GA approaches are compared using a general set of benchmark problems, including the test functions that were used in the first International Contest on Evolutionary Optimization. The GDGA was inspired by the theory of gene duplication which was first proposed by the Japanese biologist, Dr. Susumu Ohno, in the 1970's. Four variants of the GDGA were considered. Individuals with genes of each length are evolved in subpopulations based on a parameter-free GA (PfGA) and a steady-state GA (SSGA) where genes with different lengths are made by concatenation as they migrate among "demes". The effectiveness of the GDGA was confirmed. The best strategy and variant of the GDGA was determined to depend on the features of the function to be performed.

1 Introduction

The genetic algorithm (GA) [2] is an evolutionary computational paradigm that is inspired by biological evolution [1]. GAs have been successfully used in many practical applications such as functional optimization, combinatorial optimization, and the optimal design of parameters for machines [3]. However, the design of genetic parameters in a GA has to be determined by trial and error, and optimization by GA is thus *ad hoc*.

The development of self-adaptating genetic parameters and operators is one of the most important research areas in evolutionary computation. Adaptation can tune an algorithm while it is being used to solve a given problem [6]. However, the design of an optimal evolutionary strategy for such adaptation is very time-consuming because the evolutionary algorithm must be run many times for the trial and error process to work. To avoid this problem with the setting of adaptive parameters, we have developed a parameter-free genetic algorithm (PfGA) [11],[12] for which specific initial values need not be set as control parameters for genetic operations. This algorithm uses simple random values or probabilities to set almost all genetic parameters. The PfGA is inspired by the "disparity theory of evolution" which was proposed by Furusawa et al. [8]. The PfGA has been implemented as a parallel distributed process on a parallel computer[13],[14].

Gene duplication is rather different, and the principle was first advocated by a Japanese biologist, Dr. Susumu Ohno, in the 1970s [7]. Biological organisms, including viruses, plants, and animals, duplicate their own genes during the process of evolution. Inspired by the theory of gene duplication, we have developed gene-duplicating GAs (GDGAs) in several variants [15]. These variants include a gene-concatenating GA, gene-prolonging GA, gene-coupling GA and extended gene-coupling GA. We discussed the feasibility of basing a gene-duplicating GA on the PfGA, on the basis of results from test functions, at the first ICEO held in 1996 [15]. In this paper, we will continue that discussion of the performance of the GDGA, in terms of more general test fuctions [5], and explain how we have strengthened its search power by comparing the PfGA with the steady-state GA (SSGA)[4] as bases for a GA in which each subpopulation evolves.

When we solve a multi-dimensional optimization problem, each variable in each sub-dimension is first coded as a gene. Next, a fitness function is defined for each sub-dimension and the GA then produces a quasi-optimal solution. Individuals with the genes that correspond to quasi-optimal solutions are evolved by concatenating the genes which allows the optimization to be solved in a higher dimension. Concatenation of genes is realized by having individuals with genes of different lengths migrate among the sub-populations. The PfGA and SSGA are compared in terms of the evolution of the various sub-populations. The PfGA is based on a few-individuals. The SSGA is a typical population-based GA. The two different types of GAs are compared in terms of the effect of gene duplication, and whether or not the effect is dependent on

the GA that is used. We initially expected that the gene-duplicating GA may obtain the globally optimal solution more effectively and efficiently by concatenating quasi-optimal solutions from the SSGA rather than the PfGA.

To evaluate the performance of the GDGAs, we used a set of general test functions each of which we classified into one of eight categories. The categories were defined by combinations of characteristics. The defining characteristics were unimodal or multimodal, symmetrical or asymmetrical, and separable or inseparable.

2 Parameter-free GA (PfGA)

We have developed a parameter-free genetic algorithm (PfGA) [11], [12], in which no control parameters for genetic operations need to be set as constants in advance. It merely uses arbitrarily random values or probabilities for setting almost all genetic parameters. The PfGA is inspired by the *disparity theory of evolution* proposed by Furusawa et al.[8]. Their idea is based on the disparity of copy error rates in the *leading* and *lagging* strands of DNA when each strand makes its copy. This leads to *diversity* in a biological ecosystem. The search strategy in the PfGA is based on a dynamic change of subpopulation size extracted from the population. This strategy enables an adaptive search to take a delicate balance between global and local search methods.

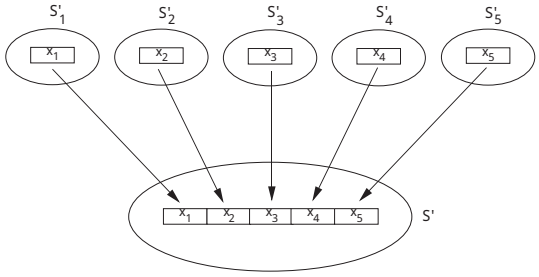

Figure 1: Subpopulations of type A for $N = 5$

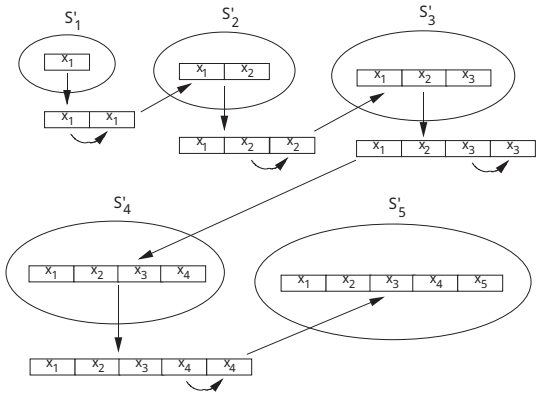

Figure 2: Subpopulations of type B for $N = 5$

3 GAs Inspired by Gene Duplication

We used four types of gene-duplicating GA.

3.1 Gene-Concatenating GA (Type A)

In this algorithm, variable x_i $(i = 1, \ldots, N)$ is encoded as an l-bit Gray code. An individual then evolved with each gene $[x_i]$ by using PfGA or SSGA. The fitness function for each individual is defined as $f(0, 0, \ldots, x_i, 0, \ldots, 0)$. When an offspring C_1 that is more fit than its two parents emerges in a subpopulation in the case of PfGA, all genes from $[x_1]$ to $[x_N]$ are concatenated to make a longer gene $[[x_1] \cdots [x_N]]$ with a rate of duplication R. This long gene belongs to the subpopulation S' as shown in Fig. 1, and the least-fit individual is removed from S' to maintain the size of S'.

3.2 Gene-Prolonging GA (Type B)

Each gene with a length $i \times l$, $[[x_1][x_2] \cdots [x_i]]$ is generated in each subpopulation S'_i $(i = 1, 2, \ldots, N)$ as the initial population. Each individual with a gene of length $i \times l$ evolves in each S'_i according to the fitness function $f(x_1, x_2, \ldots, x_i, 0, \ldots, 0)$. When an offspring C_1 that is more fit than its two parents emerges in S'_i in the case of PfGA, its gene is extended from $[[x_1] \cdots [x_i]]$ to $[[x_1] \cdots [x_{i+1}]]$ with a probability of duplication R by copying $[x_i]$ to $[x_{i+1}]$ $(i = 1, \ldots, N-1)$ as shown in Fig. 2. The individual with the longer gene by l belongs to the subpopulation S'_{i+1}. The least-fit individual is removed from S'_{i+1}.

3.3 Gene-Coupling GA (Type C)

Genes with a length of $i \times l$ are generated as the initial population in a similar way to the method in type B. Each individual in each S'_i evolves according to the PfGA and judgements by the fitness function $f(x_1, x_2, \ldots, x_i, 0, \ldots, 0)$. A gene with a length $i \times l$, $[[x_1] \cdots [x_i]]$ in S'_i is coupled to another gene $[[x_1] \cdots [x_j]]$ in S'_j to produce $[[x_1] \cdots [x_{i+j}]]$ $(2 \leq \text{i+j} \leq \text{N})$ with a probability of duplication R. This individual, with a length of $(i+j) \times l$, belongs to S'_{i+j}, as shown in Fig. 3. The least-fit individual is removed from S'_{i+j}.

3.4 Extended Gene-Coupling GA (Type D)

Type D is an extension of type C. Moreover, the loci of genes are distinguished in type D, as shown in Fig. 4. Again, genes with a length of l are generated as the initial populations. Genes are generated as the concatenation of two adjacent variables $[[x_i][x_{i+1}]]$. Next,

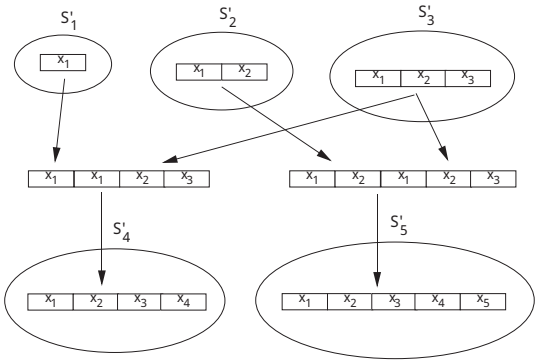

Figure 3: Subpopulations of type C for $N = 5$

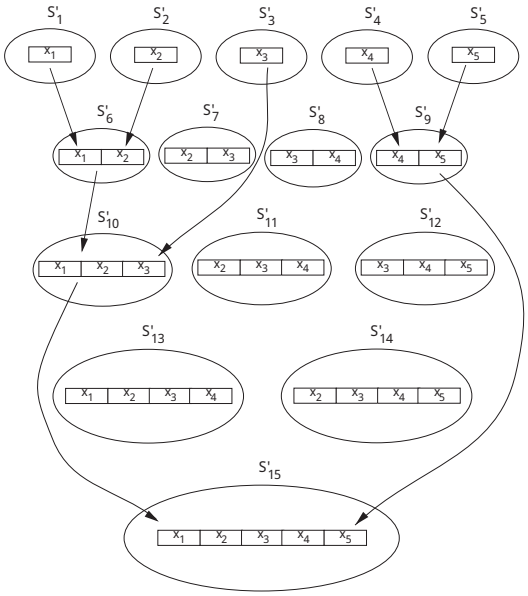

Figure 4: Subpopulations of type D for $N = 5$

genes are generated by concatenating three adjacent variables $[[x_i][x_{i+1}][x_{i+2}]]$. The final set of initial genes are $[[x_1] \cdots [x_N]]$ as shown in Fig. 4. The fitness function of individuals with two adjacenct genes, for example, is defined as $f(0, \ldots, 0, x_i, x_{i+1}, 0, \ldots, 0)$. When an offspring C_1 that is more fit than its two parents emerges in an S_i' in the case of PfGA, genes are only coupled if this will join adjacent loci of the gene, such as for $[x_1]$ and $[x_2]$, $[[x_1][x_2]]$ and $[x_3]$, $[[x_1][x_2][x_3]]$ and $[[x_4][x_5]]$. The process leads to the final gene $[[x_1] \cdots [x_5]]$, as shown in Fig. 4, with a probability of duplication R.

4 Experiment

The four types of gene duplication algorithms were applied to nine optimization (minimization) problems. The functions included benchmark functions which are frequently used and include random transformations in terms of the variable x_i using a uniformly distributed random number r_i ($0 \leq r \leq 1$). The functions are in the general form of functions that can be categorized into eight classes according to whether they are unimodal or multimodal, symmetrical or asymmetrical, and separable or inseparable, in terms of the variable x_i. Five of the functions, the Sphere model (Sp), Griewank's function (Gr), Michalewicz' function (Mi), Shekel's foxholes (Sh) and the generalized form of Langerman's function (La) were used at the first ICEO [9].

PfGA or SSGA are compared for each subpopulation S_i'. A subpopulation size of 100, a crossover rate of 0.8 and a mutation rate of 0.05 were used for the SSGA. The "delete least-fit strategy"[4] used selection of the two least-fit individuals in each generation. Other genetic operations were the same as those used for the PfGA so that a fair comparison would be made. Genes were duplicated according to each type (type A to D) of duplication with a rate of duplication R. The rate of duplication R used for the PfGAs varied from 0 to 1.0, in steps of 0.2. The rate R, which represents the number of individuals in the SSGA was varied from 0 to 50, in steps of 10. The rate R of 0 corresponds to the basic PfGA or SSGA in each case. Migration happened every one hundred evaluations. The dimension N is five. Each value x_i is encoded as a Gray code in 22 to 31 bits depending on the current function. Three hundred independent trials (a trial is defined as 10,000 evaluations) were performed for each subpopulation S_i' with different random seeds.

Criteria[9] for evaluating the performance of each algorithm are the rates of success R_s, of reaching the VTR (Value To Reach) or remaining below it, ENES (Expected Number of Evaluations), BV (Best Value) in all trials and RT (Relative Time). The VTR is defined according to the function. The RT is defined as RT=(CT-ET)/ET where CT is the total CPU time required to perform the given algorithm and ET is the CPU time required to perform one hundred evaluations of fitness. All subpopulations can evolve in parallel. The computational load is roughly in proportion to the length of the gene and is heaviest in subpopulations with genes of length N $\times$ l.

5 Results

Success rates and ENES for each function are given in this section, as a function of migration rates, for the better of the two GAs (PfGA or SSGA).

5.1 Results of ENES and Success Rates

(1)Sphere Model (Sp)

The sphere model is unimodal, symmetrical and sep-

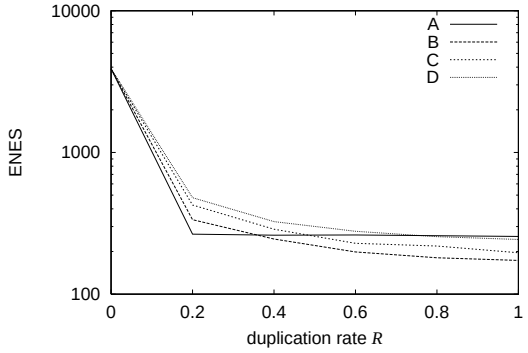

Figure 5: ENES as a function of R for the sphere model (PfGA-based)

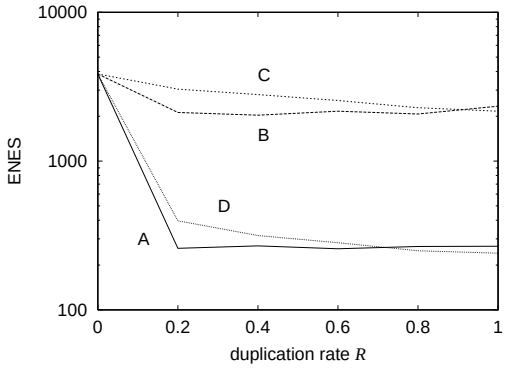

Figure 7: ENES as a function of R for the random sphere model (PfGA-based)

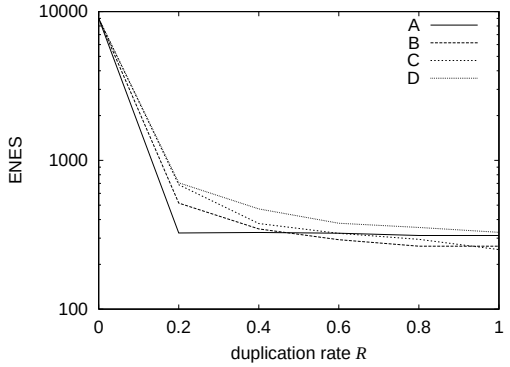

Figure 6: ENES as a function of R for Schwefel's double sum (PfGA-based)

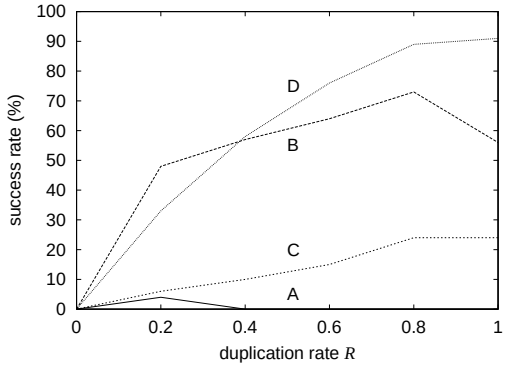

Figure 8: Success rates as a function of R for the random double sum (PfGA-based)

arable. Since the success rate R_s for this model was 100% for all GAs, the figure is omitted. Figure 5 shows the ENES for the four the PfGA-based methods. ENES based on the SSGA is not shown, but all types of gene duplication were more effective for both PfGAs and SSGAs, than for either in its basic form (i.e., for R=0). Type B was the best of the variants because the global minimum for the sphere model is at x_i =1, for any dimensions. The best ENES for the PfGA-based GDGA was 200, and for the SSGA-based GDGAs it was 750.

(2)Schwefel's Double Sum (Ds)

This function is unimodal, symmetrical and inseparable. ENES for the PfGA-based GDGAs is shown in Fig. 6. Although the success rates of only 2% without migration are not shown in the figure, the effects of migration appear immediately, and success rates with migration were 100% for all types of duplication. Types A, B, C and D, in that order, displayed the better perfromance over the range of migration rates from 0.2 to 0.5. The SSGA-based GDGAs also displayed an immediate effect from migration and the success rates reached 100% for all types of duplication from a success rate of zero without migration. The ENES for type B showed some improvement over the ENES for other types (1550 for a migration rate of 40).

(3)Random Sphere model (Rs)

This function is unimodal, asymmetrical and separable. In Fig.7, the ENES is again shown for PfGA-based GDGAs. The success rates are 100% in all cases, and the ENES for types A and D improved dramatically. Types B and C are much less effective than for the sphere model (Fig. 5) . Accordingly, types A and D can be said to be robust in terms of the asymmetry of solutions, whereas types B and C are ineffective for asymmetrical solutions. The ENES for types A and D in these results are equivalent to or slightly worse than those shown in Fig. 5. For SSGA-based GDGAs, the success rate of types A and D was 100% whereas types B and C had a success rate of zero.

(4)Random Double Sum (Rd)

This function is unimodal, asymmetrical and inseparable. Success rates and ENES for the PfGA-based GDGAs are shown in Figs. 8 and 9, respectively.

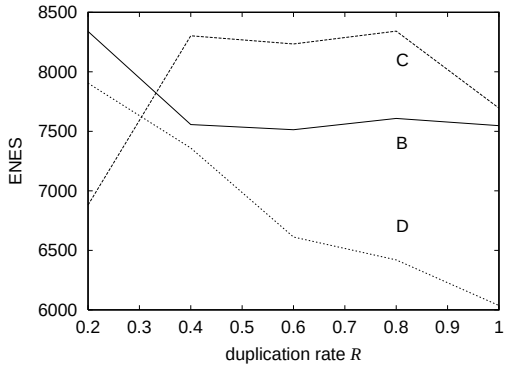

Figure 9: ENES as a function of R for the random double sum (PfGA-based)

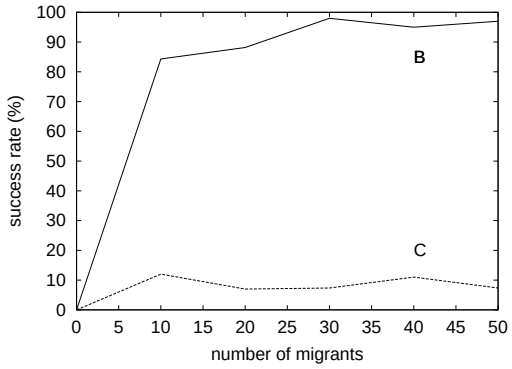

Figure 11: Success rates as a function of the number of migrants for Griewank's function (SSGA-based)

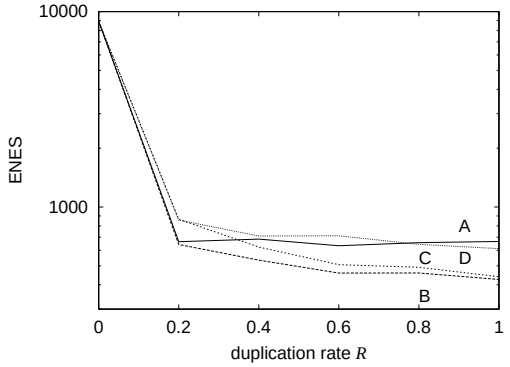

Figure 10: ENES as a function of R for Rastrigin's function (PfGA-based)

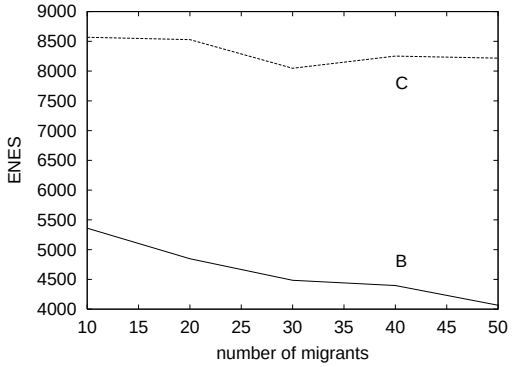

Figure 12: ENES as a function of the number of migrants for Griewank's function (SSGA-based)

As migration rates increase, the success rates also increase, and reach 91% for type D at a migration rate of 1.0. Type D also displayed the best ENES. Success rates for all SSGA-based GDGAs were zero. Type D had the best BV with a value of about 0.01.

(5)Rastrigin's function (Rs)

This function is multimodal, symmetrical and separable. ENES for the PfGA-based GDGAs are shown in Fig. 10. Success rates without migration were 20%, but the rates immediately reached 100% as soon as migration was introduced, for all four types, and the ENES also tended to decrease. Type B appeared to be the most efficient. Success rates also reached 100% for all types of SSGA-based GDGAs, and the best ENES was achieved by type B (ENES=2400 at a migration rate of 30).

(6) Griewank's Function (Gr)

This function is multimodal, symmetrical and inseparable. Success rates R_s and ENES for the SSGA-based GDGAs are shown in Figs. 11 and 12, respectively. Type B is the best with a success rate that almost reached 100%, and its ENES reached 4,000 at a migration rate of 50. The PfGA-based GDGAs showed a similar increase in the success rates for all types except type A. The ENES displayed a gradual decrease for types B and C. Type B was the best (a success rate of 50%, and ENES of about 2,500) and most easily reached the solution.

(7) Michalewicz' Function (Mi)

This function is multimodal, asymmetical and separable. The success rates and ENES for the PfGA-based GDGAs are shown in Figs. 13 and 14, respectively. The effects of gene duplication are most significant to types A and D, with success rates that reach 100% and a dramatic decrease in the ENES with migration (ENES=400). Success rates for the SSGA-based GDGAs similarly reached 100% and the ENES was 1,000 for types A and D.

(8) Shekel's foxholes (Sh)

This function is multimodal, asymmetrical and inseparable. The success rates and ENES for the PfGA-based GDGAs are shown in Figs. 15 and 16, respectively. All of the success rates were less than 10%, and gene duplication had no apparent effect. In terms of

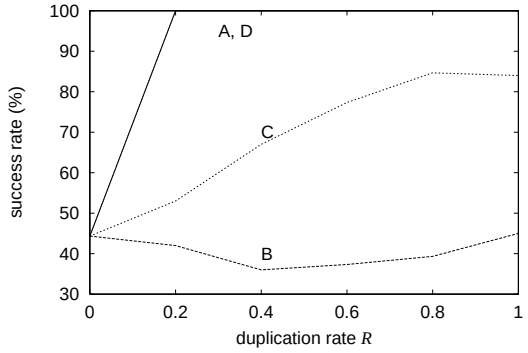

Figure 13: Success rates as a function of R for Michalewicz' function (PfGA-based)

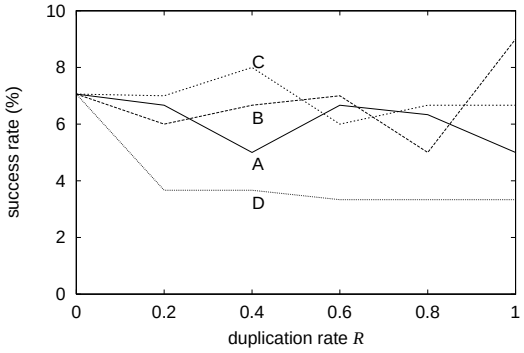

Figure 15: Success rates as a function of R for Shekel's foxholes (PfGA-based)

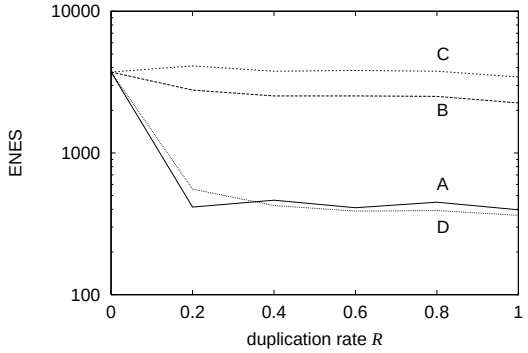

Figure 14: ENES as a function of R for Michalewicz' function (PfGA-based)

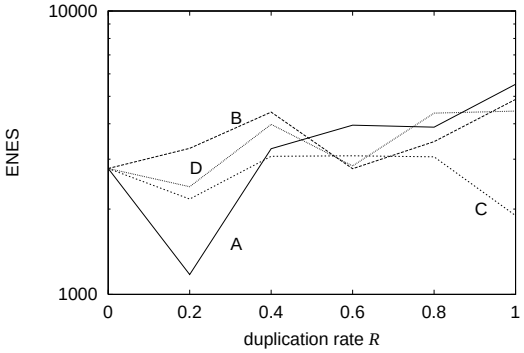

Figure 16: ENES as a function of R for Shekel's foxholes (PfGA-based)

the success rates, type D was relatively inferior to the other types. The situation was similar for SSGA-based GDGAs. This function seems to present difficulties to gene duplicated GAs in terms of finding the solution.

(9) Langerman's Function (La)

This function is also multimodal, asymmetrical and inseparable. Success rates and ENES for the SSGA-based GDGAs are shown in Figs. 17 and 18. Although the maximum success rate was 35% for PfGA-based GDGAs, the rates were around 95 to 97% for all SSGA-based GDGAs which demonstrates a strong effect from the use of a population-based GA. Significant differences don't appear clearly among the gene duplication types, however, types B, C and D displayed a better ENES than type A.

5.2 Comparison with the Algorithms Which Participated in ICEO'96

The GDGA is compared with eight of the algorithms which were entered in the first ICEO in Table 2. Each set of three lines shows the ENES, BV and RT, from top to bottom. Perfromance was judged at the ICEO on the basis of the smallest ENES over 20 runs. The results for the PfGA and GDGA in Table 2 are averaged over 20 runs for comparison with ICEO results. If the PfGA had participated in the ICEO, it would have taken second place. It is compact, has a low RT, and in general shows good performance. Although the RT is higher for the GDGA version, the ENES is significantly lower for the PfGA-based GDGA. As a whole, the PfGA-based GDGA would have taken second place and the SSGA-based GDGA would have taken third place.

In addition, the algorithm which would retain the first place, Bi-Pa, is a method called an *Inductive Search* that uses hill-climbing for the local search, and the Brent method for the global search, and so it is slightly similar to type B.

6 Discussion

The PfGA-based GDGA is now compared with the

Table 1: Comparative results for gene-duplicated GAs based on the PfGA and SSGA

function	feature			GD-PfGA			GD-SSGA			best strategy		
	uni	sym	sepa	best	%succ	ENES	best	%succ	ENES	type	base	rate
Sphere model (Sp)	Yes	Yes	Yes	B	100%	200	B	100%	750	B	PfGA	≥ 0.2
Schwefel's double sum (Ds)	Yes	Yes	No	A	100%	300	B	100%	1550	A	PfGA	≥ 0.2
Random sphere model (Rs)	Yes	No	Yes	A,D	100%	300	D	100%	1500	A,D	PfGA	≥ 0.2
Random double sum (Rd)	Yes	No	No	D	91%	6000	D	0%	-	D	PfGA	~ 1
Rastringin's function (Ra)	No	Yes	Yes	B	100%	400	B	100%	2400	B	PfGA	~ 1
Griewank's function (Gr)	No	Yes	No	B	50%	2500	B	97%	4000	B	SSGA	≥ 0.5
Michalewicz' function (Mi)	No	No	Yes	A,D	100%	400	A,D	100%	1000	A,D	PfGA	≥ 0.2
Shekel's foxholes (Sh)	No	No	No	any	8%	3000	any	1%	5000	any	PfGA	$0 \sim 1$
Langerman's function (La)	No	No	No	any	35%	4000	any	95%	4000	any	SSGA	≥ 0.5

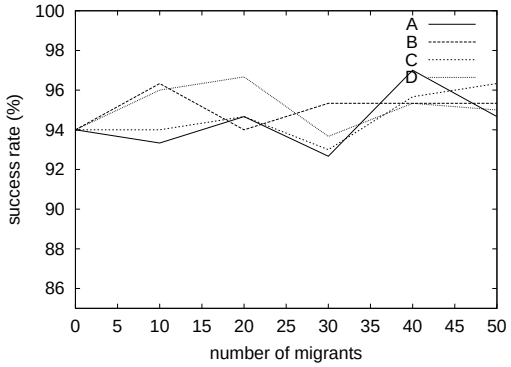

Figure 17: Success rates as a function of the number of migrants for Langerman's function (SSGA-based)

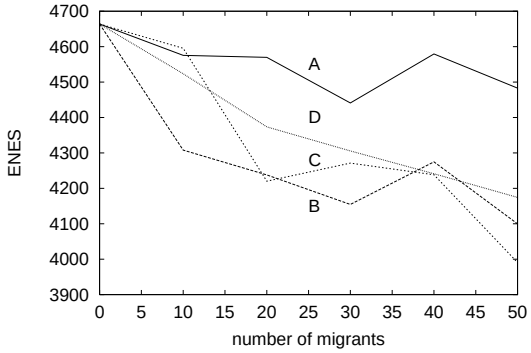

Figure 18: ENES as a function of the number of migrants for Langerman's function (SSGA-based)

SSGA-based GDGA. Table 1 shows the success rates and ENES for the best types of PfGA- and SSGA-based GDGAs, the base form of the type that had the best strategy, and its migration rate. The success rates and ENES vary according to the migration rates, so these values are approximations that use the best types. The best type for both PfGA- and SSGA-based GDGAs is the same for almost every function. The PfGA-based GDGA displays better convergence on the whole as can be seen by comparing the ENES values. Through comparison of the success rates for Griewank's and Langerman's functions (Gr and La), SSGA-based GDGAs can be seen to be superior to PfGA-based GDGAs for multimodal and inseparable functions.

As a whole, the higher the migration rate, the better the performance. The type of population should be selected according to the number of local optima in each function. The PfGA-based GDGA is the better strategy for fast convergence in unimodal functions and those with only a few local optima whereas the SSGA-based GDGA is the better strategy for multimodal functions and those with many local optima.

All types perform well on unimodal and symmetrical functions such as the sphere (Sp) model and Schwefel's double sum (Ds). However, the best type slightly differs with migration according to whether the function is separable or not, i.e., type B is best for separable functions, and type A is best for inseparable functions. Generally speaking, type B, a gene-prolonging GA, is adequate for all symmetrical functions, i.e., Sp, Ds, Ra and Gr. Conversely, types A and D are adequate for asymmetrical functions such as Rs, Rd, Mi, Sh and La, and the effects of migration appear immediately. This fact can also be understood by looking at the structures in types A to D. As a whole, a good strategy is to take type B or C for symmetrical functions and type A or D for asymmetrical functions. Although there is no great difference between the functions Sh and La, a good strategy is to use type D according to the results for Rd because those functions are all asymmetrical and inseparable.

Table 2: Results at the 1st ICEO, for PfGAs and for GDGAs

ENES BV RT	Sphe model (Sp)	Grie func (Gr)	Shek foxh (Sh)	Micha funct (Mi)	Lange func (La)
	20	41	74	120	176
Bi	3.88e-15	7.99e-6	-10.327	-4.6876	-1.499
-Pa	2	2	2	2	2
	243	21,141	6,318	6,804	4,131
Li	0.0	1.69e-5	-10.403	-4.687	-1.499
	12.7	3.1	0.25	1.28	1.62
	736	5,765	76,210	1,877	5,308
Sto	-	-	-	-	-
-Pri	4.67	1.79	0.80	1.11	1.35
	4,067	6,785	1,619	5,131	5,274
PfGA	0.0	4.66e-7	-10.40	-4.688	-1.499
	0.91	0.90	0.33	0.90	0.43
GD-	173	1367	456	240	2,330
PfGA	0.0	3.65e-10	-10.404	-4.688	-1.499
	2.39	2.35	2.40	3.54	2.12
GD-	723	2537	3632	786	3,460
SSGA	0.0	3.65e-10	-10.396	-4.688	-1.499
	71.26	70.76	41.47	79.24	52.83

7 Conclusion

We have extended and strengthened the performance of the gene-duplicating GA (GDGA) by comparing its application to PfGAs and SSGAs as the basis of a GA in which subpopulations evolve. The GDGA has four variants, a gene-concatenating, a gene-prolonging, a gene-coupling and an extended gene-coupling type. These schemes divide a given problem into subproblems and synthesize them based on GAs. Each individual concatenates sub-solutions obtained so far and makes them migrate among subpopulations. We used a set of general test functions including benchmark problems that were used at the first ICEO to evaluate the performance of the GDGA. As a result, the best strategy including the best type of duplication, good migration rates and the best basic GA were ascertained accroding to the features of the test functions. Furthermore, if the type of gene duplication is selected appropriately for a given type of problem, it is then possible to obtain globally optimal solutions faster and more efficiently than with either the PfGA or SSGA (i.e. GAs without migration), for the general set of problems.

In a future study, we will apply the scheme to combinatorial problems and to problems in dynamic environments. We are also going to implement parallel distributed processing and GP (*Gene Duplicated Genetic Programming: GDGP*) as extensions.

References

[1] Charles Darwin, "On the Origin of Species by Means of Natural Selection or the Preservation of Favoured Races in the Struggle for Life," London, John Murray, 1859.

[2] J.H.Holland, "Adaptation in Natural and Artificial System," The University of Michigan Press, 1975.

[3] D.E. Goldberg, "Genetic Algorithm in Search, Optimization, and Machine Learning," Addison Wesley, 1989.

[4] G.Syswerda, "A Study of Reproduction in Generational and Steady-State Genetic Algorithms," Foundation of Genetic Algorithms, pp94-101, Morgan Kaufmann, 1991.

[5] T. Baeck, D. Fogel and Z. Michalewicz (Eds.), "Handbook of evolutionary computation," New York: Oxford University Press, 1997.

[6] Á.E. Eiben, R. Hinterding and Z.Michalewicz, "Parameter Control in Evolutionary Algorithms," IEEE Trans. in Evolutionary Computation, Vol.3, No.2, July 1999.

[7] S.Ohno, "Evolution by Gene Duplication," Springer-Verlag, 1970.

[8] M. Furusawa and H. Doi, "Promotion of Evolution: Disparity in the Frequency of Strand-specific Misreading Between the Lagging and Leading DNA Strands Enhances Disproportionate Accumulation of Mutations," J. theor. Biol., vol. 157, pp 127-133, 1992.

[9] The Organising Committee: H.Bersini, M.Doringo, S.Langerman, G.Seront, L.Gambardella, "Results of the First International Contest on Evolutionary Optimization (1st ICEO)," 1996 IEEE Int. Conf. on Evolutionary Computation (ICEC'96), pp611-615, 1996.

[10] D.Whitley, K.Mathias, S.Rana and J.Dzubera, "Building Better Test Functions," Proc. of the Sixth Int. Conf. on Genetic Algorithms, pp 239-246, Morgan Kaufmann, 1995.

[11] S.Kizu, H.Sawai, and T.Endo, "Parameter-free Genetic Algorithm: GA without Setting Genetic Parameters," Proc. of the 1997 Int. Symp. on Nonlinear Theory and its Applications, vol.2 of 2, pp1273-1276, Dec., 1997.

[12] H.Sawai and S.Kizu, "Parameter-free Genetic Algorithm Inspired by 'Disparity Theory of Evolution,' Proc. of the 1997 Int. Conf. on Parallel Problem Solving from Nature, pp702-711, Sep. 1998.

[13] S.Kizu, H.Sawai and S.Adachi, "Parameter-free Genetic Algorithm (PfGA) Using Adaptive Search with Variable-size Local Population and its Extension to Parallel Distributed Processing," Trans. on IEICE, Vol.J82-D-II, No.3, pp1-10, Mar. 1999 (in Japanese).

[14] H.Sawai and S.Adachi, "Parallel Distributed Processing of a Parameter-free GA by Using Hierarchical Migration Methods," Proceedings of the Genetic and Evolutionary Computation Conference (GECCO)'99, vol.1, pp579-586, July 1999.

[15] H.Sawai and S.Adachi, "Genetic Algorithm Inspired by Gene Duplication," Proceedings of the Congress on Evolutionary Computation, vol.1, pp480-487, July 1999.